Akıllı Sistemlerin Endüstriyel Uygulaması I

Editor
Eyyüp GÜLBANDILAR

BİDGE

BİDGE Yayınları

Akıllı Sistemlerin Endüstriyel Uygulaması I

Editör: Prof. Dr. Eyyüp GÜLBANDILAR

ISBN: 978-625-6707-13-9

1. Baskı
Sayfa Düzeni: Gözde YÜCEL
Yayınlama Tarihi: 25.12.2023
BİDGE Yayınları

Sertifika No: 71374
Yayın hakları © BİDGE Yayınları
www.bidgeyayinlari.com.tr - bidgeyayinlari@gmail.com
Krc Bilişim Ticaret ve Organizasyon Ltd. Şti.
Güzeltepe Mahallesi Abidin Daver Sokak Sefer Apartmanı No: 7/9 Çankaya / Ankara

İÇİNDEKİLER

BÖLÜM I

Minik Makine Öğrenme (TinyML): Kavramlar, Uygulamalar ve Zorluklar

Bashar ALHAJAHMAD [1]

Giriş

Makine öğrenimi (ML), günümüzde teknolojik gelişmelerin vazgeçilmez bir parçası haline gelmiştir. Özellikle Uç Bilişim (Edge Computing) ve Nesnelerin İnterneti (IoT) bir araya geldiğinde, bu ikisi birlikte ağın kenarındaki kaynakları kısıtlı gömülü cihazlarda makine öğrenimi tekniklerini uygulama fırsatları sunmaktadır (Ray, 2022).

Edge bilişim, geleneksel bulut bilişimle ilgili sorunların üstesinden gelmek için kabul gören bir yaklaşım haline gelmiştir, özellikle IoT ortamındadır. Edge bilişim, IoT uygulamalarına hızlı

[1]) Dr. Öğr. Üyesi, Siirt Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Siirt, Türkiye.

hesaplama yanıtı sağlayarak büyük öneme sahiptir. Gelen IoT uygulamalarının bulutta merkezi sunucular tarafından desteklenmesi mümkün olmayabilmektedir. Bu nedenle, kaynakları uzak yerlere yerleştirmek yerine, İnternet'in kenarına (edge) yerleştirmek gerekmektedir (G. Nagarajan & ark., 2022).

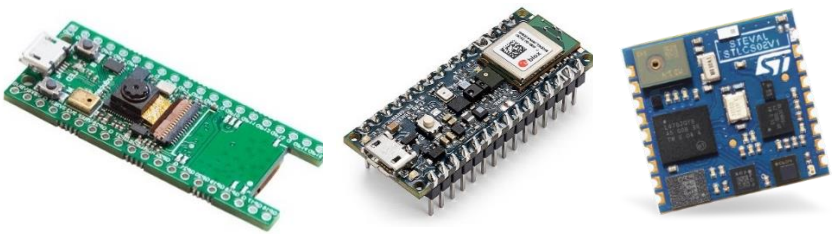
Mikrodenetleyiciler (MCU'lar), genellikle ucuz ve farklı alanlarda uygulanabilen küçük bilgisayarlar olarak kullanılmaktadır. Dünya genelinde yıllık MCU sevkiyatının 30 milyara ulaştığı tahmin edilmektedir ve talebin artmaya devam etmesi beklenmektedir. Makine öğrenimi (ML) ise IoT'nin önemli bir bileşenidir. IoT cihazlarının yaygın kullanımı ve düşük enerji tüketimi, uçta yapay zeka işlevlerinin önemini artırmaktadır. Yapay Zeka (AI) alanındaki son gelişmeler büyük veri, hesaplama ve güç tüketimi gerektirmektedir. MCU'lar genellikle pil ile çalışan ve kaynakları kısıtlı olan sistemlerdir. Bu cihazlar, 0,1 watt'tan daha az güç tüketimiyle uzun süre çalışacak şekilde tasarlanmıştır ve sınırlı kaynaklara sahiptir (örneğin, 64MHz CPU frekansı ve 256KB RAM) (H. Ren & ark., 2021).

MCU'lar ile makine öğrenimi arasındaki boşluğu doldurmak için Tiny Machine Learning (TinyML (TinyML Community, online reference)), uçtaki sinir ağı (NN) tabanlı çözümler geliştirmeye adanmıştır. Son zamanlarda, ses ve yüz tanıma gibi etkileyici sonuçlar elde edilmiştir (R. Sanchez-Iborra & ark., 2020)(B. Moons & ark., 2018).

Çoğu mikro denetleyicinin bir işletim sistemi olmadığı göz önüne alındığında, MCU'larda NN çalıştırmayı desteklemek ve aynı zamanda hesaplama yükünü ve bellek kullanımını minimize etmek için çeşitli "çıplak metal" çıkarım çerçeveleri geliştirilmiştir. Bu amaçla kullanılan bazı kütüphaneler arasında Arm'ın "CMSIS-NN" (L. Lai, 2018) ve Google'ın Mikrodenetleyiciler için TensorFlow Lite'ı (R. David & ark., 2020) bulunmaktadır. Ancak bu kütüphaneler, modelin güçlü makinelerde veya bulutta eğitildiğini ve daha sonra uç cihaza yüklendiğini varsayar. MCU'ların yalnızca çıkarım (inference) yapması gerektiği bir bağlamı ele almaktadır.

TinyML Nedir?

Bilindiği gibi yapay zeka ve onun alt alanı olan makine öğrenmesi günümüzde birçok alanda özellikle nesnelerin interneti (Alhajahmad, 2023, Alreshidi, 2019), gıda güvenliği (Akgerman vd, 2022, Ataş, 2016), eğitim (Bikmaz vd., 2016, Akkaya vd., 2021, Aher vd., 2011), sağlık (Akalın vd., 2020, Ataş vd., 2013, Yetiş vd., 2018, Yeşilnacar vd, 2005, Ataş, vd., 2010), otonom araçlar (Gürtaş, 2020, Muhammed, 2023, Çelebi, 2022), bilimsel hesaplamalar (Ataş, 2016), uzaktan algılama (Tekeli vd., 2016), vb. alanlarda çok geniş bir kullanım potansiyeline sahiptir. TinyML, makine öğrenimi kaynaklarının küçük, kaynak kısıtlı cihazlara entegre edilmesini ifade eden bir yaklaşımdır. Bu, genellikle büyük ölçekli bulut tabanlı uygulamalara odaklanan geleneksel makine öğrenimine karşı gelmektedir. TinyML, kendisi başlı başına bir teknoloji veya yöntem olmasa da, birçok açıdan makine öğrenimini, gömülü sistemleri ve performans mühendisliğini birleştiren bir protomühendislik disiplini olarak işlev görmektedir. Benzer şekilde, kimya mühendisliğinin kimyadan ve elektrik mühendisliğinin elektromanyetizmadan türediği gibi, TinyML de bulut ve mobil bilgi işlem sistemlerindeki makine öğreniminden gelişmiştir (V. J. Reddi vd., 2021). Şekil 1, TinyML alanında kullanılan bazı temel MCU'ları göstermektedir.



a) Arducam Pico4ML

b) Arduino Nano 33 BLE Sense

c) STMicroelectronics SensorTile

Şekil 1. TinyML cihazlarına örnekler

TinyML yaklaşımı, geleneksel makine öğreniminin önündeki engelleri aşmada önemli bir rol oynar. Özellikle uygun bilgi işlem donanımının yüksek maliyeti ve sınırlı veri erişimi gibi sorunları aşmaktadır. Tablo 1'de görüldüğü gibi, TinyML sistemleri geleneksel ML sistemlerine göre neredeyse iki ila üç kat daha ucuzdur ve enerji verimliliği açısından daha iyidir. Bu nedenle, TinyML yaklaşımı, gömülü cihazlarda çok düşük maliyetle veya hatta maliyetsiz bir şekilde hizmet sunabilir ve geleneksel makine öğrenimine kıyasla daha geniş bir görev yelpazesi sunabilmektedir. Ayrıca, TinyML yaklaşımı sorumlu yapay zekanın önemini vurgulamayı da kolaylaştırmaktadır.

Tablo 1. Bulut ve Mobil ML sistemleri ile TinyML sistemleri karşılaştırması. (V. J. Reddi vd., 2021).

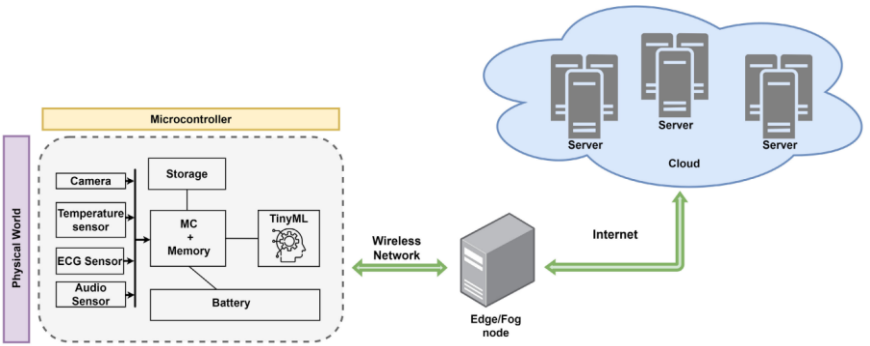
Platformu	Mimari	Bellek	Depolama	Enerji	Fiyat
Cloud E.g., Nvidia V100	GPU Nvidia Volta	HBM 16GB	SSD/disk TB–PB	250W	~$9,000
Mobile E.g., cellphone	CPU Arm Cortex-A78	DRAM 4GB	Flash 64GB	~8W	~$750
Tiny E.g., Arduino Nano 33 BLE Sense	MCU Arm Cortex-M4	SRAM 256KB	eFlash 1MB	0.05W	$3

TinyML, büyük ölçekli, dağıtılmış ve yerel makine öğrenimi görevlerini destekleme kapasitesine sahiptir. Düşük maliyetli gömülü cihazlardaki bu çözümler, ölçeklenebilirliği artırır ve düşük güç tüketimi sayesinde uzak bölgelere, hatta elektrik şebekesinden uzak noktalara dağıtım yapma imkanı sunmaktadır. Doğadaki küçük cihazların sayısı, geleneksel bulut ve mobil sistemlerin sayısını önemli ölçüde aşmaktadır (IC Insights, 2020).

TinyML'in IoT iş akışına entegrasyonu

TinyML, IoT cihazlarının yerel veri analizi yapabilmesini sağlayarak bu cihazların akıllı hale gelmesine olanak tanır ve bu da daha hızlı karar alma süreçleri sunmaktadır. Ayrıca, bağımsız

makine öğrenimi hizmetleri sunmak için farklı kullanım senaryolarına uygulanabilmektedir.
Şekil 2, TinyML'in IoT iş akışına nasıl entegre edildiğini açıklamaktadır (Y. Abadade, 2023).



Şekil 2. TinyML'in IoT iş akışına entegrasyonunu

TinyML, IoT iş akışının bir parçası olarak, sınırlı hesaplama kaynaklarına sahip cihazlarda, özellikle MCU'lar gibi IoT cihazlarında çalışacak kadar küçük ve kaynak açısından verimli ML modellerinin kullanımını ifade eder. Bu modeller, belirli görevleri gerçekleştirmek üzere eğitilmiştir, örneğin görüntü tanıma, ses sınıflandırması veya sensör veri analizi. Veriler, IoT cihazlarının yerel sensörleri tarafından toplanır ve TinyML modeli tarafından işlenir. Modelin çıktısı daha sonra cihazın kendisini kontrol etmesi, daha fazla analiz için yerel işlemciye (Edge), sis düğümüne (Fog Node) veya buluta veri göndermesi gibi çeşitli amaçlar için kullanılabilir.

TinyML Kullanım Avantajları

TinyML, IoT ve sınırlı kaynaklara sahip cihazlar için önemli avantajlar sunar. İşte bu avantajların bazıları: (Soro, S, 2021)

- **Düşük Maliyet:** TinyML, geleneksel makine öğrenimi yaklaşımlarına kıyasla daha düşük maliyetli bir çözüm sunar. Bu, özellikle maliyet duyarlı uygulamalarda ve düşük bütçeli projelerde önemlidir.

- **Enerji Verimliliği:** IoT cihazları genellikle pil ile çalışır ve enerji verimliliği kritik bir faktördür. TinyML, düşük güç tüketimi ile çalışabilir ve böylece cihazların pil ömrünü uzatır.
- **Hızlı Karar Alma:** TinyML, IoT cihazlarının yerel olarak veri analizi yapmasını sağlar, bu da daha hızlı karar alma süreçlerine olanak tanır. Bu özellik, acil tepki gerektiren uygulamalarda çok değerlidir.
- **Veri Gizliliği:** IoT verilerinin yerel olarak işlenmesi, veri gizliliğini artırabilir, çünkü hassas verilerin cihazdan çıkmadan analiz edilmesine izin verir.
- Bağımsızlık: TinyML, internet bağlantısı olmadan çalışabilir. Bu, cihazların çevrimdışı veya düşük bağlantı kalitesi olan ortamlarda da etkili olmasını sağlar.
- **Ölçeklenebilirlik:** Düşük maliyetli gömülü cihazlardaki TinyML çözümleri ölçeklenebilirlik sağlar ve bu cihazların yaygın dağıtımını kolaylaştırır.
- **İş Zekası:** TinyML, cihazların yerel olarak veri analizi yapmasını sağlayarak iş zekası uygulamaları için temel oluşturabilir.
- **Daha Geniş Uygulama Alanları:** TinyML, görüntü tanıma, ses sınıflandırma, sensör veri analizi gibi birçok farklı uygulama alanında kullanılabilir, bu da çeşitli sektörlerdeki ihtiyaçları karşılamak için kullanılabilir.

TinyML, IoT cihazlarının daha akıllı, etkili ve bağımsız hale gelmesini sağlayarak birçok endüstriye ve uygulama alanına önemli katkılarda bulunur.

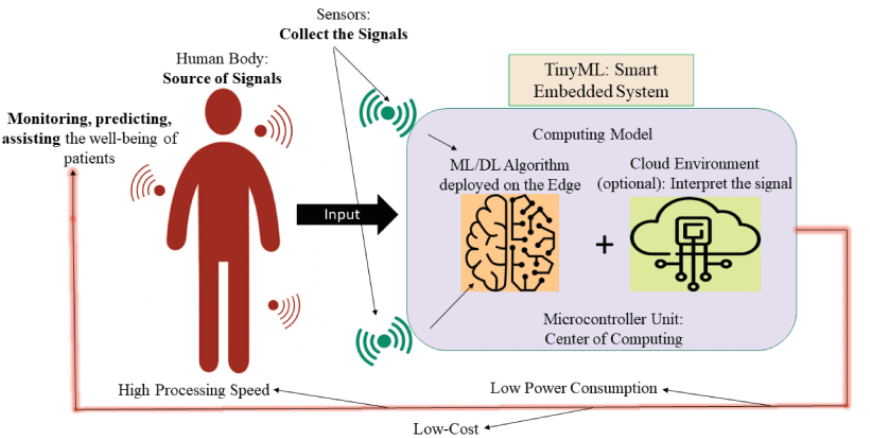
TinyML'nin Farklı Uygulama Alanları

TinyML, artık sağlık hizmetleri, gözetim ve güvenlik, akıllı nesnelerden (sensörlerden şehirlerin yönetimine kadar), endüstriyel izleme ve kontrol, finans ve yönetim, ve günlük yaşantımızın birçok alanında başarıyla entegre edilmektedir. Aşağıda bazı örnek

uygulama alanları verilmiştir: (M. Shafique vd., 2021)(Y. Abadade vd., 2023)

TinyML ve Sağlık Hizmetleri: TinyML'in gelişimi, dünya çapında her toplum için hayati bir öneme sahip olan sağlık sektörünü de etkileyerek dönüştürmektedir. İnsan vücudu, farklı organlar tarafından yayılan sinyallerin kaynağı olarak değerlendirilebilir, bu nedenle bu değerli verilerin toplanması ve sağlık hizmetleriyle ilgili sorunların azaltılması için sensörler kullanılabilmektedir. Ayrıca, bu teknoloji yeni tıbbi izleme, teşhis ve tedavi yöntemlerini mümkün kılma, bakım kalitesini artırma ve sonuçta hasta sonuçlarını iyileştirme potansiyeline sahiptir. Özellikle TinyML, gerçek zamanlı veri işleme yeteneği ve düşük profilli cihazlarda çalışabilme özelliği sayesinde sağlık profesyonellerine hastaları daha etkili ve verimli bir şekilde izleme ve tedavi etme imkanı sunar. Bu alandaki önceki çalışmaların analizi, Şekil 3'te gösterilen genel süreci benimseme fırsatını ortaya koymaktadır (C. Nicolas vd., 2022).

TinyML, sağlık sektöründe inovasyon ve daha iyi hasta bakımı için önemli bir araç haline gelmiştir.



Şekil 3. TinyML teknolojisinin sağlık sektöründe uygulanması için benimsenen genel sürecin gösterimi.

TinyML ve Akıllı Tarım: Birleşmiş Milletler'e göre (World Population Projected, online reference), 2050 yılına kadar dünya nüfusunun 9,8 milyara ulaşması beklenmektedir. Bu hızlı nüfus artışı, iklim değişikliği, toprak erozyonu ve doğal kaynakların tükenmesi gibi faktörlerle birlikte gıda üretimini olumsuz etkileyebilmektedir. Bu nedenle gıda üretimi, artan talebi karşılayabilmek ve sürdürülebilir bir şekilde büyüyebilmek için önemli bir dönüşüme ihtiyaç duymaktadır.

IoT ve TinyML, tarım sektöründe bu dönüşümde kritik bir rol oynamaktadır. IoT çözümleri, çiftliklerde mahsul ve toprak sağlığını izleme, bitki sınıflandırma (Ataş vd., 2010), bitkileri etkileyebilecek hastalıkları tespit etme (Ataş, 2016), tarlaların büyümesini drone'lar aracılığıyla kontrol etme gibi çeşitli görevleri yerine getirebilmektedir. Elektronik endüstrisi, yüksek kaliteli ve uygun maliyetli bileşenlerin bulunmasına öncülük ederek, MCU'lar, tek kartlı bilgisayarlar, sensörler ve radyo alıcı-vericiler gibi bileşenlerin gelişimine katkıda bulunmaktadır. Yeni nesil MCU'lar, sadece temel algılama ve kontrol görevlerini yerine getirmekle kalmayıp, aynı zamanda karmaşık işlemleri, ML modellerini çalıştırmak gibi destekleyerek tarım sektöründe büyük bir yenilik sunmaktadır. Ayrıca, çağdaş radyo teknolojisi, uzun menzilli iletişimin daha düşük enerji tüketimiyle gerçekleştirilebileceği bir seviyeye ulaşmıştır.

TinyML ve Anormallik Tespiti: Anormallik tespiti (AD) veya aykırı değer tespiti, verilerde beklenen davranışa uymayan kalıpları veya gözlemleri tanımlama amacı taşıyan teknikleri ifade eder. Bu teknikler, akıllı şehirler, izleme sistemleri ve enerji yönetimi gibi çeşitli IoT akıllı uygulamalarda kullanılmaktadır. Bu uygulamaların büyük bir kısmında sensörler giriş cihazı olarak kullanılır ve bu cihazlar, analiz, karar verme ve veri depolama amacıyla bulut sunucularına büyük miktarda veri iletilmesini gerektirmektedir (J. Manokaran vd., 2022).

TinyML, daha güçlü bir bilgi işleme cihazına gerek duymadan, gerçek zamanlı olarak olağandışı kalıpları veya davranışları izleyip

tanımlama yeteneği sunarak anormallik tespitinde önemli bir rol oynama potansiyeline sahiptir. Bu sayede IoT cihazlarındaki anormallikleri tespit etmek ve hızlı bir şekilde müdahale etmek, çeşitli endüstrilerde verimliliği artırmak için kullanılabilmektedir.

TinyML ve Bilgisayarlı Görme Uygulamaları: Derin öğrenme, nesne algılama (W. Ouyang vd., 2017), görüntü işleme (Ataş, 2016), hareket izleme (N. Doulamis, 2017) gibi çeşitli bilgisayarlı görme problemlerinde büyük ilerlemelere yol açmaktadır. TinyML, bu gelişmeleri kullanarak görüntüleri sınıflandırma ve görüntü ve videolardaki nesneleri tespit etme yeteneği sunmaktadır. Bu sayede TinyML, makine öğrenimi modellerini gömülü sistemlere entegre etme fırsatı sunmaktadır, bu da gözetim, güvenlik, otomasyon ve birçok diğer uygulama alanında önemli avantajlar sağlamaktadır.

TinyML Zorlukları

TinyML sistemleri, performans karşılaştırmalarını sistematik bir şekilde ölçmek için kullanılabilecek bir çerçeve tasarlarken özgün zorluklarla karşılaşır. Bu bölümde, bu özel engellerin dört temel örneği tartışılacak ve bunların nasıl aşılabileceği önerilecektir (Banbury vd., 2020).

1) Düşük Güç Tüketimi

Düşük güç tüketimi, TinyML sistemlerinin önemli bir özelliğidir ve bu nedenle enerji tüketimini adil bir şekilde ölçmek birçok zorluğu beraberinde getirmektedir. İlk olarak, farklı TinyML cihazları büyük ölçüde değişen güç tüketimine sahip olabilir, bu da doğruluğun korunmasını zorlaştırmaktadır. İkinci olarak, veri yolları ve ön işleme adımları farklı cihazlar arasında önemli ölçüde değişebilir, bu da hangi güç ölçümünün dikkate alınacağını belirlemeyi zorlaştırmaktadır.

2) Sınırlı Bellek Boyutları

Küçük boyutları nedeniyle TinyML sistemleri genellikle sıkı bellek kısıtlamalarına tabidir. Geleneksel ML sistemleri, örneğin akıllı telefonlar gibi, genellikle gigabayt seviyesinde kaynak kısıtlamalarıyla başa çıkabilirken, TinyML sistemleri genellikle bu kaynakların yarısı kadar veya daha azını kullanmaktadır. Bellek, TinyML ile geleneksel ML arasındaki önemli bir karşılaştırma noktasıdır. Geleneksel ML karşılaştırmaları, TinyML cihazlarının sağladığından çok daha yüksek (gigabayt düzeyinde) bellek gereksinimlerine sahip sonuç modelleri kullanmaktadır. Bu, aynı zamanda bir karşılaştırma paketinin dağıtımını da karmaşık hale getirir; çünkü herhangi bir ek yük, güç tüketimini önemli ölçüde etkileyebilir ve hatta karşılaştırmayı gerçekleştiremeyecek kadar büyük hale getirebilmektedir

3) Donanım Çeşitliliği

TinyML sistemleri hızla çeşitlenmektedir. Farklı cihazlar, performans, güç tüketimi ve yetenek açısından çeşitli özelliklere sahiptir. Bu cihazlar, genel amaçlı MCU'ların yanı sıra olay tabanlı sinir işlemcileri ve bellek hesaplama gibi yeni mimarilere sahip olanları içermektedir (Kim vd., 2019). Bu donanım çeşitliliği, performans testlerinin tasarımını karmaşıklaştırır, çünkü test edilen sistemlerin (SUT) belirli standart özelliklere uyması gerekmezdir. Ayrıca, farklı donanım türleri arasında performans sonuçlarını karşılaştırılabilir hale getirme görevi de zor olabilmektedir.

Günümüzdeki kıyaslama yöntemleri, bu tür donanım çeşitliliğiyle başa çıkmak için genellikle yeterli esnekliği sağlamazdır. Bu nedenle, TinyML ekosistemi için daha uygun ve özelleştirilmiş test ve değerlendirme yöntemlerine ihtiyaç vardır ve mevcut kıyaslama yaklaşımlarının yeniden yapılandırılması gerekebilmektedir.

Kod üretimi yöntemleri, önemli derecede manuel kodlama olmadan optimize edilmiş kod üretebilmektedir. Ancak kod oluşturma, farklı satıcıların kendi özel araçları ve derleyicileri

olduğundan taşınabilirlik sorunlarına yol açabilir, bu da genel karşılaştırmayı zorlaştırabilmektedir.

ML yorumlayıcıları, model yapısının soyut olması nedeniyle platformlar arasında taşınabilirliğe izin vermektedir. Örneğin, "Microcontroller için TensorFlow Lite" gibi popüler bir TinyML çerçevesi, çalışma zamanında çekirdek işlemleri gibi bireysel işlemleri çağırmak için bir yorumlayıcı kullanmaktadır. Bu yaklaşım, model mimarisinden bağımsız olduğundan yeni modelleri kolayca değiştirmeyi mümkün kılmaktadır. Ayrıca, çerçeve platforma özgü olarak optimize edilebilir ve değiştirilebilmektedir. Bu yöntem, küçük bir ek yük getirse de ikili boyut ve performans açısından avantajlar sunmaktadır.

4) Yazılım Çeşitliliği

TinyML sistemlerinin model dağıtımı için üç farklı yaklaşım vardır: manuel kodlama (Hand coding), kod üretimi (Code Generation) ve ML yorumlayıcıları (ML interpreters).

Manuel kodlama genellikle en iyi sonuçları üretir çünkü düşük seviyeli optimizasyonları mümkün kılar, ancak zaman alıcıdır ve genellikle optimizasyonların nasıl yapıldığını dışarıdan anlamak zordur. Ayrıca, bu yöntem bilgi paylaşımını ve yeni tekniklerin benimsenmesini sınırlayabilir, bu da TinyML gelişimini yavaşlatabilmektedir. Manuel kodlama, karşılaştırılabilirlik ve tekrarlanabilirlik açısından avantajlıdır, ancak zaman gerektirebilmektedir.

Sonuç

Tiny Machine Learning (TinyML), mevcut teknolojik gelişmelerin bir sonucu olarak makine öğrenimi tekniklerinin küçük, kaynak kısıtlı cihazlara uygulanmasını ifade etmektedir. IoT cihazlarının hızla yayılması ve enerji tüketiminin önemli bir faktör olması nedeniyle TinyML, endüstrinin dikkatini çekmektedir.

Bu bölümde, TinyML'nin uç bilişim ve Nesnelerin İnterneti (IoT) alanlarındaki önemi vurgulanmıştır. Düşük güç tüketimi, düşük maliyet, veri gizliliği, özerklik ve anormallik tespiti gibi avantajları, TinyML'in birçok uygulama alanında kullanılmasını teşvik etmiştir. Bu alanlardan bazıları sağlık hizmetleri, akıllı tarım, ve anormallik tespiti olarak öne çıkmiştir.

Ancak TinyML'nin bazı zorlukları da vardır. Düşük güç tüketimi ve sınırlı bellek, bu cihazların sınırlamalarını işaret ederken, donanım ve yazılım heterojenliği TinyML cihazlarının karşılaştırılmasını karmaşık hale getirmiştir. Bu zorlukları aşmak için standardizasyon ve daha iyi performans ölçümleri gerekmektedir.

Sonuç olarak, TinyML, IoT cihazlarının akıllı ve etkili hale getirilmesinde önemli bir rol oynamaktadır. Bu alandaki gelişmeler, sağlık hizmetlerinden tarıma ve endüstriyel uygulamalara kadar birçok alanda olumlu etkiler yaratabilmektedir. Ancak TinyML ekosistemi, daha fazla standartlaştırma ve karşılaştırılabilirlik için çalışmaların devam etmesini gerektirmektedir.

Kaynaklar

1. Aher, S. B., & Lobo, L. M. R. J. (2011, March). Data mining in educational system using weka. In International conference on emerging technology trends (ICETT) (Vol. 3, pp. 20-25).
2. Akalin, B., & Veranyurt, Ü. (2020). Sağlikta Dijitalleşme Ve Yapay Zekâ. Sdü Sağlık Yönetimi Dergisi, 2(2), 128-137.
3. Akgerman, A., Yavuz, E. D. Ö., Kavaslar, İ., & Güngör, S. (2022). Yapay zeka ve hemşirelik. Sağlık Bilimlerinde Yapay Zeka Dergisi (Journal of Artificial Intelligence in Health Sciences) ISSN: 2757-9646, 2(1), 21-27.
4. Akkaya, N., & Çivğin, H. (2021). Türkçe Eğitiminde Yapay Zekâ. The Journal of International Educational Sciences, 8(29), 308-322.
5. Alhajahmad, B., (2023) "Iot Based Solar Powered Smart Garden Irrigation System", 5. Anatolian Scientific Research Congress, July 21-23, Hakkari, Türkiye.
6. Alreshidi, E. (2019). Smart sustainable agriculture (SSA) solution underpinned by internet of things (IoT) and artificial intelligence (AI). arXiv preprint arXiv:1906.03106.
7. Ataş, M, Yardımcı, Y., & Temizel, A. (2010). Aflatoksinli Biberlerin Hiperspektral Görüntülerinin Siniflandirilmasi İçin Yeni Yaklaşimlar. Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, 12(3), 17-33.
8. Ataş, M. (2016). Fıstık sınıflandırma sistemi için Siirt fıstığı imgelerinden gürbüz özniteliklerin çıkarılması. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 7(1), 93-102.
9. Ataş, M. (2016). Open Cezeri Library: A novel java based matrix and computer vision framework. Computer Applications in Engineering Education, 24(5), 736-743.

10. Ataş, M. Said, (2023) “Farkli Otonom Sürüş Seviyeleri Ve Şerit Tespiti Arasindaki İlişkilerin Araştirilmasi”, 5. Anatolian Scientific Research Congress, July 21-23, Hakkari, Türkiye.
11. Ataş, M., Dikici, A., & Tümay, A. (2013) “Ahbs İstemci Yazılımı Standardizasyonu Yol Haritası”. Akademik Bilişim 2013, Akdeniz Üniversitesi, Antalya, Türkiye.
12. B. Moons, D. Bankman, L. Yang, B. Murmann and M. Verhelst, "Binar-Eye: An always-on energy-accuracy-scalable binary CNN processor with all memory on chip in 28nm CMOS", *2018 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1-4, 2018.
13. Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazel, A., Holleman, J., ... & Yadav, P. (2020). Benchmarking tinyml systems: Challenges and direction. arXiv preprint arXiv:2003.04821.
14. Bikmaz, İ., İşler, V., Kahyaoğlu, M., Akyüz, D., & Ataş, M. (2016). KARMA: Karma Gerçeklik Teknolojisi (Mixed Reality) ile Öğretmenlerin Eğitilmesi.
15. C. Nicolas, B. Naila and R.-C. Amar, "TinyML smart sensor for energy saving in Internet of Things precision agriculture platform", *Proc. 13th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, pp. 256-259, Jul. 2022.
16. Çelebi, S., & Ataş, M. (2022). “Otonom Araçlar İçin Java Tabanli Yazilim Mimarisi Önerisi” II. International Siirt Scientific Research Congress18-19 November 2022, Siirt, Türkiye.
17. G. Nagarajan, Serin V. Simpson, R.I. Minu, Chapter Thirteen - Edge computing security: Layered classification of attacks and possible countermeasures, Editor(s): Pethuru Raj, Kavita Saini, Chellammal Surianarayanan, Advances in Computers, Elsevier, Volume 127, 2022, Pages 359-377.

18. Gürtaş, S. (2020). Otonom araç sürüş destek sistemleri ve yapay zeka uygulamaları (Doctoral dissertation, Bursa Uludag University (Turkey)).

19. H. Ren, D. Anicic and T. A. Runkler, "TinyOL: TinyML with Online-Learning on Microcontrollers," *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 2021, pp. 1-8

20. IC Insights. MCUs Expected to Make Modest Comeback after 2020 Drop, 2020.

21. J. Manokaran and G. Vairavel, "Smart anomaly detection using data-driven techniques in IoT edge: A survey", *Proc. 3rd Int. Conf. Commun. Comput. Electron. Syst.*, pp. 685-702, 2022.

22. Kim, H., Chen, Q., Yoo, T., Kim, T. T.-H., and Kim, B. A 1-16b precision reconfigurable digital in-memory computing macro featuring column-mac architecture and bitserial computation. In ESSCIRC 2019-IEEE 45th European Solid State Circuits Conference (ESSCIRC), pp. 345–348. IEEE, 2019.

23. L. Lai, N. Suda and V. Chandra, "CMSIS-NN: Efficient Neural Network Kernels for Arm Cortex-M CPUs", *arXiv preprint*, 2018.

24. M. Shafique, T. Theocharides, V. J. Reddy and B. Murmann, "TinyML: Current Progress, Research Challenges, and Future Roadmap," 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 2021, pp. 1303-1306.N. Doulamis, "Adaptable deep learning structures for object labeling/tracking under dynamic visual environments," *Multimedia Tools and Applications*, pp. 1–39, 2017.

25. R. David et al., "TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems", *arXiv preprint*, 2020.

26. R. Sanchez-Iborra and A. Skarmeta, "TinyML-enabled frugal smart objects: challenges and opportunities", *IEEE Circuits and Systems Magazine*, vol. 20, no. 3, pp. 4-18, 2020.

27. Ray, P. P. (2022). A review on TinyML: State-of-the-art and prospects. Journal of King Saud University - Computer and Information Sciences, 34(4), 1595-1623.

28. Soro, S. (2021). TinyML for ubiquitous edge AI. arXiv preprint arXiv:2102.01255.

29. Tekeli, A. E., Ataş, M., Dönmez, S., & Fouli, H. (2016). Use of interactive multisensor snow and ice mapping system snow cover maps (IMS) and artificial neural networks for simulating river discharges in Eastern Turkey. Arabian Journal of Geosciences, 9, 1-17.

30. *TinyML Community*, [online] Available: https://www.tinyml.org/home.

31. V. J. Reddi, B. Plancher, S. Kennedy, L. Moroney, P. Warden, A. Agar-wal, C. Banbury, M. Banzi, M. Bennett, B. Brown et al., "Widening access to applied machine learning with tinyml", arXiv preprint, 2021.

32. W. Ouyang, X. Zeng, X. Wang et al., "DeepID-Net: Object Detection with Deformable Part Based Convolutional Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1320–1334, 2017.

33. *World Population Projected to Reach 9.8 Billion in 2050 and 11.2 Billion in 2100*, Dec. 2017, [online] Available: https://www.un.org/en/desa/world-population-projected-reach-98-billion-2050-and-112-billion-2100.

34. Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki and A. S. Hafid, "A Comprehensive Survey on TinyML," in IEEE Access, vol. 11, pp. 96892-96922, 2023.

35. Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki and A. S. Hafid, "A Comprehensive Survey on TinyML," in IEEE Access, vol. 11, pp. 96892-96922, 2023.

36. Yesilnacar, M. I., & Uyanik, S. (2005). Investigation of water quality of the world's largest irrigation tunnel system, the Sanliurfa Tunnels in Turkey. Fresenius Environmental Bulletin, 14(4), 300-306.

37. Yetiş, A. D., Yeşilnacar, M. İ., & Selek, Z. (2018). Ceylanpınar Ovası'nda yeraltı suyu tuzluluğunun coğrafi bilgi sistemi destekli incelenmesi. İklim Değişikliği ve Çevre, 3(1), 51-59.

BÖLÜM II

Doğrusal Olmayan Optimizasyon Problemlerine Kuadratik Programlama Yaklaşımı R ve Lingo Program Çözümleri

Burcu DURMUŞ[1]
Öznur İŞÇİ GÜNERİ[2]

Giriş

Optimizasyon problemleri arasında yer alan kuadratik programlama, karar verme süreçlerini hızlandırmakta ve gerçek hayatta karşılaşılan problemlerin etkin bir şekilde çözümünde kullanılmaktadır. İlk kez 1956'da Frank ve Wolfe tarafından geliştirilen kuadratik programlama problemleri; finansal portföyleri optimize etmek, en küçük kareler regresyon yöntemini uygulamak, kimyasal tesislerde çizelgelemeyi kontrol etmek, görüntü ve sinyal

[1] Dr. Öğr. Gör. Muğla Sıtkı Koçman Üniversitesi, Fen Fakültesi, İstatistik Bölümü, Muğla, Türkiye, https://orcid.org/0000-0002-0298-0802, burcudurmus@mu.edu.tr
[2] Prof. Dr. Muğla Sıtkı Koçman Üniversitesi, Fen Fakültesi, İstatistik Bölümü, Muğla, Türkiye, https://orcid.org/0000-0003-3677-7121, oznur.isci@mu.edu.tr

işleme ve daha karmaşık doğrusal olmayan programlama problemlerini çözmek için yaygın bir şekilde kullanılır (Bruce, Moskowitz & Harley, 1977; Abele, 2015).

Kuadratik programlama problemleri doğrusal olmayan programlama (NLP) problemleri arasında yer almaktadır. Bu problemler bir dizi doğrusal eşitsizlik sınırına tabi olan ikinci dereceden amaç fonksiyonunu maksimum (ya da minimum) yapan problemlerdir. Bu nedenle kuadratik problemlerin NLP problemlerinin özel bir şekli olduğu söylenebilir. Yani amaç fonksiyonu kuadratik formda ve kısıtlama fonksiyonları lineer formdadır (Hillier & Lieberman, 1980).

Araştırmacılar kuadratik programlama problemlerinin çözümü için öncelikli olarak algoritma geliştirilmesi ile ilgilenmişlerdir. Bu alandaki ilk çalışmalardan biri de Wolfe (1959) tarafından önerilen Simpleks yönteminin değiştirilmesine dayanan ve sınırlı sayıda yinelemede birleşen algoritmadır. Bu yöntem, ikinci dereceden programlama problemini doğrusal bir programlama problemine dönüştürür. Karush-Kuhn-Tucker'ın (KKT) koşullarını ekleyerek ve amaç fonksiyonunu doğrusal bir forma dönüştürerek problemi çözer. Bu koşullara literatürde kısaca Kuhn-Tucker koşulları denilmektedir (Karush, 1939; Tucker,1951).

Daha sonra yapılan çalışmalarda bu yöntem modifiye edilmiştir. Moore (1966) tarafından geliştirilen aralık analizi (interval analysis) klasik kuadratik programlamanın gelişmesine fayda sağlamıştır. Terlaky (1987) ilk elverişli çözümden başlayıp, ilk elverişli çözüme tamamlayıcı olan çift elverişli bir çözüm inşa eden aktif küme metodunu geliştirmiştir. Liu ve Wang (2007) ile Li ve Tian (2008) yine aralıklı kuadratik programlama üzerinde çalışmışlardır. Hasan (2012), kuadratik programlama problemlerinin çözümünde pivot elemanını seçmek için yeni bir teknik önermiştir.

Doğrusal olmayan optimizasyon problemlerinin çözümü için bugüne kadar elde edilenler, ikinci dereceden optimizasyon metotları ve sayısal lineer cebir teknikleri ile elde edilmiştir. Bu nedenle günümüzde kuadratik programlama problemleri ve onların

gerçek zamanlı hesapları büyük ilgi görmektedir. Teorik yöntemlerin ve bilgisayar yazılımlarının geliştirilmesi bu alanda yapılan çalışmalara ivme kazandırmıştır. Son yıllarda yapılan çalışmalar göz önüne alındığında bulanık programlama ve bilgisayar yazılımlarına bağlı araştırmaların önemli ölçüde artması bu görüşü desteklemektedir.

Özellikle son yıllarda istatistik analizlerinde sıkça tercih edilen yazılımsal programlar ile analizler yapmak, kuadratik programlama üzerine boşluğu doldurmaktadır. Bu sebeple bu çalışmada, kuadratik programlama problemleri için önerilen çözüm metotları tartışılmıştır. Farklı problemlerin kuadratik programa ile çözümleri Lingo ve R programları aracılığıyla yapılmış ve çalışma boyunca kullanılan kodlar okuyucuya sunulmuştur.

Kuadratik (Karesel) Programlama

Bir kuadratik programlama problemi, amaç fonksiyonu kuadratik (x_j^2 ya da $x_j x_i \neq 0$) olan doğrusal olmayan programlama problemlerinden biridir. Kısıtlar doğrusal programlamada olduğu gibi 1. dereceden yani doğrusaldır. Genel bir kuadratik programlama problemi şu şekilde yazılabilir (Ghadle & Pawar, 2015):

Amaç fonksiyonu:

$$Max \text{ (ya da Min) } Z = \sum_{j=1}^{n} c_j x_j + \frac{1}{2} \sum_{j=1}^{n} \sum_{k=}^{n} c_{jk} x_j x_k$$

Kısıtlar:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

$$x_j \geq 0 \quad (i = 1,2,\ldots,m;\ \mathrm{j} = 1,2,\ldots,n)$$

Tüm j'ler için $c_{jk} = c_{kj}$, tüm $i = 1,2,\ldots,m$ için k ve $b_i \geq 0$'dır.

Bu problem matris formunda ifade edilecek olursa;

Amaç fonksiyonu:

$$Max\ (ya\ da\ Min)\ Z = CX + \frac{1}{2}X^T QX$$

Kısıtlar:

$$AX \leq b, \quad X \geq 0$$

Burada;

$$X = (x_1, x_2, \dots x_n)^T$$

$$C = (c_1, c_2, \dots c_n)$$

$$b = (b_1, b_2, \dots b_m)$$

$$A = \begin{bmatrix} a_{11} & \cdots & 1_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

$$Q = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{bmatrix}$$

A matrisi mxn boyutlu ve Q matrisi nxn boyutlu simetrik bir matristir.

Q matrisi:

a. Problem maksimizasyon ise negatif tanımlıdır.

b. Problem minimizasyon ise pozitif tanımlıdır.

Bu durum Z'nin X'te minimizasyon için kesinlikle konveks, maksimizasyon için kesinlikle konkav olması anlamına gelir. Böylece konveks çözüm uzayını garanti eden kısıtların doğrusal olduğu varsayılır (Taha, 2000).

Kuadratik programlamanın doğrusal programlamadan tek farkı, $f(x) = X^T QX$ şeklinde verilen kuadratik ifadenin amaç fonksiyonuna eklenmesidir (Taha, 2000). Dolayısıyla kuadratik

programlama problemleri, Kuhn-Tucker koşulları yardımıyla Simpleks yönteminin uygulanabileceği bir forma dönüştürülebilir.

Kuadratik programlama problemlerinde kısıtların doğrusal formda olması çözüm bölgesinin konveks olmasını garanti eder. Bu nedenle çözüm için, amaç fonksiyonun tam konveks veya tam konkav olduğunu tespit etmek gereklidir.

$f(x) = X^T QX$ fonksiyonu, amaç fonksiyonunun konveks ya da konkavlığını belirlemede kullanılır (Demirel, 2009).

1. $f(x)$ fonksiyonu kuadratik formda Q matrisindeki tüm asal minörler pozitif ise;

$f(x)$ fonksiyonunun kuadratik formu pozitif belirlidir. $\Delta_i >$ i

2. $f(x)$ fonksiyonu kuadratik formda Q matrisindeki tüm asal minörler sıfır veya pozitif ise;

$f(x)$ fonksiyonunun kuadratik formu yarı pozitif belirlidir. $\Delta_i \geq$ i (Hadley, 1970).

3. $f(x)$ fonksiyonu kuadratik formda Q matrisindeki tüm asal minörler negatif ise;

$f(x)$ fonksiyonunun kuadratik formu negatif belirlidir. $\Delta_i <$ i

$f(x)$ fonksiyonu kuadratik formda Q matrisindeki tüm asal minörler sıfır veya negatif ise;

$f(x)$ fonksiyonunun kuadratik formu yarı negatif belirlidir. $\Delta_i \leq$ i

$f(x)$ fonksiyonunun kuadratik formunun belirlenmesiyle, amaç fonksiyonunun konveks ya da konkav formu ortaya çıkarılır (Taha, 2000).

a. Yarı pozitif belirli bir kuadratik fonksiyon konvekstir.
b. Yarı negatif belirli bir kuadratik fonksiyon konkavdır.
c. Pozitif belirli bir kuadratik fonksiyon kesin konvekstir.
d. Negatif belirli bir kuadratik fonksiyon kesin konkavdır.

Kuadratik Programlama için Çözüm Metotları

Kuadratik programlama problemlerinin çözümü için çok sayıda yaklaşım vardır. Simpleks yöntemi bunlardan biridir. Bu problemlerin çözümü Kuhn-Tucker koşullarını temel alır. Kuhn-Tucker koşulları Lagrange çarpanları yönteminin eşitsizlik sınırlamalarına genişletilmesidir. Bu koşullar sağlanmadığında optimum bir çözüm bulunamaz. Kuadratik programlamanın kısmi türevlerini kapsayan bu koşulların kullanılmasıyla kuadratik programlama doğrusal programlama problemi şekline dönüşür ve çözüm elde edilir.

Kuadratik programlama problemlerinin çözümünde Simpleks yöntemine dayanan Kuhn-Tucker koşullarının yanı sıra iç nokta, aktif küme, artırılmış Lagrange, eşlenik gradyan, gradyan izdüşümü ve Simpleks algoritmasının uzantılarına dayanan yöntemler yaygın olarak kullanılır. Temelleri bu yöntemlere dayanan birkaç algoritma aşağıda verilmiştir.

Lagrange Çarpanları Yöntemi

Kısıtları eşitlik halinde yazılabilen kuadratik programlama probleminin çözümü için Lagrange çarpanları yöntemi kullanılabilir. Burada Lagrange fonksiyonu:

$$\boldsymbol{L} = (x_1, x_2, \dots\ x_n, \lambda_1, \lambda_2, \dots\ \lambda_n) = \boldsymbol{f}(x_1, x_2, \dots\ x_n) + \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{\lambda_i}[\boldsymbol{b_i} - \boldsymbol{\rho_i}(x_1, x_2, \dots\ x_n)]$$

şeklinde ifade edilir. Burada $X = (x_1, x_2, \dots\ x_n)$ karar değişkeni ve $\lambda = (\lambda_1, \lambda_2, \dots\ \lambda_n)$ kısıtlara ilişkin Lagrange çarpanıdır.

Lagrange fonksiyonunun x_i ve λ_i' lere göre birinci türevleri alınırsa optimal çözümler elde edilir.

$$G = \left[\frac{\partial \rho_i(x)}{\partial x_j}\right]_{mxn} . rank(G) = m \qquad \begin{array}{l} n \geq m \text{ ve} \\ \text{en az bir } \partial_i \neq 0 \end{array}$$

$$\frac{\partial L}{\partial x_j} = \frac{\partial f(x)}{\partial x_j} - \sum \partial_i \frac{\partial \rho_i(x)}{\partial x_j} = 0 \qquad j = 1, 2, \dots, n$$

$$\frac{\partial L}{\partial x} = b_i - \rho_i(x) = 0 \qquad i = 1, 2, \dots, m$$

x_0 noktasında, modelin maksimum (veya minimum) değeri için gerekli koşullar:

$$G(x_0) = \begin{pmatrix} \frac{\partial \rho_1(x_0)}{\partial x_1} & \frac{\partial \rho_2(x_0)}{\partial x_1} & \dots & \frac{\partial \rho_m(x_0)}{\partial x_1} \\ \vdots & & & \\ \frac{\partial \rho_1(x_0)}{\partial x_n} & \frac{\partial \rho_2(x_0)}{\partial x_n} & \dots & \frac{\partial \rho_m(x_0)}{\partial x_n} \end{pmatrix}_{nxm}$$

ve

$$\boldsymbol{f_{ij}} = \frac{\partial^2 \boldsymbol{f}(x_0, \lambda_0)}{\partial x_i \partial x_j}$$

olmak üzere:

$$h(c) = \begin{vmatrix} (F_{ij})_{nxm} & -cI & g(x_0) \\ G^T(x_0) & 0 & \end{vmatrix} = 0$$

elde edilir ve $h(c) = 0$ denkleminin kökleri bulunur. Elde edilen bütün kökler:

- 0’ dan büyük ise x_0 noktası minimumdur.
- 0’ dan küçük ise, x_0 noktası maksimumdur.

Kuhn-Tucker Koşulları

1940'lı yılların ortalarında Dantzig tarafından doğrusal programlama için Simpleks yönteminin geliştirilmesi matematiksel optimizasyon konusunu başlatmıştır. Konu hakkındaki bir diğer önemli gelişme ise, 1951'de Kuhn ve Tucker tarafından günümüzde Karush-Kuhn-Tucker (KKT) koşulları ya da Kuhn-Tucker (KT) koşulları olarak bilinen doğrusal olmayan programlama problemi

için gerekli/yeterli optimallik koşullarını sağlayan yöntemin geliştirilmesidir.

Kuhn-Tucker koşulları, Lagrange çarpanları yönteminin özel bir şekli ya da Simpleks yönteminin bir uzantısı olarak tanımlanabilir.

Aşağıda doğrusal olmayan programın genel hali verilmiştir:

$$Z_{max} = f(x_1, x_2, \dots\ x_n)$$

$$q_i(x_1, x_2, \dots\ x_n) \leq 0$$

$$i = 1,2, \dots, m$$

Bu modeldeki kısıtları eşitlik haline getirmek için aylak değişkenler (negatif olmayan) kullanılabilir. Yeni model:

$$Z_{max} = f(x_1, x_2, \dots\ x_n)$$

$$q_i(x_1, x_2, \dots\ x_n) + S^2 = 0$$

şeklinde yazılır. Bu model için Lagrange fonksiyonu:

$$L(A, S, \lambda) = f(x) - \lambda[\rho(x) + S^2]$$

şeklinde ifade edilir. En iyi noktayı bulmak için Lagrange fonksiyonunda yer alan değişkenlere göre türev alınır. Bu koşullar:

I $$\frac{\partial L(X,S,\lambda)}{\partial x} = \frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial \rho_i(x)}{\partial x_j} = 0$$

II $$\frac{\partial L(X,S,\lambda)}{\partial S} = -2\lambda S = 0$$

III $$\frac{\partial L(X,S,\lambda)}{\partial \lambda} = -[\rho(x) + S^2] = 0$$

Doğrusal olmayan programlama modeli maksimizasyon amaçlı olduğundan dolayı $\lambda_i \geq 0$ olmalıdır. Bu durumda yukarıdaki eşitlikler incelendiğinde;

$\lambda_i > 0$ olduğunda, $S_i^2 = 0$ ve $q_i(x) = 0$ ya da $S_i^2 > 0$ ve $\lambda_i = 0$ ise en iyi çözüm $q_i(x) < 0$ koşulunda gerçekleşir (Taha, 2000).

Sonuç olarak, $\lambda_i q_i(x) = 0$ eşitliği elde edilir.

Bu durumda modelin optimal çözümünü elde etmek için gerekli Kuhn-Tucker koşulları:

$$\begin{aligned} \lambda_i &\geq 0 && i = 1, 2, \dots, m \\ \frac{\partial f(x)}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial \rho_i(x)}{\partial x_j} &= 0 && j = 1, 2, \dots, n \\ \lambda_i \rho_i &= 0 && i = 1, 2, \dots, m \\ \rho_i(x) &\leq 0 && 1, 2, \dots, m \end{aligned}$$

şeklinde ifade edilir.

Bu fonksiyonlar ile çözüm elde edilmek istendiğinde, negatif olmayan boş değişkenler kullanılır. Bu değişkenler, eşitsizlikleri eşitlik haline getirmeyi amaçlar ve kısıtlardaki kullanılmayan kısmı (fazlalığı) temsil eder. $y \in \mathcal{R}^n$ ve $v \in \mathcal{R}^m$ artık değişkenleri modele eklendiğinde:

$$c^T + Qx + A^T \mu^T - y = 0$$

$$Ax - b + v = 0$$

Eşitlikler düzenlendiğinde Kuhn-Tucker koşulları aşağıdaki gibi yazılır:

$$Qx + A^T \mu^T - y = -c^T$$

$$Ax + v = b$$

$$x \geq 0, \mu \geq 0, y \geq 0, v \geq 0$$

$$y^T x = 0, \mu v = 0$$

Kuadratik programlama problemlerinin çözümünde sıklıkla kullanılan bu yöntem, optimal çözümü bulma terminolojisine dayanmaktadır. Problemin çözümü için aşağıdaki adımlar izlenerek optimal çözüm elde edilebilir (Jensen & Bard, 2003):

Kısıtlar, Kuhn-Tucker koşullarını sağlayacak şekilde eşitlik haline getirilir.

Adım 1. Eşitlik negatif olması durumunda -1 ile çarpılır.

Adım 2. Minimizasyon problemleri için yapay değişkenler kullanılır. Bu değişkenler, amaç fonksiyonuna da eklenir.

Adım 3. Sonuçlar, Simpleks tablosuna işlenir.

Optimal çözüm için bu adımlar izlendiğinde amaç fonksiyonu pozitif belirli ise model oldukça iyi sonuç vermektedir. Ancak, pozitif yarı belirli amaç fonksiyonları için yapılan hesaplamalarda güçlüklerle karşılaşılabilinmektedir (Jensen & Bard, 2003).

Örnek *(Kubat & Uygun, 2023):*

Amaç fonksiyonu:

$$Max\ Z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

Kısıtlar:

$$x_1 + 2x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

Problemin matris formu:

$$Max\ Z = (4,6)\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + (x_1, x_2)\begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Kısıtlar:

$$(1,2)\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq 2$$

$$x_1, x_2 \geq 0$$

Bu durumda Kuhn-Tucker şartları:

$$(1,2)\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + S_1 = 2$$

$$Lagrange\ çarpanı: \lambda_1$$

$$x_1 - \mu_1 + R_1 = 0$$

$$x_2 - \mu_2 + R_2 = 0$$

$$\begin{bmatrix} 4 & 2 & 1 & -1 & 0 & 0 \\ 2 & 4 & 2 & 0 & -1 & 0 \\ 1 & 2 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \mu_1 \\ \mu_2 \\ S_1 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 2 \end{bmatrix}$$

$Min\, r = R_1 + R_2 = 0$ olduğunda optimal sonuç elde edilir. Optimallik için $r = 0$ olması amaçlanır.

Tablo 1. Başlangıç tablosu

temel	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	S_1	çözüm
r	6	6	3	-1	-1	0	0	0	10
R_1	4	2	1	-1	0	1	0	0	4
R_2	2	4	2	0	-1	0	1	0	6
S_1	1	2	0	0	0	0	0	1	2

Tablo 1'de $R_1 = 4$ daha büyük bir yapay değişken olduğundan dolayı x_1 temele giren, R_1 çıkan değişken olur.

Tablo 2. Birinci adım

temel	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	S_1	çözüm
r	0	3	$^3/_2$	$^1/_2$	-1	$-^3/_2$	0	0	4
x_1	1	$^1/_2$	$^1/_4$	$-^1/_4$	0	$^1/_4$	0	0	1
R_2	0	3	$^3/_2$	$^1/_2$	-1	$-^1/_2$	1	0	4
S_1	0	$^3/_2$	$-^1/_4$	$^1/_4$	0	$-^1/_4$	0	1	1

Tablo 2'de x_2 değişkeni temele girerken, S_1 çıkar.

Tablo 3. İkinci adım

temel	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	S_1	çözüm
r	0	0	2	0	-1	-1	0	-2	2
x_1	1	0	$^1/_3$	$-^1/_3$	0	$^1/_3$	0	$-^1/_3$	$^2/_3$
R_2	0	0	2	0	-1	0	1	-2	2
x_2	0	1	$-^1/_6$	$^1/_6$	0	$-^1/_6$	0	$^2/_3$	$^2/_3$

Tablo 3'te λ_1 değişkeni temele girer, R_2 çıkar.

Tablo 4. Üçüncü adım

temel	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	S_1	çözüm
r	0	0	0	0	0	-1	$-1/6$	0	0
x_1	1	0	0	$-1/3$	$1/6$	$1/3$	$-1/6$	0	$1/3$
λ_1	0	0	1	0	$-1/2$	0	$1/2$	-1	1
x_2	0	1	0	$1/6$	$-1/12$	$-1/6$	$1/12$	$1/2$	$5/6$

Tablo 4'te $r = 0$ olduğunda optimum çözümü verir:

$$x_1^* = 1/3,\ x_2^* = 5/6,\ \lambda_1^* = 1$$

Z ' nin optimum değeri, bulunan değerler amaç fonksiyonunda yerine konarak hesaplanır.

$$Z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

$$Z = 4(1/3) + 6(5/6) - 2(1/3)^2 - 2(1/3)(5/6) - 2(5/6)^2$$

$$Z = 25/6$$

Wolfe Yöntemi

Wolfe yöntemi, kuadratik programlama problemlerinin çözümünde önde gelen metotlardan biridir. Bu yöntem ikinci dereceden bir programlama problemini çözmek için Philip Wolfe (1959) tarafından geliştirilmiştir. Bu yöntem ile kuadratik programlama problemleri modifiye edilmiş Simpleks yöntemi kullanılarak çözülür. Yöntemde, problemin iki optimum çözüme sahip olduğu varsayılır.

Amaç fonksiyonu:

$$Mak\ Z = f(x) = \sum_{j=1}^{n} c_j x_j + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} c_{jk} x_j x_k$$

Kısıtlar:

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_j$$

$$x_j \geq 0$$

$$b_j \geq 0, \forall b_j$$

$\sum_{j=1}^{n}\sum_{k=1}^{n} c_{jk}x_j x_k$ yarı negatif tanımlı

$i = 1, 2, \dots, m, j = 1, 2, \dots, n$ ve $c_{jk} = c_{kj}$

Yukarıda amaç fonksiyonu ve kısıtları verilen problemin optimal çözümüne ulaşmak için aşağıdaki adımlar izlenir:

Adım 1. i. kısıttaki $i = 1, 2, \dots, m$ q_1^2 değişkenleri ve j. negatif olmayan kısıttaki $j = 1, 2, \dots, n$ r_j^2 değişkenlerine gevşeklik eklenerek eşitsizlik kısıtları denklemlere dönüştürülür.

Adım 2. Lagrange çarpanları fonksiyonu yazılır:

$$L(x, q, r, \lambda, \mu) = f(x) - \sum_{i=2}^{m} \lambda_i \left[\sum_{i=1}^{n} a_{ij}x_j - b_i + q_i^2\right] - \sum_{j=1}^{n} \mu_j\left[-x_j + r_j^2\right]$$

Burada:

$$x = (x_1, x_2, \dots\ x_n)$$

$$q = (q_1^2, q_2^2, \dots, q_m^2)$$

$$q = (r_1^2, r_2^2, \dots, r_n^2)$$

$$\lambda = (\lambda_1, \lambda_2, \dots\ \lambda_m)$$

$$\mu = (\mu_1, \mu_2, \dots\ \mu_n)$$

L' nin x, q, r, λ ve μ değişkenlerine göre kısmi türevi alınarak ve birinci dereceden kısmi türevleri sıfıra eşitlenerek Kuhn-Tucker koşulları elde edilir.

Model negatif olmayan yapay değişken v_j $(j = 1, 2, \dots, n)$ eklenerek Kuhn-Tucker koşullarına göre yazılır.

$$c_j + \sum_{k=1}^{n} c_{jk} x_k - \sum_{i=1}^{m} a_{ij} \lambda_i + \mu_j = 0$$

$$Z_v = v_1 + v_2 + v_n$$

$$i = 1, 2, \dots, m, \; j = 1, 2, \dots, n$$

Adım 3. $Z_v = v_1 + v_2 + v_n$ eşitliği altında doğrusal programlamanın başlangıç temel uygun çözümü elde edilir.

$$\sum_{k=1}^{n} x_k c_{jk} - \sum_{k=1}^{n} c_{jk} x_k - \sum_{k=1}^{n} a_{ij} \lambda_i + \mu_j + v_j = -c$$

$$\sum_{j=1}^{n} a_{ij} x_j + q_i^2 = b_i$$

$$v_j, \lambda_i, \mu_j, x_j \geq 0$$

$$i = 1, 2, \dots, m, \; j = 1, 2, \dots, n$$

Gevşetmenin sağlanması:

$$\sum_{j=1}^{n} \mu_j x_j + \sum_{i=1}^{m} \lambda_i S_i = 0$$

$$S_i = q_i^2$$

$$\lambda_i S_i = 0$$

$$\mu_j x_j = 0$$

$$i = 1, 2, \dots, m, \; j = 1, 2, \dots, n$$

Adım 4. Lineer denklemin optimum çözümünü bulmak için 2-aşamalı Simpleks yöntemi uygulanır. Adım 4'teki programlama probleminin çözümü, tanımlanan gevşeklik koşulunu sağlamalıdır.

Adım 5. 5. adımda elde edilen optimum çözüm, kuadratik programlama problemi için elde edilebilecek en uygun çözümdür.

Beale Yöntemi

Beale, klasik kuadratik programlama çözüm yöntemlerinden yeni bir yöntem geliştirmiştir. Bu yöntem, eşitliğin kısıtlı olduğu problemlere uygulanır ve temelde Simpleks yöntemini esas alarak azaltılmış gradyan yaklaşımını kullanır (Beale, 1955; Beale, 1959).

Yöntemin ilk aşamasında, herhangi bir genelliği kaybetmeden uygun bir çözümün mevcut olduğu varsayılır. Simpleks yönteminin birinci aşaması, bu yöntem için ilk adım kabul edilebilir. Önceki yöntemlerde olduğu gibi, orijinal problemde herhangi bir eşitsizlik olması durumunda, bunlar uygun gevşek değişkenler eklenerek eşitliğe dönüştürülür.

Beale yöntemi ile uygun çözüm aranırken ikinci dereceden formda herhangi bir kısıtlama getirilmez. Yani Beale yöntemi, amaç fonksiyonunun dışbükey (veya maksimizasyon durumunda içbükey) olmasına gerek duyulmadan herhangi bir ikinci dereceden programlama problemine uygulanabilir.

Bu algoritma, sabit yük problemi gibi amaç fonksiyonunun dış bükeylik durumunun bilinmediği problemlerin çözümü için kullanılabilir. Ancak global minimum ya da yerel minimum üretilmeyebilir. Beale yöntemi için kısaca aşağıdaki adımlar uygulanır:

Adım 1. Temel değişkenler ve amaç fonksiyonu temel olmayan değişkenler cinsinden ifade edilir.

Adım 2. Temel olmayan değişkenlerden hangisinin amaç fonksiyonu değerinde en büyük düşüşü sağlayacağı belirlenir.

Adım 3. Belirli temel olmayan değişken tanımlandıktan sonra, temel olmayan değişkenin negatif olmama kısıtlamalarını

ihlal etmeden artabileceği mümkün olan en büyük değer belirlenir.

Adım 4. Temel olmayan değişken temel hale gelene kadar çözüm optimallik açısından test edilir. Çözüm optimal değilse süreç tekrarlanır.

Adım 5. Optimal çözüm elde edildiğinde süreç tamamlanır.

Algoritma, dışbükey olmayan problemlerde yerel minimum elde edilebilmesi için iki ek kurala sahiptir (Beale, 1959):

1. Verilen (ya da dönüştürülen) ikinci dereceden problem köşegen dışı terimler içeriyorsa, doğrusal terimler sıfır olsa bile, serbest bir değişken temel olmayan kümeden çıkarılmalıdır.
2. Verilen (ya da dönüştürülen) ikinci dereceden problem köşegen dışı terimler içermiyor ancak köşegen terimler pozitif değilse, serbest bir değişken temel olmayan kümeden çıkarılmalıdır. Bu durumda değişken, bazı temel değişkenler (serbest değişken değil) sıfıra eşit oluncaya kadar artırılabilir veya azaltılabilir.

Bu iki kuralın uygulanması ile algoritma, son elde edilen ifadede doğrusal terimde ortadan kaybolan kısıtlanmış bir değişken olmadığı sürece yerel bir minimum nokta üretir. Bu durumda, amaç fonksiyonunun katsayılarında keyfi olarak küçük bir değişiklik yapılarak bu nokta yerel minimum yapılabilir.

Dantzig Yöntemi

1961'de G. B. Dantzig tarafından tanıtılan bu yöntem, uygun çözüm elde etmek amacıyla Simpleks yöntemine benzer bir tablo kullanır. Genel olarak bu yöntem, Simpleks yönteminin bir varyasyonu olarak düşünülebilir. Yöntem, Simpleks tekniği kullanılarak ve "standart" ve "standart dışı" tablo dizileri üzerinden ilerleyerek geliştirilmiştir. İşleyişte temel giriş ve çıkış kuralları, Kuhn-Tucker koşullarına koşullarını da karşılayan en uygun noktayı elde etmek için uygun şekilde değiştirilir.

Dantzig yöntemi, yalnızca eşitsizlik kısıtlamaları olan dışbükey problemlere uygulanabilir (Byrne, 1984). Ayrıca bu algoritma Q'nun pozitif tanımlı olduğu problemlerle sınırlı değildir, yani Q matrisinin yarı tanımlı olduğu ikinci dereceden programlama problemlerini çözmek için de kullanılabilir (Vankova, 2004).

Diğer Yöntemler

Thiel ve Panne (1960) ve Lemke (1962); amaç fonksiyonunun kolayca elde edilen kısıtlanmamış minimumundan başlayan ve daha sonra sadece bir x^* noktası bulunacak şekilde sadece kısıtlamaları sağlamakla kalmayan aynı zamanda amaç fonksiyonu için bir minimum da veren yöntemler üretmişlerdir. Thiel ve Panne' nin yöntemi, her yinelemede önceki tüm bilgileri göz ardı ettiği için Lemke' ninkinden daha fazla çalışma gerektirmektedir. Her iki yöntem de yalnızca $f^0(x)$' in kesinlikle dışbükey olması durumunda uygulanabilir.

Goldfarb (1966, 1968, 1972) ve Fletcher (1971) tarafından geliştirilen yöntemler ise, kısıtlanmamış ikinci dereceden bir fonksiyonu en aza indirmek için Newton yönteminin uzantıları olarak düşünülebilir. Yöntemler yararlı sonuçlar verir ancak hesaplama açısından karmaşıktırlar. Best ve Ritter (1976) ve Gill ve Murray (1978) tarafından bu yaklaşımların daha geliştirilmiş varyasyonları rapor edilmiştir.

Fizibilite sorununu ortadan kaldıran başka bir algoritma, ikinci dereceden programın ikilisi üzerinde Goldfarb'ın (1972) temel uygulanabilir nokta algoritmasını kullanan Goldfarb ve Idnani'nin (1983) ikili yaklaşımıdır. Başlangıçta ikili mümkün nokta kolaylıkla elde edilir (kısıtlanmamış minimum) ve ikili fizibilite kolaylıkla korunur. Açıklama ve uygulama, özellikle temel sorun açısından verildiği için biraz karmaşıktır. Powell (1983) yöntemin pozitif yarı kesin duruma genişletildiğini bildirmiştir ve oldukça başarılı sayısal sonuçlar elde edildiğini rapor etmiştir.

Bu yöntemlerin problemler üzerine uygulanmasında, öncelikle kısıtları sağlayan bir başlangıç uygun çözümünün bulunması

gerekmektedir. Bu durum çoğunlukla çok fazla hesaplama gerektirmektedir. Son yıllarda bu durumun önüne geçmek için yeni yöntemler araştırılmıştır.

Silva, Verdegay ve Yamakami (2007) kısıtlamalar kümesinde belirsizlik bulunan ikinci dereceden programlama problemlerinin bir sınıfını çözen orijinal ve yeni bulanık kümelere dayalı bir yöntem sunmuşlardır. Yöntem, portföy bağlamında değerlendirilebilecek bulanık ikinci dereceden programlama problemlerini çözmek için iki aşama kullanmaktadır. İlk aşamada bulanık problem farklı kesme seviyelerine sahip çeşitli klasik alfa problemlerine parametrelendirilir. İkinci aşamada bu alfa problemlerinin her biri geleneksel çözme teknikleri kullanılarak çözülür. Önceki problemin nihai bulanık çözümü, bu özel alfa çözümlerinin tümünün entegre edilmesiyle elde edilir.

Mirmohseni ve Nasseri (2017) kısıtlama katsayılarının ve sağ tarafların tamamının üçgensel bulanık sayılar olduğu bulanık kuadratik programlama probleminin bulanık amaç değerini türetmek için yeni bir yaklaşım önermişlerdir.

Saber ve Sulaiman (2022), amaç fonksiyonunun tek doğrusal kısıtlamalı iki doğrusal faktörün çarpımı olarak yazılabildiği ikinci dereceden programlama problemini çözmek için dinamik programlama yaklaşımını tanımlamışlardır.

Guo ve ark. (2023) ikinci dereceden programlama problemleri için aktif küme yönteminden esinlenerek yeni bir yinelemeli algoritma geliştirmişlerdir. Bu yöntemde her yinelemede, mevcut yineleme noktasının aktif kümesine göre ikinci dereceden bir programlama alt problemi oluşturulur ve çözülür. Doğrusal olmayan programlama için Karush-Kuhn-Tucker koşuluna dayanarak dikkate alınan problemin optimallik koşulu verilir ve bu, oluşturulan algoritmanın optimal bir çözümde sonlanmasını garanti eder. Bu yeni algoritmanın, önceden sunulan iki diğer yaklaşımlara göre (problemi doğrusal kısıtlara sahip sıradan bir ikinci dereceden programlama problemine dönüştürmek ve kısıtlamaları polinom fonksiyonlarıyla yumuşatmak ve probleme sıradan düzgün doğrusal

olmayan programlamayla yaklaşmak) büyük ölçekli problemler için çalışma süresi ve hesaplama doğruluğu açısından bariz bir üstünlüğü olduğu rapor edilmiştir.

R ile Kuadratik Programlama

Kuadratik programlama problemleri R'deki "quadprog" paketi ile çözülebilir. Problemin çözümündeki kilit nokta, problemin matris gösterimini bulmaktır.

Üç değişkenli bir problem için genel bir matris gösterimi aşağıdaki gibi tanımlanır:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$X^T = [x_1 \quad x_2 \quad x_3]$$

$$D = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix}$$

$$d^T = [d_1 \quad d_2 \quad d_3]$$

$$min_X \frac{1}{2} X^T D X - d^T X$$

$$= \frac{1}{2} [x_1 \quad x_2 \quad x_3] \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$- [d_1 \quad d_2 \quad d_3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$= \frac{1}{2}(d_{11}x_1^2 + d_{22}x_2^2 + dx_3^2 + (d_{12}+d_{21})x_1x_2 + (d_{13}+d_{31})x_1x_3 + (d_{23}+d_{32})x_2x_3) - d_1x_1 - d_2x_2 - d_3x_3$$

$$d_{12} = d_{21};\ d_{13} = d_{31};\ d_{23} = d_{32}$$

Kısıtlar matrisi için eşitlik kısıtları ilk sıraya yazılmalı ve eşitsizlik kısıtları "≥" şeklinde yeniden yazılmalıdır.

'quadprog' Paket Kullanımı:

$library(quadprog)$
$solve.QP(Dmat, dvec, Amat, bvec, meq)$
$Dmat$: *minimize edilecek fonksiyondaki matris*
$dvec$: *minimize edilecek fonksiyondaki vektör*
$Amat$: *kısıtları tanımlayan matris*
$bvec$: b_0 *değerlerini tutan vektör*
meq: *kısıtlardaki eşitlik (=) sayısı*

Kısıtlar matrisi için eşitlik kısıtları (=) ilk sıraya yazılmalı ve eşitsizlik kısıtları "≥" şeklinde yeniden yazılmalıdır.

R' da bir kuadratik programlama problemini çözülürken dikkat edilmesi gereken bazı hususlar vardır:

- Problem minimizasyon problemi olarak ifade edilmelidir.
- Kısıtların tamamı ≥ şeklinde yazılmalıdır.
- Eşitlik içeren kısıtlar ilk sıraya yazılmalıdır.
- Varsa, sabit terimin varlığı çözüm sonucuna eklenmelidir.
- Elde edilen sonuç, problemin maksimizasyon ya da minimizasyon problemi olup olmamasına göre yorumlanmalıdır.

Amaç fonksiyonu ve kısıtları verilen iki değişkenli bir kuadratik programlama probleminin R programı ile çözümü için izlenecek adımlar aşağıda gösterilmektedir.

Problem:

Amaç fonksiyonu:

$$\text{Max } Z = 2x_1 + 2x_2 - x_1^2 - x_1x_2 - x_2^2 + 2$$

Kısıtlar:

$$2x_1 + 2x_2 \geq 6$$

$$x_1, x_2 \geq 0$$

Çözüm:

Amaç fonksiyonu:

$$\text{Min Z} = -2x_1 - 2x_2 + x_1^2 + x_1x_2 + x_2^2 - 2$$

Kısıtlar:

$$2x_1 + 2x_2 \geq 6$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Matris notasyonları:

$$D = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$d^T = [2 \quad 2]$$

$$A^T = \begin{bmatrix} 2 & 2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$$

R Kodları:

```
Dmat <- matrix(c(2,1,1,2),2,2)
dvec <- c(2,2)
Amat <- t(matrix(c(2,1,0,2,0,1),3,2))
bvec <- c(6,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Çıktı:

$value	[1] 0.75
$unconstrained.solution	[1] 0.6666667 0.6666667
$iterations	[1] 2 0
$Lagrangian	[1] 1.25 0.00 0.00
$iact	[1] 1

$x_1 = x_2 = 0.6666667$ olduğunda kuadratik programlama problemi minimuma indirilir. Burada -2 sabiti unutulmamalıdır. Bu durumda sonuç $0.75 - 2 = -1.25$ olarak yazılır. Başlangıçta amaç fonksiyonu minimizasyon problemi olarak yeniden yazılmıştır. Bu nedenle sonuçlar buna göre yorumlanmalıdır. Orjinal kuadratik programlama problemini maksimuma çıkarmak için $x_1 = x_2 = 0.6666667$ olması durumunda, maksimize edilen değer 1.25 olacaktır.

Lingo ile Kuadratik Programlama

Lingo programı yardımıyla kuadratik programlama problemlerinin çözümü aşağıda verilen örnek problem ile açıklanmıştır.

Problem:

Amaç fonksiyonu:

$$\text{Max } Z = 2x_1 + 2x_2 - x_1^2 - x_1x_2 - x_2^2 + 2$$

Kısıtlar:

$$2x_1 + 2x_2 \geq 6$$

$$x_1, x_2 \geq 0$$

Çözüm:

Lingo Kodları:

```
MAX = 2 * X1 + 2 * X2 - X1^2 - X1 * X2 - X2^2 + 2;
2 * X1 + 2 * X2 >= 6;
X1 > 0;
X2 > 0;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		1.250000
Infeasibilities:		$0.1217653E-08$
Total solver iterations:		6
Elapsed runtime seconds:		0.06
Variable	*Value*	*Reduced Cost*
$X1$	1.5000019	0.000000
$X2$	1.499981	$-0.2414107E-08$
Row	*Slack or Surplus*	*Dual Price*
1	1.250000	1.000000
2	$-0.5114798E-08$	-1.250000
3	1.500019	0.000000
4	1.499981	0.000000

Burada $x_1 = 1.5000019$ ve $x_2 = 1.499981$ olduğunda kuadratik programlama problemi için maksimize edilen değer 1.250000 olacaktır.

R ve Lingo ile Nümerik Problem Çözümleri

Bu bölümde, nümerik şekilde ifade edilen farklı kuadratik programlama problemleri R ve Lingo programları aracılığıyla çözülmüştür. Çözümler için programlara girilen kodlar ve çıktılar raporlanmıştır. Problem 1-5 Lokhande, Khot ve Khobragade (2017); Problem 6-8 Ghadle ve Pawar (2015); Problem 9-10 Singh (2012) tarafından yapılan çalışmalardan alınmıştır. Problemlerin algoritma çözümleri için ilgili kaynaklar incelenebilir.

Problem-1

Amaç fonksiyonu:

$$\text{Max } Z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

Kısıtlar:

$$x_1 + 2x_2 \geq 2$$

$$x_1, x_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(4,2,2,4),2,2)
dvec <- c(4,6)
Amat <- t(matrix(c(1,1,0,2,0,1),3,2))
bvec <- c(2,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Çıktı:

$value	[1] −4.666667
$unconstrained.solution	[1] 0.3333333 1.3333333
$iterations	[1] 1 0
$Lagrangian	[1] 0 0 0
$iact	[1] 0

Yorum: $x_1 = 0.3333333$ ve $x_2 = 1.3333333$ olduğunda kuadratik programlama problemi minimuma indirilir. Başlangıçta amaç fonksiyonu minimizasyon problemi olarak yeniden yazılmıştır. Bu nedenle orjinal kuadratik programlama problemi için maksimum değer 4.666667 olarak elde edilir.

Lingo Kodları:

```
MAX = 4*X1 + 6*X2 - 2*X1^2 - 2*X1*X2 - 2*X2^2;
X1 + 2*X2 >= 2;
X1 > 0;
X2 > 0;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		4.666667
Infeasibilities:		0.000000
Total solver iterations:		4
Elapsed runtime seconds:		0.05
Variable	*Value*	*Reduced Cost*
X1	0.3333333	0.000000
X2	1.333333	0.000000
Row	*Slack or Surplus*	*Dual Price*
1	4.666667	1.000000
2	1.000000	0.000000
3	0.3333333	0.000000
4	1.333333	0.000000

Yorum: $x_1 = 0.3333333$ ve $x_2 = 1.3333333$ olduğunda kuadratik programlama problemi maksimuma çıkarılır. Bu durumda problem için maksimum değer 4.666667 olarak elde edilir.

Problem-2

Amaç fonksiyonu:

$$\text{Max } Z = 8x_1 + 10x_2 - x_1^2 - x_2^2$$

Kısıtlar:

$$3x_1 + 2x_2 \leq 6$$

$$x_1, x_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(2,0,0,2),2,2)
dvec <- c(8,10)
Amat <- t(matrix(c(-3,1,0,-2,0,1),3,2))
bvec <- c(-6,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Çıktı:

$value	[1] −21.30769
$unconstrained.solution	[1]4 5
$iterations	[1] 2 0
$Lagrangian	[1] 2.461538 0.00 0.00
$iact	[1] 1

Yorum: $x_1 = 4$ ve $x_2 = 5$ olduğunda kuadratik programlama problemi minimuma indirilir. Başlangıçta amaç fonksiyonu minimizasyon problemi olarak yeniden yazılmıştır. Bu nedenle orjinal kuadratik programlama problemi için maksimum değer 21.30769 olarak elde edilir.

Lingo Kodları:

```
MAX = 8 * X1 + 10 * X2 - X1^2 - X2^2;
3 * X1 + 2 * X2 <= 6;
X1 > 0;
X2 > 0;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		21.30769
Infeasibilities:		$0.1626531E-08$
Total solver iterations:		9
Elapsed runtime seconds:		0.05
Variable	*Value*	*Reduced Cost*
$X1$	0.3076748	$0.3004826E-07$
$X2$	2.538488	$-0.3577086E-08$
Row	*Slack or Surplus*	*Dual Price*
1	21.30769	1.000000
2	$-0.9754023E-08$	2.461538
3	0.3076748	0.000000
4	2.538488	0.000000

Yorum: $x_1 = 0.3076748$ ve $x_2 = 2.538488$ olduğunda kuadratik programlama problemi maksimuma çıkarılır. Bu durumda problem için maksimum değer 21.30769 olarak elde edilir.

Problem-3

Amaç fonksiyonu:

$$\text{Max } Z = 2x_1 + 3x_2 - 2x_1^2$$

Kısıtlar:

$$x_1 + 4x_2 \leq 4$$
$$x_1 + x_2 \leq 2$$
$$x_1, x_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(4,0,0,0.0000001),2,2)
dvec <- c(2,3)
Amat <- t(matrix(c(-1,-1,1,0,-4,-1,0,1),4,2))
bvec <- c(-4,-2,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Dmat matrisinde kareselliği sağlamak için 0 değeri, 0.0000001 olarak alınmıştır.

Çıktı:

$value	[1] −3.191064
$unconstrained.solution	[1] 0.5 300.0
$iterations	[1] 2 0
$Lagrangian	[1] 0.7476957 0.00 0.00 0.00
$iact	[1] 1

Yorum: $x_1 = 0.5$ ve $x_2 = 300.0$ olduğunda kuadratik programlama problemi minimuma indirilir. Başlangıçta amaç fonksiyonu minimizasyon problemi olarak yeniden yazılmıştır. Bu nedenle orjinal kuadratik programlama problemi için maksimum değer 3.191064 olarak elde edilir.

Lingo Kodları:

```
MAX = 2*X1 + 3*X2 - 2*X1^2;
X1 + 4*X2 <= 4;
X1 + X2 <= 2;
X1 > 0 ;
X2 > 0 ;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		3.195312
Infeasibilities:		$0.8978438E-08$
Total solver iterations:		5
Elapsed runtime seconds:		0.06
Variable	*Value*	*Reduced Cost*
X1	0.3125000	0.000000
X2	0.9218750	0.000000
Row	*Slack or Surplus*	*Dual Price*

1	3.195312	1.000000
2	$0.1743774E-07$	0.750000
3	0.7656250	$-0.5297208E-08$
4	0.3125000	$-0.7609481E-07$
5	0.9218750	$0.3951347E-07$

Yorum: $x_1 = 0.3125000$ ve $x_2 = 0.9218750$ olduğunda kuadratik programlama problemi maksimuma çıkarılır. Bu durumda problem için maksimum değer 3.195312 olarak elde edilir.

Problem-4

Amaç fonksiyonu:

$$\text{Max } Z = 6x_1 + 3x_2 - 4x_1x_2 - 2x_1^2 - 3x_2^2$$

Kısıtlar:

$$x_1 + x_2 \leq 1$$

$$2x_1 + 3x_2 \leq 4$$

$$x_1, x_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(4,4,4,6),2,2)
dvec <- c(6,3)
Amat <- t(matrix(c(-1,-2,1,0,-1,-3,0,1),4,2))
bvec <- c(-1,-4,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Çıktı:

$value	[1] −4
$unconstrained.solution	[1]3.0 −1.5
$iterations	[1] 3 0
$Lagrangian	[1] 2 0 0 3
$iact	[1] 4 1

Yorum: $x_1 = 3.0$ ve $x_2 = -1.5$ olduğunda kuadratik programlama problemi minimuma indirilir. Başlangıçta amaç fonksiyonu minimizasyon problemi olarak yeniden yazılmıştır. Bu nedenle orjinal kuadratik programlama problemi için maksimum değer 4 olarak elde edilir.

Lingo Kodları:

```
MAX = 6 * X1 + 3 * X2 - 4 * X1 * X2 - 2 * X1^2 - 3 * X2^2;
X1 + X2 <= 1;
2 * X1 + 3 * X2 <= 4;
X1 > 0 ;
X2 > 0 ;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		4.000000
Infeasibilities:		0.000000
Total solver iterations:		7
Elapsed runtime seconds:		0.05
Variable	*Value*	*Reduced Cost*
X1	1.000000	0.000000
X2	0.000000	3.000001
Row	*Slack or Surplus*	*Dual Price*
1	4.000000	1.000000
2	0.000000	1.999996
3	2.000000	0.000000
4	1.000000	0.000000
5	0.000000	0.000000

Yorum: $x_1 = 1.000000$ ve $x_2 = 0.000000$ olduğunda kuadratik programlama problemi maksimuma çıkarılır. Bu durumda problem için maksimum değer 4.000000 olarak elde edilir.

Problem-5

Amaç fonksiyonu:

$$\text{Max } Z = 2x_1 + x_2 - x_1^2$$

Kısıtlar:

$$2x_1 + 3x_2 \leq 6$$
$$2x_1 + x_2 \leq 4$$
$$x_1, x_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(2,0,0,0.0000001),2,2)
dvec <- c(2,1)
Amat <- t(matrix(c(-2,-2,1,0,-3,-1,0,1),4,2))
bvec <- c(-6,-4,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Dmat matrisinde kareselliği sağlamak için 0 değeri, 0.0000001 olarak alınmıştır.

Çıktı:

$value	[1] −2.444444
$unconstrained.solution	[1] 1e + 00 1e + 07
$iterations	[1] 2 0
$Lagrangian	[1] 0.333333 0.00 0.00 0.00
$iact	[1] 1

Yorum: $x_1 = 1e + 00$ ve $x_2 = 1e + 07$ olduğunda kuadratik programlama problemi minimuma indirilir. Başlangıçta amaç fonksiyonu minimizasyon problemi olarak yeniden yazılmıştır. Bu nedenle orjinal kuadratik programlama problemi için maksimum değer 2.444444 olarak elde edilir.

Lingo Kodları:

```
MAX = 2*X1 + X2 - X1^2;
2*X1 + 3*X2 <= 6;
2*X1 + X2 <= 4;
X1 > 0 ;
X2 > 0 ;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		2.444444
Infeasibilities:		0.000000
Total solver iterations:		4
Elapsed runtime seconds:		0.06
Variable	*Value*	*Reduced Cost*
$X1$	0.6666667	0.000000
$X2$	1.555556	0.000000
Row	*Slack or Surplus*	*Dual Price*
1	2.444444	1.000000
2	0.000000	0.3333333
3	1.111111	0.000000
4	0.6666667	0.000000
5	1.555556	0.000000

Yorum: $x_1 = 0.6666667$ ve $x_2 = 1.555556$ olduğunda kuadratik programlama problemi maksimuma çıkarılır. Bu durumda problem için maksimum değer 2.444444 olarak elde edilir.

Problem-6

Amaç fonksiyonu:

$$\text{Max Z} = 4\text{x}_1 + 2\text{x}_2 - \text{x}_1^2 - \text{x}_2^2 - 5$$

Kısıtlar:

$$\text{x}_1 + \text{x}_2 \leq 4$$

$$\text{x}_1, \text{x}_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(2,0,0,2),2,2)
dvec <- c(4,2)
Amat <- t(matrix(c(-1,1,0,-1,0,1),3,2))
bvec <- c(-4,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Çıktı:

$value	[1] −5
$unconstrained.solution	[1] 2 1
$iterations	[1] 1 0
$Lagrangian	[1] 0 0 0
$iact	[1] 0

Yorum: $x_1 = 2$ ve $x_2 = 1$ olduğunda kuadratik programlama problemi minimuma indirilir. Burada -5 sabiti unutulmamalıdır. Başlangıçta amaç fonksiyonu minimizasyon problemi olarak yeniden yazılmıştır. Bu durumda sonuç $-5 + 5 = 0$ olarak yazılır. Bu nedenle orjinal kuadratik programlama problemini maksimuma çıkarmak için maksimize edilen değer 0 olacaktır.

Lingo Kodları:

```
MAX = 4*X1 + 2*X2 - X1^2 - X2^2 - 5;
X1 + X2 <= 4;
X1 > 0 ;
X2 > 0 ;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		0.000000
Infeasibilities:		0.000000
Total solver iterations:		4
Elapsed runtime seconds:		0.05
Variable	*Value*	*Reduced Cost*
X1	2.000000	0.000000
X2	1.000000	0.000000
Row	*Slack or Surplus*	*Dual Price*
1	0.000000	1.000000
2	1.000000	0.000000
3	2.000000	0.000000
4	1.000000	0.000000

Yorum: $x_1 = 2.000000$ ve $x_2 = 1.000000$ olduğunda kuadratik programlama problemi maksimuma çıkarılır. Bu durumda problem için maksimum değer 0.000000 olarak elde edilir.

Problem-7

Amaç fonksiyonu:

$$\text{Min Z} = 6 - 6x_1 + 2x_1^2 - 2x_1x_2 + 2x_2^2$$

Kısıtlar:

$$x_1 + x_2 \leq 2$$

$$x_1, x_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(4,-2,-2,4),2,2)
dvec <- c(6,0)
Amat <- t(matrix(c(-1,1,0,-1,0,1),3,2))
bvec <- c(-2,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Çıktı:

$value	[1] −5.5
$unconstrained.solution	[1] 2 1
$iterations	[1] 2 0
$Lagrangian	[1] 1 0 0
$iact	[1] 1

Yorum: $x_1 = 2$ ve $x_2 = 1$ olduğunda kuadratik programlama problemi minimuma indirilir. Burada 6 sabiti unutulmamalıdır. Bu durumda problem için minimum değer $-5.5 + 6 = 0.50$ olarak elde edilir.

Lingo Kodları:

```
MIN = 6 − 6 * X1 + 2 * X1^2 − 2 * X1 * X2 + 2 * X2^2;
X1 + X2 <= 2;
X1 > 0 ;
X2 > 0 ;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		0.500000
Infeasibilities:		0.000000
Total solver iterations:		6
Elapsed runtime seconds:		0.05
Variable	*Value*	*Reduced Cost*
$X1$	1.500001	0.000000
$X2$	0.49999996	0.000000
Row	*Slack or Surplus*	*Dual Price*
1	0.500000	-1.000000
2	$0.2483473E-08$	1.000000
3	1.500001	0.000000
4	0.49999996	0.000000

Yorum: $x_1 = 1.500001$ ve $x_2 = 0.49999996$ olduğunda kuadratik programlama problemi minimuma indirilir. Bu durumda problem için minimum değer 0.500000 olarak elde edilir.

Problem-8

Amaç fonksiyonu:

$$\text{Min Z} = -4\text{x}_1 + \text{x}_1^2 - 2\text{x}_1\text{x}_2 + 2\text{x}_2^2$$

Kısıtlar:

$$2\text{x}_1 + \text{x}_2 \leq 6$$

$$\text{x}_1 - 4\text{x}_2 \leq 0$$

$$\text{x}_1, \text{x}_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(2,-2,-2,4),2,2)
dvec <- c(4,0)
Amat <- t(matrix(c(-2,-1,1,0,-1,4,0,1),4,2))
bvec <- c(-6,0,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Çıktı:

$value	[1] −6.769231
$unconstrained.solution	[1] 4 2
$iterations	[1] 2 0
$Lagrangian	[1] 0.6153846 0.00 0.00 0.00
$iact	[1] 1

Yorum: $x_1 = 4$ ve $x_2 = 2$ olduğunda kuadratik programlama problemi minimuma indirilir. Bu durumda problem için minimum değer -6.769231 olarak elde edilir.

Lingo Kodları:

```
MIN = −4 ∗ X1 + X1^2 − 2 ∗ X1 ∗ X2 + 2 ∗ X2^2;
2 ∗ X1 + X2 <= 6;
X1 > 0 ;
X2 > 0 ;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		−6.769231
Infeasibilities:		$0.8300403E-08$
Total solver iterations:		6
Elapsed runtime seconds:		0.05
Variable	*Value*	*Reduced Cost*
X1	2.461549	$-0.6996467E-08$
X2	1.076902	$-0.1095227E-07$
Row	*Slack or Surplus*	*Dual Price*
1	−6.769231	−1.000000
2	$0.6229068E-07$	0.6153846
3	2.461549	0.000000
4	1.076902	0.000000

Yorum: $x_1 = 2.461549$ ve $x_2 = 1.076902$ olduğunda kuadratik programlama problemi minimuma indirilir. Bu durumda problem için minimum değer -6.769231 olarak elde edilir.

Problem-9

Amaç fonksiyonu:

$$\text{Min } Z = x_1^2 + x_2^2$$

Kısıtlar:

$$x_1 + x_2 \geq 4$$

$$2x_1 + 2x_2 \geq 5$$

$$x_1, x_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(2,0,0,2),2,2)
dvec <- c(0,0)
Amat <- t(matrix(c(1,2,1,0,1,1,0,1),4,2))
bvec <- c(4,5,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Çıktı:

$value	[1] 8
$unconstrained.solution	[1] 2 2
$iterations	[1] 2 0
$Lagrangian	[1] 4 0 0 0
$iact	[1] 1

Yorum: $x_1 = 2$ ve $x_2 = 2$ olduğunda kuadratik programlama problemi minimuma indirilir. Bu durumda problem için minimum değer 8 olarak elde edilir.

Lingo Kodları:

```
MIN = X1^2 + X2^2;
X1 + X2 >= 4;
2 * X1 + 2 * X2 >= 5;
X1 > 0 ;
X2 > 0 ;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		8.000000
Infeasibilities:		$0.4042662E-08$
Total solver iterations:		4
Elapsed runtime seconds:		0.05
Variable	*Value*	*Reduced Cost*
$X1$	2.000000	$-0.6460022E-08$
$X2$	2.000000	$-0.6460022E-08$
Row	*Slack or Surplus*	*Dual Price*
1	8.000000	-1.000000
2	$0.8852850E-08$	-4.000000
3	3.000000	0.000000
4	2.000000	0.000000
5	2.000000	0.000000

Yorum: $x_1 = 2$ ve $x_2 = 2$ olduğunda kuadratik programlama problemi minimuma indirilir. Bu durumda problem için minimum değer 8.000000 olarak elde edilir.

Problem-10

Amaç fonksiyonu:

$$\text{Min Z} = 8\text{x}_1 + 10\text{x}_2 - 2\text{x}_1^2 - \text{x}_2^2$$

Kısıtlar:

$$3\text{x}_1 + 2\text{x}_2 \leq 6$$

$$\text{x}_1, \text{x}_2 \geq 0$$

R Kodları:

```
Dmat <- matrix(c(4,0,0,2),2,2)
dvec <- c(8,10)
Amat <- t(matrix(c(-3,1,0,-2,0,1),3,2))
bvec <- c(-6,0,0)
solve.QP(Dmat,dvec,Amat,bvec)
```

Çıktı:

$value	[1] −21.23529
$unconstrained.solution	[1] 2 5
$iterations	[1] 2 0
$Lagrangian	[1] 2.352941 0.00 0.00
$iact	[1] 1

Yorum: $x_1 = 2$ ve $x_2 = 5$ olduğunda kuadratik programlama problemi minimuma indirilir. Bu durumda problem için minimum değer -21.23529 olarak elde edilir.

Lingo Kodları:

```
MIN = −8X1 − 10X2 + 2 ∗ X1^2 + X2^2;
3 ∗ X1 + 2 ∗ X2 <= 6;
X1 > 0 ;
X2 > 0 ;
END
```

Çıktı:

Global optimal solution found.		
Objective value:		−21.23529
Infeasibilities:		$0.2838962E-08$
Total solver iterations:		9
Elapsed runtime seconds:		0.05
Variable	*Value*	*Reduced Cost*
X1	0.2352796	$0.3626311E-07$
X2	2.647081	$-0.3774450E-08$
Row	*Slack or Surplus*	*Dual Price*
1	−21.23529	−1.000000
2	$-0.8621495E-08$	2.352941
3	0.2352796	0.000000
4	2.647081	0.000000

Yorum: $x_1 = 0.2352796$ ve $x_2 = 2.647081$ olduğunda kuadratik programlama problemi minimuma indirilir. Bu durumda problem için minimum değer -21.23529 olarak elde edilir.

Sonuç ve Tartışma

Kuadratik programlama ile ilgili yapılan çalışmalar literatürde daha çok yeni algoritma geliştirilmesi yönündedir. Bu nedenle çalışmada en bilinen birkaç algoritma tanıtılarak çözüm yöntemleri üzerinde durulmuştur. Bu tür problemler için Lingo, Lindo, Matlap, GAMs ve AMPL gibi farklı programlar kullanılarak çözüm elde edilebilir.

Son yıllarda yapılan çalışmalarda farklı optimizasyon problemlerinin çözümünde R programı da yaygın olarak kullanılmaktadır. Bu çalışmada farklı kısıt ve değişken sayısına sahip 10 farklı örnek problem için R programı ile optimal çözümler elde edilmiştir.

Elde edilen çözümler Lingo programı ile aynı sonuçları vermiştir. Böylece farklı büyüklükteki problemlerin çözümü için R programı kullanılarak boyut sorunu olmadan problemler çözülebilir.

R programı son yıllarda istatistik bilimi dışında diğer pek çok alanda da sıklıkla tercih edilmektedir. Akademik çalışmalarda R programının kuadratik programlama problemleri için kullanıldığı literatürden bilinmektedir. Ancak gerçek hayat uygulamalarında ya da bu konuda yeni çalışmaya başlamış araştırmacılar tarafından daha az kullanıldığı görülmüştür. Bu çalışma ile doğrusal olmayan optimizasyon problemleri için kuadratik programlama yöntemini tanıtmak ve Lingo ve R programları yardımıyla örnek problemlerin okuyucuya açıklanması amaçlanmıştır. Bu amaç doğrultusunda örnekler için yazılan program kodları açıklamaları ile birlikte sunulmuştur. Çalışmanın özellikle konu ile yeni çalışmaya başlayan araştırmacılara ve öğrencilere yardımcı olacağı düşünülmektedir.

KAYNAKÇA

Abele, G. (2015). Quadratic programming problems. *Ilmenau University of Technology*. Web. 23 May 2015.

Beale, E. M. L. (1955). On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17, 173-184.

Beale, E. M. L. (1959). On quadratic programming. *Naval Research Logistics Quarterly*, 6 (3), 227-243.

Best, M. J. & Ritter K. (1976). An effective algorithm for quadratic minimization problems. *MRC Technical Summary Report 1691*, Mathematics Research Centre, University of Wisconsin (Madison, WI, 1976).

Byrne, S. N. (1984). Solution of quadratic programming problems. *NZOR*, 12 (2), 73-89.

Dantzig, G. B. (1961). Quadratic programming - a variant on the Wolf e-Markowitz algorithm. *2. Operations Research Centre*. University of California, Berkeley.

Dantzig, G. B. (I940). On the non-existence of tests of "Student's" hypothesis having power functions independent of σ. *Annual Mathematics Statistics*, 11, 186-192.

Demirel, K. (2009). *Kuadratik programlama ile portföy seçimi*. Yüksek Lisans Tezi, Marmara Üniversitesi, İstanbul.

Fletcher, R. (1971). A general quadratic programming algorithm. AERE Harwell Report T.P., *Journal of the Institute of Mathematics and its Applications* (401).

Frank, M. & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3 (1–2), 95-110.

Ghadle, K. P. & Pawar, T. S. (2015). New approach for Wolfe's modified Simplex method to solve quadratic programming

problems. *International Journal of Research in Engineering and Technology*, 4 (1), 371-376.

Gill, P. E. & Murray, W. (1978). *Numerically stable methods for quadratic programming*. Mathematical Programming, 14, 349-372.

Goldfarb D. & Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27, 1-33.

Goldfarb, D. (1966). *A conjugate gradient method for non-linear programming*. Ph. D. Thesis, Princeton University.

Goldfarb, D. (1968). Analogs of Newton's method for quadratic programming. *Notices of the American. Mathematics Society*, 15 (2) 400.

Goldfarb, D. (1972). Extensions of Newton's method and simplex methods for solving quadratic programs. Lootsma, F.A., *Numerical methods for non-linear optimization* (239-254), London: Academic Press.

Guo, F., Li, J., Shen, J. & Wang, J. (2023). A new algorithm for solving quadratic programming problems subject to addition-min fuzzy relation inequalities. Research Square, 1-29.

Hadley, G. (1970). *Nonlineer and dynamic programming*. London: Addison Wesley.

Hasan, M. B. (2012). A technique for solving special type quadratic programming problems. *Dhaka University Journal of Science*, 60 (2), 209-215.

Hillier, F. S. & Lieberman, G. J. (1980). *Introduction to operations research*. (Third edition). California: Holden-Day.

https://www.scribd.com/document/441470015/Karesel-quadratic-programlama Erişim Tarihi: 03.10.2023

Jensen, P. & Bard, J. (2003). *Operations research models and methods*. USA: John Wiley and Sons.

Karush, W. (1939). *Minima of functions of several variables with inequalities as side conditions*. Master's Thesis, Department of Mathematics, University of Chicago.

Kubat, C. & Uygun, Ö. (2023) *Yöneylem Araştırması*. Ders Notları: Hafta 08.

Kuhn, H. W. & Tucker, A. W. (1951). Nonlinear programming. *Proceedings of 2nd Berkeley Symposium on Mathematics, Statistics and Probability*. Berkeley: University of California Press, 481-492.

Lemke, C. E. (1962). A method of solution for quadratic programs. *Management Science*, 8 (4), 442-453.

Li W. & Tian, X. (2008). Numerical solution method for general interval quadratic programming, *Applied Mathematics and Computation*, 202 (2), 589-595.

Liu S.T. & Wang, R.T. (2007). A numerical solution method to interval quadratic programming, *Applied Mathematics and Computation*, 189 (2), 1274-1281.

Lokhande, K., Khot, P. G. & Khobragade N. W. (2017). Optimum solution of quadratic programming problem: by Wolfe's modified Simplex method. *International Journal of Latest Technology in Engineering, Management & Applied Science*, 6 (3), 11-19.

McCarl, B., Moskowitz, H. & Harley, F. (1977). Quadratic Programming Applications. *The International Journal of Management Science*, 5, 43-55.

Mirmohseni, S. & Nasseri, S. (2017). A quadratic programming with triangular fuzzy numbers. *Journal of Applied Mathematics and Physics*, 5, 2218-2227.

Moore, R. E. (1966). *Interval Analysis*, USA: Prentice-Hall.

Powell, M. J. D. (1983). *ZQPCVX: A Fortran subroutine for convex quadratic programming*. Report DAMTP/NA17,

Department of Applied Mathematics and Theoretical Physics, University of Cambridge.

Saber, N. & Sulaiman, N. (2022). Solving quadratic programming problem via dynamic programming approach. *International Journal of Nonlinear Analysis and Applications*, 13 (2), 473-478.

Silva, R. C., Verdegay J. L. & Yamakami, A. (2007). Two-phase method to solve fuzzy quadratic programming problems. *International Fuzzy Systems Conference*, London, UK, 1-6.

Singh, R. (2012). Optimization methods and quadratic programming. Master of Thesis, National Institute of Echnology, Rourkela Odisha, India.

Taha, H. A. (2000). *Yöneylem Araştırması*. (6. Basımdan çeviri), İstanbul: Litaratür Yayıncılık.

Terlaky, T. (1987). A new algorithm for quadratic programming. *European Journal of Operational Research*, 32 (2), 294-301.

Thiel, H. & Panne, V. C. (1960). Quadratic programming as an extension of classical quadratic maximization. *Management Science*, 7 (1), 1-20.

Vankova, M. (2004). *Algorithms for the solution of the quadratic programming problem*. The Degree of Magister Scientiae, University of Port Elizabeth.

Wolfe, P. (1959). The simplex method for quadratic programming. *Econometrica*, 27, 382-398.

BÖLÜM III

Metin Sınıflandırma Yöntemleri: Türkçe Uygulamalar ve İngilizce Modellerin Adaptasyonu Üzerine Kapsamlı Bir İnceleme

Halil İbrahim OKUR[1]
Kadir TOHMA[2]
Ahmet SERTBAŞ[3]

Giriş

Günümüzde, özellikle sosyal medya, web, iş dünyası ve eğitim gibi alanlarda sürekli artan metin ve belge birikimi, doğal dil işleme (NLP) alanındaki araştırmacıların dikkatini çeken önemli bir sorun haline gelmiştir. Bu geniş metin yığınlarından anlamlı bilgilerin çıkarılması ve belgelerin etkin bir şekilde sınıflandırılması, hem zahmetli hem de zaman alıcı bir süreçtir. Doğal dil işleme, yapay

[1] Arş. Gör., İskenderun Teknik Üniversitesi
[2] Arş. Gör. Dr., İskenderun Teknik Üniversitesi
[3] Prof., İstanbul Üniversitesi - Cerrahpaşa

zekanın bir alt dalı olarak, bu meydan okumayı ele almakta ve belgeleri, dilin özelliklerine dayalı olarak, otomatik olarak sınıflandırabilen sistemler geliştirmekte büyük rol oynamaktadır. Bu sistemler, metinlerin içeriğine göre sınıflandırma işlemi gerçekleştirirken, dil özelliklerini dikkate alarak daha etkin ve kullanıcı dostu sonuçlar üretmektedir. Metin sınıflandırma probleminde, klasik makine öğrenmesi algoritmalarının yanı sıra, son yıllarda derin öğrenme algoritmaları üzerine yapılan araştırmalar da büyük ilgi görmektedir, bu da alandaki gelişmeleri hızlandırmaktadır (Kaur & Bathla, 2018) (Li & ark.,, 2022).

Metin sınıflandırma problemi, benzer içerik ve bilgiye sahip belgelerin tespit edilerek listelenmesi veya sıralanması süreci olarak tanımlanabilir. Her bir doküman, içerdiği etiket bilgisi ile karakterize edilir. Bu etiket bilgisi, dokümanın özetini, başlığını veya dokümanlar arasındaki atıfları temsil eden bilgileri içerebilir. Ayrıca, metinlerin duygusal analizi de dokümanın belirli bir sınıfa veya etikete atanmasında kritik bir rol oynar. Bu süreç, dokümanların içeriğinin derinlemesine incelenmesini ve anlamlı bir şekilde sınıflandırılmasını sağlar (Han & ark.,, 2022).

Metin sınıflandırma, cümleler, paragraflar ve dokümanlar gibi çeşitli metin birimlerini etiketleyerek sınıflandırma amacı güden, doğal dil işlemenin temel problemlerinden biridir. Doğal dillerin yapısal, bölgesel, kültürel ve dönemsel çeşitlilikleri, dilin kendine has gramer kuralları da dahil olmak üzere, uygulanacak metin işleme tekniklerinin değişkenlik göstermesine neden olmaktadır. Bu bağlamda, seçilen dilin morfolojik yapısına uygun bir metin işleme yaklaşımı benimsenmelidir. Ahonen ve arkadaşlarının çalışmasına göre, kelimelerin söz dizimsel ve anlamsal özellikleri, cümle ve ifadelerin oluşturulmasında kritik bir role sahiptir. Cümleler arasındaki ilişkiler paragrafları, paragraflar arasındaki ilişkiler ise bütüncül metinleri meydana getirir (Ahonen & ark., 1997). Metinler, insanların düşüncelerini ve bilgilerini yazılı bir formatta aktarmalarını sağlar, bu sayede kişiler arası bilgi alışverişi ve etkili iletişim mümkün olur. Doğal dilin insanlar tarafından anlaşılır yapısını bilgisayar sistemlerine entegre etmek, insan-bilgisayar

etkileşiminin literatürde uzun yıllardır incelenen bir konusudur. İnsan ile bilgisayar arasındaki bu etkileşim; metinlerden bilgi çıkarımı, özetleme, soru-cevap sistemleri, metin sınıflandırma gibi çeşitli alanlara ayrılmaktadır. Mooney ve Roddick tarafından yapılan çalışmada belirtildiği üzere, metinlerin, yapay zeka sistemlerinin işleyebileceği bir formata dönüştürülmesi ve belirli ön işlem aşamalarından geçirilmesi gerekmektedir. Bu ön işlem aşamaları, yapay zeka sistemlerinin performansını önemli ölçüde etkilemektedir (Mooney & Roddick, 2013).

Resmi kaynakların dışında, özellikle sosyal medya gibi konuşma diline daha yakın platformlarda oluşturulan metinler, hızlı haberleşme arzusunun etkisiyle dilin sözdizimsel kurallarına tam olarak uymayabilir. Justicia ve arkadaşlarına ait çalışmada belirtildiği üzere, bu tür metinlerde yer alan kelimeler, sözlükteki orijinal hallerinden değiştirilmiş olabilir, hatta anlamsız ifadeler içerebilir. Metin işleme sürecinde ilk adım olarak tokenizasyon (metni parçalara ayırma) işlemi gerçekleştirilir, ardından bu tür içerikler temizlenmeli veya düzeltilmelidir. Ayrıca, işlenmesi gereksiz olan veya kelime olmayan simgeler ve işaretleri içeren istenmeyen token'lar da metinden çıkarılmalıdır. Metin sınıflandırma problemi bağlamında, dilin sık kullanılan kelimeleri ("stop words") yapay zeka yöntemleri için belirleyici bir etkiye sahip olmadığından, bu kelimelerin metinden çıkarılması gereklidir. Metnin parçalanması, kelimelerin düzeltilmesi ve fazlalıklarının temizlenmesi işlemlerinden sonra, her kelimenin eklerinden arındırılması ve kök kelimesinin elde edilmesi önemlidir. Bu yaklaşım, aynı köke sahip farklı kelimelerin farklı token olarak değerlendirilmemesini sağlar. (Justicia & ark., 2018)

Doğal dil işleme alanında, metnin detaylı bir ön işlemden geçirilmesi ve ardından vektör formuna dönüştürülmesi, daha etkin ve güvenilir bir indeksleme modelinin oluşturulmasına olanak tanır. Vijayarani ve arkadaşları tarafından da belirtilen bu yaklaşım, metinlerin yapay zeka algoritmaları tarafından anlaşılabilir bir forma sokulmasını sağlar, böylece sınıflandırma işlemi için metinler uygun hale gelir. Bu süreç, doğal dil işlemenin temel bileşenlerinden biri

olarak, metin tabanlı verilerin etkin kullanımını mümkün kılar(Vijayarani & ark., 2015).

Bu çalışmada, metin sınıflandırmasının her aşaması üzerinde literatürde yapılmış çeşitli çalışmalar ele alınmıştır. Bu bağlamda, çeşitli kullanım amaçlarına göre sınıflandırılan metin veri setleri, metinlerin ön işleme süreçleri, indeksleme yöntemleri ve metin sınıflandırma teknikleri incelenmiştir. Özellikle Türk dili morfolojisine uygun olarak metin sınıflandırma aşamalarında kullanılan modeller üzerinde durulmuş, bu modellerin detayları ve üzerinde çalışılacak olan yeni bir model sunulmuştur. Bu çalışmanın temel amacı, İngilizce dil özelliklerine göre yapılandırılmış modellerin Türkçe üzerinde uygulanmasına yönelik araştırmaları tanıtmak ve bu alanda ne tür gelişmelerin yapılabileceğine dair fikirler sunmaktır. Bu sayede, dil farklılıklarının doğal dil işleme modellerinin performansına etkileri daha iyi anlaşılabilir ve bu alandaki araştırmalar daha da ileriye taşınabilir.

Metin ön işleme

Metin sınıflandırma modellerinin performansını artıran önemli bir unsur, veri setlerinin kapsamlı bir ön işlemden geçirilmesidir, bu durum yapay zeka modellerinin genelinde de geçerlidir. Bu süreç metin veri setinin gürültüsüz hale getirilmesi, yazım hatalarının düzeltilmesi, argo ifadeler ve kısaltmaların standardize edilmesi, metinlerin bütünlük oluşturacak şekilde küçük veya büyük harfe dönüştürülmesi gibi adımları içerir. Ayrıca, metinlerin küçük parçalara ayrılma (tokenizasyon) işlemi, her kelimenin alt parçalarına ayrılarak köklerine indirgenmesi (stemming ve lemmatization) önemlidir. Bunun yanı sıra, modelin performansını olumsuz etkileyebilecek, dilde sık kullanılan ve genellikle anlamsal ağırlık taşımayan etkisiz kelimeler (stop words) olarak adlandırılan kelimelerin metin verisinden çıkarılması da bu ön işlem aşamasının kritik bir parçasıdır. Bu ön işlem adımları, metin sınıflandırma modellerinin daha doğru ve etkili çalışmasını sağlamakta önemli bir role sahiptir.

Metin veri setleri, sosyal medya, web içerikleri, iş dünyası kaynakları ve kullanıcı yorumları gibi birçok farklı platformdan elde edilebilir. Bu metinler, kaynaklarına bağlı olarak dil bilgisine uygunluk açısından farklılıklar gösterebilir. Yazı dili ile konuşma dili arasındaki farklılıklar göz önünde bulundurulduğunda, yazı dilinde dil bilgisi kurallarına uyumun daha yüksek olduğu görülür. Sosyal medyada yazılan metinlerin, konuşma diline daha yakın olması nedeniyle dilbilgisi hatalarını içerme olasılığı daha yüksektir. İş dünyasında veya resmi yazışmalarda kullanılan metinler, dil bilgisi kurallarına daha sıkı bir şekilde uyulduğundan, genellikle daha az hata içerir. Metinlerin kaynağına göre temizlenme süreçleri de farklılık gösterir. Sosyal medya metinlerinde, dilin özelliklerine uymayan kısaltmalar, argo ifadeler veya mantıksız kelimeler kullanılabilir. Bu tür metinlerin sınıflandırılması için ön işleme aşamasında bu tür ifadelerin düzeltilmesi veya temizlenmesi gerekebilir. Ancak, daha resmi bir ortamda hazırlanan metinler için bu tür ön işleme adımlarının sonuç üzerinde önemli bir etkisi olmayabilir.

Metinlerin oluşturulduğu ortamın yanı sıra, sınıflandırma probleminin çözümü için kullanılacak modelin geliştirilmesi aşamasında da metin ön işleme süreçleri çeşitlilik arz edebilir. Bu, modelin amacına ve uygulanacağı alana göre ön işleme metodolojilerinin özelleştirilmesini gerektirir. Metinlerin doğası ve modelin gereksinimleri, ön işleme adımlarının yapısını ve kapsamını belirleyen temel etmenlerdir. Dolayısıyla, sınıflandırma modelinin etkinliği ve doğruluğu, bu özelleştirilmiş ön işleme tekniklerine büyük ölçüde bağlıdır.

Metin temizleme

Metin ve belge veri setleri, genellikle gereksiz kelimeler içerebilir ve bu, özellikle istatistiksel ve olasılık temelli öğrenme algoritmalarında sistemin performansını olumsuz yönde etkileyebilir. Bu bölümde, metin temizleme ve ön işlem tekniklerine genel bir bakış sunulacaktır.

Metinlerin küçük parçalara ayrılması (tokenization): Metnin kelime, sembol, ifade veya diğer anlamlı unsurlar gibi daha küçük parçalara, yani tokenlara ayrılması işlemidir. Hem metin sınıflandırması hem de metin madenciliği için belgelerin bu şekilde ayrıştırılması gereklidir.

Örnek olarak, "Bin atın varsa bin dinlen, bir atın varsa in dinlen." cümlesi tokenlara ayrıldığında {"Bin", "atın", "varsa", "bin", "dinlen", "bir", "atın", "varsa", "in", "dinlen"} şeklinde bir dizi oluşturur.

Metinlerin tokenlara ayrılma sürecinde, kelimenin kendisi, kökü, morfolojik yapısı veya alt parçaları (karakter seviyesinde parçalama) gibi farklı yaklaşımlar kullanılabilir. Al Nahas ve arkadaşları çalışmasında, Türkçe metinler üzerinde farklı tokenizasyon tekniklerini kullanarak sinir ağı temelli sınıflandırma işlemini gerçekleştirmiş ve bu farklı tekniklerin performans üzerindeki etkilerini incelemiştir. (Al Nahas & ark., 2020)

Etkisiz kelimelerin (stop words) temizlenmesi: Dilin sık kullanılan kelimelerinin model performansı üzerindeki etkisini azaltmak amacıyla metin veri setinden çıkarılması işlemidir. Örneğin, Türkçede sık kullanılan "bir", "ve", "için" gibi kelimeler bu kategoriye girer.

Büyük-küçük harf duyarlılığı (capitalization): Metin içerisindeki tutarsızlıkları gidermek için kelime veya küçük parça metinlerin (token) büyük-küçük harf durumlarının standart bir forma getirilmesidir. Bu, metin analizindeki tutarlılığı artırır.

Argo ve kısaltmalar (slang and abbreviation): Argo ve kısaltmalar, metin ön işleme sürecinde ele alınması gereken diğer anormalliklerdir. Örneğin, “Destek Vektör Makinesi (Support Vector Machine)” teriminin kısaltması olan "DVM (SVM)" gibi kısaltmalar metin içerisinde bulunabilir. Ek olarak argo ifadeler ise gayri resmi konuşma dilinde kullanılan sözcüklerdir. Bu tür sözcüklerle başa çıkmanın yaygın yöntemi, onları resmi dile dönüştürmektir.

Gürültü temizleme (noise removal): Metinlerdeki noktalama işaretleri ve özel karakterler gibi gereksiz unsurların çıkarılmasıdır. Bu unsurlar insanlar için anlamlı olmasına rağmen, sınıflandırma algoritmaları için zararlı olabilir.

Yazım hatalarını düzeltme (spelling correction): Yazım hatalarının düzeltilmesi, özellikle sosyal medya metinlerinde sıkça rastlanan bir durumdur. Doğal dil işleme **(**NLP) alanında bu sorunu ele almak için birçok algoritma ve teknik mevcuttur (Oflazer, 1994) ve (Bölücü, 2019).

Eklerden arındırma (stemming): Doğal dil işlemede (NLP), aynı anlama sahip olan ancak farklı formlarda bulunan kelimelerin, temel formuna indirgenmesi işlemidir. Örneğin, Türkçe'de "adaletli" kelimesinin eksiz hali "adalet"tir. İngilizce metinler için Snowball stemmer (Porter, 2001) ve NLTK kütüphanesi (Perkins, 2010), Türkçe metinler için ise Snowball'ın Türkçe versiyonu (Çilden, 2006) ve Zemberek Kütüphanesi (Akın, 2007) kullanılabilir.

Kök bulma (lemmatization): Bir kelimenin temel biçimini (lemma) elde etmek amacıyla son eklerin değiştirilmesi veya tamamen kaldırılması işlemine dayanan bir NLP sürecidir. Bu işlem, kelimenin kökünü saptama aşamasını içermektedir. NLTK ve Zemberek kütüphaneleri, bu işlemler için yaygın olarak kullanılan araçlardır.

Örneğin, Kılınç ve arkadaşları tarafından gerçekleştirilen çalışmada, TTC-3600 Türkçe metin sınıflandırma veri seti tanıtılmış ve metinlerin sınıflandırma süreci öncesinde kelime eklerinden arındırılması veya kelime köklerinin bulunması üzerindeki performans değerlendirilmiştir. Çalışmada, metinleri oluşturan kelimeler farklı şekillerde işlenmiştir: ham formlarıyla, frekansı koruyan örnekleme (FPS - Frequency-Preserving Sampling) ile kelime boyutunun azaltılmasıyla veya zemberek kütüphanesi kullanılarak kelimelerin köklerinin bulunmasıyla. Bu farklı ön işleme yöntemlerinin metin sınıflandırma performanslarına etkileri incelenmiştir. Bu tür çalışmalar, kök bulma ve eklerden arındırılma

aşamaları metin sınıflandırma işlemlerindeki önemini ve etkisini vurgulamaktadır (Kılınç & ark., 2017).

Morfolojik yapılar: Morfoloji, dil biliminin bir dalı olarak, kelimelerin çeşitli biçimlerini ve kelime türlerini analiz eden, aynı zamanda sözcük yapısını da içeren bir biçim bilgisidir (Karaağaç, 2013). Bu alan, özellikle morfolojik açıdan zengin dillerde, dilin yapısının derinlemesine anlaşılmasında önemli bir rol oynar.

Güngör ve ark. tarafından yürütülen çalışmada, morfolojik analizin adlandırılmış varlık tanıma bağlamında incelenmesi amaçlanmıştır. Bu çalışmada, çeşitli dillerdeki veriler aynı doğal dil modeline tabi tutularak, kelime ve karakter düzeyinde gömme (embedding) ile birlikte morfolojik gömme kullanılarak kelime temsilleri oluşturulmuştur. Türkçe, Çekçe, Macarca, Fince ve İspanyolca gibi diller üzerinde yapılan deneyler, bu dillere ait morfolojik yapıların karşılaştırmalı analizini sunmuştur. Bu tür çalışmalar, farklı dillerin morfolojik özelliklerinin doğal dil işleme modellerine entegrasyonunun önemini ve bu entegrasyonun dil modellemesi üzerindeki etkisini göstermektedir (Güngör ve ark., 2019).

Kelime temsil yöntemleri

Araştırmacılar, kelimeler arasındaki kaybolan söz dizimsel ve anlamsal ilişkileri çözmek için metin özellik çıkarma teknikleri üzerine çalışmalar yapmaktadır. Bu alandaki araştırmalar, yeni tekniklerin geliştirilmesine odaklanmaktadır, ancak bu tekniklerin çoğu hâlâ bazı sınırlamalara sahiptir (Chalmers, 1992) (Li, 2017).

Kelime torbası (bag of words (BoW)): Kelime torbası modeli (BoW), metnin belirli bölümlerinden kelime sıklığı gibi kriterlere dayalı olarak metin belgesinin azaltılmış ve basitleştirilmiş bir temsilidir. BoW tekniği, doğal dil işleme (NLP), spam filtreleri, belge sınıflandırması ve bilgi alımı gibi çeşitli alanlarda kullanılmaktadır. BoW modelinde, bir belge veya cümle bir kelime torbası olarak düşünülür ve bu süreçte kelime listeleri oluşturulur. Bu kelimeler, bir matriste yer alır ve kelimeler arasındaki anlamsal

ilişki bu yapılandırmada genellikle göz ardı edilir. Kelimelerin sıklığı sayılır ve bu bilgi, belgelerin ana odak noktalarını belirlemek için kullanılabilir. Bu modelde dilbilgisi ve kelimelerin sıralaması göz ardı edilirken, kelime sıklığı ön plana çıkar (Zhang, 2010). Bu yaklaşım, metin analizinde önemli bir rol oynamakta ve metin içeriğinin özetlenmesi ve sınıflandırılması için kullanılan temel tekniklerden biri olarak kabul edilmektedir.

Ağırlıklı sözcük çıkarımı ve terim frekansı (term frequency-ınverse document frequency (TF)): Ağırlıklı sözcük çıkarımı, öznitelik çıkarımının temel bir biçimidir ve her sözcüğün içinde geçtiği sayıya karşılık gelen bir değere eşlendiği Terim Frekansı (TF) yöntemiyle gerçekleştirilir. Bu yöntemde, her belge, içerdiği sözcüklerin sıklığını yansıtan bir vektöre dönüştürülür. Bu yaklaşım, dilde yaygın kullanılan bazı kelimelerin aşırı temsil edilmesi sorununa yol açabilir.

Ters belge frekansı (IDF) ve TF-IDF: Yaygın kullanılan kelimelerin etkisini azaltmak için, terim sıklığına ek olarak Ters Belge Frekansı (IDF) yöntemi önerilmiştir. IDF, belgedeki kelimelerin sıklığına göre ağırlık atar ve bu TF-IDF kombinasyonu, Terim Frekansı-Ters Belge Frekansı olarak bilinir. TF-IDF, belgedeki yaygın terimlerin aşırı temsil edilmesi sorununu hafifletmeye çalışır, ancak kelimeler arasındaki ilişkileri açıklayamama gibi bazı eksikliklere sahiptir, çünkü her kelime bağımsız bir indeks olarak sunulur.

Gelişmiş modeller ve kelime gömme: Son yıllarda, kelime gömme (word embedding) gibi daha karmaşık modeller geliştirilmiştir. Bu yöntemler, kelimelerin benzerlikleri ve konuşma etiketlemesi gibi kavramları da içerebilmektedir. Kelime gömme, kelimelerin semantik ilişkilerini ve bağlamlarını dikkate alarak daha zengin ve etkili bir metin temsili sağlar (Qaiser, 2018). Bu gelişmiş modeller, metin analizi ve sınıflandırma alanlarında, kelimeler arasındaki anlamlı ilişkileri ve bağlamları daha iyi yansıtabilen teknikler sunmaktadır.

N-Gram: N-gram tekniği, metin analizinde kullanılan bir yöntemdir ve bir metin kümesindeki kelimelerin sırasına dayalı olarak n adet kelimenin bir araya getirilmesiyle oluşturulan bir kümedir. Bu, doğrudan bir metin temsili olmamakla birlikte, bir metni temsil etmek için kullanılabilecek bir özellik kümesi olarak işlev görür. Kelime torbası modeli (BOW), kelimelerin sırasını göz ardı eden bir metin temsili sağlar ve bu modelin oluşturulması kolaydır. Metin, genellikle yönetilebilir bir boyutta bir vektör aracılığıyla temsil edilir. N-gram, 1-gram, 2-gram ve 3-gram gibi farklı uzunluklardaki kelime gruplarını kullanarak metin özelliklerini çıkarır ve bu sayede 1-grama göre daha fazla bilgi sağlayabilir (Cavnar, 1994).

Örneğin, "Arkadaşım Ahmet günlerdir sınavlara çok sıkı çalışıyor." cümlesi için;

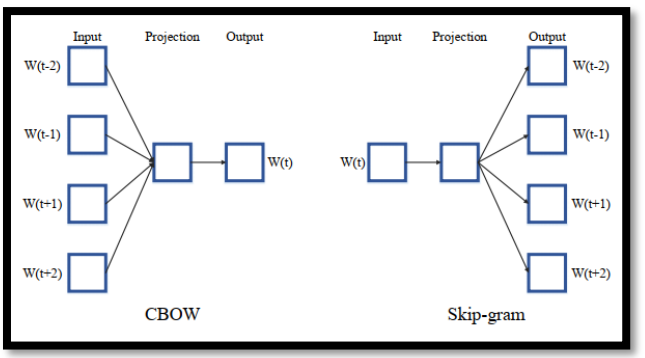
- 2-gram: {" Arkadaşım Ahmet", " Ahmet günlerdir ", " günlerdir sınavlara", " sınavlara çok", " çok sıkı ", " sıkı çalışıyor"}
- 3-gram: {" Arkadaşım Ahmet günlerdir", " Ahmet günlerdir sınavlara", " günlerdir sınavlara çok", " sınavlara çok sıkı", " çok sıkı çalışıyor"}.

Bu örnekler, n-gram tekniğinin, metinlerin daha zengin ve bağlamsal bir şekilde temsil edilmesine olanak tanıdığını gösterir. Kelimeler arasındaki sıra ve bağlamı dikkate alan bu yaklaşım, metin analizinde daha derinlemesine bir anlayış sağlamaktadır.

Kelime Gömme (word embedding): Kelimelerin anlamsal temsillerini oluşturmanın bir yoludur. Kelimelerin söz dizimsel analizleri ve farklı yazılış şekilleri, modelin kelimelerin anlamını tam olarak kavramasını zorlaştırabilir. Aynı zamanda, kelime torbası modelindeki kelimelerin sırasının göz ardı edilmesi gibi sorunlar da vardır. N-gram bu sorunu tamamen çözmemektedir ve bu nedenle, cümle içindeki her kelimenin bir benzerliği bulunmasına ihtiyaç vardır. Bu sorunu çözmek için araştırmacılar, kelime gömme üzerinde çalışmışlardır.

Kelime gömme, kelime dağarcığındaki her kelimenin veya cümlenin gerçek sayılardan oluşan N boyutlu bir vektöre eşlendiği bir özellik öğrenme tekniğidir. Bu teknik, derin öğrenme modellerinde başarıyla kullanılmakta olup, Word2Vec, GloVe ve FastText gibi yöntemler en yaygın kullanılan kelime gömme örnekleridir. Bu yöntemler, kelimelerin bağlamsal anlamlarını ve birbirleriyle olan ilişkilerini daha iyi yakalayarak, metin analizinde ve sınıflandırmada daha yüksek performansa ulaşmayı sağlar. Kelime gömme teknikleri, metin tabanlı verilerin işlenmesinde ve anlamlandırılmasında derin öğrenme yaklaşımlarının temel bileşenleri arasında yer alır.

Word2Vec, etiketsiz (unsupervised) ve tahmine dayalı bir modeldir ve kelimeleri vektör uzayında temsil etmeyi amaçlar. Google araştırmacısı Tomas Mikolov ve ekibi tarafından 2013 yılında geliştirilmiştir (Mikolov, 2013). Word2Vec, her kelime için yüksek boyutlu bir vektör oluşturmak amacıyla iki gizli katman kullanır. Bu model, sürekli kelime torbası (CBOW) ve Skip-gram modelleri üzerine inşa edilmiştir.



Şekil 1. *Word2Vec – CBOW ve Skip-Gram Mimarisi (Mikolov, 2013)*

CBOW ve Skip-Gram, input ve output'un alınma biçimi açısından farklılık gösteren iki ayrı modeldir. CBOW (Continuous Bag of Words) modeli, birden fazla kelimenin input olarak alındığı ve bu kelimelerin merkezinde yer alabilecek bir kelimenin output

olarak tahmin edilmeye çalışıldığı bir yapıdır. Skip-Gram modeli ise, tek bir kelimenin input olarak alınıp bu kelimenin çevresinde yer alabilecek kelimelerin output olarak tahmin edilmeye çalışıldığı bir yapı sunar.

Her iki model de, bir cümle üzerinde uygulanan bu işlemleri tüm cümlelere genişleterek, başlangıçta bulunan etiketsiz verilere mapping işlemi uygulayarak modelin eğitimini gerçekleştirir. Bu modeller, kelimelerin birbirleriyle olan ilişkilerini ve bağlamını yakalayarak, metin tabanlı verilerin daha etkili bir şekilde işlenmesini ve analiz edilmesini sağlar. Word2Vec, doğal dil işlemede kelimelerin vektörel temsillerini elde etmek için kullanılan önemli bir araçtır ve derin öğrenme tekniklerinin temel taşlarından biri olarak kabul edilir. Şekil 1'de bu model için genel mimari sunulmuştur.

GloVe (global vectors for word representation), metin sınıflandırmasında kullanılan etkili bir kelime gömme tekniğidir. Pennington ve ekibi tarafından 2014 yılında geliştirilen bu unsupervised öğrenme algoritması, kelimelerin vektör temsillerini elde etmek için tasarlanmıştır (Pennington & ark., 2014). GloVe modeli, bir korpustan toplanan kelimeler arası küresel birlikte bulunma istatistiklerine dayanarak eğitilir. Bu eğitim süreci sonucunda elde edilen temsiller, kelime vektör uzayını oluşturur. Bu modelde, anlamsal olarak ilişkili kelimelerin vektör temsilleri birbirine yakın konumlandırılır. Örneğin; "kral" ve "kraliçe", "Ankara" ve "şehir", "erkek" ve "kadın" gibi metinler içerisinde anlamsal ilişki içinde olan kelimeler, GloVe model uzayında birbirine yakın yerlerde bulunur. Bu yakınlaşma, kelimelerin anlamsal olarak benzer olduğunu ve metin içerisinde benzer bağlamlarda kullanıldığını gösterir. GloVe, kelime anlamlarının ve bağlamlarının daha derinlemesine anlaşılmasına olanak tanıyan önemli bir kelime gömme yöntemidir ve doğal dil işleme alanında yaygın olarak kullanılmaktadır.

FastText, Facebook AI Araştırma laboratuvarı tarafından 2016 yılında geliştirilmiş, Word2Vec modeline benzer yapısıyla

dikkat çeken bir kelime gömme yöntemidir (Joulin & ark., 2016). Bu teknik, kelimelerin morfolojik özelliklerini de içerecek şekilde tasarlanmıştır. Her kelime, w, bir dizi n-gram karakterli çanta olarak temsil edilir. FastText, kelimeleri tek tek değil, bunları birkaç harflik n-gramlara bölerek yapay sinir ağına girdi olarak sunar. Örneğin, "iyilik" kelimesi tri-gram yapısı kullanılarak analiz edildiğinde, yapay sinir ağına sunulan girdiler "iyi", "yil", "ili" ve "lik" şeklinde olacaktır. Buradaki n, tekrar derecesini ifade eder ve bir kelimenin kaç harflik parçalara bölüneceğini belirler. "iyilik" kelimesinin word vektörü, bu n-gram vektörlerinin toplamı olarak hesaplanır.

Facebook, FastText modelini Wikipedia veri seti üzerinde 300 boyutlu vektörler kullanarak eğitmiş ve 294 dil için önceden eğitilmiş kelime vektörlerini yayınlamıştır. Bu geniş kapsamlı model, kelimelerin anlamını ve yapısını derinlemesine analiz etmek için kullanılır ve doğal dil işlemede önemli bir araç olarak kabul edilir. FastText, özellikle morfolojik olarak zengin dillerde etkili olduğu için, bu dillerin incelenmesinde ve metin işlemede kritik bir rol oynamaktadır.

Bağlamsal kelime temsilleri (contextualized word representations, kelimelerin anlamını ve kullanımını bağlamsal olarak modelleyen ileri düzey bir sözcük gömme tekniğidir. Context2Vec (Melamud, 2016), bu tür bağlamsal kelime temsillerinin geliştirilmesinde kullanılan bir yöntemdir ve çift yönlü Uzun Kısa Süreli Bellek (LSTM) ağlarından yararlanır. Context2Vec, kelimenin hem sözdizimi hem de anlambilim özelliklerini dikkate alır ve ek olarak, bu özelliklerin dilin bağlamı içinde nasıl değiştiğini (yani çok anlamlılığı) modellemeye odaklanır.

Bu teknik, kelimelerin bağlamsal anlamlarını daha derinlemesine anlamak için kullanılır ve kelimelerin farklı bağlamlarda nasıl farklı anlamlar kazanabileceğini gösterir. Context2Vec'in çift yönlü LSTM kullanması, kelimelerin hem önceki hem de sonraki bağlamlarının dikkate alınmasını sağlar, bu da kelime temsillerinin zenginliğini ve doğruluğunu artırır. Bu

yöntem, doğal dil işleme ve metin analizi çalışmalarında önemli bir araç olarak kabul edilmektedir ve kelimelerin bağlamsal temsillerinin daha iyi anlaşılmasına katkıda bulunur.

Boyut azaltma (dimension reduction): Terime dayalı vektör modellerinde metin dizileri genellikle çok sayıda özellik içerir. Bu, zaman karmaşıklığı ve bellek tüketimi açısından pahalı olabilir. Bu sorunu ele almak için araştırmacılar, özellik uzayının boyutunu küçültmek amacıyla boyut azaltma tekniklerini kullanmaktadırlar. Ana boyut azaltma teknikleri arasında Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Random Projection (Fodor, 2002) ve Autoencoder (Wang & ark., 2016) bulunmaktadır.

Otomatik kodlayıcı (autoencoder), giriş verisini çıkışa mümkün olduğunca benzer şekilde kopyalamayı amaçlayan bir tür sinir ağıdır. Otomatik kodlayıcı, sinir ağlarının güçlü özelliklerini kullanarak bir boyut azaltma yöntemi olarak işlev görür. Temel fikir, girdi ve çıktı katmanları arasında yer alan gizli katmanın daha az birime sahip olmasıdır, bu da özellik uzayının boyutlarını azaltmak için kullanılabilir. Özellikle çok sayıda özelliğe sahip metinler, belgeler ve diziler için, otomatik kodlayıcı kullanımı, daha hızlı ve verimli bir veri işleme imkanı sunar. Genel yapısı, Evrişimli Sinir Ağları (CNN) ve Tekrarlayan Sinir Ağları (RNN) gibi sinir ağı mimarilerinden oluşur. Bu teknikler, metin analizi ve doğal dil işleme alanlarında, özellikle büyük veri setleriyle çalışılırken, veri işlemenin hızlanmasına ve verimliliğin artmasına katkı sağlar.

Farklı NLP alt görevleri

Adlandırılmış varlık tanıma (named entity recognition (NER)): Adlandırılmış varlık tanıma (NER), metin içerisindeki adlandırılmış varlıkları tanımlama ve bunları kişi, konum, kuruluş vb. gibi önceden tanımlanmış kategorilere ayırma görevidir. Genel alanda organizasyon, kişi ve konum adları; biyomedikal alanda ise gen, protein, ilaç ve hastalık isimleri NER için tipik örneklerdir. NER, metindeki adlandırılmış varlıkları sınıflandırarak, soru

yanıtlama, metin özetleme, makine çevirisi ve metin sınıflandırma gibi çeşitli doğal dil işleme uygulamalarında kullanılır. NER sistemleri, genellikle yüksek doğrulukla tanıma gerçekleştirse de, kuralların ve özelliklerin tasarımı için geniş çaplı insan çabası gerektirir.

İngilizce için OntoNotes (Hovy & ark., 2006) ve Winer (Ghaddar & Langlais, 2017) gibi etiketlenmiş külliyatlar (corpus) mevcuttur. Türkçe için ise, Penn-Treebank'tan alınan ve kelimelerin morfolojik netliği, adlandırılmış varlıklar, duygular ve anlamsal rol etiketlerini içeren çok katmanlı bir korpus bulunmaktadır (Yıldız & ark., 2018). Ayrıca, Türkçe için CRF tabanlı bir NER yaklaşımları çalışmalarda mevcuttur (Seker & Eryigit, 2017).

BERT ve transfer öğrenimi: Son yıllarda, BERT (Devlin & ark., 2018) gibi Transformer tabanlı sinir ağları ve transfer öğrenimi yaklaşımları NER'de kullanılmaya başlanmıştır. Bu yöntem, büyük bir korpus üzerinde önceden eğitilmiş modelleri kullanarak, geniş bir kelime yelpazesini kapsayan temsiller oluşturur. NER görevinde bu temsil yöntemlerinin kullanılması, performansı artırırken zaman ve maliyetten tasarruf sağlar. Türkçe için de BERTurk modeli (Aras & ark., 2020) gibi önceden eğitilmiş modeller mevcuttur. Bu gelişmeler, NER'in doğal dil işlemedeki rolünü ve etkinliğini önemli ölçüde artırmaktadır. (Okur & Sertbaş, 2021)

İlişki çıkarımı (relation extraction (RE)): İlişki çıkarımı, metin içerisindeki varlıklar arasındaki ilişkileri bulmak için kullanılan bir yöntemdir. Bu süreç, metinden kişi, yer, organizasyon, nesne ve fiil yapılarının çıkarılmasını ve ardından bu varlıklar arasında anlamlı bağlantılar kurulmasını içerir. Özne ve nesne arasındaki ilişki genellikle S-R-O (subject – relation - object) formatında ifade edilir (Zelenko, 2003). Örneğin, "Ankara, Türkiye'nin başkentidir." cümlesinde Ankara ve Türkiye arasındaki başkent ilişkisi (Ankara – Başkent – Türkiye) olarak modellenebilir.

Varlıklar arası ilişkiler, vektör formuna dönüştürülerek klasik makine öğrenmesi veya derin öğrenme algoritmaları ile sınıflandırılabilir (Ren, 2018). Son dönem çalışmalarında, bu ilişki

yapıları genellikle bir graf yapısı olarak temsil edilmekte ve graf sinir ağları modelleri ile sınıflandırılmaktadır. Graf üzerindeki her bir düğüm (node), kelime varlıkları veya belgelerin kendisi olarak düşünülebilir. Graf düğümleri arasındaki bağlantılar, varlık veya belgeler arasındaki ilişkileri temsil eder ve graf sinir yapıları ile bu ilişkiler sınıflandırılabilmektedir (Han, 2018).

İlişki çıkarımı işlemi, esasında iki alt kategoride incelenebilir (Sahu, 2019). Bu alt kategoriler, çeşitli doğal dil işleme görevlerinde kullanılır ve metinlerin derinlemesine analizinde kritik bir rol oynar. İlişki çıkarımı, metin sınıflandırma, bilgi çıkarımı, soru-cevap sistemleri gibi birçok alanda kullanılan önemli bir NLP görevidir. Bu süreç, metinlerden zengin anlamsal bilgilerin çıkarılmasına olanak tanır ve bu bilgilerin çeşitli uygulamalarda kullanılmasını sağlar.

Cümle seviyesi (sentence level): Cümle seviyesindeki ilişki çıkarımı, metinlerin cümlelere göre ayrıştırılması ve bu cümleler içerisindeki varlıklar arası ilişkilerin tespit edilmesine dayanır. Bu süreçte, cümle içinde doğal olarak bulunan ilişkiler çıkarılabilir veya cümlede herhangi bir ilişki bulunmadığı durumlarda, bilgi bankasından (knowledge-based - KB) alınan otomatik ilişki etiketleri kullanılabilir. KB, önceden belirlenmiş olası ilişkilere sahip varlıkların bir sözlüğü olarak düşünülebilir. Bu yönteme uzaktan denetim (distant supervision) denir (Ji, 2017) (Han, 2019).

Uzaktan denetim yönteminde, metin içerisindeki varlıklara otomatik olarak ilişkiler tanımlandığında bazı sorunlar ortaya çıkabilir. Bu sorunlardan biri, varlıkların yanlış etiketlenmesidir. Örneğin, bir metin ekonomiyle ilgiliyse ve içinde ekonomiyle ilgili önemli bir isim geçiyorsa, bu isme uzaktan denetim yoluyla ekonomi dışı, örneğin siyasetle ilgili etiketler atanabilir. Bu durum, yanlış etiketlemeye yol açar ve bu yanlış etiketlerin temizlenmesi ve düzeltilmesi gerekebilir (Ru, 2018). Cümle seviyesi ilişki çıkarımı, metin içerisindeki varlıklar arasındaki ilişkilerin daha detaylı ve hassas bir şekilde incelenmesine olanak tanır ve doğal dil işleme uygulamalarında önemli bir görevi yerine getirir. Uzaktan denetim

yöntemi, özellikle büyük veri setleri üzerinde çalışırken zaman ve kaynak tasarrufu sağlamakla birlikte, dikkatli bir şekilde uygulanmalıdır.

Belge seviyesi (document level): Belge seviyesinde ilişki çıkarımı, bireysel belgeler arasındaki ilişkilere odaklanır. Bu tür ilişkilere örnek olarak, akademik yayınlarda kullanılan atıflar verilebilir. Bu atıflar, belgeler arasında belirli bir bağlantı veya atıf ilişkisi oluşturur. Ayrıca, belgelerin başlık bilgileri veya özetlerinden çıkarılan etiketler ve nesneler de ilişki çıkarımı için kullanılabilir. Son yıllarda bu alandaki çalışmalar, belge seviyesindeki ilişkileri kullanarak artan belge/doküman sayısının otomatik etiketlenmesi veya sınıflandırılması üzerine yoğunlaşmıştır (Yao, 2019) (Han, 2019).

Belge seviyesi ilişki çıkarımının amacı, belgeleri otomatik olarak kategorize etmek ve büyük miktardaki belgeler içerisinden anlamlı bilgileri çıkarmaktır. Bu süreç, metinlerin daha geniş bir bağlamda analiz edilmesine olanak tanır ve belgelerin içeriğine bağlı olarak daha derinlemesine bir anlayış sağlar. Belge seviyesindeki ilişki çıkarımı, bilgi yönetimi, bilgi alımı ve metin madenciliği gibi alanlarda önemli bir rol oynar ve büyük veri setlerinin etkili bir şekilde işlenmesine katkıda bulunur. Bu yaklaşım, metin tabanlı verilerin daha geniş ve kapsamlı bir şekilde incelenmesini sağlayarak, doğal dil işleme ve metin analizi alanlarında yeni ufuklar açar.

Sınıflandırma teknikleri

Metin ve belge sınıflandırması, verileri önceden belirlenmiş kategorilere ayırmak için kullanılan algoritmalardan yararlanır. Günümüzde, bu alandaki algoritmalar arasında geleneksel yöntemlerin yanı sıra yapay sinir ağı algoritmaları da yaygın olarak kullanılmaktadır.

Geleneksel sınıflandırma yöntemleri:

- Rocchio Algoritması: Metin sınıflandırması için kullanılır ve belirli bir kategoriye ait belgelerin vektör temsillerini kullanarak çalışır (Selvi, 2017).
- Lojistik Regresyon: İki kategorili sınıflandırma problemlerinde etkili olan bir yöntemdir.
- Naive Bayes: Olasılığa dayalı bir sınıflandırma tekniğidir ve metin sınıflandırmasında yaygın olarak kullanılır (Xu, 2018).
- K-Nearest Neighbor (KNN): Benzerlik ölçütlerine göre sınıflandırma yapar (Wang, 2017).
- Destek Vektör Makineleri (SVM): Yüksek boyutlu veri setlerinde etkili olan bir sınıflandırma yöntemidir (Fatima, 2017).
- Karar Ağaçları ve Random Forests: Karar kuralları üzerinden sınıflandırma yapan yöntemlerdir (Selvi, 2017).

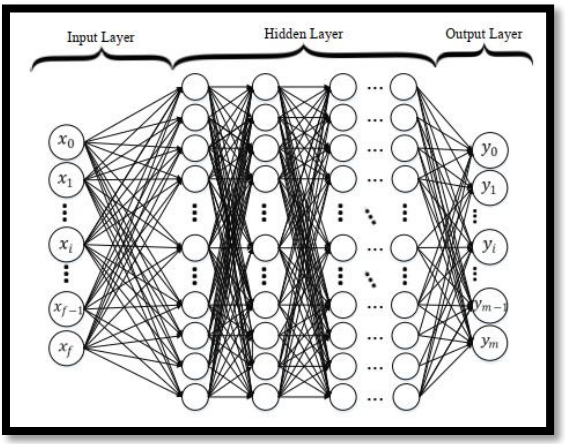
Yapay sinir ağı tabanlı algoritmalar:

- Derin Sinir Ağları (DNN): Çok katmanlı yapıları ile karmaşık sınıflandırma problemlerinde kullanılır.
- Evrişimli Sinir Ağları (CNN): Metin sınıflandırmasında, özellikle metinlerin yerel özelliklerini çıkarmada etkilidir.
- Tekrarlayan Sinir Ağları (RNN): Zamanla değişen verileri işlemek için kullanılır ve metinlerdeki bağlamsal bilgileri modelleyebilir.
- Derin İnanç Ağları (DBN): Özellik öğrenmede etkili olan bir başka sinir ağı modelidir.
- Hiyerarşik Dikkat Ağları (HAN): Metinlerin farklı seviyelerdeki önemini öğrenmek için dikkat mekanizmaları kullanır.

Sinir Ağı Tabanlı Algoritmaların Kombinasyonları**:** Farklı sinir ağı modellerinin birleşimiyle daha karmaşık ve etkili sınıflandırma modelleri oluşturulabilir.

Bu teknikler, metin ve belge sınıflandırmasında farklı yaklaşımlar ve metodolojiler sunarak, çeşitli uygulama alanlarında kullanılır. Derin öğrenme modelleri, özellikle karmaşık ve çeşitlilik gösteren NLP uygulamalarında yüksek performans sunmaktadır.

Derin sinir ağları (deep neural networks (DNN)), katmanlar arasındaki çoklu bağlantıları temel alan ve her katmanın yalnızca önceki katmandan aldığı bilgiyle bir sonraki katmana bilgi ilettiği bir yapıya sahiptir. Standart bir DNN' nin yapısı, giriş katmanından başlayarak, bir veya daha fazla gizli katman üzerinden çıktı katmanına kadar uzanır. Giriş katmanı, TF-IDF, kelime gömme veya diğer özellik çıkarma teknikleriyle elde edilen veri temsillerini alır. Çıktı katmanı, çok sınıflı sınıflandırma durumunda sınıfların sayısına, ikili sınıflandırmada ise tek bir düğüme eşittir.



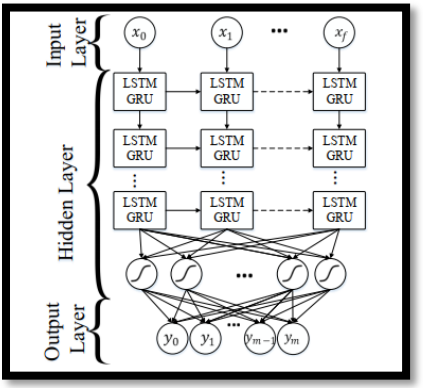
Şekil 2. *DNN Ağ Mimarisi (Schmidt-Hieber, 2020)*

Çok sınıflı DNN modellerinde, her bir öğrenme modeli (katmandaki düğüm sayısı ve katmanların sayısı gibi parametreler) tamamen rastgele belirlenir. DNN'de yaygın olarak kullanılan aktivasyon fonksiyonları arasında Sigmoid ve ReLU bulunur. Bu

ağlar, ayırt edici eğitim için standart geri yayılım algoritması kullanılarak eğitilir (Schmidt-Hieber, 2020). Derin sinir ağlarının bu yapısı ve esnekliği, karmaşık sınıflandırma problemlerinde etkin çözümler üretmelerine olanak tanır ve çok çeşitli doğal dil işleme uygulamalarında başarıyla kullanılmalarını sağlar. Bu modeller, özellikle büyük ve çeşitli veri kümeleri üzerinde derinlemesine öğrenme ve tahmin yapabilme kapasitesine sahiptir. Şekil 2'de DNN Ağ mimarisi sunulmuştur.

Tekrarlayan sinir ağı (recurrent neural network (RNN)): Tekrarlayan sinir ağları, metin sınıflandırması gibi sıralı veri işleme görevlerinde sıklıkla kullanılan bir sinir ağı mimarisidir (Yogatama, 2017). RNN' ler, bir dizinin önceki veri noktalarına ağırlık vererek, veri setinin yapısını ve bağlamsal bilgilerini dikkate alır. Bu özellik, RNN' leri metin, dizgi ve sıralı veri sınıflandırması için etkili bir yöntem haline getirir.

Bir RNN, genellikle giriş katmanı, bir veya daha fazla gizli katman ve çıktı katmanından oluşur. Giriş katmanında kelime gömme gibi özellik çıkarma yöntemleri kullanılır. RNN' nin temel özelliği, her gizli katmanın önceki katmandan gelen bilgiyi dikkate alması ve bu bilgiyi sonraki katmana aktarmasıdır. RNN mimarisi, özellikle Uzun Kısa Süreli Bellek (Long Short-Term Memory - LSTM) veya Kapı Kontrollü Birimler (Gated Recurrent Units - GRU) gibi gelişmiş varyantlarını kullanarak, metin sınıflandırmasında başarılı sonuçlar elde edebilir. RNN' ler, metinlerdeki zamanla değişen bağlamları ve ilişkileri etkili bir şekilde modelleyebilir ve böylece daha derinlemesine bir anlam analizi sağlar.



Şekil 3. *RNN Ağ Mimarisi (Yogatama, 2017)*

RNN' lerin gelişmiş varyantları olan LSTM ve GRU, sıralı verilerdeki uzun süreli bağımlılıkları daha etkili bir şekilde yakalayabilir. Bu yapılar, özellikle metinlerdeki uzun bağlamsal ilişkilerin ve anlamların kavranmasında önemli bir rol oynar. RNN ve bu gelişmiş varyantları, metin sınıflandırmasında, duygu analizinde, metin özetlemede ve diğer pek çok doğal dil işleme görevinde kritik öneme sahiptir. Bu modeller, metinlerin daha kapsamlı ve doğru bir şekilde işlenmesine olanak tanıyarak, doğal dil işleme uygulamalarının etkinliğini artırır. Şekil 3'de RNN ağ mimarisi gösterilmiştir.

Uzun Kısa Süreli Bellek (Long Short-Term Memory - LSTM), RNN' nin geliştirilmiş bir versiyonudur ve verileri rastgele aralıklarla hatırlama yeteneğine sahiptir. LSTM, özellikle sıralı verilerde uzun süreli bağımlılıkları modellemek için tasarlanmıştır. Bu mimari, öğrenilen bilgilerin uzun vadeli olarak saklanabilmesini sağlar ve değerlerin öğrenme süreci boyunca değiştirilmesini engeller. LSTM' ler, hem ileri hem de geri yönde bilgi akışına izin veren nöronlar içerir, bu sayede daha kapsamlı bir bağlam analizi yapılabilir (Liu, 2019).

Kapılı Tekrarlayan Hücreler (Gated Recurrent Units - GRU), RNN içindeki bir kapı mekanizmasıdır ve LSTM' ye benzer şekilde çalışır. GRU, LSTM' ye göre daha az parametreye sahiptir ve çıkış

kapıları bulunmaz. Bu yapısıyla GRU, LSTM' nin daha basit ve hesaplamalı olarak daha verimli bir alternatifidir (Agarap, 2018). GRU, LSTM gibi uzun süreli bağımlılıkları yakalayabilme yeteneğine sahip olmasına rağmen, daha az karmaşık yapıda olduğundan farklı uygulamalar için tercih edilebilir.

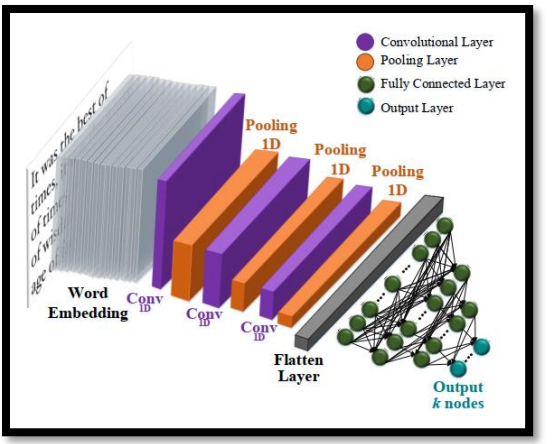
Hem LSTM hem de GRU, metin ve diğer sıralı verilerin işlenmesinde, özellikle zamanla değişen bağlamsal bilgilerin ve uzun süreli bağımlılıkların etkili bir şekilde modellemesinde önemli rol oynar. Bu mimariler, doğal dil işleme uygulamalarının yanı sıra ses tanıma, zaman serisi analizi gibi alanlarda da yaygın olarak kullanılmaktadır. Bu teknikler, metinlerin daha derinlemesine analizi ve anlaşılması için gerekli olan araçlar sağlar.

Evrişimli sinir ağları (Convolutional Neural Networks - CNN): Evrişimli sinir ağları başlangıçta görüntü işleme için tasarlanmış olmasına rağmen, metin sınıflandırmasında da etkili bir şekilde kullanılan bir derin öğrenme mimarisidir (Jacovi, 2018). Görüntü işlemede, bir görüntü tensörü, belirli bir boyutta (d × d) çekirdekler (kernels) ile işlenir. Bu işlem, özellik haritalarını oluşturur ve girişte birden çok filtre sağlanmasını mümkün kılar.

CNN'ler, hesaplama karmaşıklığını azaltmak için havuzlama (pooling) tekniklerini kullanır. Havuzlama, ağdaki bir katmandan diğerine çıktı boyutunu azaltarak, önemli özellikleri korurken çıktıların boyutunu düşürür. Maksimum havuzlama, en yaygın kullanılan havuzlama yöntemidir ve havuz oluşturma penceresindeki maksimum değerin seçildiği bir tekniktir. Özellik haritaları, bir sonraki katmana beslenmeden önce düzleştirilir ve CNN'nin son katmanları genellikle tamamen bağlantılıdır (fully connected).

Metin sınıflandırması için CNN kullanılırken, "kanalların" sayısı, yani özellik alanının boyutu (S), genellikle çok büyük olabilir. Örneğin, görüntü sınıflandırmasında sadece birkaç kanal (RGB gibi) kullanılırken, metin sınıflandırmasında bu sayı çok daha büyük olabilir (örneğin, 50.000 kelime). Bu, yüksek boyutlulukla sonuçlanabilir. Metin sınıflandırması için bir CNN mimarisi, giriş

katmanı olarak kelime gömme, 1 boyutlu evrişimli katmanlar, 1 boyutlu havuz katmanları, tamamen bağlı katmanlar ve son olarak çıktı katmanını içerir.



Şekil 4. *CNN Ağ Mimarisi (Jacovi, 2018)*

Bu mimari, metinlerdeki yerel ve global özelliklerin etkili bir şekilde yakalanmasını sağlar ve metin sınıflandırmasında önemli bir rol oynar. CNN'ler, metinlerin yapısal ve semantik özelliklerini analiz ederek, metin sınıflandırma görevlerinde yüksek performans sunar. Şekil 4'te CNN ağ mimarisi gösterilmiştir.

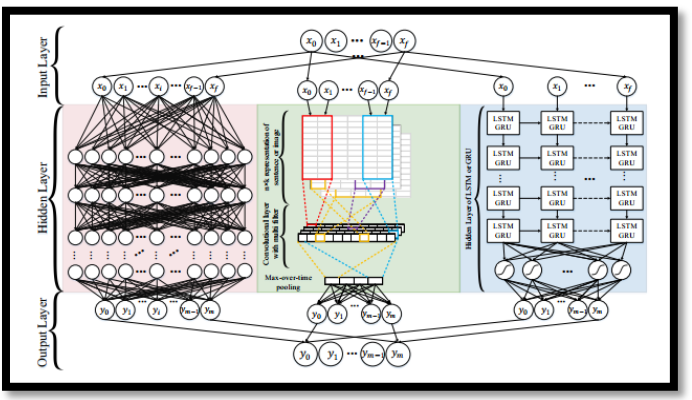
Hiyerarşik dikkat ağları (Hierarchical Attention Networks - HAN), metin ve belge sınıflandırması için geliştirilmiş başarılı derin öğrenme mimarilerinden biridir. HAN, metin sınıflandırmasını iki farklı seviyede ele alır: kelime ve cümle düzeyi. Bu mimari, alt seviyede kelime kodlama ve kelimeye yönelik dikkat mekanizmalarını; üst seviyede ise cümle kodlama ve cümleye yönelik dikkat mekanizmalarını içerir (Yang, 2016).

HAN mimarisi, metinlerin daha detaylı ve bağlamsal analizini sağlayan sıralı ve hiyerarşik bir yapıya sahiptir. Kelime düzeyindeki dikkat mekanizması, metnin her bir kelimesinin önemini değerlendirirken, cümle düzeyindeki dikkat mekanizması, metnin her bir cümlesinin bağlamdaki önemini belirler. Bu iki

seviyeli yapı, metinlerin hem mikro hem de makro düzeyde daha derinlemesine anlaşılmasını sağlar.

HAN, özellikle uzun metinlerin ve belgelerin sınıflandırılmasında etkilidir. Bu mimari, metinlerin farklı bölümlerinin önemini otomatik olarak öğrenir ve bu bilgiyi metin sınıflandırma görevinde kullanır. HAN' ın dikkat mekanizmaları, metinlerin daha doğru ve etkili bir şekilde sınıflandırılmasına katkıda bulunur, çünkü model, metinlerin hangi bölümlerinin sınıflandırma için daha önemli olduğunu belirleyebilir. Bu yaklaşım, metin sınıflandırma, duygu analizi, belge özetleme ve diğer pek çok doğal dil işleme görevinde önemli bir rol oynar ve geniş kapsamlı metinlerin etkili bir şekilde işlenmesini sağlar.

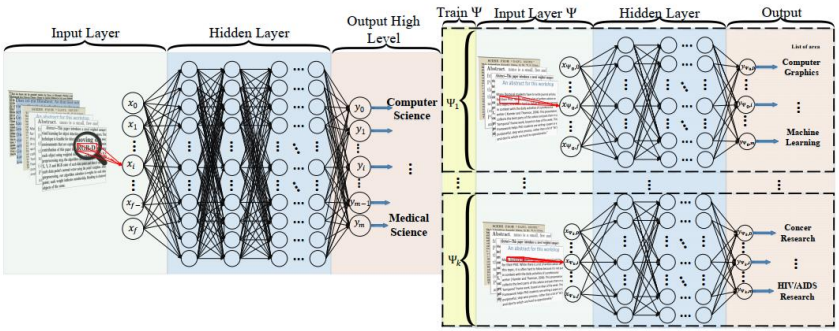
Sınıflandırma görevlerinde üstün ve kesin sonuçlara ulaşmayı hedefleyen araştırmacılar, geleneksel derin öğrenme mimarilerinin ötesine geçerek, bu alanlarda yenilikçi tekniklerin geliştirilmesine öncülük etmektedirler. Bu metodoloji, çeşitli derin öğrenme yapılarını entegre ederek, sınıflandırma yeteneklerini maksimize etmeyi hedeflemektedir. İlgili bölümde, bu entegratif yaklaşımlar ve bunların sınıflandırma performansına katkılarına ilişkin detaylı bilgiler sunulacaktır.



Şekil 5. *RMDL Mimarisi (DNN-RNN-CNN Paralel Kullanım)(Kowsari, 2018)*

Rastgele Çok Modelli Derin Öğrenme (Random Multimodel Deep Learning, RMDL), metin sınıflandırma alanında yenilikçi bir derin öğrenme metodu olarak tanımlanabilir. Bu teknik, çeşitli veri setlerinde sınıflandırma görevlerini yerine getirmek için esnek bir çerçeve sunmaktadır. RMDL, derin sinir ağı mimarisinin giriş ve çıkış katmanları arasındaki gizli katmanda, Derin Sinir Ağları (DNN), Tekrarlayan Sinir Ağları (RNN) ve Evrişimli Sinir Ağları (CNN) olmak üzere üç farklı ağ tipini paralel olarak işletir. Bu modellerin her biri için katman ve düğüm sayısı rastgele belirlenir, böylece her model benzersiz bir yapıya sahip olur. Örneğin, RMDL mimarisinde 3 DNN, 3 RNN ve 3 CNN olmak üzere toplam 9 rastgele oluşturulmuş model kullanılabilir. Çıkış katmanında, bu farklı sinir ağlarının ürettiği sonuçlar entegre edilerek sınıflandırma için nihai değerler elde edilir. Bu çok yönlü ve rastgele yapılandırılmış yaklaşım, RMDL'nin benzersiz ve etkili bir sınıflandırma performansı sunmasını sağlar (Kowsari, 2018). Şekil 5, bu kapsamlı ve çok modelli derin öğrenme mimarisini görselleştirmektedir.

Metinler için Hiyerarşik Derin Öğrenme (Hierarchical Deep Learning for Text, HDLTex) mimarisi, belgelerin hiyerarşik yapıda sınıflandırılmasını sağlayan önemli bir katkı sunar. Çoklu sınıflı sınıflandırma teknikleri, sınıf sayısı az olduğunda etkili olabilirken, belgelerin hiyerarşik yapısı göz önüne alındığında ve sınıf sayısı arttığında performanslarında azalmalar meydana gelebilir. HDLTex, bu zorluğun üstesinden gelmek için belgelerin hiyerarşik seviyelerine uygun derin öğrenme modelleri geliştirerek bu soruna çözüm getirir. Bu mimari, her bir derin öğrenme modeli için özgün bir yapı önerir ve bu yapı Kowsari (2017) tarafından detaylı bir şekilde tanımlanmıştır. HDLTex'in bu yenilikçi mimarisi Şekil 6'da görselleştirilmiştir.



Şekil 6. *HDLTex Mimarisi (DNN-RNN-CNN Paralel ve Seri Kullanım)(Kowsari, 2017)*

Derin öğrenmenin sınırlamaları: Derin Öğrenme (Deep Learning, DL) modellerinin yorumlanabilirliği, özellikle Derin Sinir Ağları (Deep Neural Networks, DNN) bağlamında, modelleme sürecinde karşılaşılan özelliklerin açıklanmasını zorlaştıran bir hususdur. Bu modellerin iç yapısının karmaşıklığı, sinir ağlarının anlaşılmasını güçleştirmekte ve bu da, derin öğrenme gibi karmaşık algoritmaların kavranmasını zorlaştırmaktadır. Derin öğrenme, yapay zeka (Artificial Intelligence, AI) alanındaki en güçlü teknikler arasında yer alır ve birçok araştırmacı, bu teknolojinin gücünü ve hesaplama kapasitesini artırmak için derin öğrenme mimarilerine odaklanmıştır. Ancak, derin öğrenme mimarilerinin sınıflandırma görevlerinde uygulanmasının bazı dezavantajları ve sınırlamaları da bulunmaktadır. Bu modellerin başlıca sorunlarından biri, derin öğrenmenin kapsamlı bir teorik anlayışını kolaylaştırmamasıdır; yani, derin öğrenme yöntemlerinin "kara kutu" doğasıdır. Bu yöntemlerin çıktılarının katmanlı yapısı, kolaylıkla anlaşılabilir değildir. Ayrıca, derin öğrenme modellerinin genellikle geleneksel makine öğrenme algoritmalarına kıyasla daha fazla veri gerektirmesi, bu tekniklerin küçük veri kümeleri üzerindeki sınıflandırma görevlerine uygun olmadığını göstermektedir. Ek olarak, derin öğrenme sınıflandırma algoritmaları için gereken büyük miktardaki veri, eğitim sürecinde hesaplama karmaşıklığını artırır, bu da donanım ve zaman maliyetlerini yükseltir.

Performans değerlendirmesi

Araştırma topluluğu içerisinde, algoritmaların değerlendirilmesi için genel kabul görmüş ve karşılaştırılabilir performans ölçütlerine sahip olmanın tercih edildiği bilinmektedir. Ancak, gerçekte bu tür standart ölçütler sadece birkaç yöntem için mevcuttur. Metin sınıflandırma yöntemlerinin değerlendirilmesinde karşılaşılan başlıca sorunlardan biri, standart veri toplama protokollerinin eksikliğidir. Ortak bir veri toplama yöntemi bulunsa dahi, farklı eğitim ve test setlerinin seçimi model performansında tutarsızlıklara yol açabilir.

Yöntemlerin değerlendirilmesi sırasında ortaya çıkan diğer bir zorluk, farklı deneylerde kullanılan çeşitli performans ölçütlerinin karşılaştırılabilirliğidir. Bu ölçütler, sınıflandırma görevlerinin performansını belirli yönlerden değerlendirir ve her zaman aynı bilgileri sunmayabilir.

Bu bölümde, değerlendirme kriterleri, performans ölçütleri ve sınıflandırıcıların performanslarının karşılaştırılabilmesi için kullanılan yöntemler vurgulanmaktadır. Farklı değerlendirme ölçütlerinin altında yatan mekanizmaların değişken olabileceği göz önünde bulundurularak, bu ölçütlerin neyi temsil ettiğini ve hangi tür bilgileri iletmeye çalıştıklarını anlamak, karşılaştırılabilirlik açısından büyük önem taşımaktadır.

Bu ölçütlerin örnekleri arasında geri çağırma, hassasiyet, doğruluk, F-skoru, Mikro-Makro Ortalama bulunmaktadır. Bu ölçümler, gerçek pozitifler (TP), yanlış pozitifler (FP), yanlış negatifler (FN) ve gerçek negatifler (TN) içeren bir "karışıklık matrisi"ne dayanmaktadır. Bu dört unsurdan her birinin önemi, sınıflandırma uygulamasının özelliğine göre değişebilir. Tüm tahminler arasındaki doğru tahminlerin oranına doğruluk; bilinen pozitifler arasındaki doğru tahmin edilenlerin fraksiyonuna duyarlılık veya geri çağırma; doğru tahmin edilen negatiflerin oranına özgüllük; ve tüm pozitifler arasındaki doğru tahmin edilenlerin oranına hassasiyet veya pozitif tahmin değeri denir.

Sonuç

Bu çalışma, günümüzde özellikle sosyal medya, web, iş dünyası ve eğitim gibi alanlarda gözlemlenen sürekli artan metin ve belge birikiminin, doğal dil işleme (NLP) alanındaki araştırmacılar için nasıl önemli bir sorun alanı haline geldiğini ele almaktadır. Metin yığınlarından anlamlı bilgilerin çıkarılması ve belgelerin etkin bir şekilde sınıflandırılması sürecinin hem zahmetli hem de zaman alıcı olduğu belirtilmiştir. Doğal dil işlemenin, bu zorlukları ele alarak otomatik sınıflandırma sistemleri geliştirme konusunda kritik bir rol oynadığı vurgulanmıştır. Ayrıca, metin sınıflandırma probleminde klasik makine öğrenmesi algoritmalarının yanı sıra, derin öğrenme algoritmalarının da giderek artan bir ilgi gördüğü ve bu ilginin alandaki gelişmeleri hızlandırdığı belirtilmiştir.

Metin sınıflandırma probleminin, benzer içerik ve bilgiye sahip belgelerin tespiti ve sıralanması süreci olarak tanımlandığı bu çalışmada, her bir dokümanın içerdiği etiket bilgisi ve bu bilginin sınıflandırma sürecinde nasıl kritik bir rol oynadığı açıklanmıştır. Ayrıca, metinlerin duygusal analizinin de bu sınıflandırma sürecinde önemli olduğu vurgulanmıştır.

Doğal dillerin yapısal, bölgesel, kültürel ve dönemsel çeşitliliklerinin, uygulanacak metin işleme tekniklerinin değişkenliğine neden olduğu ve bu çeşitliliklerin, seçilen dilin morfolojik yapısına uygun bir metin işleme yaklaşımını gerektirdiği belirtilmiştir. Bu çalışmada, doğal dil işleme alanında metnin detaylı bir ön işlemden geçirilmesinin ve ardından vektör formuna dönüştürülmesinin, daha etkin ve güvenilir bir indeksleme modeli oluşturulmasına nasıl olanak tanıdığı incelenmiştir.

Son olarak, özellikle Türk dili morfolojisine uygun olarak metin sınıflandırma aşamalarında kullanılan modellerin detayları üzerinde durulmuş ve üzerinde çalışılacak olan yeni bir model sunulmuştur. Bu çalışmanın temel amacı, İngilizce dil özelliklerine göre yapılandırılmış modellerin Türkçe üzerinde uygulanmasına yönelik araştırmaları tanıtmak ve bu alanda ne tür gelişmelerin yapılabileceğine dair fikirler sunmaktır. Bu sayede, dil

farklılıklarının doğal dil işleme modellerinin performansına etkileri daha iyi anlaşılabilir ve bu alandaki araştırmalar daha da ileriye taşınabilir. Bu çalışma, doğal dil işlemenin ve metin sınıflandırmasının, dilin zenginliği ve çeşitliliğiyle nasıl bütünleştiğini ve bu bütünleşmenin, etkili ve yenilikçi çözümler üretme potansiyelini göstermektedir.

Kaynakça

Agarap, A. F. M. (2018, February). A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. *In Proceedings of the 2018 10th international conference on machine learning and computing* (pp. 26-30).

Ahonen, H., Heinonen, O., Klemettinen, M., & Verkamo, A. I. (1997). Applying data mining techniques in text analysis. *Report C-1997-23, Dept. of Computer Science*, University of Helsinki.

Akın, A. A., & Akın, M. D. (2007). Zemberek, an open source NLP framework for Turkic languages. *Structure, 10(2007)*, 1-5.

Al Nahas, A., Kulunk, A., Gozutok, B., Kalkan, S. C., & Erdinc, H. Y. (2020, August). How to Segment Turkish Words for Neural Text Classification?. *In 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) (pp. 1-5). IEEE*.

Aras, G., Makaroğlu, D., Demir, S., & Cakir, A. (2021). An evaluation of recent neural sequence tagging models in Turkish named entity recognition. *Expert Systems with Applications*, 182, 115049.

Bölücü, N., & Can, B. (2019, April). Context based automatic spelling correction for turkish. *In 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT) (pp. 1-4). IEEE*.

Cavnar, W. B., & Trenkle, J. M. (1994, April). N-gram-based text categorization. *In Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval (Vol. 161175, p. 14).*

Çilden, E. K. (2006). Stemming Turkish words using snowball.

Chalmers, D. J. (1992). Syntactic transformations on distributed representations. *Connectionist Natural Language Processing: Readings from Connection Science, 46-55*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fatima, Shugufta, and B. Srinivasu. "Text Document categorization using support vector machine." *International Research Journal of Engineering and Technology (IRJET) 4.2 (2017): 141-147.*

Fodor, I. K. (2002). A survey of dimension reduction techniques (No. UCRL-ID-148494). *Lawrence Livermore National Lab.*, CA (US).

Ghaddar, A., & Langlais, P. (2017, November). Winer: A wikipedia annotated corpus for named entity recognition. *In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 413-422).*

Güngör, O., Güngör, T., & Üsküdarli, S. (2019). The effect of morphology in named entity recognition with sequence tagging. *Natural Language Engineering, 25(1), 147-169.*

Han, J., Pei, J., & Tong, H. (2022). Data mining: concepts and techniques. *Morgan kaufmann*.

Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., & Sun, M. (2018). Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147.*

Han, Xu, et al. "OpenNRE: An open and extensible toolkit for neural relation extraction." *arXiv preprint arXiv:1909.13078 (2019).*

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006, June). OntoNotes: the 90% solution. In Proceedings of the human language technology conference of the NAACL, *Companion Volume: Short Papers (pp. 57-60).*

Jacovi, Alon, Oren Sar Shalom, and Yoav Goldberg. "Understanding convolutional neural networks for text classification." *arXiv preprint arXiv:1809.08037 (2018).*

Ji, Guoliang, et al. "Distant supervision for relation extraction with sentence-level attention and entity descriptions." *Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 31. No. 1. 2017.*

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Justicia De La Torre, C., Sánchez, D., Blanco, I., & Martín-Bautista, M. J. (2018). Text mining: techniques, applications, and challenges. *International journal of uncertainty, fuzziness and knowledge-based systems, 26(04), 553-582.*

Karaağaç, G. (2013). Dil bilimi terimleri sözlüğü. (No Title).

Kaur, B., & Bathla, G. (2018). Document classification using various classification algorithms: a survey. *Int J Fut Revol Comput Sci Commun Eng, 4(2), 150-155.*

Kılınç, D., Özçift, A., Bozyigit, F., Yıldırım, P., Yücalar, F., & Borandag, E. (2017). TTC-3600: A new benchmark dataset for Turkish text categorization. *Journal of Information Science, 43(2), 174-185.*

Kowsari, Kamran, et al. "Hdltex: Hierarchical deep learning for text classification." *2017 16th IEEE international conference on machine learning and applications (ICMLA). IEEE, 2017*.

Kowsari, Kamran, et al. "Rmdl: Random multimodel deep learning for classification." *Proceedings of the 2nd International Conference on Information System and Data Mining. 2018.*

Li, B., Liu, T., Zhao, Z., Tang, B., Drozd, A., Rogers, A., & Du, X. (2017, September). Investigating different syntactic context types and context representations for learning word embeddings. *In*

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 2421-2431).

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ... & He, L. (2022). A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST), 13(2), 1-41.*

Liu, Gang, and Jiabao Guo. "Bidirectional LSTM with attention mechanism and convolutional layer for text classification." *Neurocomputing 337 (2019): 325-338.*

Melamud, Oren, Jacob Goldberger, and Ido Dagan. "context2vec: Learning generic context embedding with bidirectional lstm." *Proceedings of the 20th SIGNLL conference on computational natural language learning. 2016.*

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Mooney, C. H., & Roddick, J. F. (2013). Sequential pattern mining--approaches and algorithms. *ACM Computing Surveys (CSUR), 45(2), 1-39.*

Oflazer, K. (1994). Spelling correction in agglutinative languages. *arXiv preprint cmp-lg/9410004.*

Okur, H. I., & Sertbaş, A. (2021, September). Pretrained neural models for turkish text classification. *In 2021 6th International Conference on Computer Science and Engineering (UBMK) (pp. 174-179). IEEE.*

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).*

Perkins, J. (2010). Python text processing with NLTK 2.0 cookbook. *PACKT publishing*.

Porter, M. F. (2001). Snowball: A language for stemming algorithms.

Ren, F., Zhou, D., Liu, Z., Li, Y., Zhao, R., Liu, Y., & Liang, X. (2018, August). Neural relation classification with text descriptions. *In Proceedings of the 27th international conference on computational linguistics (pp. 1167-1177).*

Ru, Chengsen, et al. "Using semantic similarity to reduce wrong labels in distant supervision for relation extraction." *Information Processing & Management 54.4 (2018): 593-608.*

Sahu, Sunil Kumar, et al. "Inter-sentence relation extraction with document-level graph convolutional neural network." *arXiv preprint arXiv:1906.04684 (2019).*

Schmidt-Hieber, Johannes. "Nonparametric regression using deep neural networks with ReLU activation function." *Annals of Statistics 48.4 (2020): 1875-1897.*

Selvi, S. Thamarai, et al. "Text categorization using Rocchio algorithm and random forest algorithm." *2016 Eighth International Conference on Advanced Computing (ICoAC). IEEE, 2017.*

Seker, G. A., & Eryigit, G. (2017). Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content. *Semantic Web, 8(5), 625-642.*

Qaiser, S., & Ali, R. (2018). Text mining: use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications, 181(1), 25-29.*

Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks, 5(1), 7-16.*

Wang, Y., Yao, H., & Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing, 184, 232-242.*

Wang, Zhiguo, Wael Hamza, and Linfeng Song. "$ k $-Nearest Neighbor Augmented Neural Networks for Text Classification." *arXiv preprint arXiv:1708.07863 (2017).*

Xu, Shuo. "Bayesian Naïve Bayes classifiers to text classification." *Journal of Information Science 44.1 (2018): 48-59.*

Yang, Zichao, et al. "Hierarchical attention networks for document classification." *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. 2016.*

Yao, Yuan, et al. "DocRED: A large-scale document-level relation extraction dataset." *arXiv preprint arXiv:1906.06127 (2019).*

Yıldız, O. T., Ak, K., Ercan, G., Topsakal, O., & Asmazoğlu, C. (2018, April). A multilayer annotated corpus for turkish. *In 2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP) (pp. 1-6). IEEE*.

Yogatama, Dani, et al. "Generative and discriminative text classification with recurrent neural networks." *arXiv preprint arXiv:1703.01898 (2017).*

Zelenko, D., Aone, C., & Richardella, A. (2003). Kernel methods for relation extraction. *Journal of machine learning research, 3(Feb), 1083-1106.*

Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics, 1, 43-52.*

BÖLÜM IV

Hobi Bahçelerine Özel Yeni Bir Otonom Hidroponik Sistem

Kadir TOHMA[1]
Yakup KUTLU[2]

Giriş

Tarım, insanlık tarihinden beri var olan bir sektör olup, sürekli olarak gelişen ve değişen bir yapıya sahiptir (Bayramoğlu, Z., 2010). Verimliliğin artırılması bu sektörde önemli bir hedef olmuştur. Geleneksel tarım yöntemleriyle kazanılan deneyimler, günümüzde teknolojik ilerlemelerle birleşerek, birim alana düşen verimliliğin artmasına önemli katkılarda bulunmuştur (Ercan, T., & Kutay, M., 2016). Ancak bu ilerlemelerin tam anlamıyla sahada kullanılmadığına dair gözlemler bulunmaktadır. Bu nedenle, kontrol

[1] Arş. Gör., İskenderun Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü
[2] Doç. Dr., İskenderun Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü

edilebilir ve bilinçli tarım yöntemlerine ihtiyaç duyulmaktadır (Ağızan, K., ve ark., 2022).

Topraksız tarım, modern tarımın getirdiği yeniliklerden biridir ve birçok avantaja sahiptir. Özellikle çilek yetiştiriciliği için, topraksız tarımın, alansal verimliliği geleneksel yöntemlere göre dört katına kadar çıkardığı gözlemlenmiştir. Bu yöntem su tasarrufu, gübre kullanımının azaltılması, bitki hastalıklarının kontrolü ve ürün kalitesinin artırılması gibi avantajlar sunmaktadır (Lopez-Aranda, J. M., ve ark., 2009, Oğuz, İ.,ve ark., 2022).

Projemizde bu yöntemlerin avantajlarından yararlanarak, özellikle şehirde yaşayanlar için hobi bahçeleri ve balkonlarda kullanılabilir, mobil bir uygulamayla entegre edilen bir topraksız tarım sistemi tasarlanmıştır. Bu sistemde, otomasyonun da gücünden yararlanarak, bitkinin ihtiyaçlarına göre otomatik dozajlama ile sulama ve gübreleme gerçekleştirilmiştir. Ek olarak, bitkinin hastalık durumunu ve büyüme evrelerini izlemek üzere görüntü işleme tekniklerinden yararlanılmıştır. Bu amaçla sisteme özel olarak entegre edilmiş bir kamera ile süreci yakından takip edebilme imkânı sağlanmıştır.

Topraksız tarım, özellikle çilek yetiştiriciliğinde büyük avantajlar sağlamaktadır (Demirsoy, L., ve ark., 2017). Öne çıkan avantajlardan biri, birim alanda dikilen bitki sayısını artırarak birim alan verimini ciddi oranda yükseltmesidir. Bunun yanında, bu yöntem sayesinde 8-10 ay boyunca devamlı ürün alınabilmekte, bitki beslemesi kontrollü bir şekilde yapılabilmekte ve tarımsal ilaç kullanımı azaltılabilmektedir. Hobi bahçeleri için tasarladığımız sistem, topraksız tarımın tüm bu avantajlarını, kullanıcı dostu bir arayüzle birleştirerek, ev ortamında kolayca uygulanabilir hale getirmektedir.

Projenin öne çıkan ve diğer mevcut topraksız tarım sistemleriyle ya da otomatik dozajlama üniteleriyle karşılaştırıldığında ayırt edici özellikleri, sunduğu özgün mobil kontrol ve takip uygulamasıyla başlamaktadır. Bu uygulama, kullanıcının elini hiç sürmeden, her yönüyle bitkinin gelişim sürecini

yakından takip etmesini sağlar. Ayrıca proje, hobi bahçelerine özel bir yaklaşım sunarak, bireysel yetiştiricilerin ihtiyaçlarına daha uygun bir çözüm getirmektedir. Ekonomik olmasıyla da dikkat çeken bu sistem, yüksek maliyetli alternatiflere kıyasla daha ulaşılabilir ve geniş bir kitleye hitap edebilir. Sistem, yetiştiricilik sürecinin her aşamasında kullanıcılara sürekli geribildirim sağlar. Bu geribildirimler, tarım pratiği üzerinde kullanıcının tam kontrol sahibi olmasına imkân tanır. Yani, bu proje sadece otomasyonla sınırlı kalmayıp, aynı zamanda kullanıcının sistemle etkileşimde bulunarak, gerektiğinde müdahalede bulunabilmesine olanak tanımaktadır. Kısacası, bu yenilikçi yaklaşım, özgünlüğünü sadece teknik özelliklerinden değil, kullanıcı odaklı, interaktif ve ekonomik çözümleriyle de ortaya koymaktadır.

Özetle, geliştirdiğimiz bu inovatif topraksız tarım sistemi, tarımın sürdürülebilirliğini artırmanın yanı sıra, şehir yaşamına uyumlu, kullanıcı dostu ve verimli bir çözüm sunmaktadır. Bu sistemin getirdiği avantajlar hem bireysel kullanıcıların hem de profesyonel yetiştiricilerin tarım uygulamalarında verimlilik ve kaliteyi artırmalarına olanak tanımaktadır. Uyguladığımız bu yöntem, tarımın geleceğine dair umut verici bir yaklaşım olarak karşımıza çıkmaktadır.

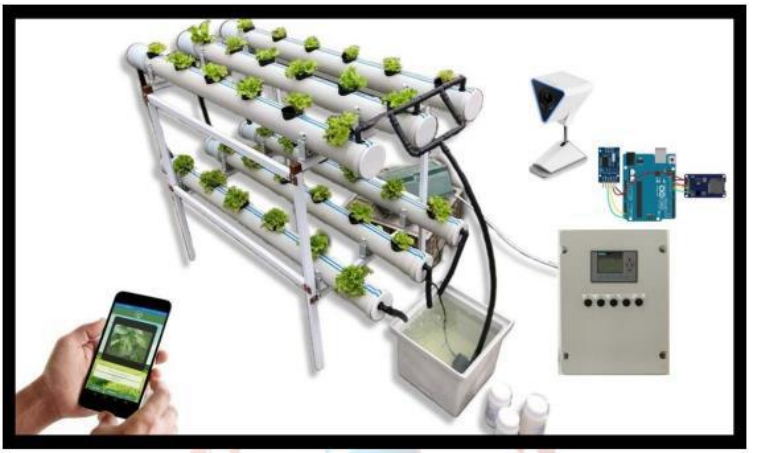
Materyal ve Yöntem

Çilek yetiştiriciliğinde topraksız tarım yöntemi, konvansiyonel tarım metotlarına kıyasla birçok avantaj sunar. Topraksız tarım, birim alana daha fazla bitki dikilmesine imkan tanıyarak alandan alınan verimi maksimize eder. Bu teknik, 8-10 ay süresince sürekli ürün elde edilmesi sayesinde genişleyen bir pazar talebi yaratır. Bunun yanında, topraksız tarımın kontrollü bir besleme ortamı sunması, çiçeklenme ve meyve kalitesini optimize etmeye yardımcı olur. Bu yöntemle su, gübre ve tarımsal ilaç kullanımında önemli ölçüde tasarruf edilebilirken; toprak yorgunluğu, toprakla ilişkilendirilen hastalıklar ve mantari hastalıklar gibi riskler asgariye indirilir (Lopez-Aranda, J. M., ve ark., 2009,).

Kentlerin genişlemesi, değerli tarım arazilerinin yok olmasına yol açmaktadır. Kentsel tarım, hobi bahçeleri ve çatı tarımı gibi inovatif yaklaşımlar, bu olumsuz etkileri hafifletmeye ve tarımın karbon ayak izini düşürmeye yönelik modern çözümler sunmaktadır (Kurban, D., & Zengin, G.2023). Hidroponik tarım sistemi bu projenin odak noktasını oluşturmaktadır. Özellikle hobi bahçesi uygulamaları için özelleştirilen bu sistem, topraksız tarım hobi setlerinin modifikasyonuyla oluşturulacaktır. Sistem tasarımında, piyasada mevcut Ovaçiftlik topraksız tarım hobi seti referans alınarak temel bir yapı oluşturulmuş (Ovaçiftlik, 2023) ve yapı geliştirilmiştir. Şekil 1'de genel mimarisi verilen benzer bir yapı üzerine kurgulanan sistem, suyun döngüsel hareketi sırasında sıvı gübre gibi besinlerin otomatik olarak dozajlanabilmesi için tasarlanmıştır. Bu otomatik dozajlama işlemi, suyun pH ve EC (elektrik iletkenlik) değerlerini Arduino geliştirme kartıyla sürekli olarak kontrol ederek gerçekleştirilmektedir. Gelişmiş bir mobil uygulama aracılığıyla bu sistem, kullanıcının bitki durumunu uzaktan takip etmesine ve kontrol etmesine olanak tanır. Sistemin izlenmesi ve yönetimi için tasarlanan mobil uygulamanın örnek bir ekran görüntüsü Şekil 2'de sunulmuştur.

Projemizde, bitkinin gelişim sürecini yakından takip edebilmek ve potansiyel hastalıklarını erken aşamada tespit edebilmek amacıyla ileri düzeyde görüntü işleme teknolojileriyle donatılmış bir kamera sistemi entegrasyonu gerçekleştirdik. Bu entegre kamera sistemi, bitkinin yaşam döngüsü boyunca geçirdiği evreleri detaylı bir şekilde analiz eder, böylece çeşitli büyüme aşamalarını belirleyebilir. Aynı zamanda, bu teknoloji sayesinde hastalıkların en erken evrelerde saptanması, bu sayede hastalığa spesifik müdahalelerin zamanında yapılabilmesi mümkün hale gelir. Böylece, çilek yetiştiriciliğinde karşılaşılan sıkıntıların önüne geçmek ve en yüksek verimi elde edebilmek için proaktif bir yaklaşım benimsenmiş olur. Ayrıca, bu yaklaşım, tarım pratiğini sadece daha verimli kılmakla kalmaz, aynı zamanda uzun vadede sürdürülebilir bir çilek yetiştiriciliği pratiği oluşturmayı da hedefler. Bu teknolojinin potansiyeli, bitki sağlığının sürekli izlenmesi ve

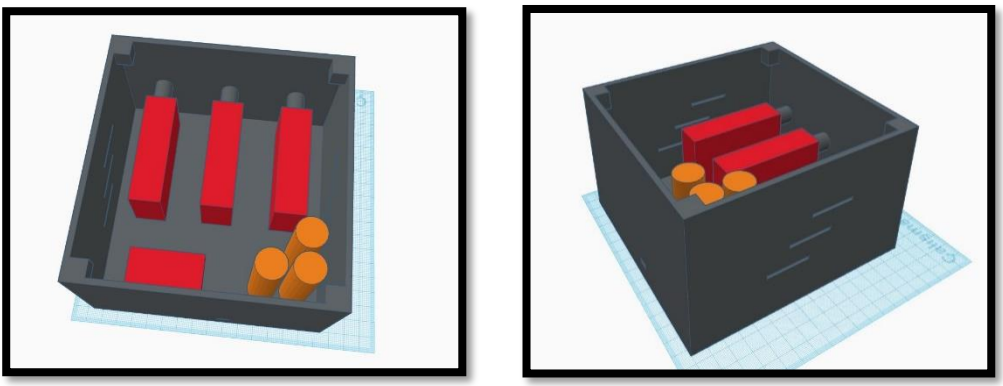
optimize edilmesiyle, sadece hastalıkların önlenmesine değil, aynı zamanda çileğin genel kalitesinin ve veriminin artırılmasına da katkıda bulunabilir.



Şekil 1. Tasarlanan Genel Sistem Mimarisi

Şekil 2. Sistemin izlenmesi ve yönetimi için tasarlanan mobil uygulamanın örnek bir ekran görüntüsü

Sistemde tasarlanan otomatik dozajlama ünitesi Şekil 3'te, 3 boyutlu yazıcı ile alınan ilk prototip görseli ise Şekil 4'te sunulmuştur.



Şekil 3. Otomatik dozajlama ünitesi tasarım görselleri

Şekil 4. otomatik dozajlama ünitesi ilk prototip görseli

Buna göre otomatik Dozajlama sistemi için ana bileşenler, çalışma prensibi, montaj ve kurulum aşağıda maddeler halinde açıklanmıştır.

Ana bileşenler:

- **Arduino Kartı (Geliştirme Kartı):** Bu kart, sistemdeki tüm bileşenleri bir araya getiren merkezdir. Sensörlerden gelen bilgileri okuma, işleme ve belirli komutları gerçekleştirme görevini üstlenir.
- **pH ve EC Sensörleri:** Suyun pH ve EC (iletkenlik) değerlerini ölçmek için kullanılır. Bu sensörler, bitkinin beslenmesi için suya eklenmesi gereken besin miktarını belirlemek için çok önemlidir.
- **Peristaltik Pompalar:** Belirlenen değerlere göre sıvı gübrenin dozajını ayarlamak için kullanılır. Bu pompalar, Arduino tarafından kontrol edilir.
- **SD Kart Modülü:** Ölçülen değerleri ve yapılan dozajlama miktarlarını kaydedebilmesi için hafıza modülü olarak eklenmiştir.
- **Wi-Fi Modülü (ESP8266):** Arduino'ya internet erişimi sağlar, böylece veriler mobil uygulamaya aktarılabilir.
- **Güç Kaynağı:** Tüm sistemi çalıştırmak için gerekli enerjiyi sağlar.

Çalışma Prensibi:

- **Ölçüm:** pH ve EC sensörleri belirli aralıklarla suyun değerlerini ölçer ve bu bilgiyi Arduino'ya iletilir.
- **Karar Mekanizması:** Arduino, okunan değerleri mobil uygulamada önceden belirlenen ideal değerlerle karşılaştırır. Eğer ölçülen değerler ideal değerlerden farklıysa, bu durumu düzeltmek için sıvı gübre eklenmesi gerekip gerekmediğine karar verilir.

- **Dozajlama:** Eğer sıvı gübre eklenmesi gerekiyorsa, Arduino peristaltik pompaları belirli bir süre boyunca çalıştırarak suya gerekli miktarda sıvı gübre ekler. Bu dozaj miktarı, pH ve EC değerlerine bağlı olarak değişkenlik gösterebilir.
- **Veri Kaydedilmesi ve İletilmesi:** Arduino, yaptığı işlemleri ve ölçümleri SD karta kaydeder. Aynı zamanda bu bilgiler Wi-Fi modülü aracılığıyla mobil uygulamaya iletilir. Böylece kullanıcı, gerçek zamanlı olarak sistemin durumunu takip edebilir ve gerektiğinde müdahale edebilir.
- **Otomatik Kalibrasyon:** Sistemin doğru çalışabilmesi için pH ve EC sensörlerinin düzenli olarak kalibre edilmesi gerekmektedir. Arduino, belirli zaman aralıklarında otomatik olarak bu sensörleri kalibre eder.

Montaj ve Kurulum:

Sistem, hobi bahçeleri için tasarlandığından kompakt bir yapıya sahip olması hedeflenmiştir. Sensörler, su tankının içine yerleştirilirken, peristaltik pompalar, sıvı gübre konteynerlarına bağlanır. Arduino, tüm bileşenleri bir araya getiren merkezi bir kutuda yer alır. Bu kutu suya dayanıklıdır ve dış etkenlere karşı korumalıdır.

Son olarak, kullanıcının kolaylıkla sisteme erişebilmesi için mobil uygulama aracılığıyla kullanıcı dostu bir arayüz sunulmuştur. Bu arayüz sayesinde kullanıcı, ölçümleri görebilir, dozajlama ayarlarını değiştirebilir ve tüm sistem hakkında bilgi alabilir.

Sonuç

Topraksız tarımın giderek popülerleşen bir yaklaşım olduğu, çevremizdeki pek çok uygulamayla gözlemlenmektedir. Özellikle sürdürülebilir tarım anlayışının bir parçası olarak değerlendirilen bu yaklaşım, doğal kaynakların korunmasında önemli bir rol oynamaktadır. Projemiz kapsamında geliştirilen hidroponik sistem, besinlerin otomatik dozajlanması özelliği ile sulama ve gübreleme

konusundaki sorunları çözmeyi amaçlamaktadır. Bu özellik, suyun PH ve EC değerlerine göre gerçek zamanlı ayarlamalar yaparak optimum bitki büyümesini desteklemektedir. Geliştirilen özgün mobil uygulama sayesinde kullanıcılar, bitkinin genel durumunu, hastalıklarını ve büyüme sürecini kolaylıkla takip edebilmektedir. Bu, bitki sağlığı ve verimliliği açısından büyük bir avantaj sağlamaktadır. Sistemin hobi bahçelerine özel tasarlanmış olması, kullanıcının tarım uygulamalarını kolaylaştırma ve ev ortamında verimli bir tarım deneyimi yaşama amacına hizmet etmektedir. Projede elde edilen %80-90 oranındaki su tasarrufu, özellikle su kaynaklarının kısıtlı olduğu bölgelerde büyük bir avantajdır. Bununla birlikte gübre tasarrufu da ekonomik ve ekolojik açıdan büyük bir katkıdır. Otomasyon teknolojilerinin entegrasyonu sayesinde, hata oranları minimize edilmiş, böylece kullanıcı hatalarından kaynaklanabilecek olumsuzlukların önüne geçilmiştir.

KAYNAKÇA

Ağızan, K., Bayramoğlu, Z., & Ağızan, S. (2022). Akıllı Tarım Teknolojilerinin Tarımsal İşletme Yöneticiliğine Sunduğu Avantajlar. *Turkish Journal of Agriculture-Food Science and Technology*, 10(9), 1697-1706.

Bayramoğlu, Z. (2010). Tarımsal Verimlilik ve Önemi. *Selcuk Journal of Agriculture and Food Sciences*, 24(3), 52-61.

Demirsoy, L., Misir, D., & Nafiye, A. (2017). Topraksız tarımda çilek yetiştiriciliği. *Anadolu Ege Tarımsal Araştırma Enstitüsü Dergisi*, 27(1), 71-80.

Ercan, T., & Kutay, M. (2016). Endüstride nesnelerin interneti (IoT) uygulamaları. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 16(3), 599-607.

Kurban, D., & Zengin, G. (2023). Sürdürülebilir Kent Yaklaşımlarından Kentsel ve Topraksız Tarım: Paris, Barselona ve İzmir Örnekler. *Balkan and Near Eastern Journal of Social Sciences, 2023(09), 90-101*.

Lopez-Aranda, J. M., Miranda, L., Medina, J. J., Soria, C., de los Santos, B., Romero, F., ... & Santos, B. M. (2009). Methyl bromide alternatives for high tunnel strawberry production in southern Spain. *HortTechnology*, 19(1), 187-192.

Oğuz, İ., İbrahim Oğuz, H., & Ebru Kafkas, N. (2022). Strawberry Cultivation Techniques. *IntechOpen*. doi: 10.5772/intechopen.104611

Ovaçiftlik (2023). Ovafarm topraksiz tarim baslangic hobi seti 2023. (23/08/2015 tarihinde http://tuik.gov.tr/PreHaberBultenleri.do?id=16198 adresinden ulaşılmıştır).

BÖLÜM V

Arama Motorları İçin Temel Gerçekleri ve Makine Öğrenmesini Kullanan Sorgu Öneri Sistemi

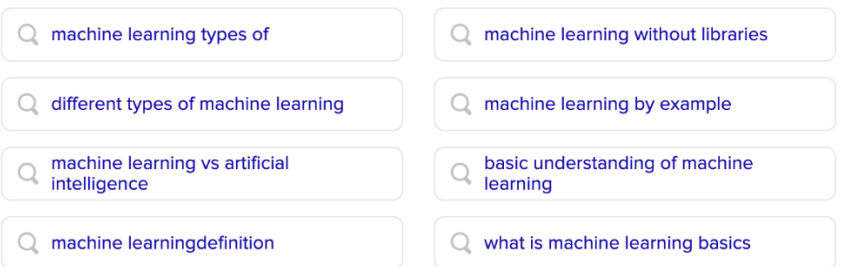
Fatih ÇELİK[1]
Sibel SENAN[2]

Giriş

Günümüzde internet yaygın bir kullanım ağına kavuşmuştur. Artık pek çok insan mobil cihazlarında yardımı ile çevrimiçi dünyada vakit geçirmeye başlamıştır. Çevrimiçi dünyadaki artan kullanıcı etkileşimi ile bu dünyadaki veri miktarında da ciddi bir artış meydana gelmiştir. Bu artışın ana nedeni, değişen dünya ile çevrimiçi dünyadaki fırsatlarını kaçırılmama arzusudur. Örneğin, kullanıcı bir televizyon araması yaptığında birçok satıcı kullanıcının televizyon aradığını bilerek buna yönelik aksiyonları geliştirmektedir. Büyüyen veri ile, kullanıcıların istediği şeyleri bulabilmeleri zorlaşmakta ve kullanıcıya istediği şeyleri doğru

[1] İstanbul Üniversitesi-Cerrahpaşa, Bilgisayar Mühendisliği Bölümü, 0000-0001-5330-978X
[2] Doç. Dr., İstanbul Üniversitesi-Cerrahpaşa, Bilgisayar Mühendisliği Bölümü, 0000-0001-6773-0428

olarak sunabilmek önem kazanmaktadır. Arama motorları, tam olarak bu önemli görevi yerine getirmek amacı ile çalışan makinelerdir. Arama motorları temel olarak kullanıcılara aradıkları şeyi hızlı ve doğru olarak sunmayı amaçlar. Bazı durumlarda kullanıcının kelimelerini anlamak, arama motorları için zor olabilir. İnsan ifadelerini makinenin anlayabileceği şekilde ifade etmek zordur. Ayrıca kullanıcılar, bazı durumlarda ne aradığını tam olarak bilmiyor olabilirler. Bunun gibi koşullarda, kullanıcılara aramak istediklerini doğru ve hızlı olarak sunmak önem kazanır. Sorgu önerileri, kullanıcı sorgularındaki anlamlardan yola çıkarak kullanıcılara aramak istediklerini en uygun şekilde sunmayı amaçlar. Google [3], DuckDuckGo [4], Yahoo [5] gibi çoğu arama motoru, kullanıcılara daha iyi bir deneyim sunmak adına sorgu önerilerini kullanır. DuckDuckGo'ya ait bir sorgu önerisi Şekil 1'de gösterilmiştir. Dijital veri miktarının giderek büyüdüğü günümüzde, etkili sorgu sistemlerini oluşturmak arama motorları için büyük önem taşımaya devam etmektedir. Bu çalışma, arama motorlarının etkinliğini artırmak için sorgu önerilerinde makine öğrenimi ve temel gerçeklerin beraber kullanımının sorgu önerilerinin başarısını artırabileceğini göstermektedir.



***Şekil 1**. DuckDuckGo arama motoruna ait sorgu önerisi.*

Arama motorları sorgu önerilerini mevcutta kullanıyor olsada, kullanıcıların arama niyetlerini okumak ve bu niyetleri

[3] https://www.google.com
[4] https://duckduckgo.com
[5] https://search.yahoo.com

sonuçlarla eşleştirmek zor olduğu için hala yeterince verimli değildir. Mevcut sorgu önerisi yöntemleri belirli seviyede başarılı sayılabilir fakat çoğu zaman kullanıcının isteklerine tam olarak cevap veremez. Bunun altında yatan ana neden kullanıcının sorgularındaki belirsizlik ve değişkenliklerdir. Çalışma, kullanıcı sorgularındaki belirsizlik ve değişkenlikleri en aza indirmek için çeşitli yöntemleri kullanarak temel gerçekleri oluşturmaktadır. Ardından farklı yöntemlerle oluşturulmuş olan temel gerçekler, bir makine öğrenmesi modeli kullanılarak değerlendirilmiştir. Çalışmanın amacı, sorgu önerisi doğasından kaynaklanan hedef belirleme zorluğunu çeşitli yöntemler uygulayarak ortadan kaldırmak ve bu çözümün ardından uygulanan yöntemlerin etkisini gözlemek üzere makine öğrenmesi modeli gerçekleştirmektir. Bu durum, temel gerçekleri belirlemede kullanılan yöntemlerin doğruluğunu test etmeye olanak tanır.

Sorgu önerileri ile ilgili olarak literatürde pek çok çalışma mevcuttur. Sorgu önerisi için farklı yöntemler geçmişten günümüze dek kullanılmıştır. Bu yöntemlerden bazıları, semantik tabanlı yaklaşımlar (Barman et al., 2020), çizge tabanlı yaklaşımlar (Syed et al., 2022) ve makine öğrenmesi tabanlı yaklaşımlardır (Puthiya Parambath et al., 2022). Çizge tabanlı yaklaşımlar, sorguları, sayfaları, kullanıcıları ve terimleri içeren çeşitli çizge yaklaşımlarını kullanır. Bu çizgeler daha sonra sorguların arasındaki benzerlikleri ve yararlı kalıpları çıkarmak için kullanılır. TaSQS (Miyanishi & Sakai, 2013) ile, belirli bir zamandaki popülariteye göre sorgu-tıklama verilerini kullanarak bir çizge oluşturulmuştur. Bu çizgeyi kullanarak sorgu önerileri, bir zaman çizelgesinde sunulmuştur. Geçmiş tarihlerdeki önerilere göre bu sorgu önerileri *HAC (Hierarchical Agglomerative Clustering)* algoritması kullanılarak kümelenmiştir. Google aramaları da dahil çeşitli basit referans değerlerinde benzer çalışmalara göre daha iyi sonuçlar elde edilmiştir. Web aramalarında çeşitlendirme ve kişiselleştirmeyi birleştiren *QS-DP (Query Suggestion with Diversification and Personalization)* (Jiang et al., 2015), sorgu önerileri için çok parçalı çizgeleri kullanmıştır. Bu çizgeler yardımıyla sorgu önerileri

çeşitlendirilmiştir. Çalışmanın kişiselleştirme bölümü için, *UPM (User Profiling Model)* kullanılmıştır. Bahsedilen iki modeli birleştirerek tek bir modelde sunan QS-DP, ticari bir arama motoruna ait verilerle yapılan testlerde birlikteliğin birbirini geliştirebileceğini göstermiştir. Bir kullanıcı oturumundaki davranışını modelleyerek, bu modeli birden fazla *Transformer* mimarisinde uygulayan çalışma (Mustar et al., 2022), sorguları farklı *Tokenizer*'ları kullanarak önceden eğitilmiş düz ve hiyerarşik mimarilere sahip *Encoder*'lar ile test eder. Yapılan değerlendirmeler, *Transformer* tabanlı mimarilerin, *RNN* tabanlı yöntemlere göre daha başarılı olduğunu göstermektedir Ayrıca, hiyerarşik yöntemlerin sorgu önerileri için genel olarak iyi performans gösterdiği ve düz modellerin karmaşık ve uzun sorgular için daha uygun olduğu belirtilmiştir. Arama önerisi için *Block-Level Optimization*'ı kullanan yaklaşım (Kohli, 2020), uçtan uca derin öğrenme mimarisinin, herhangi bir zamanda kendisine sunulan diğer tüm önerilerin, eşzamanlı olarak farkında olurken bütünsel bir şekilde öneriler sunabileceğini göstermiştir. Bu öneriler, karma amaçlı optimizasyon yöntemi ve çeşitli ödül fonksiyonları kullanılarak sıralanmıştır. Bu mimarinin faydaları geleneksel yaklaşımlara göre tartışılmıştır. Sistem, tıklama oranlarının ve çeşitliliğin eş zamanlı optimizasyonu göz önünde bulundurularak eğitilmiş ve sistemin farklı senaryolarda, herhangi bir çoklu öneri sistemine ölçeklenebileceği belirtilmiştir. Başka bir çalışmada, bilgi çizgelerini kullanarak daha ayrıntılı ve alakalı sorgu önerisi için sezgisel önerilerle genişletme yapan ve oluşturulan çizgelerde arama paradigmaları için iyi kurulmuş teknikler ile arada köprü oluşturan bir model önerilmiştir (Lissandrini et al., 2020). Yapılan çalışma ek bilgi kullanmadan, ilkeli dil modellemeyi kullanmıştır. Gerçek kullanıcılarla yapılan testler, keşif amaçlı arama yapan kullanıcılara yardımcı olmada etkinliğini ve anlamlılığını kanıtlamıştır. Çeşitli puan türlerinde rakiplerini geride bırakmıştır. Geri bildirim oturumları kullanıcıların arama sorgularına sunulan öneriler için yardımcı kaynak olarak kullanılabilmektedir. Bir diğer çalışmada, geri bildirim oturumunda tıklanan ve tıklanmayan ve belgelerde

sunulan anahtar kelimeleri belirleyen *Related Search Recommendation (RSR)* çerçevesi sunulmuştur. Geri bildirim oturumları, zenginleştirilmiş belgelere çevrilmiştir. Ardından çevrilmiş belgeler kullanılarak, sözde belgeleri oluşturulmuştur. Bu sözde belgeler birleştirilerek optimize edilmiş ve optimize edilmiş sözde belgeler oluşturulmuştur. Bu belgelerde çeşitli benzerlik yöntemleri kullanılarak sorgu önerileri oluşturulmuştur. Model çeşitli yöntemlerle kıyaslanmış, performansı değerlendirmek için *Click-Through Rate (CTR)* kullanılmıştır. Önerilen model daha yüksek *CTR* sağlamıştır.

Sorgu önerileri, kullanıcıların çevrimiçi dünyadaki taleplerini karşılamak için önemli bir görevdir. Hala gelişim potansiyeli mevcuttur ve etkinliği önemli derecede artırabilir. Sorguların karmaşık anlamları ve kullanıcı hedeflerinin belirlenmesindeki zorluklar nedeniyle hedef kümesinin olmaması yapay zekâ çalışmaları için zorlayıcıdır. Bu çalışma, kullanıcılara çeşitli yapay zekâ görevlerini kullanarak sorgu önerisi sistemleri hazırlamak için çeşitli etiketli veri hazırlama yöntemleri sunmaktadır. Bu yöntemler, makine öğrenmesi modeli üzerinde test edilerek yöntemlerin verimliliği de sunulmaktadır. Özetle bu çalışma, sorgu önerileri için doğru etiket verilerinin hazırlanarak bu verilerle makine öğrenmesi gibi yöntemleri kullanılarak sorgu önerisi sistemleri geliştirmelerine katkıda bulunur.

Yöntem

Veri Kümesi

AOL veri kümesi (Pass et al., 2006) 2006 yılında yayınlanmıştır. Veri kümesi, AOL arama motorunda kullanıcıların yaptığı aramaları, çeşitli bilgiler ile bulundurmaktadır. Veri kümesi yaklaşık olarak 36 milyon sorgu içerir ve 2006 yılının 3 aylık periyodunu kapsamaktadır. Kullanıcılar anonimleştirilmiş *AnonID* ile ifade edilirken, *Query* sorgu cümleciğini içerir. *QueryTime* kullanıcının arama yaptığı anı ifade ederken, *ItemRank* ve *ClickURL* bir tıklama eylemini içerir. Tıklama eylemi içermeyen sorgular için

bu iki alan boş bırakılmıştır. Veri kümesine ait bazı bilgiler Tablo 1'de gösterilmiştir. Ayrıca *Query* alanı boş olan veriler sorgu günlüklerinden çıkarılmıştır.

Tablo 1. AOL sorgu günlüklerine ait bazı istatistikler.

Toplam Arama	36389567
Tıklama İçeren Arama	16946938
Tıklama İçermeyen Arama	19442629
Benzersiz Tıklama	1632788

AOL sorgu günlüklerini kullanarak sorgu önerisi alanında farklı yöntemleri kullanan çeşitli çalışmalar yapılmıştır (Deepak & Santhanavijayan, 2022; Mustar et al., 2022; Venugopal & Santosh Nimbhorkar, 2020).

Temel Gerçek Tanımları

Temel gerçek, matematik bilimlerinde ve yapay zekâ uygulamalarında mutlak gerçekleri ifade etmek için kullanılır. Bir makine öğrenmesi uygulamasında temel gerçekler etiket verileri olarak adlandırılabilir. Temek gerçekler özellikle makine öğrenmesi alanında model başarısını ve performansını ciddi şekilde etkiler. Çalışmada kullanılan AOL sorgu günlüklerinin, bir makine öğrenmesi problemi olarak ifade edilmesi oldukça zordur çünkü bu veriler, etiketleri içermez. Makine öğrenmesi yöntemleri, öğrenmek için veriler ile etiketleri de göz önünde bulundurur. Çalışma makine öğrenmesi modellerinde kullanmak üzere sorgu günlüklerinden temel gerçekleri çıkararak tanımlar. Bunun için 3 farklı temel gerçek belirleme yöntemini kullanır. Birinci yöntem kullanıcı sorgularını vektör olarak ifade ederek bu sorgulara en yakın sorguları bulmaktır. Bunun için *cosine similarity* kullanılmıştır. Bir diğer yöntemde, kullanıcı oturumlarını analiz edilerek oturumların benzerliklerinden yararlanılmıştır. Son yöntemde ise kullanıcıların arama sonuçlarına göre tıklamış olduğu verilerden destek alınarak temel gerçekler

oluşturulmuştur. Temel gerçeklere ait örnekler Tablo 2'de gösterilmiştir.

$$\text{Cosine}(x, y) = \frac{x \cdot y}{|x||y|}$$

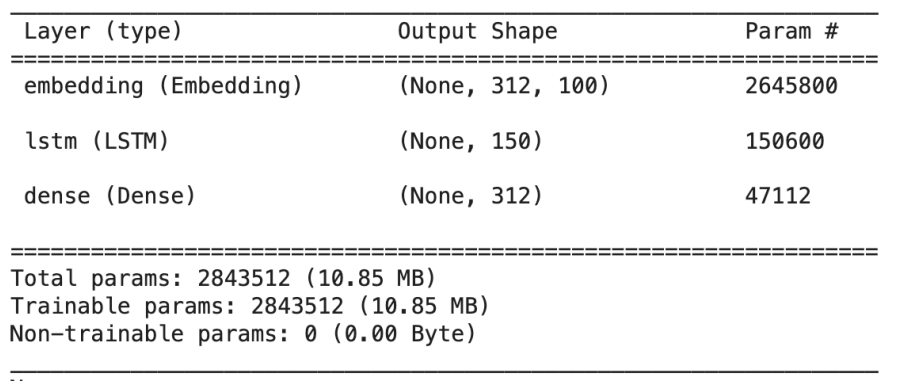
Tablo 2. Farklı yöntemler ile oluşturulan temel gerçek örnekleri.

Query	Method 1	Method 2	Method 3
funny jokes	funny videos	stupid videos	ol funny jokes
tupac	eminem	tupac fans	tupac online
virginia lottery	georgia lottery	mega millions	lottery yahoo

Makine Öğrenmesi

Makine öğrenmesi, yapay zekâ alt konu başlıklarından bir tanesidir. Makine öğrenmesi, verileri kullanarak öğrenir ve bu verileri kullanarak tahminler ve kararlar üretir. Çeşitli makine öğrenmesi yaklaşımları günümüzde kullanılmaktadır. Makine öğrenmesinin, konuşma tanıma, görüntü işleme, finans, öneri sistemleri ve sağlık hizmetleri gibi birçok alanda çok çeşitli kullanımı mevcuttur. Sorgu önerisinde de makine öğrenmesi güncel ve talep gören bir yöntem olmuştur. Çalışma, makine öğrenmesi yöntemlerinden biri olan yapay sinir ağları kullanılmıştır. Yapay sinir ağları, verilerin içerisindeki karmaşık ve daha derin anlamların anlaşılması için yaygın olarak kullanılmaktadır. Çalışma, RNN yöntemlerinin verilerin arasındaki ilişkileri yakalamadaki problemlerini ve uzun dizilerdeki dizinin önceki elamanlarının değerlendirilen eleman ile aralarındaki ilişkiyi yakalamadaki zayıflıklarını önemli ölçüde azaltan *Long-Short Term Memory (LSTM)* (Hochreiter & Schmidhuber, 1997) kullanmaktadır. Sorgu önerileri alanında, *LSTM* yöntemi oldukça yaygın olarak kullanılmaktadır. Daha önce çeşitli *LSTM* modelleri sorgu önerileri için sunulmuştur (Ahmad et al., 2018, 2019). Daha önceki aşamada oluşturulan üç farklı temel gerçek kümesi ile kullanıcı oturumlarındaki son aramaları hedef kabul eden veri kümesi, *LSTM*

yöntemi kullanılarak eğitilmiştir. Eğitim için kullanılan *LSTM* modeli Şekil 3'te gösterilmiştir.



Şekil 2. LSTM modeline ait bilgiler.

Bulgular

Bu kısımda önerilen *LSTM* modeli, oluşturulan temel gerçek tanımları kullanılarak ve bir oturum boyunca oluşan sorgulara ait tıklama sonuçları hedef olarak işaretlenerek test edilmiştir. *LSTM* modellerinde veri kümesi içerisinden tıklamaya sahip 278.000 veri seçilmiştir. Bu veri kümesi, eğitim için 222.400 doğrulama için 55.600 adet olmak üzere bölünmüştür. *LSTM* modelinin kayıp fonksiyonu için *softmax* kullanılmıştır. Model öncelikle, oluşturulan temel gerçekler kullanılarak eğitilmiştir. Daha sonra temel gerçeklerin başarısını değerlendirebilmek için tıklama verilerini kullanarak oluşturulan hedef kümesi model üzerinde eğitilmiştir. Test verileri üzerinde yapılan değerlendirmeler, Model performansı, makine çevirisinin kalitesini ölçmek için kullanılan *BLEU* ve bir modelin örneği ne kadar iyi tahmin ettiğini gösteren *Perplexity* metrikleri kullanılarak değerlendirilmiştir. Modellere ait *BLEU* ve *Perplexity* değerleri Tablo 3'te gösterilmiştir. Temel gerçekler kullanılarak oluşturulan model performansı *LSTM-wGT* ile tıklama verilerini hedef olarak kullanan model *LSTM-wCU* ile ifade edilmiştir. Hazırlanan iki modele ait performanslar ölçüm için kullanılan metrikler ile değerlendirildiğinde çalışmada hazırlanan

temel gerçek kümesinin tıklama verilerini kullanan modele göre *BLEU* (Papineni et al., 2002) ve *Perplexity* açısından daha iyi sonuç verdiği gözlemlenmiştir. Bu durumun ana etkeninin, çalışmada oluşturulan temel gerçeklerin belirlenmesi için kullanılan yöntemler olduğu düşünülmektedir. Çünkü önerilen temel gerçek belirleme yöntemleri, sadece kullanıcının tıklama verilerini değil kullanıcıların sorgularının altında yatan temel anlamları, kullanıcıların birbirleri ile benzerliklerini ve tıklama verilerini de dikkate alan hibrit bir yaklaşımı kullanır. Bu hibrit yaklaşım kullanıcı sorgularını değerlendirmekte çok yönlü bakışların daha güçlü olabileceğini göstermektedir.

Model	BLEU				Perplexity
	1	2	3	4	
LSTM-wGT	18.41	14.24	10.91	7.25	68.49
LSTM-wCU	13.24	11.77	8.33	5.90	93.41

Sonuç

Sorgu önerisi, günümüzde arama motorları için önemli bir görevdir. Bu görev için birçok farklı yöntem yaygın olarak kullanılmıştır. Bu çalışmada bu yöntemlerden biri olan makine öğrenmesi yöntemi kullanılmıştır. Bu yöntem öncesinde, veri kümesinde etiket verileri doğrudan bulunmadığı için makine öğrenmesi modelinin öğrenmesini sağlayacak olan etiket verileri iki farklı şekilde hazırlanmıştır. Bu hazırlanan iki farklı etiket veri kümesi bir makine öğrenmesi modeli olan *LSTM* kullanılarak eğitilmiştir. Eğitim aşamasının ardından, *BLEU* ve *Perplexity* değerlendirme metrikleri ile model performansı ölçülmüştür. *LSTM-wGT* modelinin, *LSTM-wCU*'ya göre daha başarılı olduğu görülmüştür. Bunun ana nedeninin çalışmada oluşturulan temel gerçeklerin belirlenmesi için kullanılan yöntemler olduğu düşünülmektedir. Çünkü önerilen temel gerçek belirleme yöntemleri, sadece kullanıcının tıklama verilerine bakarak değil, sorguların altında yatan temel anlamları anlamaya çalışan,

kullanıcıların aramaları arasındaki benzerlikleri kullanan ve kullanıcıların benzer tıklama özelliklerini ele alan hibrit bir yaklaşımı kullanır. Ölçümler ve değerlendirmeler sonucunda, çalışmada önerilen temel gerçek yönteminin başarısı kanıtlanmıştır.

Kaynakça

Ahmad, W. U., Chang, K. W., & Wang, H. (2018). Multi-task learning for document ranking and query suggestion. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Ahmad, W. U., Chang, K. W., & Wang, H. (2019). Context attentive document ranking and query suggestion. *SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://doi.org/10.1145/3331184.3331246

Barman, D., Sarkar, R., Tudu, A., & Chowdhury, N. (2020). Personalized query recommendation system : A genetic algorithm approach. *Journal of Interdisciplinary Mathematics*, *23*(2). https://doi.org/10.1080/09720502.2020.1731964

Deepak, G., & Santhanavijayan, A. (2022). UQSCM-RFD: A query–knowledge interfacing approach for diversified query recommendation in semantic search based on river flow dynamics and dynamic user interaction. *Neural Computing and Applications*, *34*(1). https://doi.org/10.1007/s00521-021-06404-w

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jiang, D., Leung, K. W. T., Yang, L., & Ng, W. (2015). Query suggestion with diversification and personalization. *Knowledge-Based Systems*, *89*. https://doi.org/10.1016/j.knosys.2015.09.003

Kohli, H. (2020). Training Mixed-Objective Pointing Decoders for Block-Level Optimization in Search Recommendation. *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://doi.org/10.1145/3397271.3401236

Lissandrini, M., Mottin, D., Palpanas, T., & Velegrakis, Y. (2020). Graph-Query Suggestions for Knowledge Graph Exploration. *The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020*. https://doi.org/10.1145/3366423.3380005

Miyanishi, T., & Sakai, T. (2013). Time-aware structured query suggestion. *SIGIR 2013 - Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://doi.org/10.1145/2484028.2484143

Mustar, A., Lamprier, S., & Piwowarski, B. (2022). On the Study of Transformers for Query Suggestion. *ACM Transactions on Information Systems*, *40*(1). https://doi.org/10.1145/3470562

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: A method for automatic evaluation of machine translation. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, *2002-July*.

Pass, G., Chowdhury, A., & Torgeson, C. (2006). A picture of search. *ACM International Conference Proceeding Series*, *152*. https://doi.org/10.1145/1146847.1146848

Puthiya Parambath, S. A., Anagnostopoulos, C., & Murray-Smith, R. (2022). *Improving Sequential Query Recommendation with Immediate User Feedback*.

Syed, M. H., Huy, T. Q. B., & Chung, S. T. (2022). Context-Aware Explainable Recommendation Based on Domain Knowledge Graph. *Big Data and Cognitive Computing*, *6*(1). https://doi.org/10.3390/bdcc6010011

Venugopal, K. R., & Santosh Nimbhorkar, S. (2020). Related search recommendation with user feedback session. In *Web Recommendations Systems*. https://doi.org/10.1007/978-981-15-2513-1_6

BÖLÜM VI

Makine Öğrenimi Temelli Regresyon Yöntemleri ile Çevrimiçi Satış Adeti Tahmini: E-ticaret İçin Ampirik Bir Çalışma

Özlem Yakar[1]
Mahamoud Brahim Adoum[2]
Alper Anapalı[3]
Batuhan Furkan Saatçi[4]
Buket Doğan[5]

1.Giriş

Günümüzün teknoloji çağında veri üretimi sürekli büyüyerek genişlemektedir. Bu durum, sürekli üretilen verilerin anlamlı hale getirilmek üzere büyük veri halinde beklemektedir. Büyük veri kavramı, verilerin saklanmasında, analiz edilmesinde ve

[1] Özlem Yakar, Dr. öğrencisi, Marmara Üniversitesi, Bilgisayar Mühendisliği
[2] Mahamoud Brahim Adoum, Yüksek Bilgisayar Mühendisi, Trakya Üniversitesi, Bilgisayar Mühendisliği
[3] Alper Anapalı, Yazılım Mühendisi, Ideasoft Yazılım San. ve Tic. A.Ş.
[4] Batuhan Furkan Saatçi, JR. İş Analisti, Ideasoft Yazılım San. ve Tic. A.Ş.
[5] Buket Doğan, Doç.Dr., Marmara Üniversitesi, Bilgisayar Mühendisliği

yönetilmesinde klasik veritabanı yönetim sistemlerinin yetersiz kaldığı durumlarda karşımıza çıkan bir problem olarak tanımlanabilir (Terzi, Sağıroğlu, & Demirezen, 2017). Bu problem için, yapılandırılmamış verilerden yapılandırılmış olanlara ilişkin büyük veri kümelerini analiz edebilen sistemler, iş dünyasının anahtar faktörlerinden biri olarak kabul edilir (Yıldız A. , 2022). Teknolojinin gelişimi sonucu giderek artan büyük verinin içeriğinin analiz edilerek verilerden anlamlı bilgi çıkarma ve karar desteği olarak kullanılmasına yönelik çalışmalar hem akademik hem de sektörel olarak birçok alanda yerini almaktadır.

Son yıllarda yapılan araştırmalarda, büyük verilerin tahminlenmesinde istatistiksel algoritma modellerinden regresyon tabanlı modellerin de tercih edildiği görülmektedir. Regresyon analizi, değişkenler arasındaki ilişkilerin araştırılmasında kullanılan bir istatistiksel yaklaşımdır (Sykes, 1993). Bu yaklaşım, iki amaç doğrultusunda kullanılır. Birinci amacı, öngörü ve tahmin problemlerini çözmektir. İkincisi amacı ise, tahminleme için kullanılacak bağımlı ve bağımsız değişkenler arasındaki neden-sonuç ilişkisini bulmaktır. Veri analizinde, genellikle çok yüksek sayıda bağımsız değişken içeren veriler bulunmaktadır. Yüksek boyutlu bu verileri işlemek için de daha iyi önişleme aşamasından geçirilmiş nitelikler, uygun parametre ve teknolojiyle uygulanmış regresyon tekniklerine ihtiyaç duyulmaktadır.

Regresyon analizindeki gelişmeler, çevrimiçi alışveriş (e-ticaret) alanında da görülmektedir. E-ticaret, hayatımıza hızlıca yerleşmiş bir kavram olup, giderek büyüyen bir sektör halindedir. Günümüzün alışveriş kavramı incelendiğinde, çevrimiçi alışverişte geçen süre ve alışveriş davranışlarının kendine özgü niteliklerinin olduğu görülmüştür. Satış adedi tahmini için, müşteri alışkanlıklarının değerlendirilmesi e-ticarette önemli bir yer tutmaktadır. Bu durumu bilen ve e-ticaretle ilgilenen işletmeler de müşterilerin bilgilerini kullanarak, büyük veriler üzerinde farklı tahminlemeler yapmaktadırlar.

Bu çalışmada, regresyon yöntemleri kullanılarak makine öğrenimine dayalı çevrimiçi satış adedi tahmini yapılmıştır. Çalışmanın amacı, modelin gerçek satış değerlerine göre ne kadar doğru tahmin yaptığını ölçmektir. Çalışmada, satış adedi tahmini için e-ticaret alanındaki büyük bir şirkete ait e-ticaret veri seti kullanılmıştır. Çalışmanın başlıca katkıları aşağıdaki gibidir:

• Bu çalışmada, e-ticaret alanındaki büyük bir şirkete ait CRM e-ticaret veri seti üzerinde çeşitli veri analizleri yapılmıştır. Bu analizlerde, veriler sayısal ve kategorik öznitelikler şeklinde ayrılmıştır. Sayısal özniteliklerde "Ortalama yöntem (Mean method)" ve kategorik özniteliklerde "Mod yöntemi (Mode method)" kullanılmıştır. Kategorik öznitelikleri sayısal değerlere çevirme işleminde de "Tek-Değer Kodlama (One-Hot Encoding) yöntemi" kullanılmıştır.

• Çalışmada, makine öğrenimine dayalı Çok Değişkenli Lineer Regresyon (MLR), Destek Vektör Regresyonu (SVR), Kement Regresyonu (Lasso regression) ve Sırt Regresyonu (Ridge regression) modelleri seçilerek satış adedi tahmini yapılmıştır.

• Veri analizinin sonuçlara ilişkin içgörü sağlamak için R^2, MSE, RMSE, MAE ve RAE performans hata ölçütleri alınarak değerlendirmeler yapılmıştır.

• Bu çalışmadaki ampirik analizin temel odak noktası, e-ticaretle ilgilenen şirketler için mevcut firma satış verileri ile gerçek satış değerlerine göre ne kadar doğru tahmin yapıldığını ortaya koymaktadır.

• Bu sayede e-ticaret alanında çalışan işletmeler, kurumlar ve şirketler, müşterilerin satın alma olasılıklarını analiz ederek ve gelecekteki satışların sayısal tahmini üzerinde fikir yürüterek şirket politikalarını uygun şekilde hazırlayabilirler.

Çalışmanın bölümleri şu şekilde tanıtılmıştır: Bölüm 2'de ilgili çalışmalara yer verilmiştir. Bölüm3, araştırma metodolojisini sunar. Bölüm 4'te materyal ve yöntem kısmının detayları incelenmiştir. Bölüm 5, deneysel sonuçların yer aldığı kısımdır ve

ayrıntılı analizlerle uygulama sonuçları açıklanmıştır. Bölüm 6'da sonuç ve gelecek çalışmalar düşüncesi tartışılmaktadır.

2.İlgili Çalışmalar

E-ticaret, genellikle tüketicilerin açık internet platformunda tarayıcı/sunucu uygulamasına dayalı olarak çeşitli ticari faaliyetleri yürüttüğü yeni bir iş operasyon modunu ifade eder (Pan & Zhou, 2020). E-ticaret platformlarında müşteriler alışveriş işlemlerini, satıcılarla çevrimiçi ortamda irtibat kurarak elektronik ödeme şeklinde yapmaktadırlar. Günümüzde, ticaretin bir göstergesi olan e-ticaret işlemleri, oldukça genişleyerek farklı kitlelerdeki sektörlere de yansımıştır. İşletmeler, müşterilere hizmet kalitesini arttırabilmek ve kurumsal açıdan kendi gelecek stratejilerini daha iyi belirleyebilmek adına e-ticaret verilerini kullanmaktadırlar. Bu verilerin anlamlı şekilde yorumlanması ve nasıl uygulanacağı da bilimsel açıdan araştırma konusudur.

Bu bölümde, e-ticaret alanında makine öğrenimine dayalı regresyon yöntemleri kullanarak yapılan satış adedi tahmin modellemelerine yer verilmektedir. Çalışmanın ilgili çalışmalar kısmı 2020-2023 yılları arasını kapsamaktadır. Springer, ScienceDirect, IEEE, ACM ve MDPI veritabanlarından yardım alınarak, büyük verilerde tahminlemeler için uygun nitelikte güncel çalışmalar incelenip seçilerek bu bölümde ele alınmıştır.

2.1.Makine öğrenimine dayalı regresyon yöntemleri ile satış tahmini

E-ticaret alanında sıkça başvurulan yöntemlerin başında gelen regresyon modelleri, makine öğrenimi yöntemlerinden denetimli öğrenme (Supervised learning) modeli olarak geçmektedir ve literatürde yayınlamış pek çok çalışma mevcuttur. Manasa ve ark. (Manasa, Gupta, & Narahari, 2020), ev fiyatı satış tahmini oluşturmak istemişlerdir. Bunun için, halka açık 9 özellikli bir ev veri seti üzerinde Çok Değişkenli Doğrusal Regresyon (MLR-En Küçük Kareler modeli), Kement-Sırt Regresyonu (Lasso-Ridge

regression), Destek Vektör Regresyonu (SVR) ve Extreme Gradient Boost Regresyon (XGBoostR) regresyon modelleri kullanılmıştır. Değerlendirme metrikleri olarak R^2, RMSE ve Kök Ortalama Kare Logaritmik Hata (RMLSE) seçilerek, kullanılan regresyon modellerinde değerlendirme metriklerinin sonuçları karşılaştırılmıştır. Deney sonuçlarından, Kement-Sırt Regresyonu (Lasso-Ridge regression) modelinin diğer modellere göre daha başarılı olduğu söylenebilir.

Li ve ark. (Li, Dong, & Han, 2020), çevrimiçi müzayedelerin bitiş fiyatlarını (müzayede satışı) tahmin etmek için MLR ile Kalman filtresini harmanlayan bir MLRKF hibrit modeli önermişlerdir. Bu model, iki veri seti üzerinde e-Bay çevrimiçi müzayedelerin bitiş fiyatlarını tahminlemede uygulanmıştır. Önerilen modelde MLR, Çok Değişkenli Lineer Kement Regresyonu (MLRR), Kement Regresyonu (Lasso regression), Rasgele Orman (RF), Destek Vektör Makinesi (SVM) ve Tekrarlayan Sinir Ağı (RNN) algoritmaları kullanılmıştır ve elde edilen sonuçlar MLRKF hibrit modeli ile karşılaştırılmıştır. Tahmin doğruluğu için MAE, RMSE ve MAPE metrikleri seçilmiştir. Deney sonuçlarına göre önerilen MLRKF modelinin, düşük zaman maliyeti ve az eğitim verisi ile daha iyi sonuçlar ürettiği gözlenmiştir.

Petroşanu ve ark. (Petroşanu, ve diğerleri, 2022), derin öğrenme mimarisi olan Yönlendirilmiş Asiklik Grafik Sinir Ağı (DAGNN) modelini kullanarak e-ticarette ürün başına satış tahmin yöntemi önermişlerdir. Önerilen model için 6 farklı veri seti (Satış rakamları veri seti (SFD), İşsizlik oranı veri seti (URD), Enflasyon oranı veri seti (IRD), Zaman damgası veri seti (TSD), Özel günler veri seti (SDD), Anormal durumlar veri seti (ASD)) birleştirilerek tek veri seti (WDS) üzerinde uygulamalar gerçekleştirilmiştir. Satış tahmininde çözüm geliştirilmesi için ADAM, SGDM ve RMSProp algoritmalarından yararlanarak, LSTM ve BiLSTM yöntemleri ile elde edilen sonuçlar karşılaştırılmıştır. Deneyin sonuçlarından, Normalleştirilmiş Kök Ortalama Karesel Hata (NRMSE) hata oranına göre önerilen DAGNN modeli, en iyi tahmin doğruluğunu

vererek satış alanında doğru ürün gelir tahminini en iyi yapan model olduğu söylenebilir.

He ve ark. (He, Wu, & Si, 2022), e-ticaretle ilgili şirketlerin satış tahmini için Parçacık Sürü Optimizasyonu (PSO) tabanlı Uzun Kısa-Süreli Bellek (LSTM) hibrit tahmin modelini (PSO-LSTM) önermişlerdir. Önerilen modelde, 4 farklı veri seti (1 gerçek e-ticaret veri seti, halka açık 3 kıyaslama veri setleri) (Daqing, 2019) (Cloud, Alibaba Cloud - TIANCHI, 2020) (Cloud, Alibaba Cloud - TIANCHI, 2020) kullanılmıştır. Lineer Regresyon (LR), Regresyon için Destek Vektör Makinesi (SMOreg), Çok Katmanlı Algılayıcı (MLP), M5 model ağaçları (M5P), RF, K-en yakın komşu (K-NN), Otoregresif Entegre Hareketli Ortalama (ARIMA), Transfer Öğrenme (TL) ve RNN modelleri seçilerek, veri setlerinde uygulamalar gerçekleştirilmiştir. Deneylerde, sonuçların hata oranı analizlerinde MAE, RMSE, RAE ve Kök bağıl kare hata (RRSE) metrikleri kullanılmıştır. Deney sonuçlarından önerilen hibrit modelin, tüm veri setlerinde satış tahmini için en başarılı sonuçları verdiği görülmüştür.

2.2.Makine öğrenimine dayalı diğer tahminlemeler

E-ticaret alanında, makine öğrenimine dayalı tahminlemeler, genelde ürün önerisinde ya da direkt satış tahmini için kullanılmaktadır. Son yıllarda e-ticaret verileri, farklı metodolojilerin kullanımı ile her alanda farklı tahminler için de kullanılmaktadır. Xiahou ve Harada (Xiahou & Harada, 2022), müşteri segmentasyonu ve müşteri kayıp tahminini veren bir model önermişlerdir. Önerilen modelde, müşteri segmentasyon analizi için k-ortalama kümeleme (k-means clustering) ve müşteri kayıp tahmini için SVM ve Lojistik Regresyon (LR) yöntemleri kullanılarak karşılaştırmalı deneyler yapılmıştır. Uygulamalar, B2C e-ticaret şirketinin 987.994 müşteri davranışını içeren veri seti üzerinde gerçekleştirilmiştir. Algoritmaların başarı performans değerlendirmesi için Doğruluk (accuracy), Hatırlama (recall), Kesinlik (precision) ve Eğri altında kalan alan (area under the curve–AUC) metrikleri seçilmiştir. Elde edilen sonuçlardan, müşteri

segmentasyonu için tercih edilen k-ortalama kümeleme yönteminin doğru bir tercih olduğu görülmüştür. Tahminlemede kullanılan iki algoritma karşılaştırıldığında ise SVM'nin tahmin performansı LR'den daha iyi sonuç verdiği söylenebilir.

Wijaya ve ark. (Wijaya, ve diğerleri, 2022), e-ticaret verilerini kullanarak makine öğrenimine dayalı yoksulluk oranı tahmin modeli önermişlerdir. Deneyler için kullanılan veri seti, Endonezya'daki OLX (olx.com) (Olx, 2023) e-ticaret platformunun ürün reklamlarını içermektedir. SVR, FCBF-SVR, Derin Sinir Ağı (DNN) ve FCBF-DNN yöntemleri kullanılarak, dört farklı model üzerinde sonuçlar karşılaştırılmıştır. Önerilen modelin performans değerlendirmesi Yanlılık faktörü (Bf), RMSE, R^2 ve Doğruluk faktörü (Af) kriterlerine göre incelenmiştir. Deney sonuçlarından, şehir düzeyinde yoksulluk oranının (OECD, 2018) e-ticaret verileriyle hesaplanabileceği ve çalışmanın sonuçları incelendiğinde ise, Endonezya için yoksulluk tahmininde DNN modelinin, SVR'dan daha olumlu sonuçlar verdiği görülmüştür.

Reddy ve ark. (Reddy, Reddy, Anil, Mohanty, & Basit, 2023), dizüstü bilgisayar fiyat tahmin modeli önermişlerdir. Bir e-ticaret sitesinden alınan gerçek zamanlı verilerle oluşturulan veri seti, bilgisayar modelinin fiyatlarını doğru şekilde tahmin etmede kullanılmıştır. Çalışmada MLR, SVR ve Karar Ağacı Regresyonu (DTR) yöntemleri kullanılmıştır. Önerilen modelin hata oranları için R^2-score, MSE ve MAE metrikleri seçilmiştir. Buna göre, kullanılan algoritmalardan elde edilen sonuçlar incelendiğinde başarı oranları SVR için %87,43 ; LR için %59,73 ve DTR için %92,72 olduğu söylenebilir.

Chen ve Long (Chen & Long, 2023), FA-PSO-LSTM derin sinir ağı öğrenme modelini kullanarak finansal alanda bir risk tahmin modelini önermişlerdir. Önerilen model için veri seti, 12 e-ticaret şirketinden elde edilen üçer aylık mali kayıtları içeren on yıllık (2012-2022) finansal verilerden oluşmaktadır. Veri setinde öncelikle, finansal olan ve olmayan verilerin arasındaki ortak bağlantıyı bulabilmek için Faktör Analizi (FA) kullanılmıştır.

Ardından, akıllı optimizasyon algoritmalarından Genetik Algoritma (GA), Diferansiyel Evrim Algoritması (DEA), Parçacık Sürü Optimizasyon (PSO) algoritması ve finansal veri analizi için de derin öğrenme yöntemlerinden Uzun Kısa-Süreli Bellek (LSTM) ve RNN modelleri seçilerek veri seti üzerinde uygulanmıştır. Son olarak, FA-PSO-LSTM modeli ile SVM, Geçitli Tekrarlayan Birim (GRU) ve RNN modellerinden elde edilen sonuçların karşılaştırmalı analizi yapılmıştır. Çalışmada, önerilen modelin başarı değerlendirmesi için RMSE, MAE, MSE, MAPE ve R^2 metrikleri seçilmiştir. Deneysel sonuçlara göre, önerilen FA-PSO-LSTM modelinin en iyi finansal tahmin modeli olduğu görülmüştür. Bu çalışma, şirketlerin finansal risklere dair doğru kararlar vermesi ve sürdürülebilir bir işletme modeli olması açısından önerilmektedir.

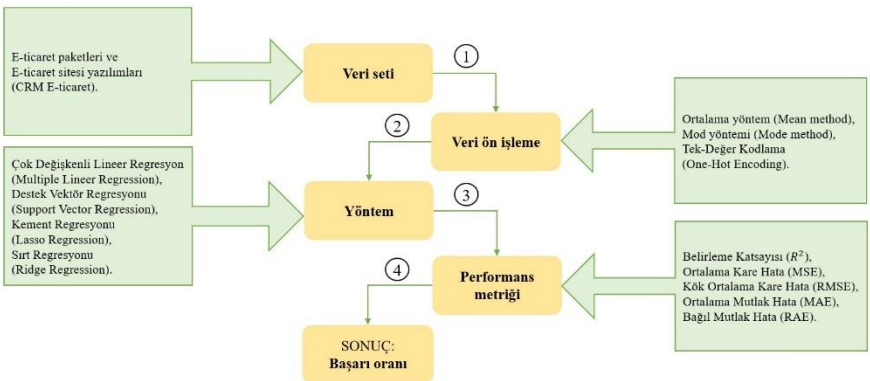
Banerjee ve ark. (Banerjee, Sinha, & Choudhury, 2021), Lineer Regresyon (LR) yöntemi kullanılarak tarım ürünlerinde fiyat tahmin modeli önermişlerdir. Çalışmada veri seti olarak Ulusal Arı Kurulu (nbb.gov.in) ((NBB) & Board, 2023) sitesindeki veriler seçilmiştir. Değerlendirme metrikleri olarak Pearson korelasyon katsayısı ve En küçük kareler yöntemi (LSM) seçilmiştir. Bu çalışma genel olarak değerlendirildiğinde, e-ticaret sektöründe tarımsal ürünler için bir fiyat tahmin modeli oluşturmada temel bir çalışma olduğu ve Lineer Regresyon modelinin, tarımsal ürün tahmininde tercih edilebileceği söylenebilir.

Pangestu ve ark. (Pangestu, Wijaya, Hernawati, & Hidayat, 2020), yoksulluk seviyesini tahmin etmeye yönelik bir model önermişlerdir. Önerilen modelde, makine öğrenimi yöntemlerinden Karar Ağacı (DT) ve Sarmalayıcı Özellik Seçimi (WFS) kullanılmıştır. Deneylerde kullanılan veri seti, Endonezya'daki bir e-ticaret firmasından alınan verilerinden oluşmaktadır. Uygulamalarda RMSE ve R^2 hata oranı metrikleri kullanılmıştır. Elde edilen sonuçlardan, veri setinde kullanılan özellik seçim algoritmalarının doğru tercih edilmesi, çalışmadaki tahmin sonuçlarını olumlu yönde etkilemiştir.

Mandal ve Maiti (Mandal & Maiti, 2022), gelecekteki ürün satışlarının tahmini için Network Promoter Score (NePS) modelini (Supriyo, 2021) önermişlerdir. Önerilen modele dayalı satış sıralamasının tespitinde LSTM (Uzun Kısa-Süreli Bellek) yöntemi (Olah, 2023) kullanılmıştır. Deneyler, Amazon.com çevrimiçi inceleme veri seti (McAuley, 2023) (Ni, 2023) üzerinde uygulanmıştır. Uygulamalarda değerlendirme metrikleri olarak R^2 ve iki Aşamalı En Küçük Kareler (2SLS) alınmıştır. Deney sonuçlarından, önerilen NePs modeli ile ürün satışları arasında kuvvetli bir bağlantı olduğu ve NePS'e dayalı gelecekteki ürün satış tahminini gözlemleyen bir model olduğu söylenebilir.

3.Araştırma Metodolojisi

Bu çalışma, e-ticaret sektöründe yer alan kurumsal firmaların gelecekteki satış adedi tahmini üzerine oluşturulmuştur. Çalışmada, gerçek hayat verilerinden oluşan bir e-ticaret veri seti, uygulamalar için kullanılmıştır. Gerçekleştirilen uygulamalardaki genel akış şeması modeli Şekil1'de gösterilmiştir.



Şekil 1. Çalışmanın genel akış şeması modeli.

Şekil 1'de, veri setinin önişleme aşamasında "Ortalama yöntem (Mean method), Mod yöntemi (Mode method) ve Tek-Değer Kodlama (One-Hot Encoding) yöntemi" kullanılarak veri seti uygulamalar için hazır hale getirilmiştir. Önişleme aşamasındaki

detaylar Bölüm 4'te ayrıntılı şekilde ifade edilmiştir. Çalışmada, uygulamalar için MLR, SVR, Kement Regresyonu (Lasso regression) ve Sırt Regresyonu (Ridge regression) modelleri ele alınarak veri seti üzerinde deneyler gerçekleştirilmiştir. Seçilen modellerde performans değerlendirmesi olarak R^2, MSE, RMSE, MAE ve RAE metrikleri alınmıştır. Bu metriklerden elde edilen sonuçlara göre, seçilen modellerde satış adedi tahmini için başarılı olup olmadığı incelenmiştir. Deney sonucunda elde edilen değerler karşılaştırılmıştır.

4.MATERYAL ve YÖNTEM

4.1.Bilgisayar ortamı

Uygulamalar 64 bit işletim sistemi, 16 GB RAM ve 3.00 GHz sahip, 11th Gen Intel(R) Core(TM) i7-1185G7 CPU işlemcili Windows 11 üzerinde çalışan bir bilgisayar üzerinde test edilmiştir. Yazılımın uygulaması, CRM e-ticaret veri seti üzerinde, Python 3.11.2 versiyon ve Spyder V5.4.3 geliştirme ortamında yapılmıştır. Yapılan tahminleme için Scikit-Learn makine öğrenimi kütüphanesinden yararlanılmıştır.

4.2.Veri seti

Bu çalışmada yer alan veriler, Türkiye'de tanınmış e-ticaret altyapı sağlayıcısı şirketi olan IdeaSoft'dan alınmıştır. Şirket, on beş yılı aşkın başarılı bir geçmişe sahip, e-ticaret ve e-ihracat alanında profesyonel şekilde çözümler üretmektedir. 2005 yılından bu yana hizmet vermeye devam eden firma, yurt içi ve yurt dışı birçok mağazasıyla, çevrimiçi sistemde müşterilere başarılı şekilde hizmet vermektedir. Çalışmada kullanılan veri seti, mevcut veya potansiyel müşterilerle çevrimiçi ortamda etkileşimi yönetmek için oluşturulmuş bir CRM e-ticaret veri setidir. Bu veri seti, 2017-2022 yılları arasında yaklaşık altı yıllık bir dönemde IdeaSoft müşterilerine ait hem müşteri verileri hem de bu müşterilerin gerçekleştirdiği e-ticaret alt yapılarına ait alışverişleri sırasında oluşan verileri kapsamaktadır. Müşteriler IdeaSoft firmasıyla telefon

satış hattı, web iletişim formu gibi farklı iletişim kanallarından iletişim kurabilmekte, demo paketlerini deneyebilmekte, satış temsilcisi ile iletişim kurabilmekte ve farklı e-ticaret alt yapı paketlerini alabilmektedir. Bu veriler, IdeaSoft CRM sisteminde kayıt altına alınmaktadır. Gerçek dünyada e-ticaret sektöründe kullanılan bu veri seti, şirketlerin internet üzerinden çevrimiçi satış takibinin yapıldığı bir ekosistemi oluşturmaktadır. Toplamda 38 öznitelik ve 242.359 örneklemden oluşan veri seti, e-ticarete dair çeşitli özellikleri barındırmasıyla, farklı tahminlemeler yapılabilir. Çalışmada kullanılan ve gerçek dünya verilerinden oluşan bu e-ticaret veri seti üzerinde, oluşturulan modelin gerçek satış değerlerine göre ne kadar doğru tahmin yapıldığı ölçülmektedir.

4.3.Veri ön işleme ve özellik seçimi

4.3.1.Veri ön işleme

Bu çalışmada, veri önişleme sürecinde özniteliklerin seçimi için "Satış adedi" hedef değişken olarak belirlenmiştir. GINI indeksine bağlı Safsızlıktaki ortalama azalma ve Çapraz doğrulama ile özyinelemeli özellik eleme yöntemleri (Recursive feature elimination with cross-validation) kullanılmıştır. GINI önem değeri hesaplanması için Python yazılım dilinde Scikit-Learn kütüphanesi kullanılarak, veri setindeki özniteliklerin önem dereceleri hesaplanmıştır. Bu hesaplamanın çıktısına bağlı olarak veri setindeki 38 öznitelik arasından önem derecesi 0.025 değerinden büyük olan 14 adet öznitelik belirlenmiştir. Özellikler üzerinde RFE (Recursive feature elimination) nesnesi yaratılarak çapraz doğrulanmış puanları hesaplanmaktadır. "Doğruluk" puanlama stratejisi, doğru sınıflandırılmış örneklerin oranını optimize etmektedir.

4.3.2.Özellik seçimi

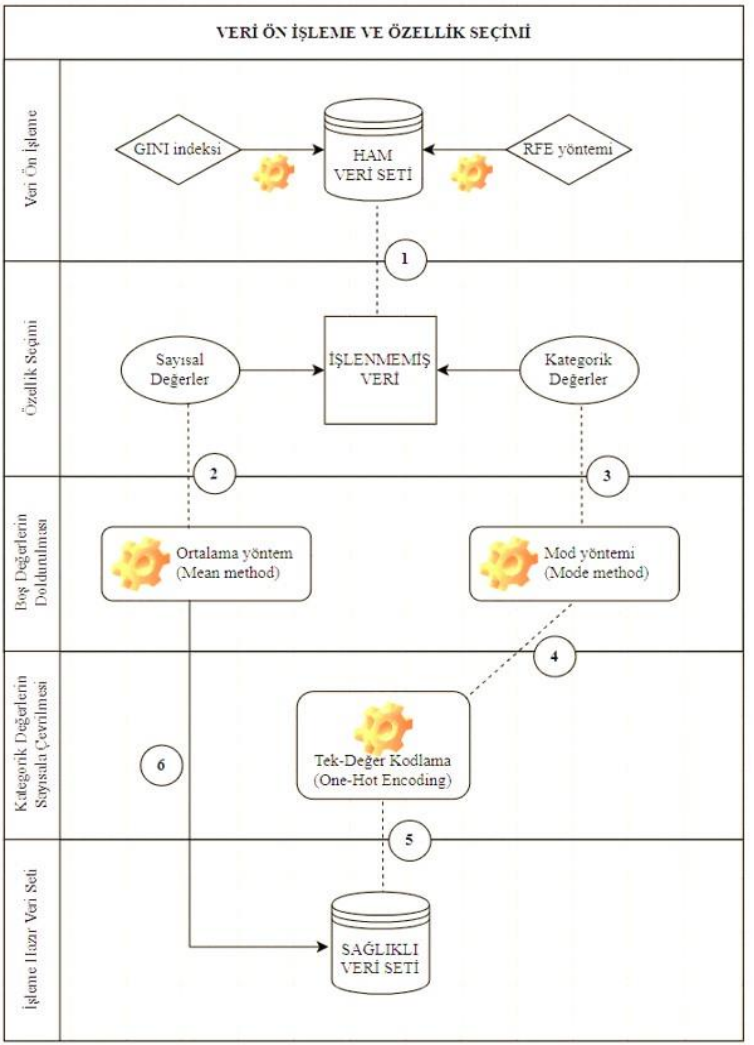
Çalışmada, GINI indeksi ve RFE yöntemine dayanarak veri önişleme sürecinde seçilen 14 öznitelik (9 adet sayısal, 5 adet kategorik öznitelikler) çalışmamız için detaylıca yeniden incelenmiştir. Analizler sonucunda, çalışmamızda yapılması

düşünülen “Satış Adedi” tahmini ile doğrudan ilişkili olan sayısal değerlerin çalışmamıza daha uygun olduğu görülmüştür. Bu nedenle, veri ön işleme sürecinde seçilen 14 öznitelikten sadece 9 adet sayısal öznitelikler son karar olarak bu çalışmada kullanılmaya karar verilmiştir. Sonuç olarak, çalışmamız için belirlenen veri seti, Tablo 1’de gösterilmiştir ve 9 adet sayısal özniteliğe sahip verilerden oluşmaktadır. Uygulamalar için belirlediğimiz veri setinde, çalışmamızda kullanılan 9 özniteliğe ait alınan özellik adı, açıklaması ve verinin tipi gösterilmiştir.

Tablo 1 – Veri seti olarak dikkate alınan özellikler ve tipleri.

No	Özellik adı	Açıklama	Tipi
1	PBX ADEDİ	Satış temsilcilerinin müşteri ile yaptığı görüşme adedini belirtir.	Tamsayı (Int)
2	FIRSAT ADEDI	Satış ekibi tarafından müşteriye atanan indirim, promosyon ve fırsat adedini belirtir.	Tamsayı (Int)
3	AKSIYON ADEDI	Satış temsilcilerinin müşteri ile aldığı aksiyon adedini belirtir.	Tamsayı (Int)
4	SATIŞ ADEDİ	Satışa dönüşen paket sayısını belirtir.	Tamsayı (Int)
5	DEMO SKORU	Müşterinin e-ticaret altyapısı demosunda ilerleme seviyesini (sipariş, ürün, promosyon oluşturma) belirtir.	Tamsayı (Int)
6	SIPARIŞ TOPLAM TUTARI	Müşterinin bir satışta yaptığı sipariş toplam tutarı bilgisini içerir.	Tamsayı (Int)
7	SEKTÖR	Müşterinin e-ticaret ürünlerinin ait olduğu sektör bilgisini verir.	Nesne (Object)
8	E-TICARET TECRÜBESI	Müşterinin daha önce bir e-ticaret sağlayıcısıyla çalışıp çalışmadığı bilgisini içerir.	Nesne (Object)
9	SATIŞ TEMSILCISI	Sipariş veya promosyon oluşturma, perakende ürün, mal ve hizmetleri müşteriye satmakla görevli kişi bilgisini içerir.	Nesne (Object)

Veri ön işleme ve özellik seçiminden yola çıkarak, veri seti üzerinde gerçekleştirilen adımlara Şekil2'de yer verilmiştir.



Şekil 2. Veri setinde veri önişleme ve özellik seçimindeki akış şeması adımları.

Şekil 2'de, veri setinde veri önişleme ve özellik seçimindeki akış şeması adımları gösterilmiştir. Çalışmada, veri ön işleme iki aşamada gerçekleştirilir. Birinci aşamada, sayısal ve kategorik özniteliklerdeki boş değerler doldurulmuştur. Sayısal öznitelikleri doldurma işleminde *"Ortalama yöntem (Mean method)"* ve kategorik öznitelikleri doldurma işleminde *"Mod yöntemi (Mode method)"* kullanılmıştır. İkinci aşamada ise, veri setinde satış adedi tahmini yapılacağı için, kategorik özniteliklerin sayısal değerlere çevirme işlemi yapılmıştır. Bunun için *"Tek-Değer Kodlama (One-Hot Encoding) yöntemi"* kullanılmıştır. Bu işlemler sonucunda, veri setindeki öznitelik sayısı toplamda 170 adete çıkmıştır.

4.4.Yöntem

Çalışmanın bu bölümünde, uygulamalarda kullanılan regresyon yöntemlerine dair açıklamalar ve bu yöntemlerin matematiksel ifadelerine yer verilmiştir. Genel olarak, regresyon yöntemleri literatürde orijinal adı ile kullanıldığı için, çalışmamızda mümkün olduğunca bu orijinal ifadelerin Türkçe isimlerine de yer verilmiştir.

4.4.1.Çok Değişkenli Lineer Regresyon (Multiple Linear Regression)

Çok Değişkenli Lineer Regresyon (MLR), tahminlemede sıkça tercih edilen istatistiksel bir yöntemdir. MLR, birden fazla bağımsız değişkene bağlı olan ve bağımlı değişkeni doğrusal artacak şekilde, değişkenler arasında bağlantı bulmaya çalışarak tahminlemeler yapmaktadır. Bunun yanında, MLR modeli kullanılarak farklı etkenlerin analizi de yapılmaktadır. Bu modelde tahminleme yapılırken, bağımlı değişken (hedef değişken) birden çok bağımsız değişkene bağlıdır. Birden fazla değişkene sahip bir regresyon denklemi Eşitlik (1)'deki (James, Witten, Hastie, Tibshirani, & Taylor, 2023) gibi tanımlanabilir:

$$Y = \beta + \beta_0 x_1 + \beta_1 x_2 + \beta_2 x_3 + \varepsilon \quad (1)$$

Burada Y bağımlı değişken (rassal/hedef değişken); x_i -ler (i=1,2,3) bağımsız (açıklayıcı) değişkenlerdir. β, bağımlı değişkenin sabit regresyon katsayısı, β_i-ler (i=1,2,3) bağımlı değişkenin kısmî regresyon katsayıları ve ε hata terimi olarak ifade edilir.

4.4.2.Destek Vektör Regresyonu (Support Vector Regression)

Destek Vektör Regresyonu (SVR), regresyon problemlerinde çözüm için kullanılan istatistiksel öğrenme teorisine dayalı bir denetimli (gözetimli) makine öğrenimi yöntemidir (Drucker, Burges, Kaufman, Smola, & Vapnik, 1997) (Vapnik, 2000). Bu yöntem, doğrusal veya doğrusal olmayan regresyon türlerinde bir ya da daha fazla değişken ile bir bağımlı değişken arasındaki bağlantıyı araştırmak için kullanılmaktadır (Zhang & O'Donnell, 2020). SVR, çalışma prensibi olarak Destek Vektör Makinesi (SVM) mantığını kullanır. SVM'nin temel mantığından hareketle tahminleme ve sınıflandırma yapan seyrek çekirdekli bir makineyi kullanarak öğrenmeyi gerçekleştirir.

SVR, genellikle tahminlemeler için yararlı bir metottur. Çünkü, model karmaşıklığı düşünüldüğünde, veri setlerinde çıkan tahmin hatasını dengeleyerek büyük boyutlu verileri işlemek için iyi bir çalışma performansı gösterir. Bunun yanında SVR, bağımlı ve bağımsız değişkenler arasındaki bağlantıyı da kategorize ederek, değişkenler arasındaki tahmin doğruluğunu en üst düzeye çıkarmaktadır.

4.4.3.Kement Regresyonu (Lasso Regression)

Kement (En Küçük Mutlak Büzülme ve Seçim Operatörü) Regresyonu, lineer regresyonun başka bir yanlı tahmin modellerinden birisidir. Tibshirani (Tibshirani, 1996) tarafından 1996 yılında önerilen bu model, Sırt Regresyon modeline benzer şekilde, katsayılar üstüne ceza terimi uygulanması ile bazı katsayıların sıfıra indirgenerek çalışması prensibine dayanmaktadır. Kement regresyonu modeli, Eşitlik (2)'deki (Tibshirani, 1996) gibi tanımlanabilir:

$$\hat{\beta_{lasso}} = argmin\{\sum_{i=1}^{N}\left(y_i - \alpha - \sum_{j=1}^{M}\beta_j x_{ij}\right)^2 + \lambda\sum_{J=1}^{M}|\beta_j|\}(2)$$

Burada N gözlem sayısı, y bağımlı değişken, M değişken sayısı, α ve $\beta_j = (\beta_1, \beta_2, \dots, \beta_M)$ bilinmeyen parametreler, $x_{ij} = (x_{i1}, x_{i2}, \dots, x_{iM})^T$ bağımsız değişkenler, λ ceza terimi (ayar parametresi) şeklinde ifade edilir. λ, regresyon için daralma (shrinkage) miktarını gösteren ayar parametresidir. λ'nın alacağı değer arttığında, daralma miktarı da aynı şekilde artar ve $\lambda > 0$ değerini almalıdır.

Kement regresyonu modeli, Eşitlik (2)'deki katsayıları sıfıra indirgeyerek, daha az değişken ile regresyon modelinin net ve kolay bir şekilde yorumlanmasını sağlar. Bu nedenle, çok sayıda değişken ve gözlem içeren ya da değişken sayısının gözlem sayısından fazla olduğu (M>N) büyük veri setlerinde yaygın olarak uygulanmaktadır.

4.4.4.Sırt Regresyonu (Ridge Regression)

Sırt Regresyonu, çok değişkenli regresyonda verileri analiz etmek için kullanılan tahmin modellerinden birisidir. Bu model, en küçük kareler yönteminde fazla uydurma problemlerini çözmek için kullanılmaktadır. Hoerl ve Kennard (Hoerl A. E., Application of ridge analysis to regression problems, 1962) (Hoerl A. E., Ridge analysis, 1964) (Hoerl & Kennard, 1968) (Hoerl & Kennard, Ridge Regression: Biased Estimation for Nonorthogonal Problems, 1970) tarafından 1970 yılında önerilen bu model, çok değişkenli lineer bağlantı problemlerini gidermek için Sırt Regresyonu tahmin modelini önermiştir. Bu sayede, daha küçük varyanslı değerlerle tahminler yapılmaktadır (Yüzbaşı & Pala, 2022). Sırt Regresyonu modeli, Eşitlik (3)'teki (Hoerl & Kennard, Ridge Regression: Biased Estimation for Nonorthogonal Problems, 1970) gibi tanımlanabilir:

$$\beta_{ridge} = \sum_{i=1}^{n}(Y_i - \alpha - \beta X_i')^2 + \mathrm{t}\sum_{i=1}^{p}\beta_j^2 \qquad (3)$$

β'nın türevi alınıp sıfıra eşitlendiğinde Eşitlik (4)'ten,

$$\hat{\beta_k} = (XX' + kI_p)^{-1}YX' \qquad (4)$$

sonucu elde edilir. Burada I_p, pxp boyutunda birim matrisi ve $t \geq 0$ ayar parametresidir. k = 0 ise $\hat{\beta_k} = \hat{\beta}$, $k = \infty$ ise $\hat{\beta_k} = 0'$dır.

4.5.Modellerin performans değerlendirmesi

Aşağıdaki tüm formüller için Y_i, i. tahmin edilen değeri; X_i, i. gerçek değeri ve m, örnek sayısını verir. $\hat{Y}$, gerçek değerin ortalaması olmak üzere Eşitlik (5)'ten,

$$\hat{Y} = \frac{1}{m}\sum_{i=1}^{m} X_i \quad (5)$$

şeklinde gösterilebilir. Regresyon yöntemi, veri seti içerisinde doğru yerdeki Yi değerine karşılık Xi değerini tahmin etmektedir.

4.5.1.Belirleme katsayısı (Coefficient of determination - R^2)

Belirleme katsayısı (R^2) (Wright, 1921), lineer regresyon için bir dizi bağımsız değişken ile tahmin edilebilen bağımlı değişkendeki varyansın oranı şeklinde ifade edilebilir. Bu katsayı, tahmin edilen modelde regresyon denkleminin başarısını (tahmin kapasitesini) ölçmektedir. R^2 performans metriği, Eşitlik (6)'daki (Chicco, Warrens, & Jurman, 2021) gibidir:

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\hat{Y} - Y_i)^2} \quad (6)$$

R^2, 0 ve 1 arasındaki değerleri alır; yüzdelik olarak %0 ile %100 şeklinde belirtilir. R^2 varyans değerinin yüksek olması, regresyon modeli tahmininde uyumlu olduğunu gösterir. Eğer R^2 değeri sıfıra yakınsıyorsa, modelin tahmini için "uyumlu değildir" çıkarımı yapılır. Bu yüzden, modelin değiştirilmesi gerekir. Eğer R^2 değeri bire yakınsıyorsa, modelin tahmini için "uyumludur" çıkarımı yapılır.

4.5.2.Ortalama Kare Hata (Mean Squared Error - MSE)

Ortalama Kare Hata (MSE), maliyet fonksiyonu (cost function) olarak adlandırılan ve lineer regresyonda en çok kullanılan

istatistiksel modellerdeki hata miktarlarını ölçen değerlendirme metriğidir (Frost, 2023). Bu metrik, veri setindeki doğru değişkenleri seçmede bir ölçüt niteliğinde önerilmiştir. MSE'de tahmin edilen değer ve gerçek değer arasındaki ortalama kare farkı verilir. MSE yaklaşımı, gelecekteki gözleme dair tahmin değişkenleri için değerlerini ve tahmin edilen varyansın büyüklüğünü kullanır (Allen, 1971). Bunun yanında, veri setinde tespit edilmesi gereken aykırı değerler olduğunda da MSE kullanılmaktadır. MSE performans metriği, Eşitlik (7)'deki (Chicco, Warrens, & Jurman, 2021) gibidir:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(Y_i - X_i)^2 \quad (7)$$

MSE, 0 ve 1 arasında değerler alır. MSE değeri ne kadar küçükse (sıfıra yakınsıyorsa) modeldeki tahminlemenin o kadar iyi olduğu söylenebilir.

4.5.3Kök Ortalama Kare Hata (Root Mean Squared Error - RMSE)

Kök Ortalama Kare Hata (RMSE), MSE'nin karekökü alınarak hesaplanan ve en çok kullanılan değerlendirme metriklerinden birisidir. MSE'de olduğu gibi, veri setinde tespit edilmesi gereken aykırı değerler (gözlemler) olduğunda RMSE kullanılabilir. RMSE performans metriği, Eşitlik(8)'deki (Chicco, Warrens, & Jurman, 2021) gibidir:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(Y_i - X_i)^2} \quad (8)$$

4.5.4.Ortalama Mutlak Hata (Mean Absolute Error - MAE)

Ortalama Mutlak Hata (MAE), tahmin değeri ve gerçek değer arasındaki mutlak farkın ortalamasını veren en basit standart hata metriğidir. Bu metrikte, modelin ortalama performans hatasını açıklarken bağıntılı yönleri incelenir (Willmott & Matsuura, 2005). MAE metriği, regresyonda genellikle model performans değerlendirmesinin iyi bir göstergesi olarak tercih edilmektedir.

MAE performans metriği, Eşitlik (9)'daki (Chicco, Warrens, & Jurman, 2021) gibidir:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|Y_i - X_i| \quad (9)$$

4.5.5.Bağıl Mutlak Hata (Relative Absolute Error - RAE)

Bağıl Mutlak Hata (RAE), tahmin edilen ve gerçek değerler arasındaki farkın toplamını, gerçek değer ile gerçek değerin ortalaması arasındaki farkın toplamına bölerek gösterilen bir hata metriğidir. Matematiksel olarak RAE performans metriği, Eşitlik (10)'daki (Kara & Şamlı, 2021) gibidir:

$$RAE = \frac{\sum_{i=1}^{m}|Y_i - X_i|}{\sum_{i=1}^{m}|X_i - \hat{X_i}|} \quad (10)$$

Burada Y_i i. tahmin değerini, X_i i. gerçek değerini, $\hat{X_i}$ gerçek değerlerin toplamını ve m, örnek sayısını göstermektedir.

5.DENEYSEL SONUÇLAR

Bu bölümde makine öğrenimine dayalı regresyon modelleri kullanılarak gerçekleştirilen deney süreçleri ve elde edilen sonuçlar paylaşılmaktadır. Çalışmada amaç, e-ticaret şirketlerinin gelecekte yapılacak olan satışlar için satış adedi tahmini üzerine yapılmıştır. Çalışma kapsamında, veri seti olarak tanınmış büyük bir şirketin e-ticaret verileri kullanılarak deneysel sonuçlar gösterilmiştir. Bu veri seti üzerinde hedef değişkenimiz olan "Satış Adedi" tahminini etkileyen özellik seçimi yapılarak 9 adet özellik elde edilmiştir. Bu özellikler üzerinde MLR, SVR, Kement Regresyonu (Lasso regression) ve Sırt Regresyonu (Ridge regression) algoritmaları kullanılarak oluşturulan modellerin eğitimi gerçekleştirilmiştir. Seçilen modellerde her bir uygulama için veri seti %70 eğitim - %30 test olarak ayrılmıştır. Çalışmada kullanılan modellerin performans ölçümleri için R^2, MSE, RMSE, MAE ve RAE metrikleri dikkate alınarak değerlendirilmiştir.

Oluşturulan modeller üzerinde tahminleme işlemleri yapılırken, veri seti üzerinde sadece test için ayrılmış olan veri seti

(%30 test verisi) dikkate alınmıştır. Deney sonuçları incelendiğinde, oluşturulan modeller için %30 test veri setine göre elde edilen sonuçların performans ölçümlerine göre karşılaştırmalı analizi Tablo 2'de yer almaktadır.

Tablo 2 – %30 Test veri setine göre algoritmaların performans ölçümlerine göre karşılaştırması.

YÖNTEMLER	**METRİKLER**				
	R^2	**MSE**	**RMSE**	**MAE**	**RAE**
Çok Değişkenli Lineer Regresyon	0.694162	0.153265	0.391490	0.117183	0.391564
Destek Vektör Regresyonu	**0.775993**	0.112257	0.335047	0.152991	0.511212
Kement Regresyonu	0.613630	0.193622	0.440025	0.140942	0.470953
Sırt Regresyonu	0.694026	0.153333	0.391577	0.117196	0.391606

Bir modelin iyi olması, varyans değerinin yüksek ve hata değerinin düşük olması anlamına gelmektedir. Buna göre, Tablo 2'deki sonuçlar incelendiğinde, SVR modelinde 0,7759 R^2 değeri ile regresyon model uyumunun oldukça iyi olduğu ve 0,1122 MSE değeri ile regresyon modelindeki tahminleyicinin en yüksek performansı sergilediği söylenebilir. Öte yandan, Kement Regresyonu (Lasso regression) modelinde 0,6136 R^2 değeri ile regresyon model uyumunun daha kötü olduğu ve 0,1936 MSE değeri ile regresyon modelindeki tahminleyicinin en düşük performansa sahip olduğu görülmüştür. Bulunan tüm bu sonuçlar, analizlerden elde edilen bulguları yansıtmaktadır.

6.Sonuç Ve Gelecek Çalışmalar

Günümüzde, alışveriş kavramı giderek yeni bir bakış açısı kazanmıştır. Teknolojinin gelişimi ile alışverişte yer alan ticaret

kavramı, yerini çevrimiçi e-ticaret olgusuna bırakmıştır. Alışverişin müşteri ile satıcı arasındaki iletişimi şu an internet üzerinden yapıldığı bir dönem olduğu için, e-ticaret alanında yer alan kuruluşlar da kendi gelecek iş stratejilerini doğru şekilde belirlemek adına farklı tahminleme yöntemlerine başvurmaktadırlar.

Bu çalışma, e-ticaret alanında makine öğrenimine dayalı regresyon yöntemlerini kullanarak "Satış Adedi tahmini" üzerine oluşturulmuştur. Çalışmada, mevcut veya potansiyel müşterilerle çevrimiçi ortamda etkileşimi daha iyi yönetmek adına, modelin gerçek satış değerlerine göre ne kadar doğru tahmin yapıldığı ölçülmüştür. Bu ölçümler için, e-ticaret verilerine dayalı satış adedi tahmininde, büyük bir şirkete ait CRM e-ticaret veri seti kullanılarak regresyon modelleri ile uygulamalar yapılmıştır. Çalışmanın ilerleyişinde uygulamalar için MLR, SVR, Kement ve Sırt Regresyonu modelleri seçilmiştir. Seçilen her bir model için R^2, MSE, RMSE, MAE ve RAE metrikleri kullanılarak, performans değerlendirmesi ile satış adedi tahmininde kullanılabilecek en iyi yöntemi bu çalışma belirlemektedir. Scikit-Learn kütüphanesinden yararlanılarak Python yazılım dilinde yapılan uygulamalar sonucu başarılı sonuçlar elde edilmiştir. Yapılan tahmin sonuçlarına göre, Satış Adedi tahmini için regresyon algoritmalarındaki hata oranlarının değişkenlik gösterdiği söylenebilir. Çalışmanın oluşturulmasında, seçilen modellerdeki uygulamalar için veri seti %70 eğitim - %30 test olarak ayrılmıştır. Çalışma, test sonucuna göre incelendiğinde, seçilen hata oranları ve veri setinde uygulanan dört farklı model arasında en iyi tahmin algoritmasını, R^2 hata oranına göre, 0,7759 ile SVR modelinde vermiştir. Bu modelin veri setinde satış adedi tahminini en iyi yapan model olduğu söylenebilir.

Çalışmanın odak noktası, e-ticaretle ilgilenen şirketler için satış adedi tahmini yapılması üzerinedir. Bu çalışma sayesinde, e-ticaret alanında satış adedi tahmini için hangi regresyon yöntemlerinin başarı olabileceği ve kullanışlılık açısından e-ticaret veri setlerine uygulandığında tahmin sonuçlarının neler olabileceği bilgisi ifade edilmiştir. Bunun yanında, e-ticaret alanındaki işletmeler ve şirketler, müşterilerin satın alma olasılıklarını doğru

şekilde analiz ederek, gelecekteki satışlarının tahmini üzerinde de başarılı sonuçlar elde edebilirler. Çalışmadaki tüm araştırma sonuçları, e-ticaret alanında çalışan işletmelerin müşteri ilişkileri yönetiminin (CRM) satış-pazarlama departmanı açısından da önem taşımaktadır. Böylece, şirketlerin internet üzerinden satış miktarlarının etkisi ölçülerek, gelecek yıllardaki doğru satış tahmini ile internet üzerinden ürünlerin satış piyasa hareketliliği de rahatlıkla öngörülebilir olması hedeflenebilir. Gelecek çalışmada, veri setinde daha detaylı incelemeler yapılarak, makine öğrenimine dayalı seçilen algoritma sayısının artırılması ile elde edilen sonuçlarda iyileştirmeye giden uygulamalar gerçekleştirilecektir.

Teşekkür

Bu çalışma 7220112 proje numarası ile Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) tarafından desteklenmiştir. Çalışmamızı finansal olarak destekleyen TÜBİTAK'a teşekkürlerimizi sunarız.

KAYNAKÇA

(NBB), & Board, N. B. (2023, 06 01). *National Bee Board (NBB)*. A book on ‘Sweet Revolution’: https://nbb.gov.in/ adresinden alındı

Allen, D. M. (1971). Mean Square Error of Prediction as a Criterion for Selecting Variables . *Technometrics*, 469-475.

Banerjee, T., Sinha, S., & Choudhury, P. (2021). Dynamic Price Prediction of Agricultural Produce for E-Commerce Business Model: A Linear Regression Model. *Data Management, Analytics and Innovation*, 493–504.

Chen, X., & Long, Z. (2023). E-Commerce Enterprises Financial Risk Prediction Based on FA-PSO-LSTM Neural Network Deep Learning Model. *Sustainability*, 1-17.

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 1-24.

Cloud, A. (2020, June 29). *Alibaba Cloud - TIANCHI*. Drug sales data: https://tianchi.aliyun.com/dataset/dataDetail?dataId=67808 adresinden alındı

Cloud, A. (2020, 06 29). *Alibaba Cloud - TIANCHI*. Car sales dataset: https://tianchi.aliyun.com/dataset/dataDetail?dataId=67543 adresinden alındı

Daqing, C. (2019, September 20). *UC Irvine Machine Learning Repository*. Online Retail II: http://archive.ics.uci.edu/ml/datasets/Online+Retail+II adresinden alındı

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Neural Information Processing Systems*, 1-7.

Frost, J. (2023, 08 07). *Mean Squared Error (MSE)*. Statistics By Jim: https://statisticsbyjim.com/?s=Mean+Squared+Error+%28MSE%29 adresinden alındı

He, Q.-Q., Wu, C., & Si, Y.-W. (2022). LSTM with particle Swam optimization for sales forecasting. *Electronic Commerce Research and Applications*, 1-19.

Hoerl, A. E. (1962). Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 54-59.

Hoerl, A. E. (1964). Ridge analysis. *Chemical Engineering Progress Symposium Series*, 67-77.

Hoerl, A. E., & Kennard, R. W. (1968). On regression analysis and biased estimation. *Technometrics*, 422-423.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 80-86.

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An introduction to Statistical Learning with applications in Python.* New York, London: Springer.

Kara, Ş. E., & Şamlı, R. (2021). Yazılım Projelerinin Maliyet Tahmini için WEKA'da Makine Öğrenmesi Algoritmalarının Karşılaştırmalı Analizi. *Avrupa Bilim ve Teknoloji Dergisi (EJOSAT)*, 419.

Li, X., Dong, H., & Han, S. (2020). Multiple Linear Regression with Kalman Filter for Predicting End Prices of Online Auctions. *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)* (s. 1-10). Calgary, AB, Canada: IEEE.

Manasa, J., Gupta, R., & Narahari, N. S. (2020). Machine Learning based Predicting House Prices using Regression Techniques. *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (s. 1-7). Bangalore, India: IEEE.

Mandal, S., & Maiti, A. (2022). Network promoter score (NePS): An indicator of product sales in E-commerce retailing sector. *Electronic Markets*, 1327–1349.

McAuley, J. (2023, 06 01). *Amazon Product Data (2014).* http://jmcauley.ucsd.edu/data/amazon/ adresinden alındı

Ni, J. (2023, 06 01). *Amazon Review Data (2018).* https://nijianmo.github.io/amazon/ adresinden alındı

OECD. (2018, July 26). *OECD Data*. Poverty rate: https://data.oecd.org/inequality/poverty-rate.htm adresinden alındı

Olah, C. (2023, 05 28). *Understanding LSTM Networks.* colah.github.io: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ adresinden alındı

Olx. (2023, May 27). *Olx*. OnLine eXchange: https://www.olx.co.id/ adresinden alındı

Pan, H., & Zhou, H. (2020). Study on convolutional neural network and its application in data mining and sales forecasting for E-commerce. *Springer*, 298.

Pangestu, A., Wijaya, D. R., Hernawati, E., & Hidayat, W. (2020). Wrapper Feature Selection for Poverty Level Prediction Based on E-Commerce Dataset. *2020 International Conference on Data Science and Its Applications (ICoDSA)* (s. 1-7). Bandung, Indonesia: IEEE.

Petroşanu, D.-M., Pîrjan, A., Căruţaşu, G., Tăbuşcă, A., Zirra, D.-L., & Perju-Mitran, A. (2022). E-Commerce Sales Revenues Forecasting by Means of Dynamically Designing, Developing and Validating a Directed Acyclic Graph (DAG) Network for Deep Learning. *Electronics*, 1-35.

Reddy, C. L., Reddy, K. B., Anil, G., Mohanty, S. N., & Basit, A. (2023). Laptop Price Prediction Using Real Time Data. *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)* (s. 1-5). Jeddah, Saudi Arabia: IEEE.

Supriyo, M. (2021, September 22). *An-Indicator-of-Product-Sales.* GitHub: https://github.com/mandalsupriyo/An-Indicator-of-Product-Sales adresinden alındı

Sykes, A. O. (1993). An Introduction to Regression Analysis. *Coase-Sandor Working Paper Series in Law and Economics*, 2.

Terzi, R., Sağıroğlu, Ş., & Demirezen, U. (2017). Büyük Veri ve Açık Veri: Temel Kavramlar. R. Terzi, Ş. Sağıroğlu, & U. Demirezen içinde, *Büyük Veri ve Açık Veri: Temel Kavramlar* (s. 16). Grafiker Yayınları.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267-288.

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory.* New York, NY: Springer.

Wijaya, D. R., Pradnya Paramita, N. L., Uluwiyah, A., Rheza, M., Zahara, A., & Puspita, D. R. (2022). Estimating city-level poverty rate based on e-commerce data with machine learning. *Electronic Commerce Research*, 195-221.

Willmott, C., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Journal of Climate Research*, 79–82.

Wright, S. (1921). Correlation and Causation. Washington, D.C., USA: National Agricultural Library.

Xiahou, X., & Harada, Y. (2022). B2C E-Commerce Customer Churn Prediction Based on K-Means and SVM. *Journal of Theoretical and Applied Electronic Commerce Research*, 458-475.

Yıldız, A. (2022). Büyük Veri'nin V'leri ve Veri Analitiği. *Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 362.

Yüzbaşı, B., & Pala, M. (2022). Ridge regresyon parametre seçimi: Türkiye'nin doğrudan yabancı yatırım örneği. *İstatistikçiler Dergisi: İstatistik & Aktüerya*, 3.

Zhang, F., & O'Donnell, L. J. (2020). Chapter 7-Support vector regression. *Machine Learning - Methods and Applications to Brain Disorders*, 123-140.

BÖLÜM VII

Log Kayıtları Üzerinden Siber Saldırı Tespit Sistemi Geliştirilmesi

Serkan ÖZARGIN[1]
Ahmet ALBAYRAK[2]

Log Kavramı

Log kavramı, bilgisayar ağları üzerinde kullanıcılar tarafından oluşturulan dosyalara verilen addır. Log kavramı İngilizce de kütük, kayıt ve günlük kayıt defteri anlamına gelmektedir. Bilgisayar sistemleri üzerinde birçok çalışma gerçekleşmektedir. Bu gerçekleşen dijital olaylar log adı altında kayıt edilmektedir.

[1] Yüksek Lisans Öğrencisi, Düzce Üniversitesi, LEE, Siber Güvenlik ABD., Orcıd:0000-0002-2166-1102

[2] Dr. Öğr. Üyesi, Düzce Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Orcıd: 0000-0003-3109-0435

Log kayıtları, bilgisayar sistemleri üzerinde gerçekleşen olayların kullanıcı bilgilerinin tutulmasını sağlar. Log, cihaz, yazılım, işletim sistemi veya sunucular üzerinden gelen olayların kayıtlarını oluşturan verileri, belirli periyotlarla haftalık, günlük, saatlik olarak tutulan dosyalardır. Log dosyalarını açmak için bir kelime işlemci programına ihtiyaç vardır. Bu programlar, LibreOffice, OpenOffice, Notepad programları kullanarak açılmaktadır (Baykara vd., 2016: 1).

5651 Sayılı Loglama Kanunu

Log kayıtları, internet erişiminin kayıt altına alınmasıyla ortaya çıkmıştır. İşlenen suçların faillerini bulmak için aynı zamanda geçmişte işlenen siber suçların çözümü için ortaya çıkmıştır. Bu kapsamda internet üzerinde işlenen suçların tespiti için işletmeler 5651 sayılı kanuna uygun olarak log tutma zorunluluğu getirilmiştir.

5651 sayılı loglama kanunu 23 Mayıs 2007 yılında kabul edilmiştir. Bu kapsamda tutulan log kayıtları kullanıcıların IP (İnternet Protokol) adresi, MAC (Media Access Control) adresi, zaman bilgisi bilgilerini kayıt altına alarak sistemlerin güvenlik seviyesini yükseltir. Tutulan log kayıtları zaman damgası vurularak 2 yıl saklanmaktadır (5651 sayılı kanun, 2016).

Logların tutulmasında iki önemli unsur ortaya çıkmaktadır. Log kayıtları ile siber saldırı tespiti arasından güçlü bir bağ bulunmaktadır. Log kayıtları ile internet üzerinde işlem yapan kullanıcıların internet izlerini çıkartarak, adli analiz yöntemleri ve siber iz sürme yöntemleri kullanılarak siber suçların tespit edilmesini sağlar. Logların siber güvenlik ve saldırı tespit yöntemleri arasındaki ilişki şekil 1 de yer almaktadır. Bu durumun kanıtı olarak log kayıtlarının tutulmasındaki amacını gösterilir. Log kayıtlarının ortaya çıkış amaçlarından biri siber suçluları tespit etmektir. Log kayıtları saldırı oranlarını düşürdüğü gibi siber güvenliğin sağlanması için en temel süreci oluşturur (5651 sayılı kanun, 2016).

Log kayıtları siber güvenliğin sağlanması için kullanılabilen dosyalardır. Şekil 2 de log kayıtlarının saldırı tespitinde kullanıldığı görülmektedir. Bu anlamda loglama sistemlerinin önemi burada ön plana çıktığı görülmektedir.

Log Kayıtlarında bulunan önemli özellikler

Log kayıtları sistemler üzerinde gerçekleşen olayların yapılan işlemlerin hangi kullanıcı tarafından yapıldığını zaman bilgisiyle kayıt altına almaktadır. Şekil 3 de Log kayıtları üzerinde yer alan açıklamalar yer almaktadır. Log kayıtları incelendiğinde bazı önemli veriler aşağıda yer almaktadır (Baykara vd., 2016: 5:7).

- Bağlanan cihazların IP/MAC Adres Bilgileri,
- Tarih ve saat bilgileri,
- Log üretilen sistem bilgileri,
- İşlem yapan kullanıcı ve cihaz bilgileri,
- Sisteme erişenlerin hangi kaynaktan geldiği,

Şekil 1. Log kayıtlarının siber güvenlik ve saldırı tespiti arasındaki ilişki.

Şekil 2. Log kayıtlarının siber saldırı ve saldırı tespiti arasındaki hareketi.

Alan Adı	**Örnek Değer**	**Açıklama**
date	2015-11-20	Aktivitenin meydana geldiği tarih
time	00:22:31	Aktivitenin meydana geldiği saat
c-ip	212.154.80.164	İstekte bulunan kullanıcının IP adresi
s-ip	193.140.180.4	Web sitesinin bulunduğu sunucunun IP adresi
cs-uri-stem	/Default.aspx	İstekte bulunulan web adresi
sc-status	200	İsteğe verilen cevabın durum kodunu içerir
cs(user-agent)	Mozilla/4.0+(compatible; +MSIE+6.0; +Windows+NT+5.1;)	İstemci tarafından kullanılan tarayıcının tipi ve diğer özellikler
cs-referrer	-	Aktif sayfaya hangi kaynaktan gelindiğini gösterir

Şekil 3. Log kayıtları üzerinde yer alan açıklamalar (Baykara vd., 2016: 6).

Log Kayıtlarının Tutulmasındaki Amaçlar

Log kayıtları, sistem yöneticileri tarafından sistemlerin kullanıldığı gibi, adli analiz uzmanları, sistem uzmanları ve siber güvenlik uzmanlarının incelediği doyalar haline gelmektedir. Bu anlamda log kayıtlarının önemi her geçen gün artmaktadır. Log kayıtları sistemlerin iyileştirilmesinden performansına kadar birçok konuda kullanılmakta olduğu görülmektedir. Bu anlamda log kayıtları birçok önemli konuda uzmanlara yarar sağlamaktadır. Siber saldırıların birçok anlamda önüne geçmektedir (Çınar ve Bilge 2016: 1).

Log Kayıtlarının Siber Güvenlik Açısından Faydaları

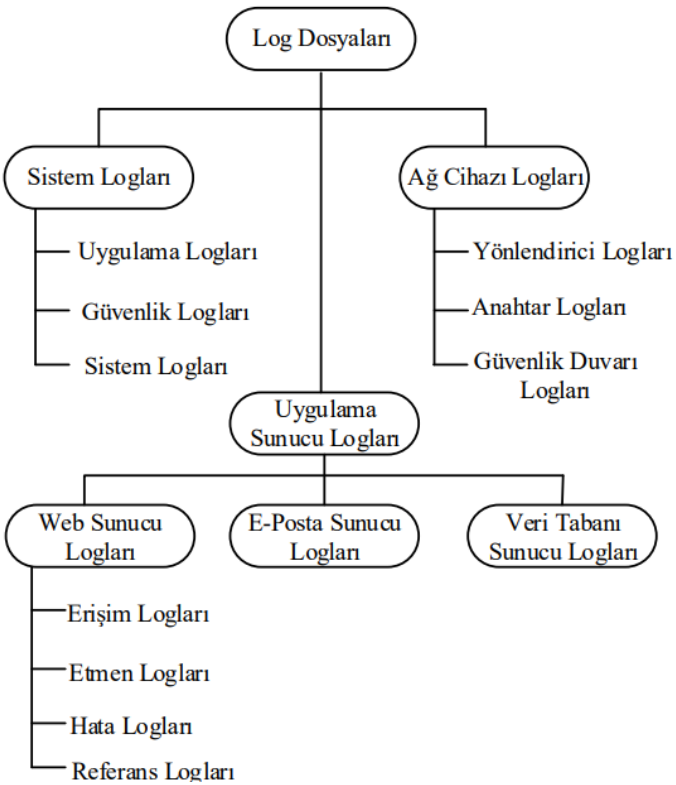
- Network sistemleri üzerinde gerçekleşen işlemlerin yol haritasını çıkartır.
- Network sistemleri üzerinde gerçekleşen kullanıcı işlemlerin kaydını tutar.
- Log kayıtları orta vadede siber saldırıların tespit edilmesini sağlar.
- kaba kuvvet (Brute Force), DDoS gibi saldırı türlerini kısa sürede tespit edebilir.
- Sistem yöneticilerine ve siber güvenlik uzmanlarına sistemin iyileştirilmesini sağlar.
- Loglama teknolojileri Siber saldırganları caydırıcı etkisi bulunmaktadır.
- Loglama cihazları güvenlik duvarı ile entegre çalışabilmektedir.
- Loglama cihazları farklı kimlik doğrulama teknolojilerini kullanmaktadır.

Log Dosya Türleri

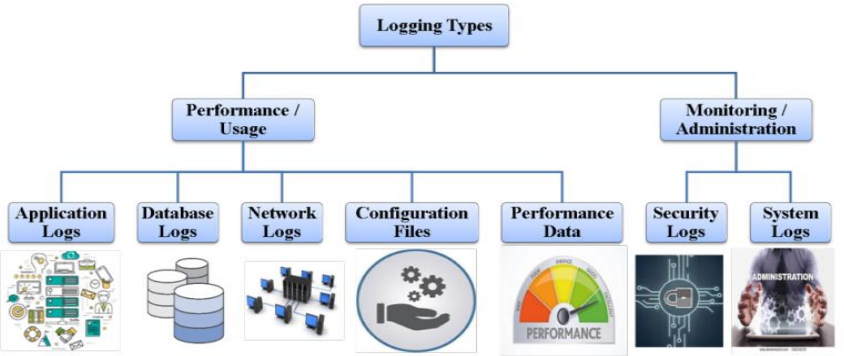
Sistemler üzerinde tutulan log kayıtlarının birçok dosya türü bulunmaktadır. Log kayıtları Kullanılan cihazlar, görevler, sunucu

türleri, bulut sistemler ve veri tabanı kullanımı, kullanıcı işlemlerinden kaynaklı işlemler, login işlemleri gibi yapılan çeşitli işlemlere göre değişiklik gösterebilir. Bu bağlamda web sunucu, veri tabanı sunucusu, işletim sistemine göre log kayıtları farklı sistemlerde farklı dosya formatlarında, gibi farklılıklar gösterebilir (Baykara vd., 2016: 2).

Farklı platformlar üzerinde log kayıt türleri farklı dosya türlerinde saklanır. Bu durum yığınla log kayıtları içinden istenilen log kaydına ulaşılmasını sağlar. Farklı kaynaklardan alınan şekil 4 ve Şekil 5 de farklı log dosya türleri yer almaktadır (Landauer 2019: 15-19).



Şekil 4. Log dosya türleri (Baykara vd., 2016:2).



Şekil 5. Log dosya türleri (Ali vd., 2021: 2).

Loglama Cihazları ve Özellikleri

Loglama cihazları bilişim sektöründe, hizmet portalı ve hotspot cihazları olarak karşımıza çıkmaktadır. Bu cihazlar internet erişimi ve sonrasında kullanıcıların birtakım bilgilerini alarak kayıt altına alan bir cihazdır. İçinde bir işletim sistemi yer alınır. Aynı zamanda donanımsal olarak bir anakart, işlemci, bellek ve harddisk gibi elektronik aksamlar yer almaktadır. Log kayıtlarının alınması için gerekli yazılımlar da mevcuttur.

Network üzerinde işlem yapmak isteyen misafirlerinizi güvenli bir şekilde internete çıkmaları sağlanır. Bu işlemler sırasında kullanıcıların, erişim ile ilgili log kayıt bilgileri 5651 Sayılı kanuna uygun olarak kayıt altına alınır. Bu işlemler kullanıcıların belirli adımları geçerek internete erişimleri sağlanır. Log kayıt cihaz türleri şekil 6 da yer almaktadır.

Log dosyalarının iyi analiz edilmesi, doğru yorumlanması, normalizasyon işleminin yapılması, işlem yapan kullanıcılar hakkında veri toplanması, farklı sistemlere entegrasyon kolaylığı, logların güvenliği, kota bant genişliği yönetimi ve kimlik doğrulama işlemleri gibi önemli adımları barındırmaktadır. Log cihazlarının önemli özellikleri aşağıda yer almaktadır (Coslat 2022).

- Web arayüzü, Türkçe dil desteği

- KVKK (Kişisel Verilerin Korunması Kanunu) ve 5651 log kanununa uygunluk
- Elektronik Zaman Damgası ile imzalama
- Gelişmiş kullanıcı ve grup yönetimi
- CSV ile çoklu kullanıcı işlemleri ve TC Kimlik ile kullanıcı kontrolu
- SMS ile kayıt, Çoklu kayıt seçeneği ve Sponsor ile kayıt seçenekleri
- Yönetici bilgilendirmesi, Kota ve bant genişliği yönetimi
- DHCP (Dinamik Ana Bilgisayar Yapılandırma Protokolü) ve Radius Sunucu entegrasyonu
- MAC ve IP işlemleri
- Reklam, Anket işlemleri ve detaylı raporlama
- Üçüncü parti uygulama ile entegrasyon
- HTTPS karşılama ve güvenlik duvarı özellikleri
- External Portal için Active Directory entegrasyon

2FA Log Cihazı Özellikleri

Kullanıcıların sistemlere erişimi sırasında şifrelerinin çalınması durumunda siber saldırganların hesabınıza veya ağınıza erişimini engeller. Kimlik Doğrulama sistemi sayesinde ağınız daha güvenli hale gelir. Kimlik doğrulama sistemi SMS (Kısa Mesaj Hizmeti) ya da Google Authentication ile 2FA (Two Factor Authentication) iki aşamalı kimlik doğrulama yaparak VPN (Sanal Özel Özel Ağ) bağlantılarınızı daha güvenli hale getirmektedir. 2FA log kayıt cihazlarının bazı özellikleri aşağıda yer almaktadır (Coslat log çözümleri 2022).

- Sistemlere SMS ile entegrasyon
- Sistemlere Google Authentication ve LDAP ile entegrasyon
- Misafir raporlama ve kural ekleme işlemleri
- AD Security gruba göre telefon numarası alma

- Farklı cihazlarda eş zamanlı çalışma

Log Cihazlarında Güvenlik Duvarı Özellikleri

Log kayıt cihazlarında güvenlik duvarı özelliğinin olması, sistemlere erişim sağlayan kullanıcıların daha güvenli erişimi söz konusudur. Bu durum adli analiz ve siber güvenliğin sağlanması açısından, sistemleri siber uzayda daha güvenli kılmaktadır.

Log kayıt cihazları, gelişmiş güvenlik duvarı özellikleri ile siber tehditlere karşı daha çok direniş sağlar. Piyasada güncel loglama cihazları içinde güvenlik duvarı eklentisi olan log cihazlarıda bulunmaktadır (Coslat log çözümleri, 2022).

Güvenlik duvarı eklentisi olan log cihazları özellikleri;

- KVKK ve 5651 Kanuna uygun loglama
- Türkçe web arayüz ve farklı dil desteği
- TC Kimlik ile Doğrulama işlemi
- Sistemlere SMS Kayıt olarak erişim sağlama
- URL (Tekdüzen Kaynak Bulucu), içerik filtreleme
- Sertifikasız Smart HTTPS (Güvenli Metin Aktarma Protokolü) filtreleme
- İnternet trafiği yönetimi ve limit işlemleri
- Yük dengeleme ve yük aktarma
- VPN desteği
- Siber saldırılara karşı daha etkili güvenlik hizmeti
- Kurumsal karşılama ekranı özelliği
- Farklı sistemlere entegrasyon

Şekil 6. Log kayıt cihaz türleri (Coslat log çözümleri, 2022).

Literatür Taraması

Literatür taramalarında yapılan siber saldırı türleri araştırılmıştır. Bu araştırmalar sonucunda en çok kullanılan saldırı türleri arasında DDoS (Dağıtık Hizmet Engelleme), otlama saldırıları, kaba kuvvet saldırılar önü çekmektedir. Bu saldırıların ülke, kurum, işletme, kişi ve kullanıcılar üzerinde birçok kayıp oluşturduğu görülmektedir. Yapılan bu siber saldırıların tespit edilmesi ve önlenmesi, log kayıtları üzerinden yapılmasının mümkün olduğu görülmektedir. Bu sistemlerin geliştirilmesinde IDS ve IPS güvenlik sistemlerinin etkin ve fazlaca kullanıldığı yapılan literatür araştırmalarında görülmektedir.

Bilgi güvenliği kapsamında siber güvenlik konusu öncelikli alan olması itibariyle çalışma konusunun ne kadar önemli olduğu bilinmektedir. Siber güvenlik anlamında ABD, RUSYA NATO, ÇİN gibi örgüt ve ülkelerde yazılım, donanım, dergi, makale, kitap gibi çeşitli çalışmalar yapılmaktadır. Ülkemizde siber güvenlik sistemleri çalışmalarına desteğini arttırması beklenmektedir (Çahmutoğlu, 2020: 13).

Sistemler üzerinde siber güvenlik unsurlarının saldırıları tespit etme ve önleme konusunda daha fazla çalışmaların yapılması gerektiği, siber saldırılara karşı alınan önlemlerin sürekli güncellenmesi gerektiği görülmektedir. Teknolojik gelişmeler

birçok alanda ileri safhada kullanılmaktadır. İşlemlerin dijital ortamda yürütülmesinde önemli seviyede fayda sağlamaktadır. Aynı zamanda işlemlerin dijitalleşmesi sistemler üzerinde zafiyetlerin oluşmasına sebep olmuştur. Ülkelerin dijital sistemler üzerinde siber anlamda zafiyetlerin giderilmesi için bu anlamda birtakım çalışmaların yapıldığı yer almaktadır. Siber zafiyetlerin giderilmesi için çeşitli teknik yöntemlerin kullanıldığı bilinmektedir. Bu yöntem ve teknikler çeşitli strateji ve politikalar yer almaktadır. Siber güvenlik, sistemler üzerindeki zafiyetlerin giderilmesi için çeşitli bilgi işlem yöntemleri kullanılmaktadır. Ülkemizde siber güvenlik anlamında çeşitli çalışmaların yapıldığı yer almaktadır (Koca 2022: 1).

Log kayıtları sistemler üzerinde yapılan işlerler hakkında birçok konuda yardımcı olmaktadır. Dijital sistemlerin her türlü işlem kayıtlarının tutulmasını sağlamaktadır. Sistemler üzerinden log kayıtlarının tutulması için birçok yazılım ve donanım araçları kullanılmaktadır. Dijital ortamda sistemler üzerinde kayıt altına alınan log kayıtları network uzmanları, adli analiz ve siber suçların tespitinde kullanılmaktadır. Bu çalışmada sistemler üzerinde tutulan log kayıtlarının analizi ve ne kadar önemli olduğuna yer verilmiştir (Bayraktaroğlu, 2009: 3-7).

Siber uzayda siber saldırılara maruz kalan birçok sistem bulunmaktadır. Saldırılar içinde en yaygın olarak bilinen ve en çok saldırı yapılan zafiyet türü SQL (Yapılandırılmış Sorgu Dili) enjeksiyon saldırılarıdır. Bu saldırıların önlenmesinde birçok sistem kullanılmaktadır. Bu çalışmada SQL enjeksiyon zafiyetleri ve yapılan saldırı türleri yer almakta ve SQL enjeksiyon saldırıları incelenmiş ve analiz edilmiştir (Çankuş, 2023: 65).

Bilgi güvenliği, bilgi sistemlerinin en önemli konuları içinde yer almaktadır. Siber güvenlik, kişi, kurum, işletme, kamu kurumlarının başlı başına ele aldığı konular arasında yer almaktadır. Bu çalışmada siber güvenlik araçları ve sızma test araçları yer almaktadır. Siber güvenlik anlamında sistemler üzerinde yer alan

siber güvenlik araçlarının yerli ve milli kaynaklar dahilinde yapılmasına yer verilmiştir (Şentürk, 2018: 129).

Loglama teknolojilerinin daha yaygın hale gelmesi ve teknoloji açıdan daha donanımlı olması gerekmektedir. Network sistemleri üzerinde kullanılan loglama teknolojileri incelenerek network analiz programları ve açık kaynak işletim sistemleri kullanılarak toplanan loglar analiz edilmesi düşünülmektedir. Log sistemleriyle ilgili çalışmaların yapıldığı görülmektedir. Güvenlik şirketlerinin siber saldırıları yazılım ve donanım altyapısını geliştirerek engellemeye çalışmaktadır. Toplanan loglar belirtilen açık kaynak işletim sistemleri ile incelenerek sistemlere sızmaya çalışan IP adresleri tespit edilmesi düşünülmektedir (Gül, 2019: 5).

Günümüz teknolojilerinde internet kullanıcılarının hızlı artışı siber güvenlik gereksinimlerini de beraberinde getirmiştir. Bilgisayar korsanlarının sistemlere saldırılarını arttırmış ve saldırı metodolojileri ve tekniklerini arttırmıştır. Bu çalışmada öğrencilerin sanal ortamda Ağ Saldırı Tespit Sistemi kurulumu ve kullanımını öğrenmeleri için bir platform geliştirilmiştir. Yapılan siber saldırıların IP adresleri üzerinden algoritmalar kullanarak siber saldırıları tespit etmesi sağlanmıştır. Bu sayede öğrenciler siber saldırıların belirlenmesi ve önlenmesi için birtakım kuralları öğrenmeleri sağlanmıştır (Kamal vd., 2022: 32-45).

IDS (Saldırı Tespit Sistemi), zararlı sistem faaliyetleriyle ilişkilendirilen IP (İnternet Protokol) adresleri gibi göstergelerin, yani IoC'lerin **(Kontrolun Tersine Çevrilmesi)** uygun ve doğru bir şekilde mevcut olmasına dayanır. Ancak, bu göstergelerin basit doğası ve sınırlı geçerliliği, siber tehditlere karşı korumayı zayıflatır. Taktikler, Teknikler ve Prosedürler, saldırgan davranışları hakkında soyut bilgiler sağlar, IDS kullanımı sayesinde otomatik algılamayı engeller ve ancak insanlar tarafı dan okunabilecek formatlarda bulunur. Bu makalede, karmaşık sistem davranışlarının algılanabilir desenlerini üreterek log verilerinden siber tehdit istihbaratı çıkaran ve IoC ve TTP (Taktik Teknik, Protokol) avantajlarını birleştiren bir yaklaşım öneriyoruz. Mevcut yaklaşımlardan farklı olarak, şüpheli

log olaylarını ortaya çıkarmak için log verileri anormallik tespiti kullanırız, bu olaylar ardışık kümeleme, desen tanıma ve iyileştirme için kullanılır. Değerlendirmelerimiz, başka bir sistemde aynı saldırının algılanması için uygun çok adımlı bir saldırıya karşılık gelen otomatik olarak çıkarılan tehdit istihbaratının uygun olduğunu göstermektedir (Chan ve Yeoh, 2017: 5-8).

Ülkemizin siber suç anlamında durum tespiti yapılmış, yapılan siber saldırılara karşı yöntemler araştırılmıştır. Siber suçları kapsayan kanun ve yasaların 2016 yılında yapılmasından dolayı, çalışma 2016 yılı öncesini kapsamamaktadır. Yöntem olarak metodoloji seçilmiştir. Makale çalışması kapsamında en çok çalışma yapılan ve aynı zamanda siber saldırı türü olan DoS (Hizmet Engelleme) ve DDoS (Dağıtık Hizmet Engelleme) saldırı türü olduğu sonucuna ulaşılmıştır. Siber saldırıların tespit yöntemi olarak Random Forest karar ağacı yöntemi kullanılmıştır (Hatipoğlu ve Tunacan, 2021: 1-2).

Log Yönetimi ve Log Tutulmasinda Yasal Zorunluluk

Log Yönetimi

Sistemler üzerinden toplanan logların yönetimi, logların toplanması kadar önemlidir. Log yönetimi, sistemler üzerinde gerçekleşen logların kayıtları, filtreleme, doğrulama, normalizasyonu ve analizi gibi işlemleri kapsamaktadır. Log yönetimi, log toplama standartları kapsamında yapılmalıdır (Gül 2019: 9-10).

Siber güvenlik anlamında log yönetimi ve analizi logların toplandığı kurumlara göre farklılık gösterebilir. Yetkili personel dahilinde loglar incelenir. Yapılan bu işlemler dahilinde loglar network haritalarının oluşturulmasını iyileştirmesini sağlamaktadır (Coslat log çözümleri, 2022).

Bütünlük (Integrity)

Log kayıtlarının yönetimi ve analizleri bağlamında herhangi bir işlemden geçmemesi, log kayıtları üzerinde bir değişikliğin söz konusu olmaması, log kayıtlarının yetkili personel tarafından işlem yapılması gibi işlemleri kapsamaktadır. Bu anlamda log güvenliği logların bütünlüğü sağlanır (Bayraktaroğlu, 2009: 15-17).

Zaman damgası (Time Stamping)

Log kayıtlarının doğruluğunu, bütünlüğünü ve siber saldırıların ortaya çıkartılabilmesi için log kayıtlarının zaman damgası ile imzalanması gerekmektedir. 5070 sayılı **"Elektronik İmza Kanunu"** uyarınca; bir elektronik verinin, üretildiği, değiştirildiği, gönderildiği, alındığı ve / veya kaydedildiği zamanın tespit edilmesi maksadıyla, elektronik sertifika hizmet sağlayıcısı tarafından elektronik imzayla doğrulanan kaydı, ifade eder." şeklinde ifade edilmektedir. Yapılan dijital işlemlerde zaman damgası kullanımı zorunlu hale getirilmiştir (Bayraktaroğlu, 2009: 15-17).

Topluma açık alanlarda internet kullanımı hükümlerinin yer aldığı **5651 yasalı kanuna** göre toplu internet kullandıran kurumlar ve mekânların bu damgayı edinmesi mecburi hale getirilmiştir. Bu kanun sayesinde artık siber suç işleyen ceza alması sağlanarak kurum sahiplerinin korunması sağlanmıştır. Tutulan log kayıtlarının işlem süreci istinaden logların kayıt tarihi ve saat bilgileri baz alınarak zaman damgası kimlik doğrulamayla imzalanır. Kullanıcılar tarafından yapılan işlemler sırasında oluşan log kayıtlarının korelasyon yapabilmesi büyük bir öneme sahiptir (Zaman damgası, 2021).

5070 sayılı kanun, elektronik verinin, üretildiği, değiştirildiği, gönderildiği, alındığı ve kaydedilmesi zamanın tespit edilmesi amacıyla, hizmet sağlayıcısı tarafından elektronik imzayla doğrulanan kayıtların resmi olarak kanıtlanmasıdır. Zaman damgaları, verinin belirtilen tarihte oluşturulduğunu, değiştirildiğini veya gönderilip alındığını bilgisini doğrulamak için kullanılmaktadır. Zaman sunucuları üzerinde açık bir anahtar

teknolojisi kullanılarak verinin bütünlüğü ve varlığını onaylanır. Zaman damgası, asıl amacı veriyi doğrulamadır. İnternet teknolojileri içinde birçok alanda kullanılmaya ve yaygınlaşmaya başlanmıştır. Dijital içerikler çoğaldıkça, marka, kurum, belediye ve devlet hizmetleri birçok işlevi bu kullanılmaya başlamıştır. Dijital işlemlerde verinin doğruluğunu, bütünlüğünü ve varlığını kanıtlamak için zaman damgası kullanılmaktadır (Zaman damgası kullanımı, 2021).

Siber saldırılar sonucu ortaya çıkartılabilecek delillerin ve adli bilişim çalışmalarının daha verimli kullanılabilmesi için zaman damgaları kullanılmaktadır. Çalışmada, logların olaylarla ilişkilendirilmesi, analizi ve yorumlanması açısından kimlik doğrulama anahtarı olarak kullanılacak en önemli eşleştirme zaman damgası olduğu açıkça ortaya çıkmaktadır (dijital içeriklerin korunması, 2021).

6868 Kişisel Verilerin Korunma Kanunu (KVKK)

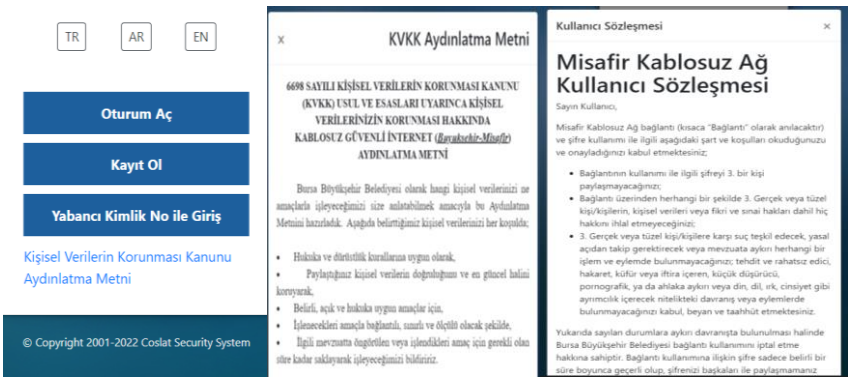
Sistemlere erişim sırasında oluşan birçok kişisel veri bulunmaktadır. Sensörler ile toplanan kişisel veriler bu bağlamda belli kurallara göre saklanmakta ve korunmaktadır. Ülkemizde, 6698 sayılı KVKK kapsamında dijital ortamda üretilen tüm kişisel veriler, kanun hükmünde korunma altına alınmıştır. Kişisel verilerin korunması temelinde, kişi haklarının yanında temel hak ve özgürlüklerinin de ihlalinin önlenmesi düşünülmekte, bunun yanı sıra kişinin mahremiyetinin korunması ön planda yer almaktadır. Bu bağlamda kişisel verilerin korunması, insanın kişiliği ve özel hayatın gizliliği hedeflenmektedir (Fidancı,2022: 5-7).

5661 Loglama Kanunu

5651 Sayılı Kanun ile **ISO 27001 (Bilgi Güvenliği Yönetim Sistemi Standardı)** ve ayrıca KVKK log kaydı tutulmasını zorunlu hale getirmiştir. 5651 Sayılı kanun kapsamında aşağıdaki sağlayıcıların log tutması zorunludur. Ülkemizde 5651 sayılı kanun ve TİB (Telekomünikasyon İletişim Başkanlığı) yönetmelikleri

gereği, kurumlar log kayıtlarını tutmakla yükümlüdür. Log kayıtları iki yıl saklanması zorundadır (Korkmaz, 2023: 13-17).

Çalışma kapsamında yapılan işlemlerden oluşan kişisel veriler Şekil 7'de yer alan (a) (b) ve (c) resimleri ile kullanıcılar kişisel verileriyle giriş sağlamadan KVKK aydınlatma metninde açıklamalar yer almakta, oluşan dijital veriler kanun kapsamında korunmakta olduğu bilgilendirilmektedir. Kişisel verilerin korunması, verilerin gizliliği bu kapsamında analizi ve işlenmesi sağlamaktır (Özargın, 2023: 88).



Şekil 7. Kişisel Verilerin Oluşturulması (a): Kişisel Verilerin giriş Paneli (b): KVKK Aydınlatma Metni (c): KVKK Misafir Kullanıcıların Sözleşmesi (Loglama sistemi, 2023).

Siber Güvenlik ve Güncel Siber Saldrilar

Siber Güvenlik

Siber güvenlik konusu siber uzayın en kapsamlı konuları arasında çalışma yapılması gereken alanların başında gelmektedir. Güvenlik her alanda olduğu gibi dijital dünya yer alan en önemli konu başlıkları arasında yer almaktadır. Sistemler üzerinde yer alan her türlü veri, yetkisi olan kişi/kurumlar tarafından kullanılmaktadır. Bu durum sistemlerin bilgi güvenliği açısından en önemli amaçları içinde yer alan bilgi güvenliğinin sağlanmasıdır. Sistemler üzerinde

yer alan verilerin teknolojik araçlarla işlenerek bilgiye dönüştürülmesi, bir çıktı alınması ise sistemlerin hedefleri arasında yer alır. Sistemlerin amaçları arasında veriyi yetkisi olan kişi veya kurumlara ulaştırmasıdır.

Siber Saldırılar

Global sistemde, hizmet sağlayan internet teknolojileri, yoğun bir siber saldırıya maruz kalmaktadır. Bu saldırılar gün geçtikçe şiddetlenmekte maddi ve manevi zararlara neden olmaktadır. Sistemler üzerinde oluşabilecek zafiyetlerin önceden belirlenmesi biraz zaman alabilmektedir. Bu anlamda yeni teknolojiler her an siber saldırı tehdidi oluşturmakta ve her an saldırı alabilecek bir potansiyele sahiptir.

Dijital sistemlerin en önemli amaçları arasında bilgi güvenliğini yer almaktadır. Bu amaca ulaşmak için çeşitli siber güvenlik altyapılarının oluşturulması gerekmektedir. Yapılan siber saldırıların loglama teknolojileri ile tespit edilmesi araştırılması yapılarak, siber saldırıların engellenmesi veya bertaraf edilmesi için bir dizi önlemlerin alınması tez kapsamında yer almaktadır. Alınacak önlemler makine öğrenmesi teknikleri ile zafiyetlere karşın geliştirilen saldırıların desenlerinin belirlenmesi ve engellenmesi ile ilgili olacaktır (Ali, 2023: 3-5).

Ülkemizde En Çok Rastlanan Siber Saldırı Türleri

Ülkemizde en çok siber saldırıya maruz kalınan saldırı türleri aşağıda yer almaktadır. Bu saldırı türleri günden güne değişiklik göstermektedir. Teknolojik cihaza göre saldırı türü değişebilir. Bu saldırıların zafiyetleri farklıdır. Maddi ve manevi kayıpları da aynı şekilde değişim göstermektedir. Bu saldırı türleri log kayıtlarının incelenmesiyle tespit edilebileceği yapılan çalışmalarda görülmektedir. Bu bağlamda log sistemlerinin geliştirilmesi güvenlik sistemlerine entegre edilmesi gerekmektedir (Kara, 2023: 14-25,69).

Ülkemizde rastlanan saldırı türleri;

- Fidye Yazılımları
- Oltalama Saldırıları
- Kredi Kartı Dolandırıcılığı
- Dağıtılmış Hizmet Reddi Saldırıları (DDos)
- Mobil Cihazlara Yapılan Tehditler
- Kaba kuvvet saldırıları

YÖNTEM

Siber Saldırıların Tespit Edilmesi Acısından Logların Önemi

Günümüz dünyasında teknoloji kullanımında bir hayli yüksek seviyelere ulaşmıştır. Neredeyse tüm işlemlerin dijitalleşmesi bilgi güvenliği sorununu ortaya çıkarmıştır. Siber güvenlik anlamında yazılan tez çalışmaların eskiye nazaran arttığı görülmektedir.

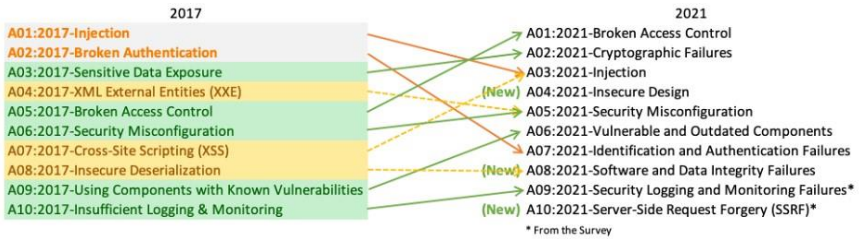
Siber saldırıların giderek arttığı ve siber saldırıların verdiği zarar ve kayıpların yüksek olması siber güvenlik konusunu akademik anlamda çalışmalar yapmaya sürüklemiştir. Siber saldırıların bertaraf edilmesi için logların incelenmesi gerektiği, bu incelemeye göre siber saldırıların tespit edilebileceği araştırılmaktadır. Son saldırılarda en çok saldırı yapılan web hizmetleri olduğu görülmektedir (Çınar ve Bilge, 2016: 1).

Ülkemizde birçok siber güvenlik modeli geliştirilmektedir. Bu modeller akademik çalışmalarla dahada güçlenmesi hedeflenmektedir. Çalışmalarda sektör paydaşlarının da payı unutulmamalıdır. Birçok ülkede olduğu gibi siber suçlar ülkemizde de gün geçtikçe artmaktadır. Siber suçlarla mücadele etmek, siber saldırı ve zafiyetlerini önlemek için birtakım çalışmaların biraz daha hız kazandırılması gerekmektedir. Siber suç anlamında birçok suç örgütü bulunmakta ve bunlara karşı hukuki işlemler yapılmaktadır. Bu anlamda siber suç, siber saldırı sonrası yaşanan kayıpların en aza indirilmesi, bu saldırıların bertaraf edilmesi gerekmektedir. Siber

güvenlik çalışmaların bu anlamda artması beklenmektedir (Sandilaç, 2022: 28-43).

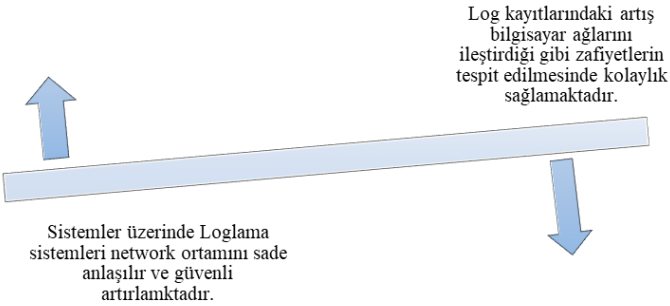
İnternet ortamında birbirleriyle iletişim halinde olan bilgisayarların iletişime geçtikleri bilgisayarlar ile ilgili loglar tutulmaktadır. Bu loglar siber saldırılar sonrası incelenmekte olduğu görülmektedir. Loglama teknolojilerinin geliştirilmesi yapılan siber saldırıları anlık olarak tespit etmesi için büyük bir öneme sahiptir. Siber saldırıların oluşturduğu maddi kayıplar büyüktür. Dolayısıyla alınan siber güvenlik önlemleri bu bağlamda maddi ve bilgi kaybı için önemlidir (Ali, 2021: 1-6).

Saldırı tespit sistemleri, en yaygın siber saldırılara göre inşa edilebilir. Fakat zamanla bu saldırı türleri değişiklik te göstermektedir. Bu durum söz ardı edilmemelidir. OWASP (Open Web Application Security Project) Açık Web Uygulama Güvenliği Projesi, raporunda bu durum ile ilgili bilgi verilmektedir. OWASP, 2017 yılında kritik 10 web uygulamasına ait güvenlik zafiyetlerini belirlenmiştir. Ayrıca 2017-2021 yılları arasındaki siber zafiyet olan uygulamaların değişimi Şekil 8 de yer almaktadır (OWASP, 2017).



Şekil 8. 2017-2021 yılları arasında en kritik web uygulamaları karşılaştırması (OWASP, 2017).

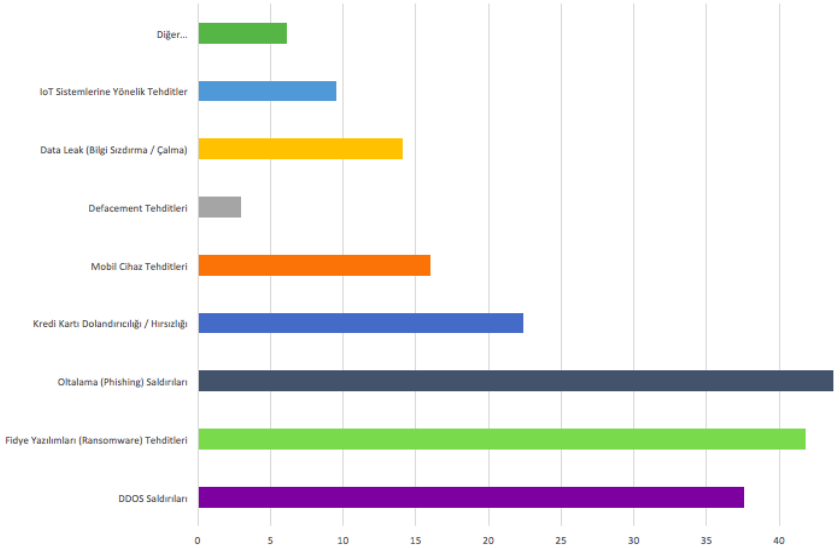
Network sistemleri üzerinde yer alan yazılım ve donanım cihazları üzerinde loglama teknolojilerinin kullanılması, sistemleri daha güvenli hale getirmektedir. Log kayıtları sistemlerde kullanılan yazılım ve donanım sistemlerindeki kullanım sıklığı sistemleri siber güvenlik açısından daha korunaklı hale getirmektedir. Şekil 2 da bu durum görselde yer almaktadır.



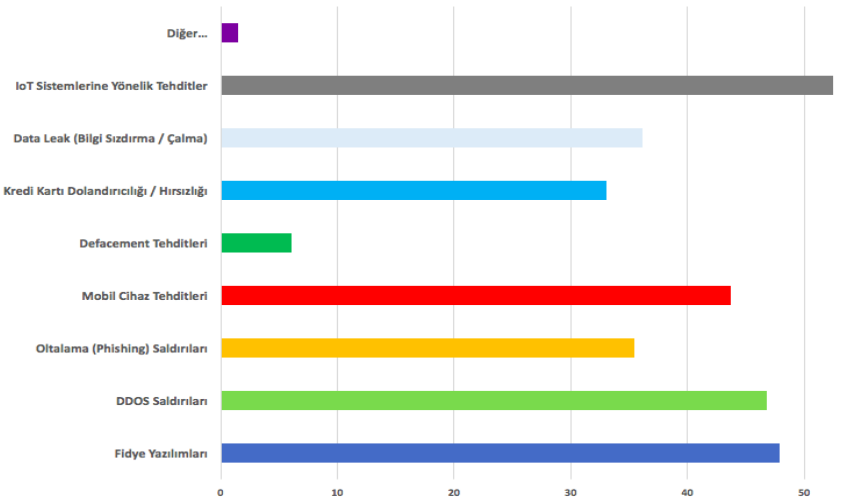
Şekil 9. Loglar ile siber güvenlik arasında kavramsal ilişki.

Microsoft Siber Saldırı Raporu

2016 Yılı baz alındığında öne çıkan siber saldırı türleri şekil 10 de yer almaktadır. 2017 Yılı baz alındığında öne çıkan siber tehditler şekil 11 de yer almaktadır. 2017 Yılında IoT (Nesnelerin İnterneti)sistemlerine yapılan siber saldırıların daha yoğun olduğu görülmektedir. Gelecek teknolojilerde canlıların hayatında IoT cihazları daha fazla yer alacağından önümüzdeki yıllarda IoT cihazlarına yapılan siber saldırılarında artacağı görülmektedir. 2016 yılına göre otlama saldırıları ilk sırada yer alırken, 2017 yılına göre IoT cihazları ilk sırada yer almaktadır (Microsoft Güvenlik Raporu, 2017).



Şekil 10. 2016 Yılı Siber Saldırı Türleri (Microsoft Güvenlik Raporu, 2017).



Şekil 11. 2017 Yılında Öne Çıkan Siber Tehditler (Microsoft Güvenlik Raporu, 2017).

Loglama teknolojilerinin daha yaygın hale gelmesi ve teknoloji açıdan daha donanımlı olması gerekmektedir. Network sistemlerinde üzerinde kullanılan loglama teknolojileri incelenerek network analiz programları ve açık kaynak işletim sistemleri kullanılarak toplanan loglar analiz edilmesi düşünülmektedir. Log sistemleriyle ilgili çalışmaların yapıldığı görülmektedir. Güvenlik şirketlerinin siber saldırıları yazılım ve donanım altyapısını geliştirerek engellemeye çalışmaktadır. Toplanan loglar belirtilen açık kaynak işletim sistemleri ile incelenerek sistemlere sızmaya çalışan IP adresleri tespit edilmesi düşünülmektedir (Özseven ve Düğenci 2011: 18-21).

Güvenlik Duvarı, sistemler üzerinde olması gereken en temel güvenlik aracıdır. Bunu yanında sistemler üzerinde saldırı tespit ve saldırı önleme sistemlerinin de olması gerektiği günümüz ötesinde ihtiyaç duyulan sistemler arasında yer almaktadır. Siber saldırı önleme ve tespit sistemleri güvenlik duvarı sistemleriyle entegreli çalışmalıdır. Günümüz teknolojilerinde, Siber güvenlik Sistemleri, güvenlik duvarı ve Saldırı tespit ve Saldırı Önleme Sistemini içinde barındıran bir topoloji olarak geliştirilmesi sağlanır.

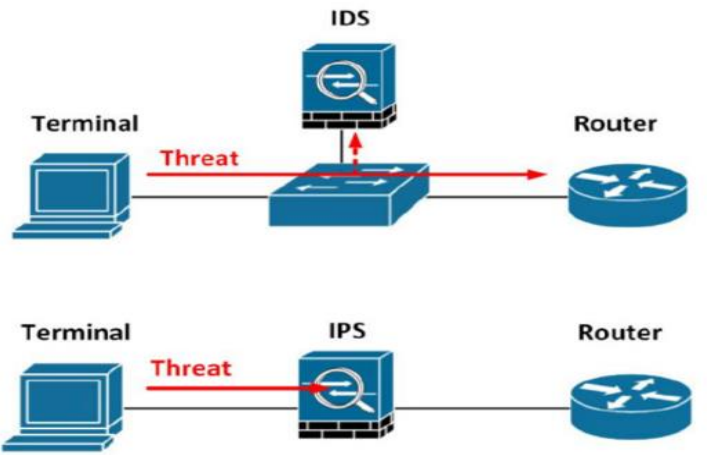
Saldırı Tespit Sistemi IDS

Yapılan siber saldırılar network üzerinde yer alan paketlerin incelenmesiyle tespit edilmektedir. IDS, network üzerindeki paketleri inceleyerek siber saldırı tespit eden sistemlerdir. Bilgisayar ağları üzerinde veri paketlerini inceleyen, network içinde güvenlik duvarı gibi konumlandırılan güvenlik sistemidir. Konumlandırıldıkları yerdeki paketleri inceleyerek siber saldırıların tespit edilmesini sağlarlar. IDS sistemlerinin bazı dezavantajları da bulunmaktadır. Büyük yapılı ağlarda bazı incelenmeyen paketler kalabilir. Kriptolu paketlerin iç yapısı incelenemeyebilir (Zitta vd., 2017: 5-9).

Saldırı Önleme Sistemi IPS

İnternet üzerinden bilgisayar sistemlerine yapılan siber saldırıların önlenmesini sağlayan sistemlerdir. Sunucu üzerine kurulu olan yazılım sistemleridir. Donanımsal cihaz şeklinde de olabilir. Ağ trafiğini çok hızlı inceleyerek saldırıların gerçek zamanda tespit edilmesini sağlar. Aynı zamanda güvenlik seviyesi yüksek olan sunucu paketlerinin incelenmesinde ve siber saldırı tespitinde kullanılmaktadır (Zitta vd., 2017: 5-7).

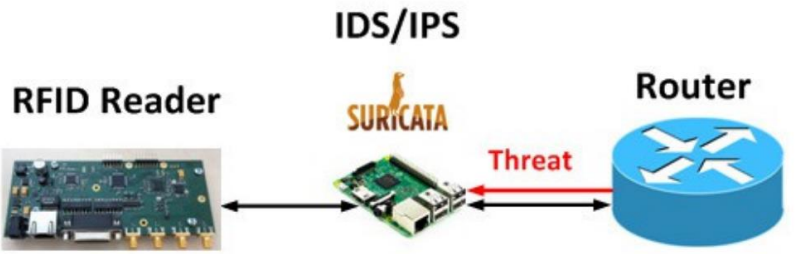
IPS sistemi Bilgisayar ağlarında hızlı çalışmaktadır. Saldırı tespit uyarı sistemi gerçek zamanlı çalışmaktadır. Ağ trafiği içinde kriptolu paketlerin içeriğini incelemektedir. Kurulu olan bilgisayarda çalışmaktadır. Şekil 13 de yer alan uygulamaların geliştirilmesi, bu uygulamalarda IDS/IPS saldırı tespit ve önleme sistemi yer almaktadır (Zitta vd., 2017: 5-9).



Şekil 12. IDS ve IPS topolojileri (Zitta vd., 2017: 5-9).

Bu çalışmada, LLRP (Düşük Seviye Okuyucu Protokolü) arayüzü için geliştirilen algılama kuralları, Raspberry Pi 3 kullanarak UHF (Ultra Yüksek Frekans) RFID (Radyo Frekansı Tanımlama) okuyucu için IDS ve IPS (Saldırı Önleme Sistemi) çözümlerinin uygulanmasını sağlamaktır. Geliştirilen güvenlik

çözümü, LLRP destekleyen mevcut RFID okuyucuların güvenliğini sağlamak için kullanılmaktadır. Ayrıca IPS/IDS çözümlerinin karşılaştırılmasını sağlamaktır. Şekil 13 de yer alan uygulamaların geliştirilmesi, bu uygulamalarda IS/IPS saldırı tespit ve önleme sistemlerinin daha etkili kullanılması, bu araştırmaların önü açık olduğu söylenebilir. IDS/IPS farklı güvenlik platformlarına entegre edilmesi araştırılabilir (Zitta vd., 2017:9-11).



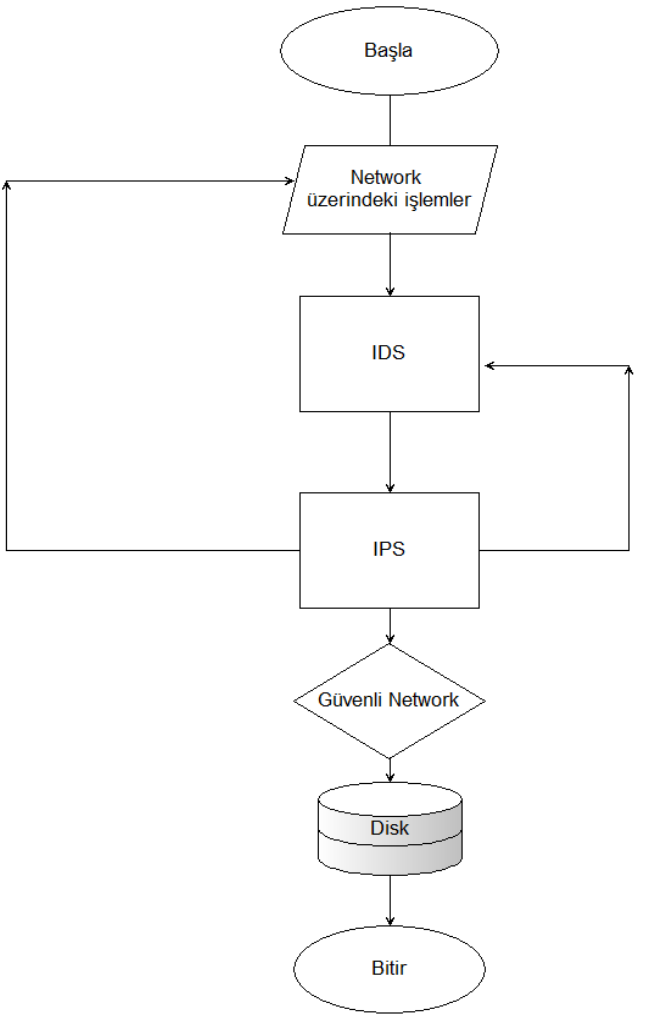
Şekil 13. IDS/IPS çözüm topolojisi (Zitta vd., 2017: 5-9).

Saldırı Tespit Sistemi ve Saldırı Önleme Sistemi Tasarımı

Günümüz teknolojilerinde, siber saldırılar gün geçtikçe artmakta ve aynı zamanda verdiği tahribatlar maddi anlamda büyük kayıplar yaşatmaktadır. Sistemlere yapılan siber saldırıların tespit edilmesi ve önlenmesi açısından log kayıtlarının hayati önemi olduğu söylenebilir. Bu kapsamda log kayıtlarının elde edilen istihbarat verilerinin kapsamlı incelenmesi sonucu geliştirilen açık kaynak uygulama sayesinde siber saldırıların tespit edilmesi ve önlenmesi sağlanabilmektedir.

Açık kaynak uygulama içerisinde IPS / IDS sistemlerinin kullanılmasıyla siber saldırı tespit ve önleme sistemi geliştirilmesi mümkündür. Açık kaynak uygulama python programlama dili kullanılabilir. Açık kaynak siber saldırı önleme sistemi IDS ve IPS sistemlerinin ne zaman nasıl çalışacağını planlayan kodlardan oluşmaktadır. Bu anlamda açık kaynak saldırı önleme sistemi, IDS ve IPS sistemlerini tetikleyerek network sistemlerinin güvenliği sağlanır. Geliştirilen açık kaynak uygulama sayesinde belirlenen

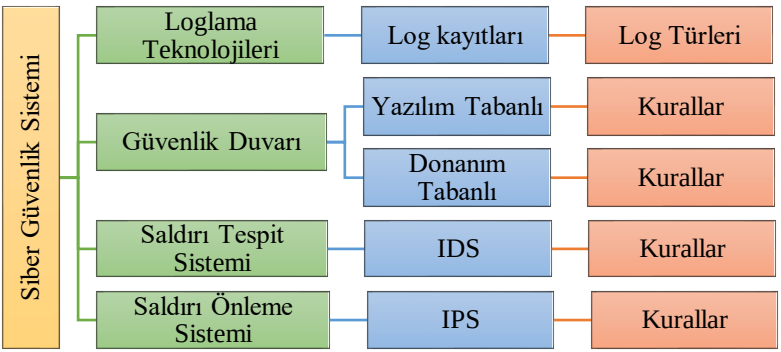
siber saldırı kuralları IPS ve IDS sistemi bünyesinde gelen saldırı IP adresine uygulanması sağlanır. Bu sayede hedef IP adresine gelen kullanıcıların saldırı yapan IP adresleri tespit edilerek, bu IP adreslerine kuralların uygulanması sağlanır. Bu kurallar sonucunda saldırgan IP adresi engellenerek sistemlere erişimi iptal edilir. Saldırı önleme sistemi çalışma sistemi şekil 14 de yer almaktadır.



Şekil 14. Siber güvenlik sistemi diyagramı.

Sistemler üzerine yapılan kaba kuvvet saldırıları kullanıcıların kullanıcı adı ve şifrelerini deneyerek sistemlere erişim sağlanması için yüzlerce deneme yaparak oluşan saldırı türüdür. Kaba kuvvet saldırıların engellenmesinde en önemli güvenlik önleme sistemlere kullanıcı bilgilerini deneme sayısı koymasıdır. Sistemlere giriş noktasında IPS sistemi devreye girerek kaba kuvvet saldırıların trafiğini engelleyebilir.

Son zamanlarda günümüz saldırılarında IPS ve IDS sistemlerinden kaçabilen siber saldırı türlerinde çıkmıştır (Zitta vd., 2017: 5-9). Şekil 16 da yer alan siber güvenlik sistemi log kayıtlarının alınması ve incelenmesi sonucu ortaya çıkmıştır. Loglama sistemleri artık siber güvenlik sistemi içinde yer alabileceği mümkündür. Bu anlamda siber güvenlik sistemi içinde loglama teknolojileri yer alarak sistemler üzerinde güvenliğin arttırılması sağlanır.

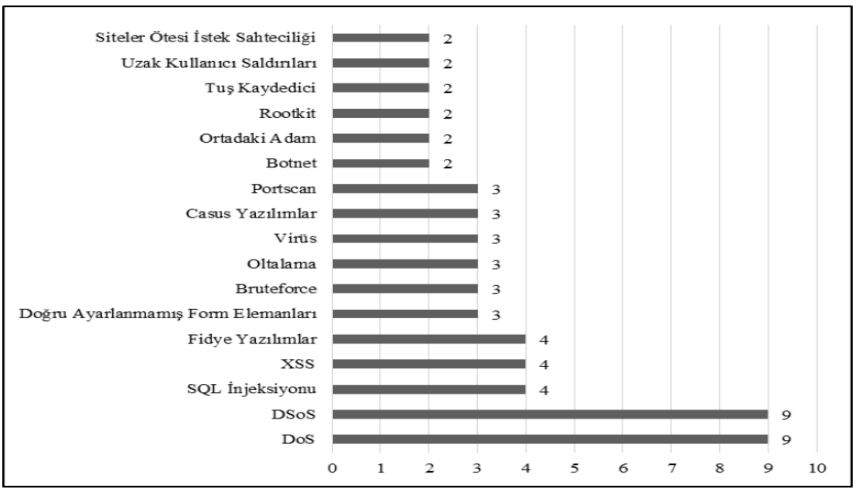


Şekil 15. Siber Güvenlik Sistemi.

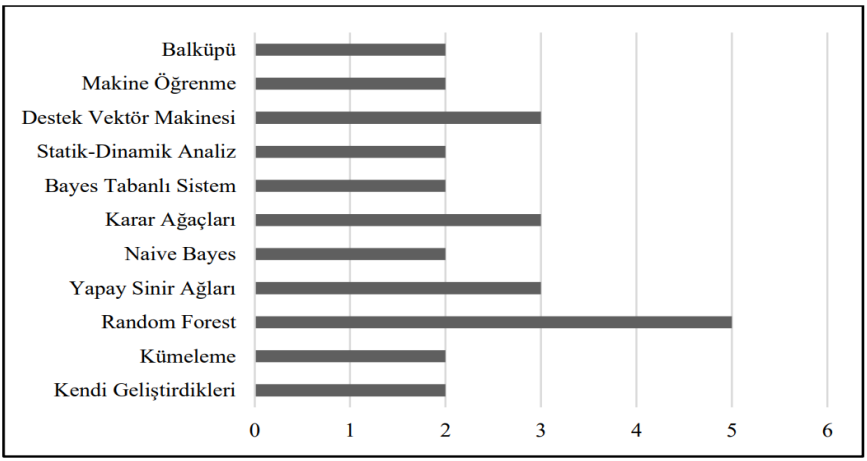
Şekil 17 de 2016-2020 yılları arasındaki incelenmiş siber saldırı türleri yer almaktadır. Siber saldırı karşılık tespit ve engellemek için kullanılan yöntemlere ait olan frekans bilgileri yer verilmiştir. Şekil 18 de en çok kullanılan siber saldırı tespit yöntemlerine ait olan frekans bilgilerini yer almaktadır.

Yapılan makale çalışmasında, ülkemizde rastlanan en çok saldır türleri ve saldırı tespit yöntemleri yer almaktadır. Bu bağlamda yapılan makale çalışması, bu çalışmada geliştirilmek istenen açık

kaynak saldırı tespit ve saldırı önleme sistemi ne katkı sağlayacağı aşikardır. Bu tür çalışmalar siber saldırı tespit ve önleme sistemi geliştirilmesinde yararlı olacağı düşünülmektedir (Hatipoğlu ve Tunacan 2021: 6).



Şekil 16. Toplamda en çok saldırı türleri (Hatipoğlu ve Tunacan, 2021: 10).



Şekil 17. En çok kullanılan saldırı tespit yöntemleri (Hatipoğlu ve Tunacan, 2021: 12).

Teknolojik gelişmelerin hız kazandığı global dünya düzeninde savaşlar yerini, siber saldırılara bırakmaktadır. Bu anlamda savaş maliyetleri en aza indirilerek siber saldırı yaparak sistemlere zarar vererek ülkelerin gelişmesini engellemektedir. Yapılan birçok siber saldırılar karşısında sistemler büyük zarar uğratılarak birçok veri kaybı ve maddi zarar oluşturmaktadır (Orak 2022:9).

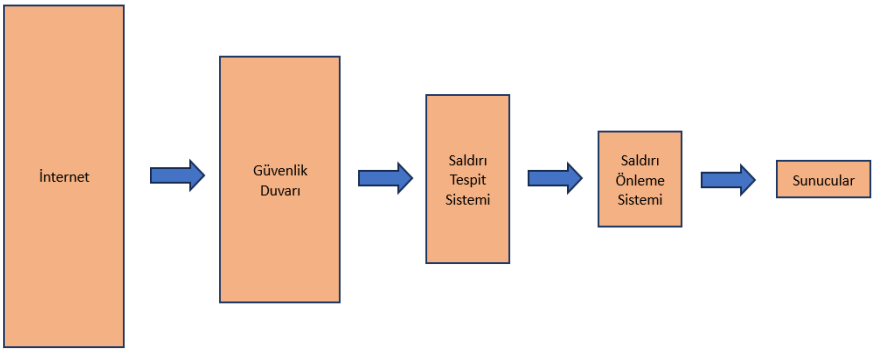
Dijital dünyada bilgi güvenliği en üst safhada yer almaktadır. Kişi, kurum, işletme, kamu ve özel kurumlar, aynı zamanda devlet, siber güvenlik anlamında siber zafiyetlerin giderilmesi çalışmaları yapmaktadır. Kurumsal anlamda bilgi güvenliği dış kaynaklardan yararlanılmaktadır.

Siber zafiyetlerin tespit edilmesinde ve önlenmesinde log kayıtlarının çok önemli bilgiler verdiği yapılan çalışmalarda yer almaktadır. Bu anlamda saldırı tespiti ve saldırı önleme sistemi geliştirilmesinde açık kaynak sistemlerin kullanıldığı görülmekte ve olumlu sonuçların verdiği görülmektedir. Bu anlamda siber saldırıların tespiti ve önlenmesinde, açık kaynak sistemler kullanılmıştır. Bu anlamda araştırmacılara, Saldırı tespit sistemi ve önleme sistemi geliştirilmesinde çeşitli teşvik ve desteklerin verilmesi gerekmektedir. Bu anlamda siber saldırı tespitinde kullanılan IDS/IPS açık kaynak sistemler kullanılmaktadır. IDS/IPS sistemlerinin kullanılması ve yaygınlaştırılması için gerekli araştırmaların daha detaylı yapılması gerekmektedir.

Saldırı Tespit Sistemi Çalışma Yapısı

Dışarıdan gelen davetsiz misafirlerin siber güvenlik sistemi üzerindeki ilerleyişi şekil 14 de yer verilmiştir. İnternet ortamından gelen misafirlerin güvenlik sistemi üzerindeki ilerleyişi adeta bir süzgeç gibi hareket etmesi sağlanır. Bu anlamda saldırganların basamaklar halinde ayrışması yapılır. Böylece zararlı yazılım ve saldırganlar siber güvenlik alt sistemleri üzerinde takılı kalmaları sağlanır. Siber güvenlik sistemi dışarıdan gelen davetsiz misafirleri bir huni misali süzerek elemesi sağlanır. Saldırı tespit ve saldırı

önleme sistemleri önümüze yazılım ya da donanım cihazı olarak da karşımıza çıkabilir.



Şekil 18. Kullanıcıların siber güvenlik sistemi içindeki hareket şeması.

Siber saldırıların tespiti ve önlenmesi IDS/IPS sistemleri yöntemleri kullanıldığı akademik çalışmalarda görülmektedir. Bu çalışmaların giderek artacağı bilinmektedir. IDS/IPS saldırı tespit ve önleme sistemi, bir güvenlik duvarı gibi, saldırılara karşı hazırda bekleyen bir yazılım ve donanım sistemi gibi çalıştığı görülmektedir.

Siber saldırılar açık kaynak sistemler üzerinden daha fazla yapılmakta ve daha fazla zarar vermekte ve itibarı zedelemektedir. Bu kapsamda anti saldırı sistemi geliştirmek için yine açık kaynak sistemlerin daha çok etkili olduğu yapılan akademik çalışmalar incelendiğinde açık kaynak sistemlerin yoğun olduğu görülmektedir. Siber saldırı tespit ve önleme sistemleri olan IDS ve IPS sistemleri yer almaktadır (Çalışkan, 2014: 8-9).

Sistemler üzerinde yer alan güvenlik duvarı siber saldırılara karşı tam olarak koruyamamaktadır. Network sistemleri üzerinde güvenlik duvarı yanında ek olarak saldırı tespit sistemi STS kullanılması gerektiği siber uzayda saldırılara karşı koyma ve sistemleri savunması daha etkili olduğu görülmektedir.

IDS ve IPS sistemleri network üzerinde çalışan sistemler üzerine ayrı yada aynı network içinde yer alabilir. Network çalışma

yapısı baz alınarak network üzerinde konumlandırılabilir. Ayrıca loglama sistemi üzerine güvenlik duvarı özelliğinin kullanılmasıyla sistemler siber uzayda güvenlik seviyesi artması sağlanır.

IPS ve IDS sistemleri network cihazları içinde kullanılması siber saldırıların tespit edilmesi ve önlenmesi konusunda daha etkili çalışan güvenlik sistemleridir. IDS ve IPS sistemleri, network sistemleri üzerinde Şekil 18 de aynı yada farklı network üzerinde konumlandırıldıkları görülmektedir. Şekil 19 da ise IPS ve IDS aynı network üzerinde yer almaktadir.

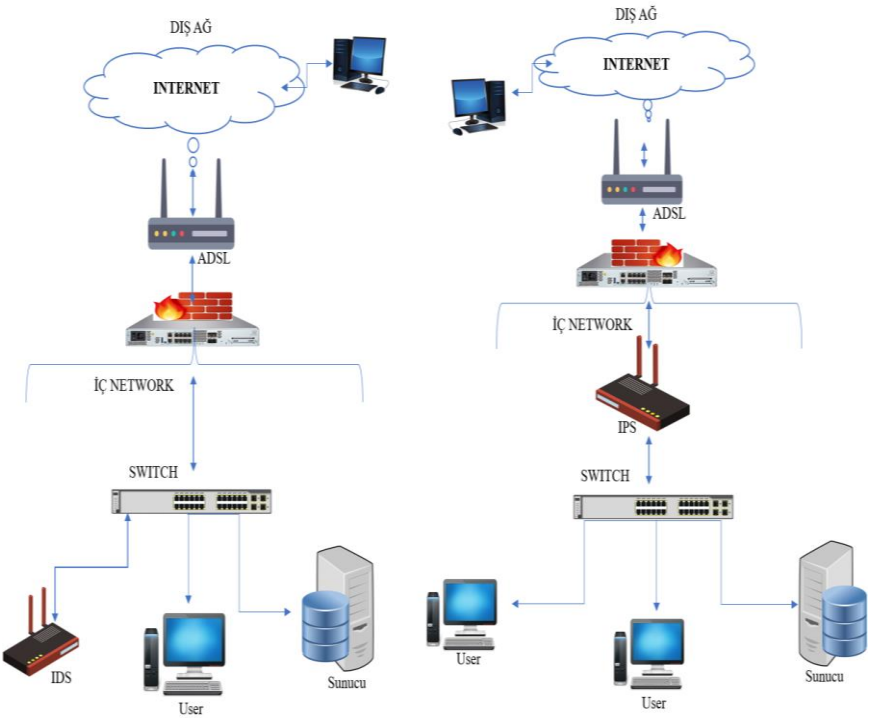
Saldırı tespit sistemi network üzerinde oluşabilecek anormal davranışları, paket hareketlerini tespit edebilecek kabiliyettedir. Saldırı önleme sistemi ise STS davranışlarını inceleyerek gereken IP/MAC adreslerine çeşitli kurallar uygulayıp saldırıları önlemeye çalışmaktadır. Bu bağlamda siber saldırı tespit ve önlenmesinde IPS ve IDS sistemlerinin güvenlik duvarıyla aynı zamanlı kullanılması gerekmektedir.

IDS ve IPS sistemi çalışma yapısı olarak aralarında çeşitli farklar bulunmaktadır. Bu farklar incelendiğinde önemli konu başlıkları ortaya çıkmaktadır. Bu konu başlıkları Tablo 1 de yer almaktadır.

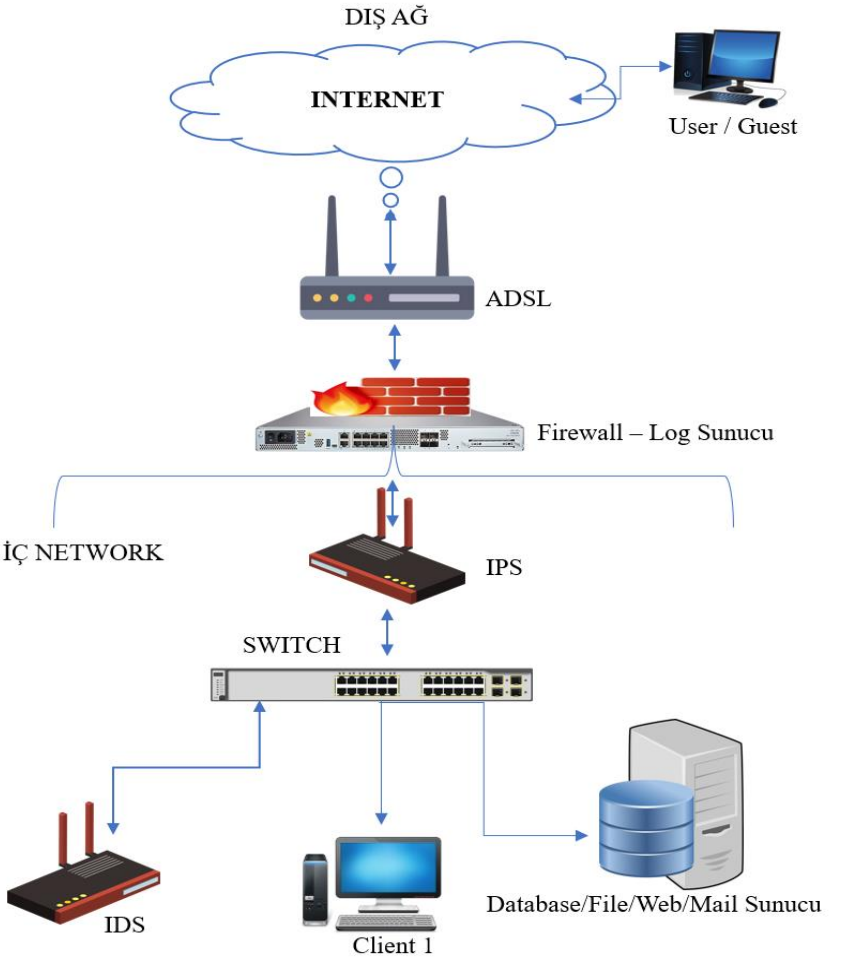
IDS ve IPS sistemleri güvenlik duvarı sonrasında aktif olarak gerçek zamanlı çalışmaktadır. Güvenlik duvarı sistemler üzerinde iç ve dış network trafiğini denetleyebilir. IDS ve IPS sistemleri iç network ortamında çalışmaktadır. Bu anlamda güvenlik duvarı arkasında kullanılır.

Saldırı Tespit Sistemi IDS	Saldırı Önleme Sistemi IPS
Saldırı tespit sistemidir.	Saldırı önleme sistemidir.
Güvenlik duvarı arkasında çalışır.	Güvenlik duvarı arkasında çalışır.
Bilgisayar ağlarında şüpheli ve risk faktörü oluşturan trafiği izler.	Bilgisayar ağlarında şüpheli ve risk faktörü oluşturan trafiği oluşturan kullanıcıları engeller.
Ağ trafiği içindeki paketleri engellemez.	Ağ trafiği kullanıcıları engelleyebilir.
Trafiği analiz ve yorum için insan faktörü gerekebilir.	Bilgisayar ağlarında risk faktörü oluşturan kullanıcıları engellemek için insan faktörü gerekmeyebilir.

Tablo 1. IDS ve IPS güvenlik sistemi karşılaştırması.



Şekil 19. Saldırı önleme sistemi a) IDS çalışma sistemi b) IPS çalışma sistemi.



Şekil 20. Saldırı önleme sistemi IDS/IPS.

SONUÇ

Siber uzayda en önemli konular arasında siber güvenlik başı çekmektedir. Bu anlamda global düzende güvenlik üzerine çalışmalar yürütülmekte, araştırmalar yapılmakta ve çeşitli tesfikler ile çalışma sayıları arttırılmaktadır. Tabiyki bu çalışmaların yapılmasında kamu kurumlarının payı yüksektir. Yapılan siber

saldırılar her gecen gün artmakta ve yeni saldırı türleri türemektedir. Bu durum dijital dünyayı tehdit etmekte ve her geçen gün bu tehtidler artmaktadır.Sistemler üzerindeki zafiyetlerin belirlenmesi, sistemlerin güvenliği açısından biraz zaman almaktadır. Her geçen gün yeni teknolojilerin sistemlere eklenmesi cihazların yazılım alt yapısı, haberleşme ve protokol açısından siber güvenliği olumsuz etkilemektedir. Bu nedenle log kayıtlarının tüm network içinde barınan yazılım ve donanım cihazlarından alınması gerektiği söylenebilir. Loglama teknolojileri aynı zamanda network sistemlerinin bir düzene girmesini ve işlemlerrin planlı programlı yapılmasını sağlar. Log kayıtları üzerinden siber saldırıların anlık engellenmesi mümkün değildir. Fakat log kayıtları üzerinden yola çıkarak siber saldırıların tespit edilmesi ve engellenmesi mümkündür.

Siber saldırıların büyük çoğunluğu kullanıcı hatalarından kaynaklanmaktadır. Kullanıcıların bilerek veya bilmeyerek sistemler üzerinde gerçekleştirdiği hatalı işlemler siber saldırıların önünü açmaktadır. Bu anlamda kullanıcıların log kayıtları sistem uzmanlarına ve siber güvenlik uzmanlarına, kullanıcı hatalarından oluşan zafiyetlerin tespit edilmesi ve bu zafiyetlerin kapatılmasına yardımcı olacaktır. Bu anlamda sistemler üzerinde işlem yapan kullanıcıların yoğun bir şekilde log kayıtlarının tutulması gerekmektedir. Tutulan bu log kayıtları düzenli bir şekilde incelenmesi sonucu kullanıcıdan kaynaklı hataların tespit edilmesi ve giderilmesi sağlanır. Bu bağlamada log kayıtları sistemler üzerinde sistemlerin normalize edilmesinde, hataların giderilmesi ve sistemler üzerinde oluşan zafiyetlerin önlenmesi önemli yere sahip olduğu görülmektedir. Buna bağlı olarak sistemlere yapılan saldırıların önüne geçilmesi sağlanır. Bu anlamda network sistemleri daha güvenli hale getirilmiş olunur.

Log kayıtları üzerinden siber güvenlik sistemlerinin tasarlanmasına yardımcı olmaktadır. Log kayıtları siber saldırıları anlık olarak tespit edemez. Ancak siber saldırıların tespit edilmesine yardımcı olur. Network sistemlerinin diğer sistemler ile çalışma durumunun da incelenmesi hakkında bilgi verir. Log kayıtları

uzmanlar tarafından incelendiğinde network sistemlerinin dizaynından iyileştirilmesine kullanıcı ve sistem hatalarından siber güvenliğine kadar birçok konuda önemli eklentilerin yapılmasını sağlamaktadır. Orta ve uzun vadede siber saldırıların tespit edilmesinde etkili olduğu yapılan çalışmalarda görülmektedir. Kanuni düzenlemeler log kayıtlarının tutulmasından yanadır. Bu anlamda tutulan log kayıtları saldırılara karşı caydırıcı etkisi olmaktadır.

Sistemlere yapılan siber saldırıların çoğunu loglama teknolojileri ile ortadan kalmasına sağlayabildiği görülmektedir. Bu saldırı türleri, Kötü Amaçlı Yazılım (Malware), Phishing (oltalama), Brute Force (Kaba kuvvet), DDoS (Dağıtılmış Ağ Saldırıları) olarak literatürde yerini almıştır. Siber saldırıların tespit edilmesinde açık kaynak kodlu bir uygulama geliştirilme çalışması yapılmıştır. Bu anlamda açık kaynak destekli IPS ve IDS sistemleri kullanılmıştır. Kullanılan IPS ve IDS sistemi açık kaynak kod yapısı ile geliştirildiği için açık kaynak kod yapıları ile geliştirilmeye ve kullanılmaya elverişlidir. Bu bağlamda IPS ve IDS sistemi içinde açık kaynak desteği olan Python dili kullanılmıştır. Ayrıca makine öğrenmesi yöntemleri kullanılarak daha etkili ve hızlı saldırı tespit etme ve önleme sağlayacağı düşünülmektedir. Yapılan çalışma siber güvenlik sistemleri anlamında siber saldırı tespit sistemi geliştirilmesine yardımcı olacağı düşünülmektedir. Siber saldırı tespit sistemi geliştiricilere bir kaynak olabilir. Açık kaynak sistemlerin güvenlik tespiti ve önlenmesi konusunda daha etkilidir. Bu anlamda yapılan birkaç çalışma bulunmaktadır. Bu çalışmanın araştırmacılara kaynak olacağı düşünülmektedir.

Bilgi

Bu bildiri çalışması, *"Siber Saldırıların Loglar Üzerinden Tespit Edilmesi ve Önlenmesi"*, yüksek lisans tez konusundan oluşturulmuştur.

KAYNAKÇA

Ali, A. A. (2021). Audit Logs Management and Security-A Survey. *Kuwait Journal of Science*, s. 48(3).

Baykara, M. R. (2016). Web Sunucu Erişim Kütüklerinden Web Ataklarının Tespitine Yönelik Web Tabanlı Log Analiz Platformu. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 28(2)*, s. 291-302.

Bayraktaroğlu, E. (2009). Bilgi Sistemlerinde Log Yönetimi ve Logların Değerlendirilmesi. *FBE, Bahçeşehir Üniversitesi, İstanbul*.

Chan, C. K. (2017). Development of a Platform to Explore Network Intrusion Detection System (NIDS) for Cybersecurity. *Journal of Computer and Communications*, s. 6(1),1-11.

Çahmutoğlu, E. (2020). Hibrit Savaşın Bir Boyutu Olarak Siber Saldırılar ve Türkiye'nin Durumu. *Yüksek Lisans Tezi, Atatürk Stratejik Araştırmalar Enstitüsü Milli Saunma Üniversitesi İstanbul*.

Çankuş, R. (2022). Veriyükü Üzerinden Sql Enjeksiyon Zafiyetlerin Belirlenmesi. *Yüksek Lisans Tezi, FBE, Düzce Üniversitesi, Düzce*.

Çınar, I. &. (2016). Web Madenciliği Yöntemleri ile Web Loglarının İstatistiksel Analizi ve Saldırı Tespiti. *Bilişim Teknolojileri Dergisi 9(2),* , s. 125-135.

Dijital içeriklerin korunması. (Erişim tarihi: 10.09.2023). https://kamusm.bilgem.tubitak.gov.tr/urunler/zaman_damgasi/ adresinden alındı

Fidancı, Ö. Ş. (2022). Kurumlar İçin Bilgi Güvenliği Yönetim Sisteminin Oluşturulması. Yüksek Lisans Tezi. *FBE, KTO Karatay Üniversitesi, Konya.*

Gül, E. (2019). Log Yönetimi ile Siber Güvenlik Araçlarının Geliştirilmesi. *Yüksek Lisans Tezi, FBE Gazi Üniversitesi, Ankara*.

Hatipoğlu, C. v. (2021). Türkiye'de Siber Saldırı ve Tespit Yöntemleri: Bir Literatür Taraması. . *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, s. 8(1), 430-445.

Kamal, A. H. (2022, December). AA Log Necropsy: A Web-Based Log Analysis Tool. *IEEE 10th Conference on Systems, Process and Control (ICSPC). IEEE.*, (s. pp. 176-179).

kanunu, 5. S. (Erişim Tarihi: 10.10.2023). *Dumlupınar Üniversitesi*. https://birimler.dpu.edu.tr/app/views/panel/ckfinder/userfiles/2/files/mevzuatlar/5651_Say_l__Kanun.pdf adresinden alındı

Kara, M. (2013). Siber Saldırılar Siber Savaşlar, Yüksek Lisans Tezi,. *SBE, İstanbul Bilgi Üniversitesi, İstanbul.*

Koca, H. (2022). Türkiye'de Siber Güvenlik Uygulamaları, Yüksek Lisans Tezi,. *SBE, Hatay Mustafa Kemal Üniversitesi, Hatay.*

Korkmaz, G. (2023). Bir Yönetim Sistemi Olan ISO 27001 Bilgi Güvenliği Sistemi Konulu Yayınların WOS Veri Tabanına Dayalı Bibliyometrik Analizi. *Yüksek Lisans Tezi, LEE, Düzce Üniversitesi, Düzce.*

Landauer, M. S. (2019). December). A framework for cyber threat intelligence extraction from raw log data. *International Conference on Big DataIEEE*, s. 3200-3209.

Loglama cihazları. (2016). https://coslat.com/ adresinden alındı

Loglama cihazları ve özellikleri. (Erişim tarihi: 10.09.2023). https://coslat.com/5651-log-server-cozumleri adresinden alındı

Loglama sistemleri arayüzü. (Erişim tarihi: 10.09.2023). http://172.16.47.254:8000/index.php?zone=kablosuzmisafir&lang=tr adresinden alındı

Manfred, V. (2019). Human-as-a-security-sensor for. *Vielberth et al. Cybersecurity* , s. 2-15.

Max Landauer A. (2020). System log clustering approaches for cyber security applications: A. *Computers & Security*.

Microsoft Güvenlik Raporu. (Erişim tarihi: 10.09.2023). https://www.microsoft.com/en-us/security/business/microsoft-digital-defense-report 2021#areaheading-oc4f81 adresinden alındı

OWASP Raporu. (Erişim tarihi: 10.09.2023). https://owasp.org/www-project-top-ten/ adresinden alındı

Özargın, S. (2023). IoT Teknolojilerinin Eğitim Alanında Kullanılması: Akıllı Okul Örneği. *Yüksek Lisans Tezi Sosyal Bilimler Enstitüsü Bandırma Onyedi Eylül Üniversitesi Balıkesir*.

Özseven, T. &. (2021). LOG Analizi: Erişim Kayıt Dosyaları Analiz Yazılımı ve GOP Üniversitesi Uygulaması. *Bilişim Teknolojileri Dergisi, ,* s. 4 (2).

Raut, U. K. (2018). Log Based Intrusion Detection System. IOSR Journal of Computer Engineering, s. 15-22.

Sandilaç, N. (2022). Siber Suç, Siber Terör ve Siber Savaş Üçgeninde Siber Dünya. *Bilişim Hukuku Dergisi*, s. 4(1), 81-140.

Şentürk, M. Y. (2019). Güncel Siber Saldırı Yöntemleri, Sızma Testi Araçları ve Temsili Bir Kurumsal Ağ Üzerinde Uygulanması, Yüksek Lisans Tezi, . *FBE, Türk Hava Kurumu Üniversitesi, Ankara.*

Zaman damgası kullanımı. (Erişim tarihi: 10.09.2023). https://kamusm.bilgem.tubitak.gov.tr/urunler/zaman_damgasi/ adresinden alındı

Zitta, T. N. (2017). The Security of RFID Readers With IDS/IPS Solution Using Raspberry Pi. *18th International Carpathian Control Conference (ICCC) 3 IEEE.*, s. 16-320.

BÖLÜM VIII

Türkçe E-Maillerin Duygu Analizi ve Makine Öğrenmesi Yöntemleri ile Morfolojik Analizi

Yunus Emre PALAVAR[1]
Ahmet ALBAYRAK[2]

Giriş

E-mail ya da diğer değişle e-posta, internet üzerinden gönderilen dijital mektuplardır. E-posta, geleneksel postaya nispeten daha ucuz, daha pratik ve daha hızlı olduğundan günlük hayatımızda geleneksel postanın yerini almaktadır. E-posta üzerinden her türlü özel ve resmi yazışmalar yapılmaktadır. Günümüzde bu yazışmaların yanında istenmeyen e-maillere maruz kalınmaktadır. Bu istenmeyen e-mailler ise spam e-posta olarak adlandırılmaktadır (Eryılmaz ve Kılıç, 2020:978).

[1] Yüksek Lisans Öğrencisi, Düzce Üniversitesi, Siber Güvenlik ABD, Orcıd: 0000-0001-8737-0722
[2] Dr. Öğretim Üyesi, Düzce Üniversitesi, Bilgisayar Mühendisliği Bölümü, Orcıd: 0000-0002-2166-1102

2023 yılı itibariyle 4.2 milyar e-posta kullanıcısı varken, günlük olarak alınan ve gönderilen e-posta sayısı ise 333 milyar olmuştur. Bu gönderilen e-postaları %56.5 spam olarak gruplanmaktadır (Prodanoff, 2023). Bu spam e-postaların kullanıcılar üzerinde olumsuz etkileri mevcuttur. Bunlardan bazıları aldatmak, dolandırmak, sanal zorbalık ya da sanal hırsızlık vb. olarak sıralanabilir (Kumari ve Nagaraju, 2023:1; Udogwu, 2021:4). Kullanıcıların bu durumlardan korunması amacıyla e-posta sağlayıcıları bu spam maillerin engellemesine yönelik çalışmalar gerçekleştirmektedir (Eryılmaz ve Kılıç, 2020:978).

Literatürde yer alan çalışmalarda yabancı kaynaklı e-postalar üzerinde yapılan çalışmalar yaygınken Türkçe kaynaklı e-postalar üzerinde yapılan çalışmalar daha kısıtlıdır (Güven, 2023:2). Güven'in yapmış olduğu çalışmada Türkçe e-postalarda spam içeren e-postaların tespiti için Rastgele Orman, Lojistik Regresyon, Naive Bayes, Yapay Sinir Ağları makine öğrenme yöntemleri ve BERT, ELECTRA, ALBERT, DistilBERT dil modelleri analiz edilmiştir (Güven, 2023:1). Makine öğrenme yöntemlerinden yapay sinir ağları %90.15 doğrulu değeri elde ederken, en başarılı dil modelleri %94.08 doğruluk değeri ile BERT ve ELECTRA olmuştur (Güven, 2023:1). Literatürde yer alan başka bir çalışmada ise yapay bağışıklık algoritmaları ile spam e-postaların tespit edilmesi üzerine çalışılmıştır. AIRS1 (Artificial Immune Recognition System 1), AIRS2 (Artificial Immune Recognition System 2), AIRS2PARALLEL (Parallel Artificial Immune Recognition System 2), CLONALG (Clonal Selection Algorithm) ve CSCA (Crow Search Algorithm) algoritmaları bu bağlamda incelenmiştir. Bu algoritmalar arasında CSCA, %86 sınıflandırma başarısı ile en iyi sınıflandırma performansı göstermiştir (Şimşek ve Aydemir, 2022:1). Karim ve diğerlerinin yapmış oldukları çalışmada spam e-posta tespitinde Yapay Zeka ve Makine Öğrenmesi yöntemlerine odaklanmış bir literatür araştırmasını açıklanmaktadır (Karim vd., 2019:168261-168295). Karim ve diğerlerinin yapmış olduğu çalışmanın sonucunda SVM ve Naive Bayes algoritmalarının yüksek talep gördüğü ve tek algoritmalı anti-spam uygulamalarının yaygın

olduğu gözlemlenmiştir (Karim vd., 2019:168290). Thanarattananakin ve diğerlerinin yapmış oldukları çalışmada Duygu Analizi'ni kullanarak spam e-mail tespiti önermişlerdir. Thanarattananakin ve diğerlerinin yapmış oldukları çalışmada 5,572 mesajdan oluşan veri setini Bag of words, Hashing ve Long short-term memory algoritması (LSTM) ile test etmişlerdir. LSTM ile test edilen veri setinden %98 doğruluk elde edilmiştir (Thanarattananakin vd., 2022:1).

Bu çalışma kapsamında normal ve spam maillerde oluşan veri setleri üç farklı adımda incelenmiştir. İlk adımda klasik veri önişleme işlemleri yapılmıştır. İkinci adımda ilk adıma ek olarak verilerden Türkçe kökenli olmayan kelimeler, bağlaçlar ve dört kelimeden kısa cümleler çıkarılmıştır. Üçüncü adımda ise ilk adımda elde edilen veri seti ile ikinci adımda elde edilen veri setinin kesişim kümesinden yeni bir veri seti elde edilmiştir. Üç adım sonucunda elde edilen üç veri seti K-means, Isolation Forest, duygu analizi, Naive Bayes, Random Forest, Logistic Regression ve Support Vector Machine yöntemleri ile analiz edilmiştir.

Bu çalışma da literatürde yer alan Türkçe veri seti kullanılan çalışmalardan farklı olarak Türkçe veri setlerinin içerisindeki yer alan Türkçe kökenli olmayan kelimelerde analiz edilmiştir. K-means, Isolation Forest, duygu analizi, Naive Bayes, Random Forest, Logistic Regression ve Support Vector Machine yöntemleri ile hem geleneksel veri ön işleme adımları uygulanan veriler değerlendirilmiş hem de geleneksel yöntemlere ek olarak Türkçe kökenli olmayan kelimelerin çıkarılması ile elde edilen veri setleri de değerlendirilmiştir.

Yöntem

Çalışma kapsamında Türkçe maillerden oluşan iki veri seti kullanılmıştır. Bu veri setlerinden ilki kaggle yer alan Türkçe spam veri setidir. Bu veri seti 330 spam e-mail ve 496 ham e-mailden oluşmak üzere toplamda 823 Türkçe e-mailden oluşmaktadır. Çalışma kapsamında kullanılan ikinci veri seti kaggle bulunan

Türkçe mail veri setidir. Bu veri seti 502 ham e-mail ve 515 spam e-mail olmak üzere toplamda 1017 Türkçe e-mailden oluşmaktadır. Bu iki veri tek veri seti birleştirilerek 2 adet veri seti oluşturulmuştur. Bu veri setlerinden ilki ham maillerden, ikincisi spam maillerden oluşmaktadır.

Elde edilen veriler makine öğrenmesi yöntemleri ile çalışılabilmesi için veri önişleme adımlarından geçirilmiştir. Böylelikle makine öğrenmesi tekniklerinin ve duygu analizinin uygulanmasına veriler hazırlanmıştır. Çalışma kapsamında sırasıyla veri önişleme, görselleştirme, kümeleme, duygu analizi ve sınıflandırma işlemleri yapılmıştır.

Veri Önişleme.

Makine öğrenmesi tekniklerinin başarısı genellikle üzerinde çalıştıkları verinin kalitesine bağlıdır. Eğer veri kümesi yetersiz, gereksiz veya ilgisiz verilerden oluşuyorsa makine öğrenmesi tekniklerinin başarı oranı düşmektedir (Huang vd., 2015:108-109; Kotsiantis vd., 2006:111-112). Veri küme içerisinden olabildiğince ilgisiz, tekrarlanan gürültüye sebep veren verilerin temizlenmesi gereklidir. Bu işlemlerde veri önişleme başlığı altında gerçekleştirilmektedir. Veri önişleme aşamasında aşağıdaki işlemler yapılmıştır.

a) Verilerin küçük harflere dönüştürülmesi.
b) Web sitesi adresleri “website” ve e-mail adresleri “email” olarak yeniden adlandırıldı.
c) Noktalama işaretleri ve sayısal ifadelerin kaldırılması.

Şekil 1’de spam maillerden oluşan veri setine veri önişleme adımları uygulanmadan ve uygulandıktan sonraki veri setinin durumu yer almaktadır.

	Data
0	Sayın Yetkili ;\n Gelişen ve değişen günümüz ...
1	Sayın Yetkili,\n \n28 Kasım 2010 tarihli KPSS ...
2	Sayın Yetkili ;\n Gelişen ve değişen günümüz ...
3	T.C.\nURLA KAYMAKAMLIĞI\nURLA HAKAN ÇEKEN ANAD...
4	Sayın Yetkili ;Hızla büyüyen ve gelişmekte ola...
..	...
510	Yılbaşı Özel İndirimleri Başladı! 16,95 TL'den...
511	Kaçırılmayacak Şahane İndirimler Başladı! Pier...
512	Kadınlar Günü sürprizin hazır mı? Çiçek& Bonny...
513	Aradığın Sürprize En Uygun Hediyeler ÇiçekSepe...
514	Canlı yayında alışveriş etkinliği için son saa...

[844 rows x 1 columns]

	Data
0	sayın yetkili \n gelişen ve değişen günümüz \...
1	sayın yetkili\n \n kasım tarihli kpss sınavı ...
2	sayın yetkili \n gelişen ve değişen günümüz \...
3	tc\nurla kaymakamliği\nurla hakan çeken anadol...
4	sayın yetkili hızla büyüyen ve gelişmekte olan...
..	...
839	yılbaşı özel indirimleri başladı tlden başlay...
840	kaçırılmayacak şahane indirimler başladı pierr...
841	kadınlar günü sürprizin hazır mı çiçek bonnyfo...
842	aradığın sürprize en uygun hediyeler çiçeksepe...
843	canlı yayında alışveriş etkinliği için son saa...

[844 rows x 1 columns]

Şekil 3. Spam mail veri setine veri önişleme uygulanmadan önce ve sonrası

Veri Görselleştirme.

Veri önişleme adımları uygulanan veri setleri kelime bulutu yöntemiyle görselleştirilmiştir. Verilerin görselleştirilmesi ile sonuçlar daha kolay okunabilir ve daha anlaşılabilir hale gelmiştir.

Kelime bulutu ya da etiket bulutu, metin verilerini görselleştirmenin anlaşılır bir yoludur. Bir kelime bulutuna bakan gözlemci, metnin yoğunluğu, sıklığı ve dağılımı konusunda kolaylıkla çıkarım yapabilir (DePaola ve Wilkinson, 2014:38; Atenstaedt, 2012:148).

Şekil 2'de veri ön işleme adımlarına ek olarak, bağlaçların ve Türkçe kökenli olmayan kelimelerin veri setinden temizlenmesi ile oluşturulan kelime bulutu yer almaktadır.

Şekil 2. Bağlaçların ve Türkçe kökenli olmayan kelimelerin veri setinden temizlenmesi ile oluşturulan kelime bulutu

Verilerin Kümelenmesi.

Kelime bulutu ile görselleştirilen spam ve ham veri setleri hakkında kelime bulutu ve bar grafiği sonuçlarıyla bir çıkarım yapılabilir ancak bu çıkarımlar tek başına yeterli değildir. Spam maillerin ve ham maillerin kümelenmesi ile yönelimleri arasında daha tutarlı çıkarımlar yapmak mümkündür. Verilerin kümelenmesinde K-means ve Isolation Forest kümeleme yöntemleri kullanılmıştır.

K-means, 1967 yılında MacQueen tarafından önerilmiştir (Macqueen, 1967:281-297). Arama verimliliğini artırmak için sezgisel bilgiyi kullanır ve aramanın daha objektif olmasını sağlar. Temel fikri, küme sayısı K'nin atanmasıdır. İlk olarak, başlangıçta bir bölüm oluşturularak, ardından küme merkezini sürekli olarak hareket ettirerek bölümü iyileştirmek için iterasyon yöntemi kullanılır (Wang vd., 2006:188-190). Aslında, bu arama yöntemi kullanılarak en iyi çözüm gerekli değildir. Ancak sezgisel bilgiyi kullanarak, her kümenin merkezini belirtmek için ortalama değeri kullanarak hesaplama karmaşıklığını azaltıp arama verimliliğini artırır. Bu, belirli bir verimlilik kısıtlaması altında büyük verilerin en iyi çözümünü elde etmeyi mümkün kılar (Zang vd., 2016:169-172). K-means algoritmasının formülsel ifadesi için formül 1'de verilen denklemlerden yararlanılır (Takaoğlu M. Ve Takaoğlu F., 2019:304).

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{ci} (||x_i - v_j||)^2 \quad (1)$$

Formül 1 için, '$||x_i - v_j||$', x ve y arasındaki Öklid mesafesi ' c_i ', i^{th} kümesindeki ver, noktalarının sayısı, c ise küme merkezlerinin sayısıdır.

K-means kümelemenin algoritmik adımları:

$\{x_1, x_2, x_3, \dots, x_n\}$ kümesi veri noktalarının, $V = \{v_1, v_2, v_3, \dots, v_c\}$ ise merkez noktalarının kümesi olsun.

Rastgele 'c' küme merkezlerini seç.

Her veri ile küme merkezlerinin arasındaki mesafeyi hesapla.

Küme merkeziyle arasındaki mesafe, diğer küme merkezleriyle olan mesafeden daha az olan veriyi, yakın olan o küme merkezine ata.

Yeni küme merkezini aşağıdaki denklemle yeniden hesapla:

$$v_i = (1/c_i) \sum_{j=1}^{ci} x_i \quad (2)$$

Her veri noktasıyla, yeni küme merkezleri arasındaki mesafeyi yeniden hesapla.

Eğer hiçbir veri noktası atanmadıysa dur, diğer durumda üçüncü adımdan itibaren tekrar et.

Isolation forest 2008 yılında Fei Tony Liu ve Zhi-Hua Zhou tarafından geliştirilmiştir. Isolation forest anomalilerin tespiti için kullanılmaktadır (Liu vd., 2008:415). Formül 3'te Isolation Forest bir x örneğinin anomali puanı hesaplanmasında kullanılan denklem verilmiştir (Hariri vd., 2019:1481).

$$s(x, n) = 2^{\frac{-E(h(x))}{c(n)}} \quad (3)$$

Formül 3 için x örneğimiz, n harici düğümlerin sayısını, h(x) x veri noktasının yol uzunluğunu, E(h(x)) yol uzunluğunun beklenen veya ortalama değeri ve c(n) ise bir ikili arama ağacında başarısız aramanın ortalama yol uzunluğunu ifade eder ve formül 4'teki gibi ifade edilir (Hariri vd., 2019:1481; Liu vd., 2008:415).

$$c(n) = 2H(n-1) - (2(n-1)/n) \quad (4)$$

İki farklı metot kullanmaktaki amaç farklı kümeleme yöntemlerinin morfoloji belirlemedeki etkilerini görmektir. Bu yöntemler sırasıyla veri önişleme adımları yapılmış veri seti üzerinde, veri önişleme adımlarına ek olarak veri setinden Türkçe kökenli olmayan kelimelerin ve bağlaçların çıkarılması ile oluşturulan veri seti üzerinde ve son olarak ilk adımda kullanılan ver seti ile ikinci adımda kullanılan veri setinin kesişim kümesi olan

veriler üzerinde test edilmiştir. Bu işlemlerden geçirilen veriler grup 1, grup 2 ve grup 3 olarak adlandırılmıştır.

Duygu Analizi.

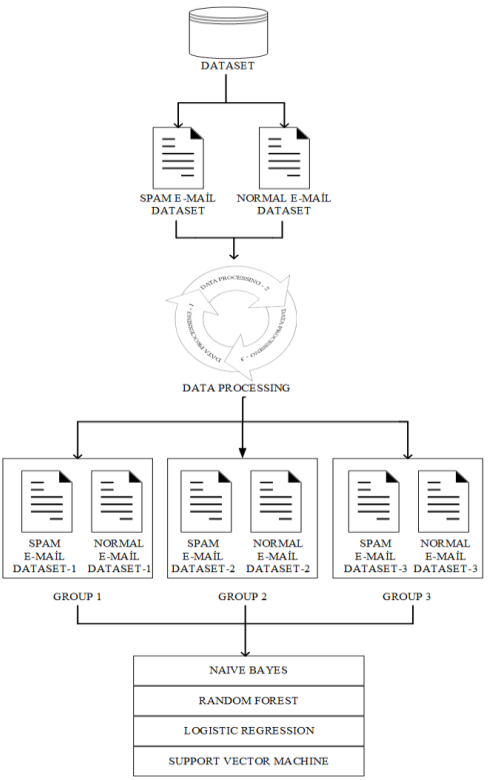
Duygu analizi, öznel tercihleri ve duygusal durumları sistematik olarak tanımlamak, çıkartmak, ölçmek ve incelemek için doğal dil işleme, hesaplamalı dilbilim ve biyometri tekniklerini kullanır (Medhat vd., 2014:1093; Liu,2012:7; Agarwal,2011:30). Genel olarak bir metnin, yazarının bir konuya, bir bağlama ya da bir belgeye kutupsal yaklaşımını ve tutumunu belirlemeyi amaçlar (Saif, 2012:508; Bostancı ve Albayrak, 2021:54). Spam ve normal maillerin tespitinde duygu tonunun belirlenmesi oldukça önemlidir. Spam mailler dolandırıcılık, kişisel bilgilerin çalınması vb. amaçlı kullanılabilmektedir. (Peng ve Zhong, 2014:2065-2066; Ezpeleta vd., 2020:83-84). Dolandırıcılık gibi ciddi suç amacı barındıran spam mailler içerikleri yönünden muhatabını korkutmak, caydırmak amaçlı olduklarından genel itibariyle olumsuz duygu tonuna sahiptirler (Ezpeleta vd., 2020:83-84). Ama aynı zamanda olumlu duygu durumuna sahip mailler aracılığıyla bu amaca hizmet eden spam mailler mevcuttur (Karim vd., 2019:168261-168262).

Çalışma kapsamında elde edilen veri setleri genel kanıya göre dolandırıcılık gibi suç amacı güden spam mailler, normal maillere göre duygu tonu doğrusal mı ya da doğrusal olmayan bir eğilim mi sergilemektedir incelenmiştir.

Verilerin Sınıflandırılması.

Çalışma kapsamında elde edilen veri setlerinin sınıflandırılmasında Naive Bayes, SVM, Random Forest, Logistic Regression yöntemleri kullanılmıştır. Bu kullanılan algoritmalarının performansları karşılaştırılmış ve aralarından en iyi yöntem saptanmıştır. Çalışma kapsamın veriler ham ve spam veri setleri olarak ikiye ayrılmıştır. Bu ayrılan veri setleri üç işlem adımına tabi tutulmuştur. Bu tabi tutulma işlemi sonrası, üç ham ve üç spam mail veri seti olmak üzere toplamda altı veri seti elde edilmiştir. İlk işlem

adımında veriler üzerinde temel veri önişleme adımları uygulanmıştır. İkinci işlem adımında ilk işlem adımına ek olarak verilerden Türkçe kökenli olmayan kelimeler, dört harften kısa kelimeler ve dört kelimeden kısa cümleler çıkarılmıştır. Üçüncü işlem adımında ikinci işlem adımı sonucu veri setinden çıkarılan veri satırları, ilk işlem adımı sonucu oluşturulan veri setinden de çıkarılmıştır. İlk işlem adımına ait veri setleri grup 1'de, ikinci işlem adımına ait veriler grup 2'de, üçüncü işlem adımına ait veriler grup 3'te listelenmiştir. Her veri grubu sırasıyla Naive Bayes, Random Forest, Logistic Regression ve SVM yöntemleri ile sınıflandırılmıştır. Şekil 3'te yapılan bu işlem adımlarına ait blok şema yer almaktadır.



Şekil 3. Veri sınıflandırma işlemi aşamasında yapılan işlemlerin blok şeması

Naive Bayes (NB), sınıflandırma için en iyi bilinen veri madenciliği algoritmalarından biridir (Zhang vd., 2016:138). Naive Bayes belirli bir sınıfa ait tüm niteliklerin birbirinden bağımsız olduğu varsayımına dayanarak yeni bir örneğin belirli bir sınıfa ait olma olasılığını ortaya çıkarır (Langley ve Sage, 1998:399-400). Bu varsayım, eğitim verilerinden çok değişkenli olasılıkları tahmin etme ihtiyacından kaynaklanmaktadır (Chen vd., 2020:1). Uygulamada çoğu nitelik değeri kombinasyonu ya eğitim verilerinde mevcut değildir ya da yeterli sayıda mevcut değildir (Chen vd., 2020:1). Bu nedenle ilgili çok değişkenli olasılıkların doğrudan tahminleri güvenilmez olacaktır (Chen vd., 2020:1). Naive Bayes, koşullu bağımsızlığı varsayarak bu çıkmazı aşmaktadır (Chen vd., 2020:1). Katı bağımsızlık varsayımına rağmen, Naive Bayes birçok gerçek dünya uygulamasında gerçekten yetkin bir sınıflandırıcıdır (Bermejo vd., 2014:140; Chen vd., 2020:1).

Random forest, regresyon ve sınıflandırma problemlerinde kullanılır. Bu algoritmanın amacı, çoklu karar ağaçları oluşturarak sınıflandırma sürecindeki sınıflandırma değerini iyileştirmektir. Random Forest algoritması, birbirinden bağımsız çalışan birçok karar ağacının birleştirilerek aralarından en yüksek puana sahip olanın seçilmesi işlemidir (Biau ve Scornet, 2016:197).

Lojistik regresyon, sınıflandırma problemlerini öğrenmek için geliştirilmiş denetimli bir makine öğrenme algoritmasıdır. Hedef değişken kategorik bir değişken olduğunda sınıflandırma öğrenme sorunları ortaya çıkar. Lojistik regresyonun amacı, yeni bir örneğin hedef kategorilerden birine ait olma olasılığını tahmin etmek için veri kümesi özelliklerinin bir fonksiyonunu bir hedefe eşlemektir (Keerthi vd., 2005:151-152; Zou vd., 2019:135-136).

Destek vektör makinesi (SVM), özellikle sınıflandırma ve regresyon problemlerinde kullanılan güçlü bir makine öğrenme algoritmasıdır. Temel amacı, veri kümesindeki öğeleri farklı sınıflara ayrıştırmak için en uygun hiper düzlemi bulmaktır. Bu hiper düzlemi belirlerken SVM, bir veri noktasının kendi sınıfına olan mesafesini maksimuma çıkarmaya çalışır, böylece sınıflar

arasındaki ayrım maksimuma çıkar. Bu, veri noktalarını daha iyi sınıflandırmak ve özetlemek için etkili bir yol sağlar (Wang, 2005; Hearts vd., 1998:18).

Sonuçlar ve Tartışma

K-means ile Verilerin Kümelenmesi.

K-means ile veriler kümelendirilirken normal ve spam mailler bir veri seti altında birleştirilmiştir. K-means ile etiketlenmemiş verilerin 2 küme altında dağılımı gözlemlenmiştir.

Tablo 1'de grup 1, grup 2 ve grup 3'e ait verilerin spam, normal, küme 0 ve küme 1 dağılımları gösterilmiştir.

Tablo 2. Grup 1, grup 2 ve grup 3'teki verilerin dağılımı

	Normal	Spam	Küme 0	Küme 1
Grup 1	998	844	174	1668
Grup 2	847	776	1601	22
Grup 3	847	776	199	1421

Tablo 1'deki veriler incelendiğinde k-means sonucu verilerin kümelenmesi ekseriyetle grup 1'de küme 1, grup 2'de küme 1 ve grup 3'te küme 2 üzerinde olmuştur. K-means ile kümeleme sonucunda veriler normal ve spam dağılımlarına yakın bir kümelenme gerçekleşmemiştir. Veri setinden bağlaçların, Türkçe kökenli olmayan kelimelerin ve dört kelimeden kısa cümlelerin çıkarılması kümelenme eğilimini değitirmiştir. K-means verilerin spam veya normal kümelenmesinde, gerçek dağılıma nazaran başarısız olmuştur.

Isolation Forest ile Verilerin Kümelenmesi.

Isolation Forest anomali tespiti için kullanılan bir algoritmadır. Normal veriler, tipik veya beklenen davranışları sergileyen verileri ifade eder. Anomali ise beklenmeyen, aykırı davranışta bulunan verileri ifade eder. Çalışma kapsamında bu anomali tespitinden

yararlanarak grup 1, grup 2 ve grup 3'teki verilerin normal maillerin ve spam maillerin dağılımlarının anomali (olası spam) ve normal olarak dağılımları karşılaştırılmıştır. Isolatin Forest algoritmasının contamination değeri 0.1, eşik değeri ise sıfır olarak ayarlanmıştır.

Tablo 2'de grup 1, grup 2 ve grup 3'e ait normal ve spam e-maillerin Isolation Forest ile kümelenmesi sonucu dağılımları yer almaktadır.

Tablo 2. Grup 1, grup 2 ve grup 3 veri setlerine ait anomali dağılımları

	Normal	Spam	Anomali (Olası Spam)	Normal
Grup 1	998	844	184	1657
Grup 2	847	776	163	1460
Grup 3	847	776	163	1460

Tablo 2'deki veriler incelendiğinde anomali ve normal verilerin dağılımları spam ve normal etiketlerinden gözle görülür bir şekilde farklı dağılmıştır. Normal ve spam etiketlerinde dağılımlar daha homojen yapıya sahipken Isolation Forest kümelemesi sonucu anomali yüzdeleri yaklaşık olarak grup 1 için %10, grup 2 için %10.05 ve grup 3 için %10.05 olmuştur. Sonuç olarak veri setinden, Türkçe kökenli olmayan kelimelerin, dört kelimeden kısa cümlelerin ve bağlaçların veri setinden çıkarılması anomali (olası spam) oranını %0.05 artırmıştır.

Isolation Forest algoritması da K-means ile yakın sonuçları üretmiştir. K-means ve Isolation Forest mailleri ayırmada istenilen başarıyı sağlayamamıştır. Burada çıkarılacak başka bir sonuç ise spam maillerin büyük bir çoğunluğu normal mailler aynı yapıya sahiptirler. K-means ve Isolation Forest spam maillerin büyük bir kısmını normal mail olarak algılamıştır.

Duygu Analizi.

Normal ve spam mail tespitinde duygu analizi maillerin duygusal tonunun belirlenmesinde kullanılır. Kişisel bilgilerin çalınmasında veya dolandırıcılık amacıyla kullanılan spam mailler çoğunlukla olumsuz duygu tonuna sahiptirler. Aksi bir durumda söz konuda mevcuttur. Bu gibi sebeplerden spam ve normal maillerde duygu tonunun karşılaştırmalı olarak inceledik.

Çalışma kapsamında elde edilen grup 1, grup 2 ve grup 3'e ait veri setlerine NLTK kütüphanesine ait Sentiment Intensity Analyzer sınıfı ile duygu analizi yapılmıştır. Bu sınıf, metinlerin duygu analizinin yapılmasın olanak sağlar. Bu analiz sonucu veriler pozitif, negatif ve nötr olmak üzere sınıflandırılmıştır. Tablo 3'te verilerin duygu analizi sonucu dağılımı gösterilmektedir. İlk işlem adımına ait veri setleri grup 1'de, ikinci işlem adımına ait veriler grup 2'de, üçüncü işlem adımına ait veriler grup 3'te listelenmiştir.

Tablo 3. Veri setlerine duygu analizi uygulaması sonuçları

	Grup 1		Grup 2		Grup 3	
	Normal	Spam	Normal	Spam	Normal	Spam
Pozitif	147	107	0	0	107	94
Nötr	753	649	843	776	659	599
Negatif	97	88	4	0	81	83

Tablo 3'teki verileri incelediğimizde göze çarpan ilk sonuç veri setlerinden Türkçe olmayan kelimelerin, bağlaçların ve dört kelimeden kısa maillerin çıkarılması, duygu dağılımının nötr ekseninde yoğunlaşmasına sebep olmuştur. Tablo 3'ten çıkartılan bir başka sonuçta normal ve spam verilerin duygu durumlarının paralel bir dağılım içerisinde olmuş olmasıdır. Normal ve spam mailler duygu analizi sonucunda aynı duygu grupları ekseninde toplanmıştır.

Naive Bayes ile Verilerin Sınıflandırılması.

Verilerden elde edilen altı veri seti çalışma kapsamında ilk olarak Naive Bayes ile sınıflandırılmıştır. Naive Bayes ile

sınıflandırma sırasında aynı işlem adımları uygulanan veri setleri birleştirilmiş ve veriler spam ya da normal olarak etiketlenmiştir. Burada ki amaç veriler üzerinde yapılan işlem adımlarının sonuçlar üzerine etkisini gözlemlenmesidir. Naive Bayes sonucu modelin performansını değerlendirmek için accuracy, precision, recall ve f1 score parametreleri kullanılmıştır. Tablo 4'te veri setlerine Naive Bayes uygulanması sonucu accuracy, precision, recall ve f1 score parametrelerinin sonuçları gösterilmiştir.

Tablo 4. Naive Bayes uygulanması sonucu accuracy, precision, recall ve f1 score parametrelerinin sonuçları

	Grup 1	Grup 2	Grup 3
Accuracy	0.92	0.85	0.90
Precision	0.92	0.85	0.91
Recall	0.92	0.85	0.90
F1 Score	0.92	0.85	0.90

Tablo 4'teki veriler incelendiğinde en yüksek performansa sahip model grup 1 olmuştur. En düşük performansa sahip model ise grup 2 olmuştur. Bu değerler göz önüne alındığında veriler içerisinden Türkçe kökenli olmayan bağlaçların ve 4 kelimeden kısa cümlelerin çıkarılması modelin performansını olumsuz etkilemektedir.

Random Forest ile Verilerin Sınıflandırılması.

Random Forest modeli oluşturulurken sırasıyla yapılan işlemler, bağımsız değişkenler ve bağımlı değişkenler belirlenmiştir. Bağımsız değişkenler mail verilerimizdir, bağımlı değişkenlerimiz ise mail etiketleridir. Etiket değerleri sayısal değerlere dönüştürülmüştür. Verilerin %80'i eğitim %20'si test verisi olacak şekilde ayarlanmıştır. Metin verilerini sayısal özelliklere dönüştürmek için TfidfVectorizer kullanılmıştır. TfidfVectorizer, max_features değerini 5000, n_estimators değeri 100 ve random_state değeri 42 olarak ayarlanmıştır. Modelin sonuçlarını değerlendirmek için accuary, precision, recall, f1-score parametrelerinden faydalanılmıştır. Tablo 5'te Random Forest ile

eğitilen modelin accuary, precision, recall ve f1-score parametrelerine göre değerlendirme sonuçları yer almaktadır.

Tablo 5. Random forest ile eğitilen modelin accuracy, precision, recall ve f1 score parametrelerinin sonuçları

	Grup 1	Grup 2	Grup 3
Accuracy	0.89	0.84	0.88
Precision	0.93	0.92	0.89
Recall	0.82	0.68	0.82
F1 Score	0.87	0.78	0.85

Tablo 5'teki ki veriler incelendiğinde grup 1 altında ki veriler diğer gruplara nispeten daha yüksek sonuçlar vermiştir. Türkçe kökenli olmayan kelimelerin, bağlaçların ve 4 kelimeden kısa cümlelerin veri setinden çıkarılması modelin performansını olumsuz etkilemektedir.

Logistic Regression ile Verilerin Sınıflandırılması.

Logistic Regression modeli oluşturulurken sırasıyla yapılan işlemler, metin verileri (X) ve etiketler (y) olarak belirlenmiştir. Etiket değerleri label encoder ile sayısal değerlere dönüştürülmüştür. Verilerin %80'i eğitim %20'si test verisi olacak şekilde ayarlanmıştır. Metin verilerini sayısal özelliklere dönüştürmek TfidfVectorizer kullanılmıştır. TfidfVectorizer, max_features değerini 5000, max_iter değeri 100 ve random_state değeri 42 olarak ayarlanmıştır. Modelin sonuçlarını değerlendirmek için accuary, precision, recall, f1-score parametrelerinden faydalanılmıştır. Tablo 6'da Logistic Regression ile eğitilen modelin accuary, precision, recall ve f1-score parametrelerine göre değerlendirme sonuçları yer almaktadır.

Tablo 6. Logistic regression ile eğitilen modelin accuracy, precision, recall ve f1 score parametrelerinin sonuçları

	Grup 1	Grup 2	Grup 3
Accuracy	0.90	0.88	0.90
Precision	0.93	0.91	0.93
Recall	0.83	0.81	0.85
F1 Score	0.88	0.86	0.88

Tablo 6'da ki veriler incelendiğinde grup 3 altındaki veriler diğer gruplara nispeten daha yüksek sonuçlar vermiştir. 3 grup altındaki veriler incelendiğinde modellerin doğruluk değerleri birbirine oldukça yakındır. Türkçe kökenli olmayan ve 4 kelimeden kısa cümlelerin veri setinden çıkarılması modelin performansını olumsuz etkilemektedir. Ancak grup 2 aşamasında veri setinden çıkarılan veri satırları grup 1'den çıkarılması ile oluşturulan grup 3'te ki modelin performansını artırmaktadır. Dolaylı yoldan Türkçe kökenli olmayan ve 4 kelimeden kısa cümlelerin çıkarılması modelin performansını artırmıştır.

Support Vector Machine ile Verilerin Sınıflandırılması.

SVM modeli oluşturulurken sırasıyla yapılan işlemler, metin verileri (X) ve etiketler (y) olarak belirlenmiştir. Etiket değerleri label encoder ile sayısal değerlere dönüştürülmüştür. Verilerin %80'i eğitim %20'si test verisi olacak şekilde ayarlanmıştır. Metin verilerini sayısal özelliklere dönüştürmek TfidfVectorizer kullanılmıştır. TfidfVectorizer, max_features değerini 5000, max_iter değeri 100 ve random_state değeri 42 olarak ayarlanmıştır. Model çekirdeği olarak linear seçilmiştir. Linear çekirdek veriler doğrusal olarak ayrılabilir olan veri setlerinde kullanılabilir bir çekirdektir. Çalışma kapsamında veriler normal ve spam olarak ayrıştırılabilir durumdadır. Modelin sonuçlarını değerlendirmek için accuary, precision, recall, f1-score parametrelerinden faydalanılmıştır. Tablo 7'de SVM ile eğitilen modelin accuary, precision, recall ve f1-score parametrelerine göre değerlendirme sonuçları yer almaktadır.

Tablo 7. SVM ile eğitilen modelin accuracy, precision, recall ve f1 score parametrelerinin sonuçları

	Grup 1	Grup 2	Grup 3
Accuracy	0.92	0.87	0.90
Precision	0.94	0.87	0.92
Recall	0.87	0.82	0.85
F1 Score	0.90	0.84	0.88

Tablo 7'de ki veriler incelendiğinde grup 1 altındaki veriler diğer gruplara nispeten daha yüksek sonuçlar vermiştir. Türkçe kökenli olmayan kelimelerin, bağlaçların ve 4 kelimeden kısa cümlelerin veri setinden çıkarılması modelin performansını olumsuz etkilemektedir.

Sonuçlar ve Öneriler

Bu çalışmada Türkçe e-maillerden oluşan veri setleri, K-means, Isolation Forest, duygu analizi, Naive Bayes, Random Forest, Logistic Regression ve Support Vector Machine yöntemleri ile test edilmiştir. Test edilen veri setine uygulanan üç işlem adımından sonra üç veri seti elde edilmiştir. Elde edilen ilk veri seti, verilerin küçük harflere dönüştürülmesi, web sitesi adreslerinin "website" ve e-mail adresleri "email" olarak yeniden olarak yeniden adlandırılması, noktalama işaretlerinin ve sayısal ifadelerin kaldırılması ile elde edilmiştir. İkinci veri seti ilk veri setinden bağlaçların, Türkçe kökenli olmayan kelimelerin ve dört kelimeden kısa cümlelerin çıkarılması ile elde edilmiştir. Üçüncü veri seti ilk veri seti ile ikinci veri setinin kesişim kümesidir.

Bu veri setlerine K-means ile iki grup olacak şekilde test edilmiştir. Bu test işlemi sonucu veriler spam ve normal etiketlerine göre iki gruba orantısal dağılmamıştır. K-means beklendiği gibi verileri bölememiştir. Isolation forest ile veri setleri üzerinde anomali tespiti yaptığımızda normal etiketine sahip mailler üç veri seti içinde %10.04 anomali değerine sahipken, spam etiketli mailler ise sırasıyla %9.96, %10.06, %9.93 anomali değerlerine sahiptir. Türkçe kökenli olmayan kelimelerin ve bağlaçların veri setlerinden

çıkarılması spam veriler üzerinde anomaliyi artırmıştır. Duygu analizi sonuçlarında ise veri setlerinden Türkçe olmayan kelimelerin, bağlaçların ve dört kelimeden kısa maillerin çıkarılması, duygu dağılımının nötr ekseninde yoğunlaşmasına sebep olmuştur. Naive Bayes, Random Forest, Logistic Regression ve Support Vector Machine yöntemleri veri setlerinin test edilmesi sonucunda ilk veri setini üzerinde Naive Bayes ve SVM 0.92 doğruluk değeri ile en başarılı sonucu verirken Logistic Regression ile 0.90 ve Random Forest ile 0.89 doğruluk değerleri elde edilmiştir. İkinci veri seti üzerinde ise Logistic Regression 0.88 doğruluk değeri ile en başarılı yöntem olurken, SVM 0.87, Naive Bayes 0.85 ve Random Forest 0.84 doğruluk değerleri elde edilmiştir. Üçüncü ve son veri setinde ise Naive Bayes, SVM ve Logistic Regression 0.90 doğruluk değerleri ile en başarılı sonucu vermiş Random Forest ile 0.88 doğruluk değeri elde edilmiştir.

Veri setinden bağlaçların, Türkçe kökenli olmayan kelimelerin ve dört kelimeden kısa cümlelerin çıkarılması Naive Bayes, Random Forest, Logistic Regression ve Support Vector Machine modellerinin doğruluk değerini düşürmektedir. Sadece mail içeriklerinin değerlendirilmesinden ziyade konu başlıklarının değerlendirilmesi spam ve normal maillerin eğilimlerinin daha geniş bir çerçeveden değerlendirilmesine olanak sağlayabilir.

Kaynakça

Prodanoff, J. T. (2023). *21 Must-read stats about how many emails are sent per day*. Erişim Tarihi: 20.09.2023, https://webtribunal.net/blog/how-many-emails-are-sent-per-day/.

Güven, Z. A. (2023). Türkçe e-postalarda spam tespiti için makine öğrenme yöntemlerinin ve dil modellerinin analizi. *Avrupa Bilim ve Teknoloji Dergisi, (47)*, 1-6.

Eryılmaz, E. E., & Kılıç, E. (2020). İstenmeyen e-postaların tespiti için kullanılan yöntemlerin incelenmesi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 11(3),* 977-987.

Özdemir, C., Ataş, M., & Özer, A. B. (2013, April). Classification of Turkish spam e-mails with artificial immune system. *In 2013 21st Signal Processing and Communications Applications Conference (SIU)*, 1-4.

Kumari, B. A., & Nagaraju, C. (2023). Robust machine learning technique for detection and classification of spam mails. *EasyChair*, 1-8

Thanarattananakin, S., Bulao, S., Visitsilp, B., & Maliyaem, M. (2022, January). Spam detection using word embedding-based LSTM. *In 2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 227-231

Udogwu, C. T. (2021). Ensemble Classification Method for Email Spam Prediction. *Doctoral dissertation, Dublin, National College of Ireland.*

DePaolo C.A. and Wilkinson K. (2014 April). Get your head into the clouds: using word clouds for analyzing qualitative assessment data. *TechTrends, vol. 58, no. 3*, 38–44.

Atenstaedt R. (2012, Mar.). Word cloud analysis of the BJGP. *The British Journal of General Practice : the Journal of the Royal College of General Practitioners, vol. 62, no. 596,* 148–148.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal, 5(4),* 1093-1113.

Liu, B. (2022). Sentiment analysis and opinion mining. *Springer Nature.*

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. J. (2011, June). Sentiment analysis of twitter data. *In Proceedings of the workshop on language in social media (LSM 2011),* 30-38.

Saif, H., He, Y., & Alani, H. (2012). Semantic sentiment analysis of twitter. In *The Semantic Web–ISWC 2012: 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I 11*, 508-524.

Bostancı, B., & Albayrak, A. (2021). Duygu analizi ile kişiye özel içerik önermek. *Veri Bilimi, 4(1),* 53-60.

MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. *In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, No. 14,* 281-297.

Wang, J., Zhang, X., & Zhou, H. (2006). A genetic k-means algorithm for spatial clustering. *Computer Engineering, 3,* 188-190.

Zhang, Z., Zhang, J., & Xue, H. (2008, May). Improved K-means clustering algorithm. *In 2008 Congress on Image and Signal Processing Vol. 5,* 169-172.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. *In 2008 eighth ieee international conference on data mining,* 413-422).

Zhang, L., Jiang, L., Li, C., & Kong, G. (2016). Two feature weighting approaches for naive Bayes text classifiers. *Knowledge-Based Systems, 100*, 137-144.

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. *In Uncertainty Proceedings 1994*, 399-406.

Bermejo, P., Gámez, J. A., & Puerta, J. M. (2014). Speeding up incremental wrapper feature subset selection with Naive Bayes classifier. *Knowledge-Based Systems, 55*, 140-147.

Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. *Knowledge-Based Systems, 192, 105361*, 1-12.

Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test, 25*, 197-227.

Keerthi, S. S., Duan, K. B., Shevade, S. K., & Poo, A. N. (2005). A fast dual algorithm for kernel logistic regression. *Machine learning, 61*, 151-165.

Zou, X., Hu, Y., Tian, Z., & Shen, K. (2019, October). Logistic regression model optimization and case analysis. *In 2019 IEEE 7th international conference on computer science and network technology (ICCSNT)*, 135-139.

Wang, L. (Ed.). (2005). Support vector machines: theory and applications. *Springer Science & Business Media.*

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications, 13(4)*, 18-28.

Huang, J., Li, Y. F., & Xie, M. (2015). An empirical analysis of data preprocessing for machine learning-based software cost estimation. *Information and Software Technology, 67*, 108-127.

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International Journal of Computer Science, 1(2)*, 111-117.

Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. *IEEE Access, 7*, 168261-168295.

Peng, Q., & Zhong, M. (2014). Detecting spam review through sentiment analysis. *J. Softw., 9(8)*, 2065-2072.

Ezpeleta, E., Velez de Mendizabal, I., Hidalgo, J. M. G., & Zurutuza, U. (2020). Novel email spam detection method using sentiment analysis and personality recognition. *Logic Journal of the IGPL, 28(1)*, 83-94.

Takaoğlu, M. ve Takaoğlu, F. (2019). K-means ve hiyerarşik kümeleme algoritmanın weka ve matlab platformlarında karşılaştırılması. *İstanbul Aydın Üniversitesi Dergisi,11(3)*, 303-317.

Hariri, S., Kind, M. C., & Brunner, R. J. (2019). Extended isolation forest. *IEEE transactions on knowledge and data engineering, 33(4)*, 1479-1489.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. *In 2008 eighth ieee international conference on data mining*, 413-422.

BÖLÜM IX

Çevrimiçi Reklamcılıkta Reklam Trafiği Satın Alma Optimizasyonu

Zeynep KOBAL KOÇBULUT[1]
Wojtek PRZEDZİMİRSKİ[2]
Fatih ÇOLAK[3]
Esma GÜNEŞ KAYA[4]

Giriş

Teknolojinin gelişmesi ve tüketici alışkanlıklarındaki değişimlerle birlikte pazarlama ve reklam araçları da gelişmiştir. Bu geçiş, geleneksel medya üzerinden yapılan pazarlama faaliyetlerinden internet ve dijital platformlara doğru yönelen reklam stratejilerine doğru bir değişim sürecidir. İnternetin ve dijital

[1] Triodor Ar-Ge Merkezi, İstanbul, Türkiye
[2] Azerion, Amsterdam, Hollanda
[3] Triodor Ar-Ge Merkezi, İstanbul, Türkiye
[4] Triodor Ar-Ge Merkezi, İstanbul, Türkiye

teknolojilerin gelişmesi, reklamverenlere daha hedeflenmiş ve ölçülebilir reklam seçenekleri sunmaktadır. Son yıllarda cirosu her yıl artmakta olan çevrimiçi reklamcılık sektörü, web sitesi sahiplerinin en önemli gelir kaynaklarından biri haline gelmiştir (Fuchs, 2018). Özellikle küçük ve orta ölçekli işletmeler, gelirlerinin önemli bir kısmını çevrimiçi reklamcılık yoluyla elde etmektedir. Geleneksel kanallar ile yapılan reklamların aksine çevrimiçi reklamcılık daha düşük maliyetlidir ve sahip olduğu geniş erişim ağı sayesinde hedef kitleye anında ulaşabilmektedir.

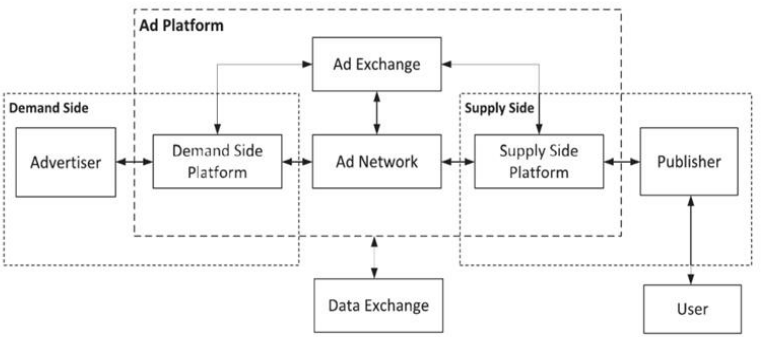
Eskiden yayıncılar ve reklamverenler önceden belirlenmiş sözleşmeler aracılığıyla birbirine bağlı olurlardı. Son yıllarda sözleşme temelli reklamcılığın yerini, bilgisayar programlarının yayıncıları ve reklamverenleri birbirine bağlama sürecini devraldığı "programatik reklamcılık" almıştır (Busch, 2016). Geleneksel reklamcılığa yeni bir boyut kazandıran bu yenilik, gerçek zamanlı veriyi işleyerek çevrimiçi reklam trafiği alım satımını gerçekleştirirken aynı zamanda tüketici ilgisine göre anlık reklam teklifleri oluşturabilmektedir (Zeren, 2019). Yapay zeka ve makine öğrenimi yöntemleri, karar verme görevleri için veri odaklı yaklaşımların geliştirilmesine yardımcı olur ve reklam alanlarının satış süreci birkaç milisaniyede gerçekleştirilir (Busch, 2016). Çevrimiçi reklamcılık süreci, bir web sitesine iFrame gibi bloklar yerleştirmek ve bunları bir hizmet veya ürün(ler) için reklam veren partilere satmaktan meydana gelir (Ryan ve Graham, 2014: 85-100). Bu bloklara reklam alanları denir ve son kullanıcı tarafından görüntülenen web sayfasında gösterimler oluştururlar (Ha, 2008: 31-48).

Bilgisayar yazılımlarının çalıştırılması, yayıncılar ve reklamverenler arasında bulunan aracı oluşumlarda gerçekleştirilir. Bu aracılara "Reklam Ağları" veya ''Ad Exchange''ler (AdX'ler) denir ve yayıncıları ve reklamverenleri doğrudan veya yardımcı platformlar aracılığıyla birbirine bağlarlar. Bu sistemde yayıncılar satıcılardır, gösterimler ise satılan öğelerdir. Reklamverenler ise alıcılardır ve bu gösterimleri satın alırlar. Yayıncıların ve reklamverenlerin doğrudan AdX'e bağlı olduğu sistemde yayıncı,

her gösterim için, AdX'e gösterimin ve son kullanıcının bilgilerini içeren bir reklam isteği gönderir. AdX, her iki tarafın karşılıklı tercihlerine göre bir reklamveren bulur. Yanıt, yayıncıya geri iletilir ve seçilen reklam, yayıncının reklam alanında gösterilir.

YÖNTEM

İnternet üzerinde çok sayıda reklam (AD) ağı bulunduğundan, yayıncılar web sitelerinde ayırdıkları reklam slotlarını yönetebilmek ve reklam verenlere satmak için sıklıkla mevcut arz tarafı platformlara (supply side platforms, SSP) ihtiyaç duymaktadır. Sistemin diğer tarafında ise reklamverenler, talep tarafı platformlar (demand side platforms, DSP) aracılığıyla kendi kampanyaları için teklif vermektedir. "Programmatic buying" adı verilen bu yapıda SSP ve DSP ajansları reklam trafiğini yönlendirmektedir. Şekil 1'de çevrimiçi reklam açık artırma pazarındaki aktörlerin konumlanmaları gösterilmektedir.

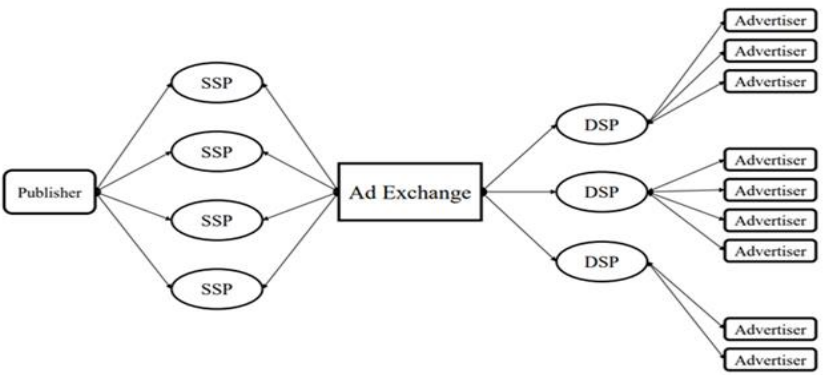


Şekil 1. Programatik reklam pazarındaki aktörler

Programatik reklamcılık, akıllı sistemlerde en hızlı ilerleyen teknolojilerden biridir. "Gerçek Zamanlı Teklif Verme" (RTB/Real-Time Bidding), yayıncıların gösterimlerini gerçek zamanlı açık arttırmalarla satan popüler bir programatik reklamcılık yöntemidir (Yuan vd., 2013: 3). Bu gerçek zamanlı açık artırmalar, yayıncıları ve reklamverenleri birbirine bağlayarak aracı görevi gören bazı

aktörler tarafından gerçekleştirilir. Bu şekilde, yayıncılar ve reklamverenlerin birbirlerini bulmak için özel bir çaba harcamasına gerek kalmamaktadır.

Reklam ağları, yayıncıların reklam alanlarını reklamverenlere satmak için açık artırmalar düzenler (Graham, 2010); yayıncılardan gösterim bilgilerini içeren reklam isteklerini alır ve reklam verenlerin gösterimler için teklifler verdiği çevrimiçi açık artırmalar gerçekleştirir. En yüksek teklifi veren reklamveren, reklam(lar)ını yayıncının web sitesinde göstermek üzere seçilir (Wang vd., 2017: 297-435). Yayıncıların ve reklamverenlerin açık artırmalara katılmalarına yardımcı olmak için iki oluşum tanıtılmıştır. Bir yandan Arz Tarafı Platformları (SSP'ler) reklam isteklerini yönetir ve yayıncıların AdX açık artırmaları aracılığıyla gösterim satmasına yardımcı olur. Diğer yandan, Talep Tarafı Platformları (DSP'ler) reklam verenlere bağlıdır ve açık artırmalarda teklif vermelerine yardımcı olur (Refaei Afshar, 2022). Şekil 2'de RTB sistemine genel bir bakış gösterilmektedir.



Şekil 2. Gerçek Zamanlı Teklif Verme Sistemi [10]

Birçok RTB sisteminde yayıncı, her gösterim için satmak istediği minimum miktarı belirten rezerv fiyat veya taban fiyat adlı bir tutar belirler. Bir yayıncı için farklı kârlar sağlayabilecek çeşitli AdX'ler vardır. Bu nedenle, gelir açısından en uygun AdX'e karar vermek yayıncı için rezerv fiyatı belirlemeden önce alınması

gereken önemli bir karardır. Açık artırmalara katılmak için bir yayıncının kullanabileceği yöntemler [10] çalışmasında anlatılmıştır.

Bu çalışmada, reklamverenleri reklam alanı sağlayan yayıncı iş ortaklarıyla buluşturan platformumuz için beklenen geliri en üst düzeye çıkarmak adına iki farklı optimizasyon yöntemi incelenmiştir. Bunlardan ilki reklam trafiği satın almak için dinamik bir reklam "tıklanma/kullanıcı kazanım" oranını tahmin eden bir model geliştirmek üzerinedir:

Çoğu durumda, reklam trafiğini yayıncıdan satın almak için kullanılan fiyatlandırma modeli, reklamverenin kampanyasında kullanılan fiyatlandırma modeliyle aynı değildir. Yayıncılar ağırlıklı olarak reklam gösterimi (rendered impression) başına ödeme almak isterken, reklamveren reklamın bir kullanıcı tarafından görüntülenmesi (viewable impression), reklama tıklanması veya reklam gösterildikten sonra gerçekleşen bazı olaylar için ödeme yapmak istemektedir (kullanıcıya bir ürünün satılması, kullanıcının kaydolması, vb.). Sistem, bu ikisi arasında kaldığından reklam gelirinin hesaplanmasında arbitraj riski yüklenmektedir.

Bu çalışma ile arbitrajlı trafik satın alma algoritmaları için aradaki dönüş oranını tahmin edecek bir yöntem geliştirilmesi amaçlanmıştır. Buradaki en kritik sorun, sonuca etki edecek verinin seçimidir. Oranın hem yayıncı trafik kriterlerine hem de kullanıcıya sunulan reklam kampanyasının özelliklerine bağlı olması beklenir. Ayrıca, seçilen özelliklerin belirleyicilik düzeyinin ve reklamı gören kullanıcı tepkisinin zaman içinde değiştiği göz önüne alındığında; tahmin modelinin koşullardaki değişimlere yeterince hızlı yanıt verebilecek şekilde geliştirilmesi de gerekir. Bu sorunlarla baş etmek amacıyla envanter ve reklam özelliklerine dayalı geniş bir girdi yelpazesi ile gradyan artırmaya dayalı bir regresyon öğrenim modeli geliştirilmiştir. Modelin özelliklerdeki değişikliklere hızlı yanıt verdiğinden emin olmak için, model eğitiminde girdi olarak dar günlük veri zaman pencereleri (günün kırılımları) kullanılmıştır.

Bu model aynı zamanda kampanya, cihaz ve diğer envanter özelliklerine bağlı olarak tıklama oranını başarıyla tahmin edebilecek seviyededir. Tahmini tıklama oranı, Lua tabanlı mantık katmanı tarafından yakalanmakta ve fiyat dönüşüm hesaplamasına dahil edilerek, reklam açık artırmalarında kullanılan bir gösterim başı ücret değeri oluşturmaktadır. Dağıtılmış veri işleme görevlerindeki gün içinde eğitim boru hatlarımız tarafından üretilen yeni tahmin modellerinin reklam sunucularına otomatik dağıtılmaları ve anında yüklenmeleri sağlanmaktadır.

Yapılan bir diğer çalışma da çoklu envanter kanallarında reklam trafiği satın alma üzerinedir:

Bir reklamveren için, bir reklam kampanyasını bir SSP ile doğrudan rezerve etmek, mevcut envanter ve özellikleri hakkında daha iyi bir iç gözlem sağlar, ancak bu SSP ile bağlantılı olmayan envanter kaynaklarına olan erişimi sınırlar. Diğer yandan bir DSP üzerinden kampanya rezervasyonu yaptığında bunun tersi söz konusudur: daha fazla envantere erişilebilir fakat tüm özelliklerini inceleme yeteneği yoktur. Geliştirmiş olduğumuz sistem, birden fazla SSP ve DSP ile entegre olduğundan reklamverene hem DSP'lerin hem de SSP'lerin birden fazla platformunda yayınlanmak üzere tek bir kampanya rezervasyonu yapma imkanı sunmak istemektedir. Çoklu dağıtılmış kampanya harcamalarını kontrol eden merkezi bir "bankacı" hizmeti içeren bir hizmet mimarisi tasarlanıp geliştirilmiştir.

Aktif reklam kampanyaları, bunların teslim süreci ve harcama seviyeleri ile ilgili olarak platformlar arası gerçek zamanlı iletişim kanalları bulunmamaktadır. Farklı reklam platformları farklı hedefleme ve envanter belirleme özelliklerini yönetebilmek ve reklamverenin ihtiyaçlarını bu platformların hepsinde sunulan kampanyalarla karşılaştırmalı karşılamak için "bankacının", farklı platformdaki kaynaklardan mikro kampanya siparişleri üretmesine olanak tanıyan bir mesajlaşma veri yolu kurulmuştur. Reklam bilgileri, Kafka tabanlı bir hizmet aracılığıyla, bankacı için teslimat geri bildirimini saniyeler mertebesinde toplayan bir veri hattına

aktarılmaktadır. Reklamverenin önceliklerine göre belirlediği kampanya hedefleme kriterlerinin gereklilik oranını içeren bir kampanya meta bilgi seti platformlardan hangilerinin kampanyayı reklamvereni tatmin edecek şekilde sunma kapasitesine sahip olduğuna karar vermesini sağlamaktadır.

Tartışma Ve Sonuç

Reklam ihalelerine ilişkin çevrimiçi yayıncılar için karar desteğinin pratik öneminin büyük olmasına rağmen, çalışmalar ve mevcut platformlardan (SSP'ler) yararlanmak için destek araçlarının geliştirilmesi üzerine yapılanlar şimdiye kadar araştırma toplulukları tarafından az ilgi çekmiştir. Reklam ihalelerine ilişkin mevcut araştırmalar ise çoğunlukla reklam pazarının analizi ve tasarımı üzerine yoğunlaşmış ve Google, Yahoo ve Facebook gibi dev çevrimiçi oyuncuların sponsorluğunda gerçekleştirilmiştir. Çoğunluk genel fiyatlama modellerinin içeriği, market tasarımı ve analizi, ihale tasarımı algoritmalardır. Yayıncıların garantili sözleşmeler için fiyatlandırma modelleri tasarlamasına yardımcı olmak için yalnızca birkaç çalışma yapılmıştır. Mevcut SSP'lerin en iyi şekilde kullanılmasını ile ilgilenen KOBİ'lere imkan verecek çok az bilgi mevcuttur. RTB sistemlerinin belirsizlik ve dinamiklik gibi özelliklere sahip yapısından dolayı birçok optimizasyon problemini barındırmaktadır. Tepki süresi, ağ kalitesi ve talep dağılımlarının önceden belirlenmesinin mümkün olmaması gibi faktörler göz önünde bulundurularak problemlerin çözümlerine yaklaşılması gerekmektedir.

Bu çalışmadaki problem için geliştirilen yeni model, herhangi bir şekilde yeniden yüklemede gecikme veya çalışmamaya sebep olmadan tahminleri yürütmektedir. Saatlik olarak tekrar eğitilen ve reklam sunucusu çalışma süresine kesintisiz olarak entegre edilmiş bir XGBoost H2O tabanlı tahmin modeli oluşturulmuştur. Bu, tıklama oranını %11'lik ortalama mutlak sapma ile tahmin etmeye olanak tanımakta ve reklam sunucusuna, tıklama başı fiyatlandırmayı görüntüleme başı fiyatlandırmaya bu yüksek doğruluk seviyesinde çevirme imkanı vermektedir.

Kaynakça

Busch, O. (Ed.). (2016). Programmatic advertising: the successful transformation to automated,

data-driven marketing in real-time. New York: Springer Cham.

Fuchs, C. (2018). The online advertising tax as the foundation of a public service internet. London:

University of Westminster Press.

Graham, R. (2010). A brief history of digital ad buying and selling. Retrieved on 27.11.2023 from

https://www.clickz.com/a-brief-history-of-digital-ad-buying-and-selling/55414/

Ha, L. (2012, Mayıs). Online advertising research in advertising journals: a review. Journal of

Current Issues & Research in Advertising. 30, 31-48. doi: 10.1080/10641734.2008.10505236

Ryan, K.M. & Graham, R.S. (2014). Taking Down Goliath: digital marketing strategies for

beating competitors with 100 times your spending power. New York: Palgrave Macmillan

Refaei Afshar, R. (2022). Machine learning for ad publishers in real time bidding. [Phd Thesis 1

(Research TU/e / Graduation TU/e), Industrial Engineering and Innovation Sciences]. Eindhoven University of Technology.

Wang, J. & Zhang, W. & Yuan, S. (2016). Display advertising with real-time bidding (RTB) and

behavioural targeting. Foundations and Trends in Information Retrieval, 11, 297-435. doi: 10.1561/1500000049

Yuan, S & Wang, J & Zhao, X. (2013, August). Real-time bidding for online advertising:

measurement and analysis. ADKDD '13: Proceedings of the Seventh International Workshop on Data Mining for Online Advertising, 1, 1-8. doi: 10.1145/2501040.2501980

Zeren, D. & Keşlikli, İ. (2019, Ekim). Programatik reklamcılık: kavram, işleyiş ve potansiyeli

açısından değerlendirmesi. Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 28 (2), 312-326. doi: 10.35379/cusosbil.628647

BÖLÜM X

Mobil Uygulama Geliştirmede Dart, Flutter, Kotlin, React Native ve Swift Yolculuğu

Funda AKAR[1]
Uğur KOLÇAK[2]

Giriş

Dijital çağın hızla ilerlemesiyle birlikte mobil cihazlar günlük yaşantımızın ayrılmaz bir parçası haline gelmiştir. Artık sadece iletişim aracı olmaktan çok daha fazlasını temsil ediyorlar. Mobil cihazlar hayatımızı kolaylaştıran ve daha eğlenceli hale getiren birçok uygulamayı sunuyor. Bu uygulamalar bize her an her yerde bilgiye erişme, alışveriş yapma, eğlence yaşama ve işlerimizi yönetme fırsatı veriyor. Ancak günümüz uygulamaların arkasındaki geliştirme süreci ve teknolojiler her birimizin bu deneyimleri elde etmemizi sağlıyor (Brito vd., 2019).

[1] Dr.Öğr.Üyesi, Erzincan Binali Yıldırım Üniversitesi, fakar@erzincan.edu.tr
[2] Öğrenci, Erzincan Binali Yıldırım Üniversitesi, ugurkolcak06@gmail.com

Bu makale mobil uygulama geliştirme dünyasına derinlemesine bir bakış atıyor. İster işinizi büyütmek ister kişisel bir projeyi hayata geçirmek için mobil uygulama geliştirmeyle ilgilenen bir geliştirici olun, doğru teknolojiyi ve yaklaşımı seçmek önemlidir. Bu yazıda, dört ana programlama dillerine odaklanacağız.

Dart, Kotlin, React Native ve Swift. Her bir teknolojinin kendine özgü yetenekleri ve avantajları olan programlama dilleri ve çerçeveleri mobil uygulama geliştirmenin çeşitli yönlerini kapsıyor. Dart, Google tarafından geliştirilen bir programlama dilidir ve Flutter çerçevesi ile bir araya gelerek hızlı ve kullanıcı dostu Android ve iOS uygulamaları oluşturmanıza olanak sağlar (Dart overview | Dart). Kotlin, Android uygulama geliştirmek için mükemmel bir seçenektir ve Java'ya göre birçok avantaj sunar. React Native, JavaScript temelli bir çerçeve olarak farklı platformlarda uygulama geliştirmeyi kolaylaştırırken Swift, iOS uygulamaları için hızlı ve etkileyici sonuçlar elde etmenizi sağlar . Bu makale boyunca, bu dört teknolojiyi ayrıntılı bir şekilde inceleyeceğiz ve hangi durumda hangi seçeneği tercih etmeniz gerektiği konusunda size rehberlik edeceğiz. Her bir teknolojinin özellikleri, avantajları ve dezavantajları hakkında daha fazla bilgi edinecek ve mobil uygulama geliştirmenin dünyasına bir adım daha yaklaşacaksınız.

Dart ve Flutter ile Mobil Uygulama Geliştirme

Mobil uygulama geliştirme dünyasında yeni ufuklara açılan teknolojiler olan Dart programlama dili ve Flutter çerçevesi, geliştiricilere heyecan verici olanaklar sunuyor. Flutter, 2017 yılında Google tarafından sunulan açık kaynaklı ve ücretsiz bir geliştirme yazılımıdır. Bu çerçeve, web, mobil ve masaüstü uygulamalarını geliştirmek için kullanılır ve hem Android hem de iOS tabanlı uygulamaların geliştirilmesini kolaylaştırır. Dart ise Google tarafından geliştirilen açık kaynaklı bir programlama dilidir. Mobil uygulamaların ötesinde, Dart programlama dili ile web, IoT cihazları için uygulamalar ve sunucu tarafı geliştirme de mümkündür(Sharma vd., 2022). Dart ve Flutter'ın en önemli özelliklerinden biri, tek bir kod tabanı ile hem Android hem de iOS için uygulama

geliştirebilmesidir. Bu, geliştiricilerin aynı uygulamayı iki farklı platform için tekrar tekrar yazmalarına gerek kalmaması anlamına gelir. Bu da hem zaman hem de maliyet tasarrufu sağlar.

Ayrıca, Flutter'ın yüksek performanslı ve güzel arayüzler sunması da geliştiricilerin tercih sebepleri arasındadır. Flutter, widget adı verilen bileşenlerle uygulamanın arayüzünü oluşturur. Bu widget'lar, platforma özgü özellikleri taklit edebilir veya kendi özgün tasarımlarını yansıtabilir. Flutter, widget'ları hızlı bir şekilde render ederek, uygulamanın akıcı ve pürüzsüz çalışmasını sağlar. Dart ve Flutter ile geliştirilen uygulamalar, hot reload ve hot restart gibi özellikler sayesinde, kodda yapılan değişiklikleri anında görüntüleyebilir ve test edebilir. Buda geliştirme sürecini daha verimli ve kolay hale getirir(Syaifudin vd., 2022)(Brito vd., 2019)

Avantajları arasında hızlı sonuçlanma süresi vermesi önde gelir. Flutter uygulama geliştirirken yapılan değişiklikleri, Stateful Hot Reload özelliği sayesinde değişiklikleri anlık olarak görüntülemeyi sağlar. Böylece yaptığımız değişiklikten sonra uygulamamızı tekrar başlatmadan sonuçları anında görüntülemeyi sağlar. Dart bünyesinde birçok widget kütüphanesi bulundurmaktadır. Böylece diğer SDK kıyasla daha hızlı bir geliştirme ortamı sağlar (Boukhary & Colmenares, 2019). Flutter, kod kalitesini ve uygulama güvenliğini artırmak için de birçok özellik sunar. Flutter, null safety adı verilen bir özelliği sayesinde, kodda null değerlerine karşı koruma sağlar. Bu, uygulamanın çalışma zamanında çökmesini önler ve hata ayıklamayı kolaylaştırır. Ayrıca, Flutter, test ve debug araçları ile geliştiricilere uygulamalarını daha kolay test etme ve hataları bulma imkânı verir. Flutter, Firebase gibi popüler servislerle de uyumlu çalışır. Firebase, uygulamaların veri depolama, kimlik doğrulama, analitik, bildirim ve daha birçok işlevini kolayca yönetmelerine olanak sağlar.

Flutter ile Kullanıcı Arayüzü Tasarlama ve Widget Kullanımı

Flutter ile kullanıcı arayüzü tasarlamak için, widget adı verilen bileşenleri kullanmak gerekir. Widget, uygulamanın görünümünü ve

davranışını belirleyen bir arayüz öğesidir. Flutter, iki tür widget sunar: stateless widget ve stateful widget. Stateless widget, durumunu değiştirmeyen, sabit bir widgettir. Stateful widget, durumunu değiştirebilen, dinamik bir widgettir. Flutter, widgetleri bir ağaç yapısı şeklinde düzenler. Widget ağacının en üstünde MaterialApp widgeti bulunur. MaterialApp widgeti, uygulamanın temel özelliklerini tanımlar. Widget ağacının altında Scaffold widgeti bulunur. Scaffold widgeti, uygulamanın ana iskeletini oluşturur. Scaffold widgetinin içinde AppBar, Drawer, Body, BottomNavigationBar gibi widgetler yer alabilir. Uygulamanın içeriğini göstermek için Body widgetinin içine Container, Column, Row, Text, Image, Icon, Button, List, Card gibi widgetler yer alabilir (Lohani, 2022).

Kotlin ile Android Uygulama Geliştirme ve Kotlin Programlamanın Özellikleri

Kotlin, Java sanal makinesi (JVM) üzerinde çalışan ve ayrıca JavaScript kaynak koduna derlenebilir, statik tipli bir programlama dilidir. İlk geliştirme Sankt-Peterburg, Rusya merkezli JetBrains programcıları tarafından yapılmıştır İsmi Kotlin Adasından gelmektedir. Java ile uyumlu söz dizimi olmasa da Kotlin Java kodu ile çalışmak üzere tasarlanmıştır. Kotlin ile Android uygulama geliştirmek, birçok geliştirici için cazip bir seçenektir. Çünkü Kotlin, Java'ya göre daha kısa, daha anlaşılır ve daha güvenli bir kod yazma imkânı sunar. Kotlin, Java ile tam uyumlu olduğu için, mevcut Java kodlarını Kotlin'e kolayca dönüştürebilir veya Java ve Kotlin kodlarını birlikte kullanabilirsiniz. Ayrıca, Kotlin, Android Studio gibi popüler geliştirme ortamlarıyla da entegre çalışır. Kotlin programlama dilinin sunduğu diğer özellikler ise şunlardır, Null safety, Kotlin, null değerlerini kontrol etmek için özel bir sözdizimi kullanır. Bu sayede, null pointer exception gibi hataları önler ve uygulamanın çökmesini engeller. Data classes, Kotlin, veri tutan sınıflar için data class adı verilen bir yapı sunar. Bu yapı sayesinde, sınıfların eşitlik, hash code, toString gibi fonksiyonlarını otomatik olarak oluşturabilirsiniz. Extension functions, Kotlin, mevcut

sınıflara yeni fonksiyonlar eklemek için extension function adı verilen bir özellik sağlar(*Kotlin for Android Developers*, 2015). Bu sayede, sınıfları değiştirmeden onlara yeni işlevler kazandırabilirsiniz. Coroutines, Kotlin, eş zamanlı programlama için coroutines adı verilen bir kavram kullanır. Coroutines, hafif ağırlıklı thread'ler olarak düşünülebilir. Coroutines sayesinde, uzun süren işlemleri arka planda çalıştırabilir ve uygulamanın yanıt vermesini sağlayabilirsiniz.

Kotlin'in geliştirilme aşamasında JetBrains bu dili neden tasarladıklarını sade bir şekilde açıklıyor. Kotlin, Java'ya oranla daha kısa ve daha anlaşılır, kendine özgü şekilde kodlama yapabilmektedir. En önemli özelliği ise Java ve Android ile tam performans uyumlu çalışabilmektedir (Hassan, 2019). JVM teknolojisi ile derlenir. Bu dili popüler yapan en önemli kısım ise Google tarafından duyuruldu. Android uygulamalar geliştirmek için tam performans Kotlin diline destek verdiklerini ve Android uygulama geliştirmek için resmi dil olduğunu tüm geliştiricilere paylaşıldı. Kotlin, Java'ya oranla daha modern, daha temiz ve daha esnek bir programlama dilidir. Kotlin, Java'da bulunan bazı sorunlu ve gereksiz özellikleri ortadan kaldırarak, geliştiricilere daha kolay ve daha güvenli bir kod yazma deneyimi sunar. Kotlin, Java ile tam uyumlu olduğu için, mevcut Java projelerine kolayca entegre edilebilir veya Java ve Kotlin kodları birlikte kullanılabilir. Kotlin, Android Studio gibi popüler geliştirme ortamlarıyla da sorunsuz bir şekilde çalışır (Syaifudin vd., 2022).

Kotlin programlama dilinin avantajları şunlardır: Daha az kod, Kotlin, Java'ya göre daha az kod satırı ile aynı işlevi gerçekleştirebilir. Bu, kodun okunabilirliğini ve bakımını artırırken, hata olasılığını azaltır. Daha güvenli, Kotlin, null safety adı verilen bir özellik sayesinde, null pointer exception gibi hataları önler ve uygulamanın çökmesini engeller. Ayrıca, Kotlin, data class, sealed class gibi yapılarla veri tutan sınıfları daha güvenli bir şekilde tanımlamayı sağlar. Daha esnek, Kotlin, extension function, infix function, operator overloading gibi özelliklerle mevcut sınıflara yeni fonksiyonlar eklemeyi veya var olan fonksiyonları değiştirmeyi

mümkün kılar. Bu da kodun daha esnek ve daha anlaşılır olmasını sağlar. Daha fonksiyonel, Kotlin, lambda expression, higher-order function, collection operation gibi özelliklerle fonksiyonel programlama paradigmasını destekler. Bu da kodun daha kısa ve daha ifade gücü yüksek olmasını sağlar(Martinez & Gois Mateus, 2022).

Kotlin ile Android uygulama geliştirmek için, Android Studio adı verilen entegre geliştirme ortamını (IDE) kullanmak gerekir. Android Studio, Kotlin desteği sunan, Google tarafından geliştirilen, Android uygulama geliştirme için özelleştirilmiş bir IDE'dir. Android Studio ile Kotlin projesi oluşturmak için, yeni proje sihirbazını kullanmak gerekir. Sihirbaz, proje adı, paket adı, minimum SDK seviyesi, uygulama türü, aktivite türü ve tema gibi seçenekleri sunar. Sihirbaz tamamlandığında, Android Studio, proje dosyalarını, klasörlerini ve kodlarını otomatik olarak oluşturur (*Kotlin for Android Developers,* 2015).

Kotlin ile Android Uygulamalarında Veri Yapıları, Nesne Yönelimli Programlama ve Fonksiyonel Programlama

Kotlin ile Android uygulamalarında veri yapısı olarak, diziler, listeler, kümeler ve haritalar kullanılabilir. Diziler, sabit boyutlu, aynı tipte elemanlardan oluşan veri yapısıdır. Listeler, değişken boyutlu, aynı veya farklı tipte elemanlardan oluşan veri yapısıdır. Kümeler, tekrarlanmayan, aynı veya farklı tipte elemanlardan oluşan veri yapısıdır. Haritalar, anahtar-değer çiftlerinden oluşan veri yapısıdır. Kotlin ile nesne yönelimli programlama yapmak için, sınıflar, nesneler, miras, soyutlama, polimorfizm ve enkapsülasyon gibi kavramlar kullanılır. Sınıflar, nesnelerin özelliklerini ve davranışlarını tanımlayan şablonlardır. Nesneler, sınıflardan türetilen, bellekte yer tutan örneklerdir. Miras, bir sınıfın başka bir sınıfın özelliklerini ve davranışlarını devralmasıdır. Soyutlama, karmaşık bir sistemi basit bir şekilde ifade etmedir. Polimorfizm, bir nesnenin farklı şekillerde davranabilmesidir. Enkapsülasyon, bir sınıfın iç detaylarını dış dünyadan gizlemesidir. Kotlin ile fonksiyonel programlama yapmak için, lambda ifadeleri, yüksek

seviyeli fonksiyonlar, saf fonksiyonlar, değişmez veriler ve gecikmeli değerlendirme gibi kavramlar kullanılır. Lambda ifadeleri, isimsiz fonksiyonlardır. Yüksek seviyeli fonksiyonlar, başka fonksiyonları parametre olarak alabilen veya döndürebilen fonksiyonlardır. Saf fonksiyonlar, yan etkisi olmayan, aynı girdiye aynı çıktıyı veren fonksiyonlardır. Değişmez veriler, değiştirilemeyen veri tipleridir. Gecikmeli değerlendirme, bir ifadenin ihtiyaç duyulduğu anda değerlendirilmesidir(*Kotlin for Android Developers,* 2015; Martinez & Gois Mateus, 2022).

React Native ile Uygulama Geliştirme ve React Native Programlamanın Özellikleri

React ilkel yerel platform kullanıcı arayüzüne render, yani uygulamanız aynı yerel platform API'leri diğer uygulamalar yapar. Birçok platform, tek bir React. Tek bir kod tabanının platformlar arasında kod paylaşabilmesi için bileşenlerin platforma özgü sürümlerini oluşturun. React Native ile bir ekip birden fazla platformu yönetebilir ve ortak bir teknolojiyi paylaşabilir. React Native, gerçek anlamda yerel uygulamalar oluşturmanızı sağlar ve kullanıcılarınızın deneyimlerinden ödün vermez. Doğrudan platformun yerel UI yapı taşlarıyla eşleşen View, Text ve Image gibi platformdan bağımsız yerel bileşenlerden oluşan bir çekirdek set sağlar. Bu paragrafı daha kapsamlı hale getirmek ve uzatmak için, React Native'in ne olduğunu ve nasıl çalıştığını daha detaylı bir şekilde açıklayabilirsiniz. Örneğin, şu şekilde devam edebilirsiniz, React Native, Facebook tarafından geliştirilen ve JavaScript ile platformlar arası mobil uygulamalar oluşturmayı sağlayan bir yazılım çerçevesidir (Kadrija vd., 2022). React Native, web geliştiricilerinin mevcut bilgilerini kullanarak iOS ve Android uygulamaları geliştirmelerine olanak tanır. React Native uygulamaları, yerel uygulamalara benzer performans ve kullanıcı deneyimi sunar. React Native, web geliştiricilerinin bildiği React kütüphanesini kullanarak, yerel platformun API'lerini çağırır ve kullanıcı arayüzü öğelerini oluşturur. Bu sayede, web görünümü kullanmadan, doğrudan yerel bileşenlerle çalışır. React Native,

ayrıca tek bir kod tabanı ile hem iOS hem de Android için uygulama geliştirmeye olanak tanır. Bu da geliştirme süresini ve maliyetini azaltır. React Native'in sunduğu avantajlar arasında, kod tekrar kullanımı, performans, bileşen tabanlı yapı ve hızlı geliştirme süreci sayılabilir(Kishore vd., 2022).

React Native, 2015 yılında GitHub'da paylaşılarak kullanıma sunulmuştur. React Native'i tercih etmenin birçok nedeni vardır. Bunlardan bazıları şunlardır,

- Platformlar arası uyumluluk: React Native, tek bir kod tabanı ile hem Android hem de iOS için uygulama geliştirmeye olanak tanır. Bu da geliştirme süresini ve maliyetini azaltır ve kod tekrarından kaçınır**.**
- Emülatör kullanımı: React Native, emülatör kullanarak uygulamaları test etmeyi ve hata ayıklamak oldukça kolaydır. Ayrıca, sıcak yeniden yükleme (hot reload) ve sıcak modül değiştirme (hot module replacement) gibi özellikler sayesinde kodda yapılan değişiklikleri anında görüntüleyebilir ve test edebilirsiniz**.**
- Güvenlik: React Native, güvenli bir şekilde veri depolama, kimlik doğrulama, bildirim ve daha birçok işlevi kolayca yönetmenize olanak sağlayan Firebase gibi popüler servislerle uyumlu çalışır. Ayrıca, React Native, uygulamaların güvenliğini artırmak için HTTPS protokolünü destekler.
- Hızlı prototipleme: React Native, hızlı bir şekilde çalıştığınız projede hızlı bir prototipleme yapmanızı sağlar. Böylece, uygulamanın işlevselliğini ve kullanılabilirliğini daha erken test edebilir ve geri bildirim alabilirsiniz.
- Tasarım özgünlüğü: React Native, yerel platformun arayüz öğelerini taklit edebilen veya kendi özgün tasarımlarını yansıtabilen widget adı verilen bileşenlerle uygulamanın arayüzünü oluşturur. Böylece, uygulamanızın tasarımını istediğiniz gibi özelleştirebilirsiniz.
- Süreklilik: React Native, web geliştiricilerinin bildiği JavaScript dilini kullanarak, web uygulamalarından mobil

uygulamalara kolayca geçiş yapmalarını sağlar. Bu da geliştiricilerin sürekli olarak yeni diller veya teknolojiler öğrenmelerine gerek kalmadan projelerine devam etmelerini sağlar.

- Topluluk desteği: React Native, arkasında çok büyük bir topluluğa sahiptir. Bu topluluk sayesinde, React Native ile ilgili sorularınıza cevap bulabilir, kaynaklara ulaşabilir ve yeni özelliklerden haberdar olabilirsiniz. Ayrıca, React Native ile geliştirilen birçok açık kaynaklı kütüphane ve modül de topluluk tarafından paylaşılmaktadır.
- Büyük firmaların tercihi: React Native, dünyada milyonlarca kullanıcıya sahip bazı popüler uygulamaların geliştirilmesinde kullanılmıştır. Bunlardan bazıları Facebook, Instagram, Skype, Uber Eats ve Pinterest'tir. Bu da React Native'in güvenilirliğini ve kalitesini göstermektedir.

React Native ile Kullanıcı Arayüzü Tasarlama ve Bileşen Kullanımı

React Native ile kullanıcı arayüzü tasarlamak için, bileşen adı verilen arayüz öğelerini kullanmak gerekir. Bileşen, uygulamanın görünümünü ve davranışını belirleyen bir kod parçasıdır. React Native, yerel platformlara uyumlu, hazır bileşenler sunar. Bunlar arasında View, Text, Image, Button, TextInput, ScrollView, FlatList, SectionList, Switch, Slider, Picker, Modal, Alert, StatusBar, ActivityIndicator gibi bileşenler bulunur. React Native, ayrıca, kendi bileşenlerini oluşturmak için, bileşenleri birleştirmeye veya özelleştirmeye olanak sağlar. React Native, bileşenleri bir ağaç yapısı şeklinde düzenler. Bileşen ağacının en üstünde App bileşeni bulunur. App bileşeni, uygulamanın ana bileşenidir. App bileşeninin altında, uygulamanın içeriğini gösteren diğer bileşenler yer alır(Azizah vd., 2021; GÜLCÜOĞLU vd., 2021).

Çapraz Platform Uygulama Geliştirmenin Avantajları ve Dezavantajları

Çapraz platform uygulama geliştirmek hem iOS hem de Android için aynı kodu kullanarak uygulama geliştirmek anlamına gelir. Çapraz platform uygulama geliştirmenin avantajları arasında, geliştirme süresinin ve maliyetinin azalması, kodun yeniden kullanılabilirliği, bakımın kolaylığı, uygulamanın daha geniş bir kitleye ulaşması, uygulamanın tutarlılığı sayılabilir. Çapraz platform uygulama geliştirmenin dezavantajları arasında ise, performansın düşmesi, yerel özelliklere erişimde sınırlılık, uygulamanın platformlara uyum sağlamada zorlanması, uygulamanın güvenliğinin azalması, uygulamanın boyutunun büyümesi sayılabilir (Tunalı vd., 2015).

Swift ile iOS Uygulama Geliştirme ve Swift Programlamanın Özellikleri

Swift programlama dili, Apple tarafından 2014 yılında tanıtılan ve Objective-C'nin yerini alması planlanan bir dildir. Swift, modern, güçlü ve sezgisel bir dildir ve iOS, macOS, watchOS, tvOS ve Linux gibi platformlarda uygulama geliştirmek için kullanılır (Schneider & Schultes, 2022). Swift, açık kaynak kodlu bir dildir ve geliştiricilerin kodu incelemesine, değiştirmesine ve katkıda bulunmasına olanak tanır (Swift history | Swift). Swift programlama dilinin özellikleri şunlardır:

- Nesne yönelimli ve işlevsel: Swift, nesne yönelimli programlama paradigmasını destekler ve sınıf, yapı, protokol gibi kavramları kullanır. Ayrıca, işlevsel programlama paradigmasını da destekler ve lambda ifadeleri, yüksek seviyeli fonksiyonlar, koleksiyon işlemleri gibi özellikleri kullanır.
- Statik tipli: Swift, statik tipli bir dildir ve değişkenlerin tipini derleme zamanında belirler. Bu, hata olasılığını azaltır ve performansı artırır. Ayrıca, Swift, tip çıkarımı adı verilen bir özellik sayesinde, değişkenlerin tipini belirtmeden de tanımlayabilirsiniz.

- Güvenli: Swift, güvenli bir dildir ve kodda hataları önlemek için birçok özellik sunar. Örneğin, opsiyonel değerler adı verilen bir yapı sayesinde, null değerlerini kontrol edebilir ve null pointer exception gibi hataları engelleyebilirsiniz. Ayrıca, Swift, bellek yönetimini otomatik olarak yapar ve bellek sızıntısı gibi sorunları önler.
- Hızlı: Swift, hızlı bir dildir ve C ve Objective-C ile karşılaştırıldığında daha yüksek bir performans sunar. Bu da uygulamaların daha hızlı çalışmasını sağlar. Ayrıca, Swift'in sözdizimi daha temiz ve anlaşılır olduğundan, uygulama geliştirme süreci de daha hızlı olur.
- Modern: Swift, modern bir dildir ve pek çok yeni özellik sunar. Örneğin, eklenti fonksiyonlar adı verilen bir yapı sayesinde, mevcut sınıflara yeni fonksiyonlar ekleyebilir veya var olan fonksiyonları değiştirebilirsiniz. Ayrıca, Swift, jenerikler, türetilmiş sınıflar, çevrim içi fonksiyonlar gibi pek çok gelişmiş özellik sunar.
- Kolay öğrenme: Swift, kolay öğrenilebilir bir dildir ve hiç kodlama bilmeyen biri bile kolayca Swift dilini tanıyabilir. Swift'in sözdizimi basit ve anlaşılır olduğu için kod yazmak çok kolaydır.
- Platformlar arası uyumluluk: Swift ile geliştirilen uygulamalar hem iOS hem de macOS platformlarında çalışabilir. Ayrıca, Swift ile Linux platformunda da uygulama geliştirebilirsiniz. Bu da geliştiricilerin farklı platformlara uyum sağlamasını kolaylaştırır.
- Xcode entegrasyonu: Swift ile uygulama geliştirmek için Xcode adı verilen Apple'ın entegre geliştirme ortamını kullanabilirsiniz. Xcode ile uygulama tasarlama, test etme ve hata ayıklama gibi pek çok işlevi yerine getirebilirsiniz. Ayrıca Xcode ile Objective-C kodlarını da Swift'e kolayca dönüştürebilir veya Swift ve Objective-C kodlarını birlikte kullanabilirsiniz.
- Güçlü topluluk desteği: Swift, güçlü bir topluluk desteğine sahiptir. Swift ile ilgili sorularınıza cevap bulabilir, kaynaklara ulaşabilir ve yeni özelliklerden haberdar olabilirsiniz. Ayrıca,

Swift ile geliştirilen birçok açık kaynaklı kütüphane ve modül de topluluk tarafından paylaşılmaktadır.

- Geleceğe yönelik: Swift, Apple'ın gelecekteki platformları için de uyumlu olacak şekilde tasarlanmıştır. Apple, Swift'i uzun vadeli bir programlama dili olarak planladığı için, Swift ile geliştirilen uygulamalar gelecekteki güncellemelere de uyumlu olacak şekilde tasarlanmıştır.

Swift ile iOS uygulama geliştirmek için, Xcode adı verilen entegre geliştirme ortamını (IDE) kullanmak gerekir. Xcode, Swift desteği sunan, Apple tarafından geliştirilen, iOS uygulama geliştirme için özelleştirilmiş bir IDE'dir. Xcode ile Swift projesi oluşturmak için, yeni proje sihirbazını kullanmak gerekir. Sihirbaz, proje adı, takım adı, organizasyon adı, organizasyon tanımlayıcısı, paket adı, arayüz türü, yaşam döngüsü türü, minimum iOS sürümü, uygulama türü, aktivite türü ve tema gibi seçenekleri sunar. Sihirbaz tamamlandığında, Xcode, proje dosyalarını, klasörlerini ve kodlarını otomatik olarak oluşturur(Ansari, 2017).

Swift ile iOS Uygulamalarında Veri Yapıları, Nesne Yönelimli Programlama ve Protokol Yönelimli Programlama

Swift ile iOS uygulamalarında veri yapısı olarak, diziler, kümeler ve haritalar kullanılabilir. Diziler, değişken boyutlu, aynı tipte elemanlardan oluşan veri yapısıdır. Kümeler, tekrarlanmayan, aynı tipte elemanlardan oluşan veri yapısıdır. Haritalar, anahtar-değer çiftlerinden oluşan veri yapısıdır. Swift ile nesne yönelimli programlama yapmak için, sınıflar, nesneler, miras, soyutlama, polimorfizm ve enkapsülasyon gibi kavramlar kullanılır. Sınıflar, nesnelerin özelliklerini ve davranışlarını tanımlayan şablonlardır. Nesneler, sınıflardan türetilen, bellekte yer tutan örneklerdir. Miras, bir sınıfın başka bir sınıfın özelliklerini ve davranışlarını devralmasıdır. Soyutlama, karmaşık bir sistemi basit bir şekilde ifade etmedir. Polimorfizm, bir nesnenin farklı şekillerde davranabilmesidir. Enkapsülasyon, bir sınıfın iç detaylarını dış dünyadan gizlemesidir. Swift ile protokol yönelimli programlama

yapmak için, protokoller, uzantılar, delegasyon, kapanışlar, özellik gözlemcileri, delegasyon, kapanışlar, özellik gözlemcileri gibi kavramlar kullanılır. Protokoller, bir sınıfın, yapının veya enumun uyması gereken özellikleri ve davranışları tanımlayan sözleşmelerdir(Nunes vd., 2017). Uzantılar, bir sınıfın, yapının veya enumun özelliklerini ve davranışlarını genişletmeye yarayan araçlardır. Delegasyon, bir nesnenin başka bir nesneye bazı sorumluluklarını devretmesidir. Kapanışlar, isimsiz fonksiyonlardır. Özellik gözlemcileri, bir özelliğin değerinin değişmesini izleyen ve buna göre işlem yapan araçlardır.

Uygulama Geliştirme Adımları

Uygulama geliştirme, bir fikri gerçekleştirilebilmesi için izlenmesi gereken süreçlerden bir tanesidir. Uygulama geliştirme, web, mobil, masaüstü veya giyilebilir gibi farklı platformlar için uygulamalar oluşturmayı sağlar. Uygulama geliştirme, hangi programlama dilini veya hangi alanı seçtiğinize göre farklılık gösterir. Örneğin, Android için Java veya Kotlin, iOS için Swift veya Objective-C, cross platform için Flutter, React Native , web için HTML, CSS, JavaScript gibi farklı diller seçimlere bağlık olarak kullanılabilirdir. Uygulama geliştirme süreci, genellikle şu adımlardan oluşur,

Fikir belirleme, Uygulama geliştirmenin ilk adımı, uygulamanın ne yapacağına ve hangi sorunu çözeceğine karar vermektir. Uygulamanın fikri, kullanıcıların ihtiyaçlarını, beklentilerini ve ilgilerini karşılamalıdır. Ayrıca, uygulamanın rakiplerinden farklılaşması ve özgün olması da önemlidir.

Araştırma yapma, Uygulama geliştirmenin ikinci adımı, uygulamanın pazarını, hedef kitlesini, rakiplerini ve teknik gereksinimlerini araştırmaktır. Bu aşamada, uygulamanın başarılı olması için hangi faktörlerin etkili olduğunu belirlemek ve uygulamanın güçlü ve zayıf yönlerini analiz etmek gerekir.

Tasarım oluşturma, Uygulama geliştirmenin üçüncü adımı, uygulamanın kullanıcı arayüzünü ve işlevselliğini tasarlamaktır. Bu

aşamada, uygulamanın nasıl görüneceği ve çalışacağı belirlenir. Uygulamanın tasarımı, kullanıcı deneyimini iyileştirmek ve kullanıcı memnuniyetini artırmak için basit, şık ve kullanışlı olmalıdır.

Kodlama yapma, Uygulama geliştirmenin dördüncü adımı, uygulamanın kodunu yazmaktır. Bu aşamada, seçilen programlama dili ve platforma uygun olarak uygulamanın iş mantığı ve algoritmaları kodlanır. Uygulamanın kodu, hata içermemek ve performanslı çalışmak için temiz, anlaşılır ve optimize edilmiş olmalıdır.

Test etme, Uygulama geliştirmenin beşinci adımı, uygulamanın test edilmesidir. Bu aşamada, uygulamanın çalışmasında herhangi bir sorun olup olmadığı kontrol edilir. Uygulamanın testi, farklı senaryolar, cihazlar ve koşullar altında yapılmalıdır. Uygulamanın testi, hataları bulmak ve düzeltmek için önemlidir.

Yayınlama, Uygulama geliştirmenin altıncı adımı, uygulamanın yayınlanmasıdır. Bu aşamada, uygulamanın son halinin seçilen platformun mağazasına yüklenmesi ve kullanıcıların erişimine sunulması sağlanır. Uygulamanın yayınlanması için platformun kurallarına ve standartlarına uygun olması gerekir.

Güncelleme, Uygulama geliştirmenin son adımı, uygulamanın güncellenmesidir. Bu aşamada, uygulamanın performansını artırmak, yeni özellikler eklemek veya var olan özellikleri iyileştirmek için uygulamaya düzenli olarak güncellemeler yapılır. Ayrıca, kullanıcıların geri bildirimleri de dikkate alınarak uygulamanın kalitesi yükseltilir.

Her Dil ve Framework İçin Temel Uygulama Geliştirme Adımları

Flutter, Swift, Kotlin ve React Native, çapraz platform uygulama geliştirmek için kullanılan popüler programlama dilleri ve çerçevelerdir. Çapraz platform uygulama geliştirme, tek bir kod

tabanı ile hem Android hem de iOS için uygulama geliştirmeyi sağlar. Bu da geliştirme süresini ve maliyetini azaltır. Ancak, bu diller ve çerçeveler arasında bazı farklılıklar vardır. Bu farklılıklar, uygulama geliştirme sürecini de etkiler. Aşağıda, her biri için uygulama geliştirme adımlarını bulabilirsiniz,

Flutter, Google tarafından geliştirilen ve Dart programlama dilinde yazılan bir mobil uygulama geliştirme çerçevesidir. Flutter ile uygulama geliştirmek için şu adımları izleyebilirsiniz,

Flutter SDK'sını indirin ve kurun. Flutter destekli bir IDE (örneğin Android Studio veya Visual Studio Code) kurun ve eklentilerini yükleyin. Yeni bir Flutter projesi oluşturun veya mevcut bir projeyi açın. Uygulamanızın tasarımını widget adı verilen bileşenlerle oluşturun. Uygulamanızın iş mantığını ve algoritmalarını Dart dilinde kodlayın. Uygulamanızı emülatörde veya fiziksel cihazda test edin ve hata ayıklayın. Uygulamanızı Android veya iOS platformuna derleyin ve yayınlayın .

Swift, Apple tarafından geliştirilen ve iOS, macOS, watchOS ve tvOS için uygulama geliştirmek için kullanılan modern bir programlama dilidir. Swift ile uygulama geliştirmek için şu adımları izleyebilirsiniz,

Xcode adı verilen Apple'ın bütünleşmiş geliştirme ortamını indirin ve kurun. Yeni bir Xcode projesi oluşturun veya mevcut bir projeyi açın. Uygulamanızın tasarımını storyboard adı verilen arayüz editörüyle oluşturun. Uygulamanızın iş mantığını ve algoritmalarını Swift dilinde kodlayın. Uygulamanızı simülatörde veya fiziksel cihazda test edin ve hata ayıklayın. Uygulamanızı iOS platformuna derleyin ve yayınlayın.

Kotlin, JetBrains tarafından geliştirilen ve Java ile uyumlu olan genel amaçlı bir programlama dilidir. Kotlin ile hem Android hem de iOS için uygulama geliştirmek mümkündür. Kotlin ile uygulama geliştirmek için şu adımları izleyebilirsiniz,

Android Studio veya IntelliJ IDEA gibi Kotlin destekli bir IDE kurun ve eklentilerini yükleyin. Yeni bir Kotlin projesi oluşturun

veya mevcut bir projeyi açın. Uygulamanızın tasarımını XML dosyalarında veya Jetpack Compose gibi arayüz kütüphanelerinde oluşturun. Uygulamanızın iş mantığını ve algoritmalarını Kotlin dilinde kodlayın. Uygulamanızı emülatörde veya fiziksel cihazda test edin ve hata ayıklayın. Uygulamanızı Android veya iOS platformuna derleyin ve yayınlayın(*Kotlin for Android Developers*, 2015).

React Native, Facebook tarafından geliştirilen ve JavaScript programlama dilinde yazılan bir mobil uygulama geliştirme çerçevesidir. React Native ile uygulama geliştirmek için şu adımları izleyebilirsiniz,

React Native CLI veya Expo gibi bir araç kullanarak React Native ortamını kurun. Yeni bir React Native projesi oluşturun veya mevcut bir projeyi açın. Uygulamanızın tasarımını React Native bileşenleriyle oluşturun. Uygulamanızın iş mantığını ve algoritmalarını JavaScript dilinde kodlayın. Uygulamanızı emülatörde veya fiziksel cihazda test edin ve hata ayıklayın. Uygulamanızı Android veya iOS platformuna derleyin ve yayınlayın (Kishore vd., 2022).

Uygulama Fikri Belirleme ve Pazar Araştırması Yapma

Uygulama fikri belirlemek, uygulama geliştirme sürecinin en önemli adımlarından biridir. Uygulama fikri, uygulamanın amacını, hedef kitlesini, iş modelini ve farklılaştırıcı özelliğini tanımlar. Uygulama fikri belirlemek için, ilgi alanlarını, tutkularını, problemlerini, ihtiyaçlarını, beklentilerini ve isteklerini düşünmek gerekir. Uygulama fikri belirledikten sonra, pazar araştırması yapmak gerekir. Pazar araştırması, uygulamanın rekabet gücünü, potansiyel müşterilerini, pazar büyüklüğünü, pazar trendlerini ve pazar ihtiyaçlarını analiz etmeye yarar. Pazar araştırması yapmak için, rakip uygulamaları incelemek, müşteri anketleri yapmak, pazar raporları okumak, pazar uzmanları ile görüşmek gibi yöntemler kullanılabilir.

Uygulama Tasarımı Oluşturma ve Kullanıcı Deneyimi Planlama

Uygulama tasarımı oluşturmak, uygulamanın görünümünü, işlevselliğini, gezinmesini ve etkileşimini belirlemeye yarar. Uygulama tasarımı oluşturmak için, kullanıcı arayüzü (UI) ve kullanıcı deneyimi (UX) tasarımı yapmak gerekir. UI tasarımı, uygulamanın renklerini, fontlarını, ikonlarını, resimlerini, düğmelerini, menülerini, formlarını, animasyonlarını ve geçişlerini tanımlar. UX tasarımı, uygulamanın kullanıcıların ihtiyaçlarını, beklentilerini, duygularını ve memnuniyetini nasıl karşıladığını tanımlar. UI ve UX tasarımı yapmak için, kullanıcı araştırması yapmak, kullanıcı hikayeleri yazmak, kullanıcı akışları çizmek, bilgi mimarisi oluşturmak, tel çerçeveler hazırlamak, prototipler yapmak, testler yapmak gibi yöntemler kullanılabilir.

Uygulama Kodlama, Test Etme ve Yayınlama

Uygulama kodlama, uygulamanın tasarımını, iş mantığını, veri tabanını ve arka plan işlemlerini kodlamaya yarar. Uygulama kodlama yapmak için, uygun programlama dili, framework, kütüphane, IDE, SDK ve API seçmek gerekir. Uygulama kodlama yaparken, kod kalitesini, performansını, güvenliğini ve hata önleme yöntemlerini göz önünde bulundurmak gerekir. Uygulama kodlama bittikten sonra, uygulama test etmek gerekir. Uygulama test etmek, uygulamanın çalışabilirliğini, işlevselliğini, kullanılabilirliğini, uyumluluğunu, güvenliğini ve performansını kontrol etmeye yarar. Uygulama test etmek için, birim testleri, entegrasyon testleri, sistem testleri, kabul testleri, regresyon testleri, stres testleri, yük testleri, güvenlik testleri, kullanılabilirlik testleri gibi yöntemler kullanılabilir. Uygulama test edildikten sonra, uygulama yayınlamak gerekir. Uygulama yayınlamak, uygulamanın hedef platformlara dağıtılmasını sağlar. Uygulama yayınlamak için, uygulamanın sürümünü, sertifikasını, imzasını, açıklamasını, ikonunu, ekran görüntülerini, videolarını, kategorisini, fiyatını, gizlilik politikasını ve kullanım koşullarını belirlemek gerekir. Uygulama yayınlamak

için, hedef platformların geliştirici hesaplarına kaydolmak, uygulamayı yüklemek, incelemeye göndermek ve onay almak gerekir (ELİBOL & SELÇUKCAN EROL, 2017; KESKİN & KILINÇ, 2015).

Performans ve Optimizasyon

Flutter Performans ve Optimizasyon Flutter, hızlı bir geliştirme süreci sunar ve uygulama geliştiricilerine avantajlar sağlar.

Hızlı Geliştirme, Flutter, Stateful Hot Reload özelliği sayesinde kod değişikliklerini anında görüntülemeyi sağlar. Bu, geliştirme sürecini hızlandırır.

Cihazlar Arası Uyum Hem Android hem de iOS için aynı kod tabanını kullanabilirsiniz. Bu, projelerin hızla platformlar arası geçiş yapmasını sağlar.

Özelleştirilebilirlik, Flutter, özelleştirilebilir ve animasyonlu kullanıcı arayüzleri oluşturmak için geniş bir widget kütüphanesi sunar. Kotlin Performans ve Optimizasyon Kotlin, Android uygulama geliştirmek için giderek daha popüler hale geliyor. İşte Kotlin'in performans ve optimizasyon yetenekleri, Java ile Uyum, Kotlin, Java ile sorunsuz bir şekilde entegre olur ve Android uygulamalarını daha hızlı geliştirmeyi sağlar. Okunabilir Kod, Kotlin, daha kısa ve daha anlaşılır kodlar yazmanıza olanak tanır, bu da geliştirme sürecini hızlandırır.

Performans, Kotlin, Android platformuyla mükemmel bir şekilde uyumlu çalışır ve uygulamaların hızlı çalışmasını sağlar. React Native Performans ve Optimizasyon React Native, platformlar arası geliştirmeyi kolaylaştırır, ancak bazı özel performans ve optimizasyon özellikleri sunar, Platformlar Arası Kullanılabilirlik hem Android hem de iOS için tek bir kod tabanı kullanabilirsiniz, bu da geliştirme sürecini hızlandırır.

Hızlı Prototipleme, React Native, hızlı bir şekilde çalışan projelerde hızlı prototipler oluşturmanızı sağlar. Tasarım

Özgünlüğü, React Native, yerel platformların UI bileşenlerine uygunluğu sayesinde kullanıcı deneyimini artırır. Swift Performans ve Optimizasyon Swift, Apple platformları için özel olarak tasarlanmıştır ve yüksek performans sağlar, Optimize Edilmiş Çalışma, Swift, Apple platformları için optimize edilmiş bir dil olarak tasarlanmıştır ve uygulamaların yüksek performans sergilemesini sağlar.

Güçlü ve Sezgisel Dil, Swift, geliştiricilerin hızlı ve verimli bir şekilde kod yazmasına olanak tanır. Çeşitli Çerçeveler ve API'ler, Apple, uygulama geliştiricilerine benzersiz ve eğlenceli deneyimler sunmak için çok sayıda çerçeve ve API sağlar. Karşılaştırma sonuçları her dört dilin performansı ve optimizasyon yetenekleri, projelerinizin gereksinimlerine bağlı olarak değişebilir. Eğer hızlı prototipleme ve çapraz platform uyumluluğu önemliyse, React Native veya Flutter seçenekleri cazip olabilir. Özellikle Android için geliştirme yapacaksanız, Kotlin hızlı ve kullanışlı bir seçenektir. Ancak, tamamen Apple platformlarına odaklanacaksanız, Swift en popüler performansı sunabilir. Sonuç olarak, her dilin kendine özgü avantajları ve zorlukları vardır. Projenizin özel gereksinimlerini ve hedeflerinizi dikkate alarak doğru dil seçimini yapmanız önemlidir.

Çapraz Platform Desteği

Flutter, Teknolojinin öncüsü konumunda yer alır, Google tarafından geliştirilen açık kaynaklı bir mobil uygulama geliştirme çerçevesidir. İşte Flutter'ın avantajları ve dezavantajları,

Avantajlar hem Android hem de iOS için tek bir kod tabanı kullanımı. Hızlı prototipleme ve geliştirme süreçleri. Zengin ve özelleştirilebilir bir widget kütüphanesi vardır. Google tarafından desteklenen açık kaynak bir SDK sunar.

Dezavantajlar, Platforma özgü performansı optimize etme yeteneği sınırlı olabilir.

Kotlin Multiplatform Mobile (KMM), Kotlin'in Gücü KMM, JetBrains tarafından geliştirilen Kotlin programlama dilini

kullanarak Android ve iOS için çapraz platform desteği sağlar. İşte KMM'nin avantajları ve dezavantajları,

Avantajlar, Android ve iOS için tek bir kod tabanını paylaşma. Java ile sorunsuz entegrasyon. Temiz ve okunabilir kod yazma olanağı. Kapsamlı bir dil ve geliştirici topluluğu.

Dezavantajlar, KMM gelişmekte olan bir teknoloji olup bazı eksiklikler içerebilir.

React Native, JavaScript'in Gücü React Native, Facebook tarafından geliştirilen bir çapraz platform çerçevesidir.

React Native'in avantajları ve dezavantajları,

Avantajlar hem Android hem de iOS için tek bir kod tabanı kullanımı. JavaScript tabanlı olması ve geniş bir geliştirici topluluğu. Hızlı prototipleme ve güncelleme olanağı. Facebook'un desteği ve sürekli güncellemeler.

Dezavantajlar, Yerel uygulamalara göre performans ve kullanıcı deneyimi bazen düşebilir.

Xamarin, .NET Ekosistemi Xamarin, Microsoft tarafından desteklenen bir çapraz platform geliştirme çerçevesidir.

Xamarin'in avantajları ve dezavantajları, Avantajlar Hem Android hem de iOS için tek bir C# kod tabanını kullanma. .NET ekosistemi ile entegrasyon. Yerel platform API'lerine erişim ve kullanıcı deneyimi özelleştirme yeteneği sunar.

Dezavantajlar, Projeyi başlatma süreci diğerlerine göre daha karmaşık olabilir. Xamarin'ın bazı sürümleri lisans maliyeti gerektirebilir.

Hangi programlama dilini seçeceğinizi belirlerken, projenizin karmaşıklığı, hedef kitlesi, ekibinizin becerileri, mevcut altyapı ve kaynaklar, performans ve kullanıcı deneyimi gereksinimleri gibi faktörleri dikkate almalısınız. Her bir çapraz platform destekli dilin kendi avantajları ve dezavantajları vardır, bu nedenle projenizin özel gereksinimlerine En Popüler şekilde uyacak olanı seçmek için

dikkatli bir değerlendirme gerekmektedir. Sonuç olarak, her proje benzersizdir ve en uygun dil seçimi, proje gereksinimlerinize bağlı olacaktır (Yilmaz & Üstün, 2021).

Performans ve Optimizasyon

Performans, uygulamanın hızını, verimliliğini, tüketimini ve kararlılığını ifade eder. Performans, uygulamanın kullanıcı memnuniyetini, pazar payını ve gelirini etkiler. Performansı etkileyen faktörler arasında, kod kalitesi, algoritma seçimi, veri yapısı seçimi, bellek yönetimi, ağ bağlantısı, donanım özellikleri, platform özellikleri sayılabilir. Performansı ölçmek için, performans testleri, profillemeler, analizler, izlemeler, raporlamalar gibi yöntemler kullanılabilir. Optimizasyon, uygulamanın performansını artırmak için yapılan iyileştirme işlemleridir. Optimizasyon için En Popüler uygulamalar arasında, gereksiz kodlardan, resimlerden, dosyalardan kurtulmak, kodu sadeleştirmek, yorumlamak, yeniden kullanmak, parçalamak, önbelleğe almak, sıkıştırmak, asenkron çalıştırmak, hata yakalamak, güncellemek, güvenliğini sağlamak sayılabilir. Optimizasyon için araçlar arasında, IDE'ler, SDK'lar, API'lar, kütüphaneler, frameworkler, test araçları, analiz araçları, izleme araçları, raporlama araçları sayılabilir.

Çapraz Platform Uygulama Geliştirme için Popüler Frameworkler ve Platformlar

Çapraz platform uygulama geliştirme için popüler frameworkler ve platformlar arasında, Flutter, React Native, Xamarin, Ionic, Cordova, PhoneGap, NativeScript, Unity, Unreal Engine, Corona SDK, Appcelerator, Qt sayılabilir. Bu frameworkler ve platformlar, farklı programlama dilleri, kütüphaneler, araçlar, özellikler, avantajlar ve dezavantajlar sunar. Bu frameworkler ve platformlar, uygulamanın türüne, amacına, hedef kitlesine, iş modeline, bütçesine, zamanına, beklentilerine ve gereksinimlerine göre seçilebilir.

Çapraz Platform Uygulama Geliştirme için En Popüler Uygulama Örnekleri ve İpuçları

Çapraz platform uygulama geliştirme için En Popüler uygulama örnekleri arasında, Facebook, Instagram, Skype, Uber, Airbnb, Netflix, Spotify, Pinterest, Slack, Discord, LinkedIn, Walmart, Tesla, Khan Academy, Duolingo sayılabilir. Bu uygulamalar, çapraz platform uygulama geliştirme frameworklerini ve platformlarını başarıyla kullanan, milyonlarca kullanıcıya sahip, yüksek performanslı, güzel tasarımlı, işlevsel, kullanışlı ve popüler uygulamalardır. Çapraz platform uygulama geliştirme için ipuçları arasında, uygun framework ve platform seçmek, tek kod tabanını iyi yönetmek, yerel özelliklere erişmek için köprüler kullanmak, uygulamayı farklı platformlarda test etmek, uygulamayı optimize etmek, uygulamayı güncel tutmak, uygulamayı izlemek ve raporlamak sayılabilir.

İş Dünyasında Kullanım

Flutter İş Dünyasında Kullanımı

Şirketler, Flutter, iş dünyasında oldukça çeşitli ve popüler şirketler tarafından kullanılmaya başlanmıştır. Özellikle Google, Alibaba, Tencent ve Square gibi büyük teknoloji şirketleri Flutter'ı kullanmaktadır. Google, kendi ürün ve hizmetlerinin birçoğunu Flutter ile geliştirmektedir. Alibaba, e-ticaret platformlarının mobil uygulamaları için Flutter'ı tercih ederken Tencent, oyunlar ve eğlence uygulamaları için Flutter'ı kullanır.

Kullanım Alanları, Flutter, çeşitli sektörlerde kullanılmaktadır. Büyük e-ticaret platformları, özellikle mobil alışveriş uygulamaları ve ödeme sistemleri için Flutter'ı kullanır. Google, Android ve iOS platformları için uygulama geliştirmek için Flutter'ı kullanırken, finansal hizmetler sunan şirketler kullanıcı dostu arayüzler oluşturmak için Flutter'ı tercih edilir.

Kotlin İş Dünyasında Kullanımı

Şirketler, Kotlin, özellikle Android uygulama geliştirmek isteyen şirketler tarafından yaygın olarak kullanılmaktadır. Pinterest, Trello, Uber, Coursera gibi büyük şirketler Kotlin'i Android istemcileri ve sunucu tarafı uygulamaları için kullanmaktadır. Kotlin, özellikle büyük şirketler tarafından takdir edilen güçlü ve esnek bir dil olarak öne çıkmıştır.

Kullanım Alanları, Kotlin, özellikle Android uygulama geliştirmek isteyen şirketler arasında yaygın olarak tercih edilir. Sosyal medya platformları, büyük veri işleme gereksinimlerini hızlı bir şekilde karşılayabilmek için Kotlin'i kullanırken, taşıma hizmetleri sunan şirketler ise sunucu tarafı hizmetlerini geliştirmek için Kotlin'i tercih eder.

React Native İş Dünyasında Kullanımı

Şirketler, React Native, özellikle büyük şirketler tarafından kullanılan bir çerçeve olarak öne çıkar. Facebook, Instagram, Airbnb, Walmart gibi büyük şirketler, React Native'i tercih eden önde gelen şirketlerdir. Instagram, React Native'i kullanarak Android ve iOS için tek bir kod tabanında uygulama geliştirmiştir.

Kullanım Alanları, React Native, çeşitli sektörlerde kullanılmaktadır. Airbnb, çok platformlu mobil uygulamalarını hızlı bir şekilde geliştirmek için React Native'i kullanırken, perakende ve eğitim gibi sektörlerde faaliyet gösteren büyük şirketler, kullanıcı deneyimlerini artırmak ve etkileşimli uygulamalar oluşturmak amacıyla React Native'i kullanmaktadır.

Swift İş Dünyasında Kullanımı,

Şirketler, Swift, özellikle Apple ekosistemine odaklanmış şirketler tarafından tercih edilir. Apple, Airbnb, LinkedIn, Eventbrite gibi şirketler Swift'i kullanmaktadır. Apple, Swift'i iOS, macOS ve diğer Apple platformları için ana programlama dili olarak benimsemiştir.

Kullanım Alanları, Swift, iOS ve macOS uygulamalarını geliştirmek için ideal bir seçenektir. Airbnb gibi şirketler, iOS ve macOS platformlarında mükemmel bir kullanıcı deneyimi sunmak için Swift'i kullanmaktadır. Ayrıca, etkinlik yönetimi platformları, Swift'i etkinlik yönetimi uygulamaları inşa etmek için tercih eder.

Her bir dil ve çerçeve, iş dünyasında farklı kullanım alanlarına sahiptir ve şirketler, projelerinin ihtiyaçlarına uygun olanı seçerler. İşinizin gereksinimlerini ve hedeflerini dikkate alarak doğru dil ve çerçeve seçimi yapmak önemlidir.

Mobil Uygulama Geliştirme için Popüler Kitaplar

- “The Pragmatic Programmer”: Bu kitap, yazılım mühendisliği prensiplerine odaklanarak, genel yazılım geliştirme becerilerinizi geliştirmenize yardımcı olur. Mobil uygulama geliştirme konusunda temel prensipleri anlamak için harika bir kaynaktır.
- "Head First Android Development”: Android uygulama geliştirmeye başlamak isteyenler için özellikle faydalıdır. Hem temel hem de ileri düzey konuları kapsar, bu yüzden Android geliştirme yolculuğunuzda size rehberlik edebilir.
- "iOS Programming-The Big Nerd Ranch Guide”: iOS uygulama geliştirmeye odaklanır ve temel iOS geliştirme konularını ele alır. Bu kitap, iOS uygulama geliştirmeye yeni başlayanlar için önemli bir kaynaktır.
- "Flutter in Action”: Flutter'ı kullanarak hem Android hem de iOS platformları için uygulamalar geliştirmek isteyenler için mükemmel bir kaynaktır. Flutter'ın temellerini öğrenmek için idealdir.
- "React Native in Action”: React Native kullanarak çok platformlu mobil uygulamalar geliştirmek isteyenler için önerilen bir kitaptır. React Native'ı öğrenmek için detaylı bir kılavuz sunar.

Popüler Makaleler

- Medium: Mobil uygulama geliştirmeyle ilgili geniş bir makale koleksiyonu sunar. Medium'da, deneyimli geliştiricilerin yazdığı rehberler ve incelemeleri bulabilirsiniz.
- Smashing Magazine: Bu kaynak, kullanıcı arayüzü (UI) ve kullanıcı deneyimi (UX) tasarımı hakkında önemli makaleler ve kaynaklar sunar. Mobil uygulamaların etkili tasarımı için faydalıdır.
- Android Developers Blog ve Apple Developer Blog: İlgili platformların resmi blogları, en son gelişmeler, güncellemeler ve önerilerle dolu makaleler sunar. Bu kaynaklar, platformların güncellemelerini ve özelliklerini takip etmek için harikadır.

En Popüler Forumlar ve Topluluklar

- Stack Overflow: Sorularınıza yanıtlar bulabileceğiniz ve diğer geliştiricilerle etkileşimde bulunabileceğiniz popüler bir yazılım geliştirme forumudur.
- GitHub: Mobil uygulama geliştirme projelerinizi yönetmek ve kod paylaşımı için mükemmel bir platformdur. Proje belgelerine ve sorunlarına erişim sağlar.
- Reddit: /r/androiddev ve /r/iOSProgramming gibi subreddit'ler, mobil uygulama geliştirme topluluklarına katılmak ve sorularınızı sormak için harika yerlerdir. Ayrıca, güncel haberleri ve projeleri takip etmek için kullanabilirsiniz.
- Slack ve Discord Sunucuları: Mobil uygulama geliştirme toplulukları, Slack ve Discord gibi anlık ileti uygulamalarında özel sohbet sunucuları oluştururlar. Bu sunucular, diğer geliştiricilerle iletişim kurmanız ve sorularınıza yanıt bulmanız için harikadır.

- Meetup: Mobil uygulama geliştirme ile ilgili yerel etkinlikler ve toplantılar hakkında bilgi almak ve diğer geliştiricilerle yüz yüze tanışmak için kullanışlıdır. Yerel geliştirici topluluklarına katılarak deneyiminizi paylaşabilirsiniz.

Popüler IDE'ler ve Editörler

- Android Studio: Android uygulama geliştirme için Google tarafından geliştirilen resmi bir IDE'dir. Android platformuna özgü özellikler ve hızlı emülatör entegrasyonu sunar. Android Studio, Kotlin programlama dilini de destekler.
- Xcode: iOS ve macOS uygulama geliştirme için Apple tarafından sunulan resmi IDE'dir. Xcode, Objective-C ve Swift dillerini destekler ve uygulamaları hızlı bir şekilde oluşturmanızı sağlar.
- Visual Studio Code (VS Code): Hem Android hem de iOS için kullanılabilen ücretsiz ve açık kaynaklı bir kod düzenleyicidir. Geniş bir eklenti ekosistemi ile kullanıcı dostu ve özelleştirilebilirdir. Hem native hem de çapraz platform geliştirme için kullanılabilir.
- Flutter ve Dart Editörleri: Flutter uygulamaları geliştirmek için, Dart programlama dilini kullanan resmi Flutter ve Dart editörlerini kullanabilirsiniz. Bu editörler, Flutter projelerinizi optimize etmek için özel olarak tasarlanmıştır.

Popüler SDK'lar ve API'lar

- Android SDK ve API'lar: Android uygulama geliştirmek için kullanılan resmi yazılım geliştirme kiti (SDK) ve API'lar, Android platformunun tüm özelliklerine erişim sağlar. Bu kütüphane, cihaz özellikleri, veritabanı yönetimi, kullanıcı girişi ve daha fazlasını içerir.
- iOS SDK ve API'lar: iOS uygulama geliştirme için kullanılan resmi SDK ve API'lar, iOS cihazlarının özelliklerine erişim

sağlar. Bu kütüphane, kullanıcı arayüzü, ses ve görüntü işleme, haritalar ve daha fazlasını içerir.

- Firebase: Google tarafından sunulan bulut hizmet platformu olan Firebase, uygulamanızın analitiği, kimlik doğrulama, gerçek zamanlı veritabanı, sunucu işlevleri ve depolama gibi bir dizi hizmeti içerir. Firebase, uygulamanızın gelişimini hızlandırmak için kullanışlıdır.
- React Native: React Native, hem Android hem de iOS için kullanılabilen bir JavaScript çerçevesidir. Native uygulama geliştirmeyle web geliştirme deneyimini birleştirir ve çok platformlu uygulamalar oluşturmanıza olanak tanır.

Mobil Uygulama Geliştirme için En Popüler Kütüphaneler ve Frameworkler

- Flutter: Google tarafından geliştirilen Flutter, tek bir kod tabanıyla hem Android hem de iOS için etkileyici çok platformlu uygulamalar oluşturmanızı sağlar. Zengin bir widget kitaplığına sahiptir ve hızlı bir geliştirme süreci sunar.
- React Native: Facebook'un açık kaynaklı React Native çerçevesi, JavaScript kullanarak çok platformlu uygulamalar geliştirmenize olanak tanır. Kendi modülleri ve topluluk desteği ile güçlüdür.
- Kotlin MultiPlatform: Kotlin Multiplatform projeleri, her iki platformda çalıştırılacak ortak kodu içerirken, aynı zamanda platforma özgü kodu da destekler. Bu, her platformun özel gereksinimlerini karşılamak için gerektiğinde platforma özgü kodu eklemenizi sağlar.
- JQuery Mobile: HTML, CSS ve JavaScript kullanarak mobil web uygulamaları oluşturmanıza yardımcı olur. Basit ve kullanışlı bir çerçeve olarak bilinir.
- Ionic Framework: Ionic, web teknolojilerini kullanarak Android ve iOS için hızlı ve kullanıcı dostu mobil uygulamalar

geliştirmenizi sağlayan bir çerçevedir. Angular, React veya Vue.js gibi popüler çerçeveleri destekler.

- UIKit: UIKit, Apple'ın iOS, iPadOS, macOS, watchOS ve tvOS gibi işletim sistemlerinde kullanılan bir kullanıcı arayüzü (UI) framework'üdür. UIKit, grafiksel kullanıcı arayüzü oluşturmak, olayları yönetmek ve kullanıcı etkileşimlerini işlemek için geniş bir set sunar.

Mobil Uygulama Geliştirme için Popüler Test, Analiz ve İzleme Araçları

- Firebase Test Lab: Android uygulamalarını farklı cihazlar ve OS sürümlerinde test etmek için kullanılır. Ayrıca, performans ve uyumluluk testleri yapmanızı sağlar.
- App Center Test Cloud: Microsoft'un sağladığı hizmet, uygulamanızı farklı cihazlarda ve platformlarda test etmek için kullanışlıdır.
- Crashlytics: Firebase tarafından sunulan bir hata izleme ve analiz aracıdır. Uygulamanızdaki hataları izler, raporlar ve analiz eder.
- Google Analytics for Mobile: Uygulama analitiği için kullanılan popüler bir araçtır. Kullanıcı davranışını, dönüşümleri ve diğer analizleri izlemek için kullanılır.
- Mobil Uygulama Geliştirme için Popüler Tasarım, Prototipleme ve Mockup Araçları
- Sketch: Özellikle iOS uygulamaları için tasarım yapmak için kullanılan vektör tabanlı bir tasarım aracıdır. Özelleştirilebilir bileşenler, simgeler ve artboards içerir.
- Figma: Çevrimiçi bir tasarım aracıdır ve çoklu kullanıcı iş birliği, prototipler oluşturma ve paylaşma özelliklerine sahiptir. Herhangi bir platformda çalışır.

- Adobe XD: Adobe tarafından sunulan bir prototipleme ve tasarım aracıdır. Basit ve kullanıcı dostu bir arayüze sahiptir. Yaratıcı bulma, prototipler ve iş birliği özelliklerini içerir.
- InVision: Prototipler oluşturmanıza, kullanıcı testleri yapmanıza ve iş birliği yapmanıza olanak tanır. Tasarım prototiplerini paylaşmanıza ve yorumları toplamanıza yardımcı olur.

Mobil Uygulama Geliştirme için Popüler Uygulama Kodlama Standartları ve Kalite

Uygulama kodlama standartları ve kalite yöntemleri, kodunuzu daha düzenli ve sürdürülebilir hale getirmenize yardımcı olur. Model-View-Controller (MVC) veya Model-View-ViewModel (MVVM) gibi tasarım desenleri kullanarak kodunuzu modülerleştirebilirsiniz. Ayrıca, kod incelemeleri ve test odaklı geliştirme (TDD) gibi uygulamaları hataları erkenden tespit etmek ve kodunuzu iyileştirmek için kullanabilirsiniz.

Uygulama Kodlama, Hata Önleme ve Çözme Yöntemleri

Hata önleme ve çözme, uygulama geliştirme sürecinin önemli bir parçasıdır. Hata izleme araçları, uygulama hatalarını izlemek ve analiz etmek için kullanılır. Firebase Crashlytics veya App Center Crash Reporting gibi araçlar, hataların kaynağını ve sıklığını belirlemenize yardımcı olur. Ayrıca, IDE'nizdeki hata ayıklama araçları ve kod incelemeleri hataları bulmanıza ve çözmenize yardımcı olur.

Uygulama Kodlama Örnekleri ve Kaynak Kodları

Mobil uygulama geliştirme öğrenirken, gerçek projeleri incelemek ve örnek kodları gözden geçirmek çok önemlidir. GitHub gibi platformlar, açık kaynaklı mobil uygulama projelerini barındırır. Udacity, Coursera ve benzeri eğitim platformları, mobil uygulama geliştirme kursları ve örnek projeler sunar. Resmi dökümantasyonlar, Google Codelabs ve Apple Developer

Dökümantasyonu gibi kaynaklar, platform özellikleri ve en popüler uygulamalar hakkında bilgi edinmenize yardımcı olabilir. Stack Overflow ve Reddit gibi topluluklar, diğer geliştiricilerle etkileşimde bulunmanıza ve gerçek dünya uygulama kodlarına dair deneyimler ve öneriler elde etmenize olanak tanır. Video öğreticiler ve yazılı rehberler, görsel öğrenmeyi tercih edenler için uygulama kodlama teknikleri hakkında daha fazla içerik sunar.

KAYNAKÇA

Ansari, M. A. (2017). Swift or Objective-C-Which One is Better? Içinde *IJSTE-International Journal of Science Technology & Engineering* | (C. 3, Sayı 10). www.ijste.org

Azizah, A. H., Faidah, S. Z., Ulum, M. B., & Handayani, P. (2021). Exploration of React Native Framework in designing a Rule-Based Application for healthy lifestyle education. *Proceedings of 2021 1st International Conference on Computer Science and Artificial Intelligence, ICCSAI 2021*, 391-394. https://doi.org/10.1109/ICCSAI53272.2021.9609763

Boukhary, S., & Colmenares, E. (2019). A clean approach to flutter development through the flutter clean architecture package. *Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019*, 1115-1120. https://doi.org/10.1109/CSCI49370.2019.00211

Brito, H., Santos, Á., Bernardino, J., & Gomes, A. (2019). Mobile development in Swift, Java and React Native: an experimental evaluation in audioguides. *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6. https://doi.org/10.23919/CISTI.2019.8760864

Dart overview | Dart. (t.y.). Geliş tarihi 05 Aralık 2023, gönderen https://dart.dev/overview

ELİBOL, M., & SELÇUKCAN EROL, Ç. (2017). Scrum Metodu Kullanılarak Bir Mobil Uygulama Geliştirme Sürecinin Gerçekleştirilmesi. *Bilişim Teknolojileri Dergisi*, 169-169. https://doi.org/10.17671/gazibtd.309299

GÜLCÜOĞLU, E., USTUN, A. B., & SEYHAN, N. (2021). Comparison of Flutter and React Native Platforms. *Journal of Internet Applications and Management*, *12*(2), 129-143. https://doi.org/10.34231/iuyd.888243

Hassan, A. M. (2019). JAVA and DART programming languages: Conceptual comparison. *Indonesian Journal of Electrical*

Engineering and Computer Science, *17*(2), 845-849. https://doi.org/10.11591/ijeecs.v17.i2.pp845-849

Kadrija, S., Memeti, A., & Luma-Osmani, S. (2022). Development of mobile app through React Native hybrid framework. *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 1-6. https://doi.org/10.1109/MECO55406.2022.9797173

KESKİN, Yrd. Doç. Dr. N. Ö., & KILINÇ, Araş. Gör. H. (2015). Mobil öğrenme uygulamalarına yönelik geliştirme platformlarının karşılaştırılması ve örnek uygulamalar. *Açıköğretim Uygulamaları ve Araştırmaları Dergisi*, *1*(3), 68-90.

Kishore, K., Khare, S., Uniyal, V., & Verma, S. (2022). Performance and stability Comparison of React and Flutter: Cross-platform Application Development. *International Conference on Cyber Resilience, ICCR 2022*. https://doi.org/10.1109/ICCR56254.2022.9996039

Kotlin for Android Developers. (2015). http://leanpub.com/kotlin-for-android-developers

Lohani, D. (2022). *Taking Flutter to the Web: Learn How to Build Cross-Platform UIs for Web and Mobile Platforms Using Flutter for Web*. Packt Publishing, Limited.

Martinez, M., & Gois Mateus, B. (2022). Why Did Developers Migrate Android Applications From Java to Kotlin? *IEEE Transactions on Software Engineering*, *48*(11), 4521-4534. https://doi.org/10.1109/TSE.2021.3120367

Nunes, R., Reboucas, M., Soares-Neto, F., & Castor, F. (2017). Visualizing swift projects as cities. *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017*, 368-370. https://doi.org/10.1109/ICSE-C.2017.115

Schneider, L., & Schultes, D. (2022). Evaluating Swift-to-Kotlin and Kotlin-to-Swift Transpilers. *Proceedings - 9th*

IEEE/ACM International Conference on Mobile Software Engineering and Systems, MOBILESoft 2022, 102-106. https://doi.org/10.1145/3524613.3527811

Sharma, S., Khare, S., Unival, V., & Verma, S. (2022). Hybrid Development in Flutter and its Widgits. *2022 International Conference on Cyber Resilience (ICCR)*, 1-4. https://doi.org/10.1109/ICCR56254.2022.9995973

Swift history | Swift. (t.y.). Geliş tarihi 05 Aralık 2023, gönderen https://www.swift.com/about-us/history

Syaifudin, Y. W., Hatjrianto, A. S., Funabiki, N., Liliana, D. Y., Kaswar, A. B., & Nurhasan, U. (2022). An Implementation of Automatic Dart Code Verification for Mobile Application Programming Learning Assistance System Using Flutter. *Proceedings - IEIT 2022: 2022 International Conference on Electrical and Information Technology*, 322-326. https://doi.org/10.1109/IEIT56384.2022.9967902

Tunalı, V., Zafer Erdogan, S., Volkan TUNALI, A., & Şenol Zafer ERDOĞAN, A. (2015). *Comparison of Popular Cross-Platform Mobile Application Development Tools*. https://www.researchgate.net/publication/282816272

Yilmaz, Ö., & Üstün, A. B. (2021). App Inventor ve Alternatif Blok Tabanlı Mobil Uygulama Geliştirme Platformlarının Karşılaştırmalı İncelenmesi Comparative Review of App Inventor and Alternative Block Based Mobile Application Development Platforms. *Disiplinlerarası Eğitim Araştırmaları Dergisi Journal of Interdisciplinary Educational Research*, *5*(9), 1-11.

BÖLÜM XI

Nesnelerin İnternetinde Güvenlik Tehditleri ve Korunma Stratejileri

Abdullah Erhan AKKAYA[1]

1. IoT AĞ GÜVENLİĞİNE GİRİŞ

İnternete bağlı cihazların sayısı gün geçtikçe artmaktadır. IHS Markit raporuna göre dünya çapında internete bağlı cihazların sayısının 2017 yılında 27 milyara, 2030 yılında ise yıllık ortalama %12'lik bir artış ile 125 milyara ulaşacağı öngörülmektedir (Otoum et al., 2022). Nesnelerin İnterneti (Internet of Things-IoT) olarak adlandırılan bu ağ, ev aletlerinden endüstriyel cihazlara kadar çeşitli nesnelerin internet üzerinden birbirleriyle bağlantı kurmasını sağlamaktadır. IoT ağları sayesinde bu cihazlar, verimli bir veri alışverişi ve iş birliği gerçekleştirerek kullanıcı deneyimini iyileştirmekte ve yeni hizmetler sunabilmektedir. Örneğin, akıllı ev sistemleri, kullanıcıların uzaktan ısıtma, aydınlatma ve güvenlik

[1] Dr. Öğr. Üyesi, İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Orcid: 0000-0001-6193-5166.

kontrolleri yapmasına olanak tanımaktadır. Endüstriyel IoT ise makineler arası verimliliği artırmakta, arıza tespiti ve önleyici bakım gibi uygulamaları mümkün kılmaktadır (Zhong et al., 2017).

Cihaz sayısındaki ani büyüme, özellikle güvenlik açısından birtakım riskleri de beraberinde getirmektedir. Artan cihaz sayısıyla birlikte, kişisel verilerin gizliliği, siber saldırılar ve diğer güvenlik tehditleri daha büyük bir endişe kaynağı halini almıştır. İnternete bağlı cihazlar tarafından toplanan veriler, kişisel veya iş ile ilgili hassas bilgiler içerebileceğinden, siber suçlular için her zaman ilgi çekici olmuştur. Veri gizliliğinin korunması adına IoT cihazlarını ve ağlarını korumak için güçlü güvenlik önlemlerinin alınması büyük önem taşımaktadır. Verilerin güvenli bir şekilde saklanması ve iletilmesi, yazılımların düzenli olarak güncellenmesi ve sürekli bir güvenlik izleme sisteminin oluşturulması gerekmektedir. IoT'nin getirdiği faydalar yanında, güvenlik ve gizlilik riskleri de söz konusudur. Ancak, güvenli haberleşme protokolleri ve yapay zekâ gibi çözümler ile bu risklerin azaltılabileceği görülmektedir.

1.1. IoT Ağ Güvenliği

IoT ağ güvenliği, internete bağlı cihazların donanımını, yazılımını, verilerini ve iletişim altyapısını yetkisiz erişimlere, siber saldırılara ve diğer tehditlere karşı korumak için uygulanan siber güvenliğe ait protokolleri, politikaları ve önlemleri kapsamaktadır (Abomhara & M. Køien, 2015). Sürecin temel amacı, şifreleme, erişim kontrolü, tehdit izleme ve kullanıcı doğrulama gibi çeşitli araçlar kullanarak IoT ağlarının gizliliğini, bütünlüğünü ve sürekli erişilebilirliğini sağlamaktır (Samaila et al., 2018). IoT ağ güvenliği, evlerdeki akıllı IoT cihazlardan fabrikalardaki otomasyon sistemlerine kadar her türlü IoT cihazının korunmasını kapsamaktadır. Veri şifreleme, verilerin güvenli bir şekilde saklanmasını ve aktarılmasını sağlamaktadır, böylece yalnızca yetkili kişilerin erişebileceği bir ortam oluşturulur (Jorge Granjal et al., 2015). Erişim kontrolü, ağa bağlanacak cihazları sınırlayarak yabancı cihazların ağa istediği şekilde girmesini engelleyerek ağ güvenliğini artırmaktadır (Sicari et al., 2016). Tehdit izleme, sürekli

olarak ağı gözlemleyerek herhangi bir şüpheli aktiviteyi tespit ederek duruma hızlı bir şekilde müdahale edebilmektedir (Miettinen et al., 2017). Kullanıcı doğrulaması ise, ağa erişim sağlamak isteyen herkesin kimliklerini doğrulayarak ek bir güvenlik katmanı oluşturmaktadır (Ferrag et al., 2018).

IoT cihazlarının ve ağlarının günlük yaşantımızın ve iş süreçlerimizin ayrılmaz bir parçası haline geldiği bu dönemde, bahsedilen güvenlik önlemleri, gizliliğimizi ve veri güvenliğimizi korumak için hayati öneme sahiptir. Her geçen gün artan IoT cihazlarına karşı proaktif bir güvenlik yaklaşımı benimsemek, siber tehditlerin ve güvenlik açıklarının önüne geçilmesinde kritik bir rol oynamaktadır.

Bilişim teknolojileri (BT) ağları, bilgisayarlar, sunucular, anahtarlar, yönlendiriciler ve diğer bilişim teknolojisi donanımlarını içeren, veri ve bilgi alışverişini sağlayan dijital ağlar anlamına gelmektedir. BT ağları, ofislerde, kurumsal ortamlarda, kamu sektöründe ve hatta evlerde yaygın olarak kullanılmaktadır. Bu ağlar, internete bağlanmak, veri paylaşmak, ağ üzerinden iletişim kurmak ve çeşitli dijital işlemleri gerçekleştirmek için temel bir altyapı sağlamaktadır. Özellikle, Geleneksel BT ağlarına göre IoT ağ güvenliğini sağlamak daha zorlayıcıdır (Sicari et al., 2016). Bunun nedenleri arasında cihazların kaynak kısıtlamaları, ağların geniş ve dinamik ölçeği ve kullanılan çeşitli özel ve eski protokoller sayılabilir.

Kullanıcıların mahremiyetini korumak ve veri ihlallerini önlemek; IoT düğümleri, ağlar ve bulut arasındaki iletişimi güvence altına almak; yazılım güvenliğini güncellemeler ve zafiyetleri gidermek yoluyla sağlamak; malware ve virüsler gibi siber tehditlere karşı nesnelerin ağına dayanıklılık kazandırmak; saldırıları ve sistemdeki anomalileri hızlı bir şekilde tespit ederek yanıt vermek; hacking, veri hırsızlığı gibi güvenlik olaylarından hızla toparlanmak, IoT ağ güvenliğinin temel amaçları arasındadır.

1.2.IoT Ağ Güvenliğinin Önemi

Milyarlarca sensör, cihaz ve sistemin birbirine bağlandığı bir dünyada, IoT güvenliği hayati öneme sahiptir. Bu cihazlar arasında sağlık izleme cihazları, ev güvenlik sistemleri, endüstriyel kontrol sistemleri ve daha pek çoğu bulunmaktadır. Kullanıcıların sağlık metriklerinin yetkisiz kişilerin eline geçmesi kişisel sağlık durumları hakkında hassas bilgiler içerebilir (Kumar, 2023). Güvenlik kamerası görüntülerine ulaşılarak (Khan & Salah, 2018) kişilerle ile ilgili özel bilgiler ifşa edilebilir. Termik santral veya nükleer santral gibi kritik altyapılara sahip enerji şebekelerine yönelik saldırılar hayati tehlikelere varan olaylarla sonuçlanabilir. Veri ihlalleri nedeniyle büyük finansal kayıplar ortaya çıkabilir (Aranuwa et al., 2022). Analitik veya yapay zekâ modelleri için kullanılan IoT verilerinin doğruluğu ve bütünlüğünün bozulması sonucunda sistemler hatalı sonuçlar üretebilir. Verilen örneklerden de görüldüğü üzere IoT ağlarının güvenliğini ihmal etmek, akıllı sistemlerin gizliliğini, güvenliğini ve güvenilirliğini ciddi şekilde tehlikeye atabilmektedir.

1.3. IoT Güvenliğindeki Zorluklar ve Temel Problemler

IoT ağları için sağlam bir güvenlik sistemi oluşturmanın önünde bir dizi teknolojik zorluklar bulunmaktadır. IoT cihazlarının düşük hafıza, sınırlı işlem gücü ve kısıtlı pil ömrü gibi özellikleri, karmaşık güvenlik algoritmalarının bu cihazlarda etkin bir şekilde çalışmasını zorlaştırmaktadır (Alrubayyi et al., 2023; Alrubayyi et al., 2021; Ojo et al., 2018). Bu cihazlar, genellikle basit işlemler için tasarlanmıştır ve bu nedenle, daha fazla kaynak gerektiren gelişmiş güvenlik protokollerini desteklemekte yetersiz kalabilirler. Bu durum, IoT cihazlarının güvenliğini sağlamak için daha az kaynağa ihtiyaç duyan güvenlik çözümlerinin geliştirilmesini gerektirir. Bu tür çözümler hem cihazların sınırlı kapasitelerini göz önünde bulundurarak tasarlanmalı hem de etkili bir güvenlik seviyesi sağlamalıdır. IoT cihazları, ağ geçitleri, ağlar ve bulut arasında uçtan uca veri şifrelemesi sağlandığı takdirde güvenli veri iletişimi gerçekleştirebilirler. IoT cihazlarının ve ağlarının karmaşık yapısı ve

sürekli değişen doğası, IoT cihazlarının çeşitliliği ve sayısının artması, ağ üzerindeki veri trafiğini ve potansiyel güvenlik tehditlerini artırmaktadır.

Farklı IoT platformları ve teknolojileri arasında ortak standartların olmaması (Ekpenyong et al., 2022), IoT ağları için etkili bir güvenlik sistemi geliştirmeyi zorlaştıran önemli bir teknik engeldir. Bu durum, çeşitli üreticiler tarafından geliştirilen cihaz ve sistemlerin farklı protokoller ve teknolojiler kullanmasından kaynaklanmaktadır. Bu çeşitlilik, cihazların birbirleriyle ve merkezi sistemlerle uyum içinde iletişim kurmasını güçleştirmektedir. Standart protokollerin eksikliği, güvenlik zafiyetlerine de yol açabilmektedir. Bu nedenle, farklı cihaz ve sistemler arasında uyumluluğu ve güvenliği artıracak şekilde standartlaştırılmış protokollerin oluşturulması büyük önem taşımaktadır. Bu durum IoT üreticileri ve sektör grupları arasında iş birliği ve koordinasyon gerektiren bir konudur.

Geniş ve dinamik yapıdaki IoT ağlarında tehditleri hızlı bir şekilde tespit etmek ve buna etkin bir şekilde yanıt vermek, IoT ağları için güçlü bir güvenlik sistemi geliştirmenin önündeki zorluklardan bir diğeridir. Bu tür zorluklar, IoT ağlarının karmaşık yapısından ve sürekli değişen doğasından kaynaklanmaktadır. IoT cihazlarının çeşitliliği ve sayısının artması, ağ üzerindeki veri trafiğini ve potansiyel güvenlik tehditlerini artırdığından güvenlik sistemlerinin sürekli olarak güncellenmesi ve geliştirilmesi gerekmektedir. IoT ağlarının genişlemesi ve dinamik yapısı, siber saldırıların daha karmaşık ve çeşitli hale gelmesine yol açmaktadır (Bhale et al., 2023). Bu durumun gereği olarak tehditleri hızlı ve etkili bir şekilde tespit etmek için gelişmiş analiz araçları ve algoritmaların kullanılması zorunlu hale gelmektedir.

IoT cihaz üreticileri ve satıcıları arasındaki yetersiz güvenlik uzmanlığı (Mazhar et al., 2021), IoT cihazlarının ve ağlarının güvenliğini sağlamak için gerekli bilgi ve becerilerin eksikliğinden kaynaklanmaktadır. Üreticiler ve satıcılar genellikle cihazların işlevselliğine ve performansına odaklanırken, güvenlik genellikle

ikincil planda kalmaktadır. Bu yaklaşım, cihazların ve ağların siber saldırılara karşı savunmasız hale gelmesine yol açmaktadır. IoT cihazlarının ve ağlarının karmaşıklığı ve sürekli gelişen doğası, güvenlik uzmanlarının sürekli olarak yeni tehditler ve savunma stratejileri hakkında bilgi sahibi olmalarını gerektirmektedir. Ancak, bu alandaki uzman eksikliği, cihazların ve ağların güvenliğini sağlamak için gerekli güvenlik önlemlerinin uygulanmasını zorlaştırabilir. Bu durum, IoT cihazlarının ve ağlarının siber saldırılara karşı daha savunmasız hale gelmesine neden olmaktadır.

IoT güvenlik uygulamalarında yaşanan temel sorunlar, genellikle cihaz üreticileri tarafından önemsenmemektedir. Tasarım aşamasında, güvenlikle ilgili maliyetlerin bütçeye göre fazla olduğu ve pazarda birinci olmak için yapılan yarışta güvenlik özelliklerinin geride kaldığı düşünülmektedir (Classen, 2020). Ürün piyasaya sürüldükten sonra yazılım yoluyla birçok sorun çözülebilir olsa da bağlantılı bir cihazın genel ekosistemini ve güvenlik varsayımlarını daha sonradan değiştirmek oldukça zor bir işlem haline gelebilir.

IoT satıcıları arasında güvenlik güncellemeleri ve düzeltmeleri sağlama konusunda yeterli teşviklerin bulunmaması, IoT ağlarının güvenliğini sağlamak için önemli bir zorluktur. Bu durum, IoT cihazlarının ve ağlarının güncel tehditlere karşı savunmasız kalmasına yol açabilmektedir. IoT cihazlarının sürekli gelişen siber tehditlere karşı korunması için düzenli güvenlik güncellemeleri ve düzeltmeleri hayati önem taşımaktadır. Ancak, bazı durumlarda, IoT cihaz üreticileri ve satıcıları, maliyet ve kaynak kısıtlamaları nedeniyle bu güncellemeleri zamanında sağlamakta yetersiz kalabilirler. Bu eksiklik, cihazların güvenlik açıklarını kapatamayacağından, cihazları siber saldırılara karşı daha savunmasız hale getirebilir. Ayrıca, IoT cihazlarının uzun ömürlü olması ve sürekli olarak güncellenmesi gerektiği gerçeği, satıcıların bu güncellemeleri sağlama konusundaki isteksizliğini daha da problematik hale getirir. Bu durum, kullanıcıların ve ağların güvenliğini riske atabilmektedir.

ZigBee, Bluetooth, WiFi gibi popüler IoT iletişim protokollerinin de güvenlik açıklarını artıran belirli güvenlik sınırlamaları vardır (Garcia-Morchon & Wehrle, 2010). Bu protokollerin her biri, IoT cihazlarının ve ağlarının güvenliğini sağlamak için özel dikkat ve güvenlik önlemleri gerektirmektedir. ZigBee, düşük güç tüketimi ve uzun batarya ömrü ile bilinmesine rağmen güvenlik açısından bazı zayıflıklara sahiptir. ZigBee ağları sık sık zayıf şifreleme ve anahtar yönetimi sorunlarından muzdariptir (Allakany et al., 2023). Ayrıca, ZigBee cihazları arasındaki iletişim sırasında ortaya çıkabilecek güvenlik açıkları, siber saldırılara yol açabilmektedir. Bluetooth, özellikle eski sürümlerde, güvenlik açıklarına karşı savunmasız olabilmektedir. Bluetooth bağlantıları, verilerin güvenli bir şekilde aktarılması için şifreleme kullanır. Ancak, eski Bluetooth sürümlerinde kullanılan şifreleme algoritmaları, modern standartlara göre daha zayıf olabilir. Ayrıca veri aktarımı sırasında kullanılan şifreleme yöntemlerinin kırılabilir olması, güvenlik risklerini artırmaktadır (Tschirschnitz et al., 2021). Eski versiyona sahip Bluetooth cihazları, güvenlik açıklarını gidermek için gereken yazılım güncellemelerini almayabilir. Bu, cihazları yeni türdeki saldırılara karşı savunmasız bırakır.

2. BAŞLICA IoT AĞ GÜVENLİĞİ TEHDİTLERİ VE SALDIRILARI

İnternet Nesneleri (IoT) cihazlarının ve sistemlerinin hızlı yayılması, siber suçluların istismar edebileceği saldırı yüzeylerini de genişletmiştir. IoT ağları, kullanıcı mahremiyetini ciddi şekilde tehlikeye atabilecek, kritik sistemleri tehlikeye sokabilecek ve geniş çaplı siber saldırılara yol açabilecek çok çeşitli güvenlik tehditleri ve zafiyetleri ile karşı karşıya kalmaktadır. Bu durum, akıllı ev aletlerinden sağlık durumu takip sistemlerine, iş yerlerindeki sunuculardan üniversite evrak otomasyon sistemlerine kadar hayatımızın birçok alanında yer alan IoT cihazlarının güvenliğini sağlamanın ne kadar önemli olduğunu gözler önüne seriyor. IoT cihazlar günlük yaşamımızın bir parçası haline geldikçe, bu cihazları siber suçluların saldırılarından korumak için daha fazla çaba sarf etmemiz gerekiyor. Kullanıcı verilerinin gizliliğini korumak, kritik

altyapıları güvende tutmak ve olası büyük siber saldırıları engellemek için IoT ağlarının güvenliğine öncelik vermek zorundayız.

IoT cihazlarının yaygınlaşmasıyla birlikte, güvenlik zafiyetlerini azaltmak ve siber tehditlere karşı dayanıklılık oluşturmak için teknoloji üreticileri, hizmet sağlayıcılar ve son kullanıcıların bir araya gelerek toplu bir çaba göstermesi gerekmektedir.

2.1.Kötü Amaçlı Yazılım (Malware) Tehditleri

Zararlı yazılımlar (Malware), bilgisayar sistemlerine ve ağlarına enfekte olmak, zarar vermek veya yetkisiz erişim sağlamak amacıyla tasarlanmış kötü niyetli yazılım programlarını ifade etmektedir. Özellikle fidye yazılımı saldırıları büyük bir endişe kaynağıdır. Fidye yazılımları, sistem dosyalarını ve sistem verilerini şifreleyerek yetkili kullanıcıların erişimini engeller. Genellikle şifre çözme anahtarını elde etmek için saldırgan fidye talep etmektedir.

Virüsler, solucanlar, casus yazılımlar ve Truva atları; dosya transferleri, güncellenmemiş güvenlik açıkları ve enfekte olan çıkarılabilir medya aygıtları yoluyla hızla yayılabilen diğer zararlı yazılım çeşitleridir. Bu tür zararlı yazılımlar:

- Kullanıcıların kişisel ve hassas verilerini çalabilir. Çalınan veriler finansal bilgiler, şifreler, kişisel kimlik bilgileri ve diğer önemli veriler olabilir.
- Önemli dosyaları silerek veya bozarak veri kaybına neden olabilir. Bu durum, özellikle yedekleme yapılmamışsa, geri dönüşü olmayan zararlara yol açabilir.
- Cihazların normal işleyişini bozarak onları kullanılamaz hale getirebilir. Bu durum, sistem kaynaklarını aşırı kullanma, işletim sistemi bileşenlerini bozma veya cihazın tamamen çökmesine neden olabilir.

- Enfekte cihazları spam veya istenmeyen e-posta göndermek için kullanabilir. Bu durum, kullanıcının bilgisi dışında gerçekleşebilir ve cihazın veya kullanıcının itibarını zedeleyebilir.
- Enfekte cihazları diğer cihazlara saldırmak veya daha fazla zararlı yazılım yaymak için bir platform olarak kullanabilir. Bu durum, ağ güvenliğini tehlikeye atar ve geniş çaplı siber saldırılara yol açabilir.

Zararlı yazılımlara karşı korunmak için, güncel antivirüs yazılımları kullanmak, düzenli güvenlik güncellemeleri yapmak, güçlü şifreler kullanmak ve şüpheli e-posta eklerini veya bağlantılarını açmamak önemlidir. Ayrıca, kullanıcıların ve ağ yöneticilerinin zararlı yazılımların yayılma yöntemleri ve bunlara karşı alınabilecek önlemler konusunda bilinçlendirilmesi ve eğitilmesi de büyük önem taşımaktadır.

Gelişmiş sürekli tehditler (Advanced Persistent Threats, APT) ise, siber casusluk ve veri sızdırma amacıyla BT sistemlerinde uzun süreli bir yer edinmeyi hedefleyen gizli zararlı yazılımları kapsar. APT'ler, aylar boyunca operasyonları gizlice izledikten sonra zarar vererek, endüstriyel IoT ve kritik altyapılar için büyük bir tehdit oluşturur.

IoT zararlı yazılımları, sağlam güvenlik yeteneklerinden yoksun birçok eski cihazı ve kısıtlı uç noktaları istismar edebilir. Saldırganlar sıklıkla IoT ağlarına ilk erişim sağlamak için phishing, sosyal mühendislik ve kimlik bilgisi doldurma gibi teknikler kullanır ve daha sonra zararlı yazılım yükler.

2.2.Ağ ve İletişim Tabanlı Saldırılar

Hacking, siber suçluların bilgisayar sistemlerine, ağlara veya cihazlara yetkisiz erişim elde etmek için kötü niyetli kod enjekte etme, kimlik bilgilerini tehlikeye atma ve güvenlik açıklarından yararlanma gibi teknik çabaları ifade eder. IoT ekosistemleri, birkaç yaygın hacking tehdidi ile karşı karşıyadır.

2.2.1. Dinleme (Eavesdropping) Saldırıları

Eavesdropping saldırıları, IoT cihazlarının ve ağlarının güvenliğinde ciddi bir tehdit oluşturur. Bu saldırılar, veri aktarımı sırasında gerçekleşir ve genellikle şifrelenmemiş veya zayıf şifrelenmiş ağ trafiğini hedef almaktadır (Bout et al., 2022). IoT cihazları genellikle sürekli veri aktarımı yapar ve bu da onları dinleme saldırılarına karşı savunmasız hale getirir. Birçok IoT cihazı, özellikle tüketici sınıfı ürünler, zayıf şifreleme veya hiç şifreleme kullanmayabilir, bu da verilerin kolayca ele geçirilmesine yol açar (Qiu et al., 2021).

Eavesdropping saldırılarının etkileri arasında gizlilik ihlali, veri manipülasyonu ve güven kaybı bulunmaktadır. Kullanıcıların kişisel ve hassas bilgileri ele geçirilebilir ve ele geçirilen veriler yanıltıcı veya zararlı amaçlar için kullanılabilir(Singh et al., 2021). Bu tür saldırılara karşı korunmak için güçlü şifreleme, güvenli ağ kullanımı ve sürekli güvenlik güncellemeleri önemlidir (Zhou et al., 2022).

2.2.2. Ortadaki Adam Saldırıları

Ortadaki adam (Man-in-the-Middle, MitM) saldırısı, bir saldırganın iki taraf arasındaki iletişimi dinleyip gelen ve giden verileri değiştirdiği bir durumdur. Bu, özellikle IoT cihazları ve sunucular arasında gerçekleşmektedir. IoT cihazları sürekli veri aktarımı yaptığı için, bu onları MitM saldırılarına karşı daha savunmasız hale getirir (Bhardwaj et al., 2023). Saldırganlar, cihaz ve ağ arasına girerek veri paketlerini ele geçirebilirler. Veri paketlerinin ele geçirilmesi özellikle şifrelenmemiş veya zayıf şifrelenmiş ağlarda daha kolay olmaktadır (Yang et al., 2023). IoT cihazlarının sürekli veri aktarması, onları MitM saldırılarına karşı hassas kılmaktadır (Kollipara et al., 2023). Ayrıca, birçok IoT cihazı, özellikle tüketici sınıfı ürünler, zayıf şifreleme kullanabilir veya eski güvenlik protokolleriyle çalışabilir, bu durum ayrıca güvenlik risklerini artırır (Li et al., 2022).

MitM saldırıları sonucunda, kullanıcı adları, şifreler ve finansal bilgiler gibi hassas bilgiler, ele geçirilebilir (Yang et al., 2023). Saldırganlar, iletişimi manipüle ederek yanıltıcı bilgiler gönderebilir veya alabilir(Kollipara et al., 2023). Ayrıca kullanıcılara ait cihazların gizliliği tehlikeye gireceğinden ve cihaz güvenliği tamamen zayıflayabilir (Bhardwaj et al., 2023).

2.2.3. Dağıtık Hizmet Reddi (Distributed Deniel of Service) Saldırıları

Dağıtık Hizmet Reddi (DDos) saldırılarında hedeflenen sunucular veya ağlar, çok sayıda istekle boğularak işlevsiz hale getirilir. IoT cihazları, genellikle zayıf güvenlik önlemleri nedeniyle DDoS saldırılarında *zombi host* olarak kullanılabilir. Mirai botnet'in 600,000'den fazla cihazı ele geçirerek gerçekleştirdiği DDoS saldırıları, IoT zararlı yazılımlarının yıkıcı potansiyelini açıkça göstermiştir (De Donno et al., 2018). Mirai, genellikle zayıf şifrelerle korunan IoT cihazlarını hedef alarak bu cihazları kontrol altına almış ve büyük ölçekli DDoS saldırıları için kullanmıştır. Bu saldırılar, hedeflenen sistemlerin ve hizmetlerin felç olmasına neden olmuş ve geniş çaplı kesintilere yol açmıştır. Mirai'nin etkisi, IoT cihazlarının güvenliğinin ne kadar önemli olduğunu ve bu cihazların güvenlik açıklarının nasıl ciddi sonuçlar doğurabileceğini göstermiştir. Mirai Etkisi olarak adlandırılabilecek olay, IoT cihazlarında güvenliğin artırılması adına birtakım dersler içermektedir. Öncelikle IoT cihazlarının güçlü şifreler ve iki faktörlü kimlik doğrulama gibi güçlü güvenlik önlemleriyle korunması gerekmektedir. IoT cihazların düzenli olarak güvenlik güncellemeleri alması ve bilinen güvenlik açıklarının hızlı bir şekilde giderilmesi ayrıca önemlidir. IoT cihazlarının bağlı olduğu ağların güvenliğinin sağlanması ve cihazların yetkisiz erişimlere karşı korunması gerekmektedir. Kullanıcıların ve ağ yöneticilerinin IoT güvenliği konusunda bilinçlendirilmesi ve eğitilmesi, bu tür tehditlere karşı farkındalığın artmasında ve korunmada kritik öneme sahiptir.

2.2.4. Yanıltıcı Bilgi (Spoofing) Saldırıları

Siber güvenlikte önemli bir tehdit olan sahtecilik (spoofing) saldırıları, son yıllarda giderek yaygınlaşmaktadır. Sahtecilik saldırılarında saldırgan, bir ağa veya sisteme meşru bir kullanıcıymış gibi davranarak erişim sağlamaya çalışır. IP spoofing, e-posta spoofing ve web sitesi spoofing en yaygın sahtecilik saldırı türleri arasındadır. IP spoofing'de saldırgan, kendi gerçek IP adresini ağda dolaşan paketlerde gizleyerek farklı bir adresten geliyormuş gibi davranır. E-posta spoofing ise bir e-posta adresinin sahte bir e-posta gibi görünmesidir; böylece alıcı, e-postanın güvenilir bir kaynaktan geldiğini düşünerek içeriğine güvenir. Web sitesi spoofing'de ise saldırgan, hedef web sitesine çok benzer sahte bir web sitesi oluşturarak kullanıcıları tuzağa düşürür (Ahmed et al., 2023).

Sahtecilik saldırılarının temel amacı, yetkisiz erişim ve veri hırsızlığıdır. Saldırgan, ele geçirdiği bilgileri finansal dolandırıcılık, kimlik hırsızlığı gibi suçlarda kullanabilir. Ayrıca bu saldırılar, ileri düzey siber suçların habercisi de olmaktadır. Sahtecilik saldırılarına karşı önlem almak için ağ trafiğinin ve erişim loglarının düzenli olarak izlenmesi, şifreleme ve çok faktörlü kimlik doğrulamanın (2FA) kullanılması önerilmektedir (Meng et al., 2023). Bununla birlikte, insan faktörü de göz ardı edilmemelidir. Çalışanların farkındalığının artırılması ve düzenli eğitimler ile sahtecilik saldırılarının önüne geçilebilir. Unutulmamalıdır ki teknoloji ne kadar gelişmiş olursa olsun, insan hataları siber güvenliğin en zayıf halkası olmaya devam edecektir.

2.2.5. Side-Channel Saldırıları

IoT ağlarında side-channel saldırıları, cihazların yan kanallarından bilgi sızdırarak güvenlik açıklarını ortaya çıkaran saldırılardır. Bu saldırılar, cihazların güç tüketimi, elektromanyetik emisyonlar veya işlemci zamanlamaları gibi fiziksel özelliklerini analiz ederek hassas bilgilere ulaşmayı hedefler.

IoT cihazlarının yan kanal güç verilerinin siber güvenlikte, özellikle de saldırı tespitinde kullanılması mümkündür. Raspberry Pi

3 model B ve DragonBoard 410c gibi iki popüler IoT cihazı üzerinde normal koşullar altında ve saldırı altında yan kanal güç davranışları incelendiğinde elde edilen yan kanal güç imzaları, saldırıları tespit etmede kullanılabilmektedir. (Lightbody et al., 2023).Christopher Liptak ve arkadaşlarının 2022'de yaptığı bir başka çalışma, IoT cihazlarının güvenliğinin, özellikle şifreleme uygulamalarının yan kanal saldırılarına karşı savunmasızlığını ele almıştır. Bu çalışma, cihazların güç tüketimini izleyerek kriptografik anahtarları keşfetmeye yönelik yan kanal saldırılarını ve bu saldırılara karşı alınabilecek önlemleri incelemiştir (Liptak et al., 2022). Suvrima Datta ve arkadaşlarının 2022'de gerçekleştirdiği araştırma, IoT ağlarında hacimli saldırıları tespit etmek ve hafifletmek için dağıtılmış, öz öğrenen ve otonom bir sistem olan iDAM'ı önermiştir. Bu sistem, MUD uyumlu IoT cihazlarının davranış profillerini izleyerek ve özel cihaz türleri için OC-SVM modelleri oluşturarak çalışır. Bu çalışma, IoT altyapısının çeşitli seviyelerinde meydana gelebilecek hacimli saldırıları etkili bir şekilde hafifletmeyi amaçlamıştır (Datta et al., 2022).

2.3.Büyük Saldırı Vaka Çalışmalarından Bazıları

Son zamanlarda gerçekleşen büyük ölçekli saldırılar, özellikle Mirai ve Ukrayna enerji şebekesi hacklemesi, IoT güvenlik başarısızlıklarının gerçek dünyada nasıl etkiler yaratabileceğini gözler önüne sermektedir. 2016'nın sonlarında Mirai botnet'i, varsayılan şifreleri kullanan 600,000'den fazla IoT cihazını enfekte ederek, Avrupa ve Kuzey Amerika'daki DNS sunucularını ve web trafiğini bozan büyük DDoS saldırıları başlattı. Bu olay, güvensiz tüketici IoT cihazlarının yıkıcı potansiyelini gösterdi (De Donno et al., 2018).

2015 yılında, Ukrayna enerji şirketlerinin BT sistemlerine sızan saldırganlar, BlackEnergy zararlı yazılımını kullanarak SCADA sistemlerini ele geçirdi. Alt istasyonları devre dışı bırakarak, silici zararlı yazılımla sistemleri çökerttiler ve 200,000'den fazla müşteriye birkaç saat boyunca elektrik verilememesine neden oldular. Bu saldırı, güvensiz endüstriyel

IoT'nin kritik altyapılara karşı gerçek hayattaki tehditlerini ortaya koydu (Lee et al., 2016).

Son zamanlarda, evcil hayvan takip cihazlarındaki güvenlik açıkları konusunda "evcil hayvanlarımızın bizden bilgi çalabileceğine" dair bir uyarı yayınlanmıştır. Bu açıklar, saldırganların konumları sahtelemesine kadar varabilmektedir. Konum sahtelemesi, bir cihazın veya kullanıcının gerçek konumunun yanıltıcı bir şekilde değiştirilmesi veya manipüle edilmesi işlemidir. Bu, genellikle GPS sinyallerini taklit ederek veya dijital olarak cihazın konum bilgisini değiştirerek gerçekleştirilir. Ayrıca saldırganlar, cihazları uzaktan etkinleştirerek pil ömrünü hızla tüketmesine ve müşteri verilerini çalmasına olanak tanıyabilir. Bu, yaygın IoT ürün hatalarından kaynaklanan gizlilik, güvenlik ve güvenilirlik risklerini göstermektedir (Harper et al., 2022).

Bu olaylar, sektörler arası şirketlerin, kritik sistemler ve altyapıları etkileyebilecek büyüyen cihaz sayıları öncesinde IoT güvenliğine öncelik vermelerinin nedenini vurgulamaktadır.

3. IoT AĞ PROTOKOLLERİ VE GÜVENLİĞİ

IoT ağları, cihazlar, sensörler, kontrolörler ve bulut platformları arasında bağlantı ve veri aktarımını sağlamak için iletişim protokollerini kullanmaktadır. Farklı kullanım durumlarına, çevrelere ve performans gereksinimlerine göre optimize edilmiş çok çeşitli IoT protokolleri bulunmaktadır. Ancak, birçok popüler eski protokol, güçlü güvenlik mekanizmalarından yoksun olup, siber saldırı riskleri taşımaktadır. Bu bölümde, önemli IoT protokollerine genel bir bakış sunarak, güvenlik özelliklerini ve eksikliklerini değerlendirip, gelişmekte olan teknolojileri kullanarak nasıl iyileştirmeler yapılabileceği üzerinde durulmuştur. Bu yaklaşımın hem mevcut sistemlerin güvenliğini artıracağı hem de IoT teknolojisinin gelecekteki gelişimine katkıda bulunacağına inanılmaktadır.

3.1.Başlıca IoT Protokollerine Genel Bakış

ZigBee, Bluetooth/BLE, Hücresel, WiFi, LoRaWAN, Z-Wave, 6LoWPAN protokolleri tüketici, endüstriyel ve altyapı uygulamalarında sıkça kullanılan önemli IoT protokollerindedir (Jamal Rashid et al., 2021). Bu protokoller, akıllı evlerden hastanelerde kullanılan cihazlara kadar, hayatımızın her alanında IoT cihazlarının nasıl iletişim kurduğunu belirlemektedir. Her biri, belirli bir ortama ve ihtiyaca göre tasarlanmıştır. ZigBee ev otomasyonu için ideal bir protokol iken, LoRaWAN akıllı şehirlerdeki pil gücüyle çalışan cihazlar için tercih edilmektedir. Bu çeşitlilik, IoT teknolojisinin farklı alanlarda nasıl uyarlanabileceğini ve optimize edilebileceğini göstermektedir. Bu protokoller, IoT dünyasının çeşitliliğini ve zenginliğini temsil eder ve her birinin kendine has özellikleri vardır.

ZigBee, IEEE 802.15.4 standardı üzerine kurulu düşük güçlü ağ oluşturma protokolü, düşük veri oranlı sensör/kontrol ağları için optimize edilmiştir. Akıllı evler, sağlık, enerji izleme gibi alanlarda kullanılmaktadır. ZigBee, düşük maliyetli, düşük enerji tüketimi ve düşük veri hızı sunan bir teknolojidir (Chi et al., 2016). Ayrıca akıllı tarım sistemleri, akıllı sağlık sistemleri, akıllı evler ve izleme ortamları gibi çeşitli IoT uygulamalarında kullanılan kablosuz sensör ağlarının ömrünü uzatabilecek bir teknolojidir (Duy Tan et al., 2023).

Bluetooth, kısa menzilli kablosuz iletişim teknolojisi olarak IoT cihazlarının haberleşmesi için yaygın bir şekilde kullanılmaktadır. Özellikle BLE (Bluetooth Low Energy) versiyonu, pil ömrü kısıtlı IoT sensörleri ve cihazları için uygun bir çözüm sunmaktadır (C. Gomez et al., 2012). BLE, klassik Bluetooth'a göre daha düşük güç tüketimi sunarken, aynı menzil ve veri hızını koruyabilmektedir. BLE çipsetleri, yıllarca pil ömrü sunabilen düşük güç modlarına sahiptir. Bu sayede pil veya enerji hasadı ile çalışan IoT cihazları için idealdir. BLE üzerinden tipik IoT veri iletişimi, küçük paketler halinde ve düşük veri hızlarında gerçekleşmektedir. Buna rağmen, BLE mesh ağları gibi çözümlerle binlerce IoT cihazı

ölçeklenebilir şekilde bağlanabilmektedir. BLE'nin en önemli güvenlik mekanizmaları 128 bit AES şifreleme, kimlik doğrulama ve yetkilendirme protokolleridir. Bu sayede IoT verilerinin şifrelenmesi ve cihaz erişimlerinin kontrolü sağlanabilmektedir (Abomhara & M. Køien, 2015). Ancak bazı zafiyetler mevcuttur ve güncellenmesi gerekmektedir. Sonuç olarak, düşük maliyetli, düşük güç tüketimli bir haberleşme çözümü sunan BLE, IoT ekosistemi için kaçınılmaz bir teknolojidir. BLE, güvenlik zaafiyetlerinin giderilmesiyle, geleceğin en yaygın IoT protokollerinden biri olmaya adaydır.

Hücresel ağlar, dünya çapında yaygın kapsama alanları ve güvenilirlikleri sayesinde, geniş alan IoT uygulamaları için ideal bir haberleşme çözümü sunmaktadır (Naik, 2017). Özellikle endüstriyel IoT sensörleri, akıllı şehir altyapısı, filo yönetimi ve benzeri uzak konumlardaki IoT cihazları için tercih edilmektedirler. Hücresel IoT, mevcut 2G, 3G, 4G ve gelişmekte olan 5G hücresel ağ altyapılarını kullanmaktadır. 2G ve 3G düşük bant genişliği IoT sensörleri için yeterliyken, 4G LTE daha yüksek veri hızları, düşük gecikme süreleri ve artan bağlantı kapasitesi ile daha zengin IoT uygulamalarına olanak tanımaktadır (Mehmood et al., 2017). 5G ise sunduğu çok yüksek bant genişliği, milisaniye mertebesinde ultra düşük gecikme, 1 milyon cihaz/km2 yoğunlukta bağlantı ve kesintisiz hizmet sürekliliği ile IoT ekosistemi için büyük fırsatlar sunacaktır (Shafiq et al., 2013). Özellikle gerçek zamanlı kritik IoT uygulamalarında 5G devrim yaratacaktır. Hücresel IoT için NB-IoT ve LTE-M gibi düşük güçlü geniş alan (LPWA) teknolojileri de öne çıkmaktadır. 10 yıla kadar pil ömrü sunabilen bu teknolojiler, düşük maliyetli IoT sensörleri ve cihazları için uygundur (Raza et al., 2017).Hücresel IoT, SIM kart tabanlı güçlü kimlik doğrulama, şifreleme, bütünlük kontrolü gibi yerleşik güvenlik mekanizmalarına sahiptir. Ancak 5G ile yeni IoT tehditleri de ortaya çıkacaktır. Bu nedenle çok katmanlı güvenlik yaklaşımları şarttır (Jorge Granjal et al., 2015).

WiFi (Wireless Fidelity), IEEE 802.11 standardına dayanan yerel alan ağı (LAN) teknolojisidir ve IoT cihazları arasında yüksek

bant genişliğine sahip kablosuz iletişim sağlamaktadır. WiFi, IoT sensörleri, kameralar, robotlar, tıbbi cihazlar gibi veri yoğun uygulamalar için tercih edilen bir protokoldür. WiFi, 2.4 GHz ve 5 GHz frekans bandında çalışır ve 11 Mbps'den 54 Mbps'ye kadar veri hızı sunmaktadır. WiFi ağlar genellikle yıldız veya örgü topolojisine sahiptir ve RC4 akış şifreleme ile AES blok şifreleme gibi güvenlik mekanizmalarını desteklemektedir. IoT cihazları için WiFi'ın en büyük avantajlarından biri, mevcut altyapının ve WiFi çipsetlerinin uygun maliyetli ve yaygın olarak bulunabilir olmasıdır. WiFi Alliance'ın sürekli geliştirdiği 802.11 ah gibi yeni standartlar, daha uzun menzilli ve düşük güç tüketimli IoT bağlantısı vaat etmektedir (Adame et al., 2014). Ancak WiFi'ın IoT için bazı kısıtlamaları da vardır. Örneğin, çok sayıda eşzamanlı cihaz bağlantısını desteklemekte zorlanır ve pil gücü kısıtlı cihazlar için yüksek enerji tüketimi söz konusudur. Ayrıca açık WiFi ağları kolaylıkla dinlenebilir. Bu nedenle IoT verilerinin gizliliği için ek güvenlik tedbirleri gereklidir (J. Granjal et al., 2015). WiFi'ın yüksek bant genişliği, gelişmiş güvenlik özellikleri ve mevcut altyapı ile uyumluluğu, onu veri yoğun IoT uygulamaları için uygun bir haberleşme teknolojisi haline getirmektedir. Ancak özellikle güvenlik, ölçeklenebilirlik ve enerji verimliliği konularında iyileştirmelere ihtiyaç vardır.

LoRaWAN (Long Range Wide Area Network), düşük güçlü geniş alan ağı (LPWAN) teknolojilerinden biridir ve uzun menzil, düşük bant genişliği haberleşmesi gereken IoT uygulamaları için uygundur. LoRa İttifakı tarafından geliştirilmiştir ve açık bir standarttır. LoRaWAN, 10 yıla kadar pil ömrü sunan düşük maliyetli IoT sensörleri ve cihazları için idealdir (Raza et al., 2017). Kırsal alanlarda 15 km'ye kadar menzil sunabilmekte, kentsel alanlarda ise 2-5 km aralığında bağlantı sağlamaktadır (Mekki et al., 2019). Bant genişliği 125/250 kHz'dir ve veri hızı 0,3-50 kbps aralığındadır. LoRaWAN topolojisi yıldız tipindedir. Veri güvenliği için 128 bit AES şifrelemesi kullanılır. Ayrıca cihazların ağa katılımı da yetkilendirme ile kontrol edilir (Naik, 2017). LoRaWAN'ın en

yaygın kullanım alanları akıllı şehirler, akıllı bina ve ev otomasyonu, tarım takibi, filo yönetimidir (Mekki et al., 2019).

Z-Wave, akıllı ev otomasyonu ve kontrolü için yaygın olarak kullanılan bir kablosuz iletişim protokolüdür (Fargas & Petersen, 2017). Z-Wave genellikle ışıklandırma, ısıtma, havalandırma, güvenlik sistemleri ve ev aletlerinin uzaktan kontrolü gibi uygulamalarda tercih edilen bir protokoldür. Z-Wave, 900 MHz frekans bandında mesh ağ topolojisine dayalı olarak çalışır. Maksimum 100 metre menzile ve 100 kbps veri hızına sahiptir. Ağdaki cihazlar, merkezi bir kontrol birimi ile haberleşirler. Z-Wave Alliance tarafından standartlaştırılmıştır ve AES 128 bit şifreleme gibi güvenlik mekanizmalarını destekler (J. Granjal et al., 2015). Z-Wave'in en büyük avantajlarından biri, mevcut kablolu sistemlere kolay entegrasyon imkânı sunmasıdır. Ayrıca düşük maliyetli olması ve düşük güç tüketimi sayesinde pil ile çalışan sensörler için uygundur. Ancak menzil ve veri hızı kısıtlamaları, Z-Wave'i yüksek bant genişliği gerektiren uygulamalar için uygun kılmamaktadır.

6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks), düşük güç tüketimli IoT cihazlarının IPv6 tabanlı internet bağlantısı için tasarlanmış bir iletişim protokolüdür, özellikle pil gücüyle çalışan sensörler, aktüatörler ve diğer kısıtlı kabiliyetli cihazlar için uygundur. 6LoWPAN, IEEE 802.15.4 standardı üzerine inşa edilmiştir ve mesh ağ mimarisine dayanır (Gabriel et al., 2022). Böylece IPv6 paketleri, parçalara ayrılarak (fragmente edilerek) düşük MTU'ya sahip IEEE 802.15.4 bağlantısı üzerinden aktarılabilir (Gaur et al., 2021). 6LoWPAN ayrıca güvenli otantiklik, şifreleme, ağ katmanı sıkıştırması gibi optimizasyonlar da sağlar (Yang et al., 2020). 6LoWPAN protokolü, akıllı şehirler, bina/ev otomasyonu, endüstriyel IoT, tarım takibi gibi birçok alanda kullanılmaktadır (Mazlan et al., 2020). Düşük maliyet, düşük güç tüketimi, IPv6 desteği gibi avantajları bulunmaktadır (Chen et al., 2022). Ancak ölçeklenebilirlik, güvenlik ve gerçek zamanlı veri aktarımı konularında bazı sınırlamaları vardır (Shelby et al., 2012).

3.2. Farklı Protokollerin Güvenlik Mekanizmaları

IoT protokolleri, cihazlar arası ve cihazdan buluta iletişimlerde gizlilik, bütünlük ve erişilebilirlik sağlamak için şifreleme, kimlik doğrulama ve yetkilendirme gibi güvenlik tekniklerini kullanmaktadır (Sicari et al., 2016). ZigBee, veri transferlerini korumak için 128-bit AES şifrelemesi kullanır ve ağ içindeki IoT cihazları ve ağ geçitleri arasında kimlik doğrulamasını destekleyen güvenlik modlarına sahiptir. Ancak uçtan uca güvenlik eksikliği, açıkların oluşmasına neden olabilir (Garcia-Morchon & Wehrle, 2010). Bluetooth, cihaz doğrulaması, şifreleme ve veri imzalama gibi çeşitli yöntemler kullanarak güvenlik sağlar. BLE varyantı, IoT cihazları arasındaki bağlantıyı korumak için 128-bit AES şifrelemesi kullanmaktadır (Carles Gomez et al., 2012).

Hücresel protokoller, LTE gibi hücresel standartlardan SIM kart tabanlı kimlik doğrulama, şifreleme ve bütünlük kontrolü gibi iyi tanımlanmış güvenlik çerçevelerini miras alarak cihaz kimliğini ve trafiğini korur (Naik, 2017). WiFi, WPA2 şifreleme protokolü kullanarak ağ erişimini ve verileri korur. Bu protokol, geçici anahtar bütünlük protokolü ve AES CCMP şifre paketleri gibi mekanizmaları içerir. WPA3, IoT WiFi ağları için daha gelişmiş güvenlik vaat etmektedir (Padgette et al., 2017).

LoRaWAN, yalnızca yetkili son cihazların bir ağa katılmasını sağlamak ve dinlemeyi önlemek için AES 128-bit şifreleme anahtarları kullanır. Ağ ve uygulama oturum anahtarları, verileri şifrelemeyi sağlar (Haxhibeqiri et al., 2018). Z-Wave, akıllı ev ortamlarında kablosuz iletişimi korumak için AES 128-bit şifrelemeyi kullanır. Bu, erişimi yöneten ve anahtarların dağıtımını sağlayan denetleyicilere dayanan merkezileştirilmiş bir güvenlik modeline dayanmaktadır (Khanji et al., 2019).

6LoWPAN, cihazlar ve internet ana bilgisayarları arasında güvenli iletişim kurmak için kriptografik anahtarlar ve kimlik doğrulama mekanizmaları kullanarak başlangıç yöntemlerini tanımlamaktadır (Raza et al., 2013). Bu protokoller, IoT cihazlarının güvenli bir şekilde iletişim kurmasını sağlamak için farklı güvenlik

önlemlerine sahiptir. Her biri, kendi kullanım alanına ve ihtiyaçlarına göre özelleştirilmiş güvenlik çözümleri sunar. Bu, IoT ağlarının karmaşıklığını ve güvenlik alanındaki çeşitliliğini gösterir. Her protokol, IoT dünyasında veri güvenliğinin nasıl sağlanabileceğine dair kendi yaklaşımını sunmaktadır.

3.3. Teknolojik Boşluklar ve Potansiyel İyileştirmeler

Mevcut güvenlik tekniklerine rağmen, IoT protokolleri, teknolojik kapasite sınırlamalarına sahiptir ve çeşitli ekosistemleri kapsayan standartlara dayalı uçtan uca güvenlik modellerinden yoksundur. Bu durum, istismarlardan, yetkisiz erişimden, veri sızıntılarından ve diğer saldırı türlerinden kaynaklanan potansiyel zafiyetlere yol açabilir (Jing et al., 2014).

Blockchain teknolojisi, IoT ağlarında çoklu katmanlar ve katılımcılar arasında merkezi olmayan kimlik ve erişim yönetimi sağlayarak, IoT sistemlerinin güvenliğini ve verimliliğini artırmada önemli bir rol oynamaktadır. Bu teknoloji, IoT cihazlarının ve ağlarının güvenliğini artırırken, aynı zamanda tek noktadan arıza risklerini de önemli ölçüde azaltır. Blockchain, IoT cihazları arasında güvenli ve şeffaf bir iletişim sağlar, böylece her bir cihazın kimliği ve işlemleri doğrulanabilir ve güvenilir bir şekilde kaydedilir. Bu, özellikle akıllı şehirler, sağlık hizmetleri ve endüstriyel otomasyon gibi alanlarda kritik öneme sahiptir (Malhotra, 2023). Merkezi olmayan yapısı sayesinde, blockchain, tek bir noktadan hizmet veren sistemlerin aksine, ağın bütünü üzerinde güvenlik ve dayanıklılığı artırır. Bu, özellikle büyük ölçekli IoT uygulamalarında, sistemlerin sürekli ve kesintisiz çalışmasını sağlamaktadır (Giaretta et al., 2019). Blockchain tabanlı sistemler, akıllı sözleşmeler aracılığıyla otomatikleştirilmiş işlemler ve protokoller sunabilmektedir. Bu, IoT cihazlarının ve kullanıcılarının kimlik doğrulama süreçlerini basitleştirerek güvenliği artırmaktadır. Ayrıca, blockchain, IoT cihazları ve ağları arasında güvenli veri paylaşımını ve işlemleri kolaylaştırır. Bu, özellikle tedarik zinciri yönetimi ve varlık izleme gibi uygulamalarda verimliliği ve şeffaflığı artırır (Xu et al., 2018).

Kuantum sonrası kriptografi, gelecek nesil kriptografik protokoller olarak, kuantum bilgisayar saldırılarına karşı gelecekte koruma sağlamak için geliştirilmiştir. Bu alandaki çalışmalar, ızgara tabanlı, karma tabanlı ve kod tabanlı teknikler gibi çeşitli yöntemler üzerinde yoğunlaşmaktadır. Suparna Kundu vd. (2023) ızgara tabanlı kriptografiye odaklanmış ve bu tekniklerin kuantum sonrası kriptografide nasıl kullanılabileceğini incelemiştir. Çalışma, Scabbard adlı bir ızgara tabanlı post-kuantum anahtar kapsülleme mekanizmaları setini ele almış ve bu mekanizmaların maskeleme teknikleriyle entegrasyonunu incelemiştir (Kundu et al., 2023). Alvaro Cintas Canto vd. (2023) kod tabanlı kriptografinin güvenilirliğini artırmak için hata tespit tekniklerini ele almıştır. Bu çalışma, Niederreiter kripto sistemi gibi kod tabanlı algoritmaların güvenilirliğini artırmak için FPGA üzerinde uygulanan hata tespit mekanizmalarını incelemiştir (Cintas-Canto et al., 2023). Yousef Fazea (2023) ızgara tabanlı ızgara kripto sistemlerindeki yan kanal saldırılarını incelemiştir. Bu çalışma, Discrete Ziggurat (DZ) yönteminin yan kanal saldırılarına karşı savunmasızlıklarını ve bu saldırılara karşı alınabilecek önlemlerden bahsetmektedir (Fazea et al., 2023).

Hafif şifreleme, kaynakları kısıtlı olan IoT cihazları için özel olarak tasarlanmış bir kriptografi çözümüdür. IoT cihazlar genellikle düşük işlem gücüne, sınırlı belleğe ve enerji kısıtlamalarına sahip olduğundan, hafif şifreleme yöntemleri, bu tür cihazların güvenliğini sağlamak için kaynak yükünü en aza indirirken yeterli savunma sağlamayı hedeflemektedir. Q-SECURE adıyla geliştirilen post-kuantum dirençli güvenlik sistemi (Ngouen et al., 2023) özellikle kaynakları kısıtlı olan IoT cihazlarının şifrelemesini güçlendirmek amacıyla tasarlanmıştır. Sistem, bir IoT ağında bulunan diğer cihazların yardımıyla, herhangi bir boyuttaki post-kuantum kriptografik anahtarların üretilmesini mümkün kılmaktadır. Bu yenilikçi yaklaşım, IoT cihazlarının güvenliğini artırırken, aynı zamanda gelecekteki kuantum bilgisayar saldırılarına karşı koruma sağlamayı hedeflemektedir. Arduino platformunda geleneksel ağır şifreleme algoritmaları oldukça zayıf bir performansa sahiptir. Hafif

şifreleme algoritmaları ise bu tür cihazlarda çalışmak için daha uygundur. Hafif şifreleme algoritmaları, kaynak kısıtlamaları göz önünde bulundurularak tasarlanmıştır ve bu cihazların güvenliği sağlanırken aynı zamanda enerji verimliliği de artırılmaktadır (Weng, 2023).

Yapay zekâ tabanlı analizler, bağlantı desenlerini dinamik olarak analiz etmek, anormallikleri tespit etmek ve hızla gelişen heterojen IoT ağlarında güvenlik politikalarını uyarlamak için kullanılarak IoT ağ güvenliğine önemli katkılar sağlamaktadır. Özellikle normal ağ trafiği ile potansiyel saldırganlardan gelen ağ trafiği arasındaki farkları ayırt etmeye yardımcı olan derin giriş tespit sistemleri (Deep IDS) bulunmaktadır (Darius et al., 2023). Deep IDS sistemleri özellikle akıllı şehirlerdeki IoT ağlarının güvenliğini artırmak için etkin bir şekilde kullanılabilir olduğunu göstermiştir.

4. YAPAY ZEKÂ KULLANARAK IoT AĞLARININ GÜVENLİĞİNİ SAĞLAMA

IoT cihazlarının ve sistemlerinin hızla yaygınlaşması, bu ağları siber saldırılara karşı savunmasız hale getirmektedir. 2021 yılının ilk altı ayında, 1,5 milyarın üzerinde IoT güvenlik ihlali gerçekleşti, ve bu ihlallerin büyük bir kısmı Telnet uzaktan erişim protokolü üzerinden yapılmıştır (*Cost of cyber security. (n.d.). Purples*). Pandemi sürecinin etkisiyle, IoT cihazlarına yönelik hackleme olaylarında bir artış gözlemlenmiştir. Değişen saldırı biçimlerine karşı geleneksel güvenlik yöntemleri yetersiz kalmaktadır. Yapay zekâ ve derin öğrenme, IoT güvenliğini geliştirmede büyük potansiyele sahiptir. Yapay zekâ, saldırıları gerçek zamanlı olarak tespit edip engelleyebilir, şüpheli etkinlikleri ayırt edebilir ve siber tehditlere hızla yanıt verebilir (Meidan et al., 2018). Örneğin, destek vektör makineleri ağ geçidi trafiğinde anomalileri %99 doğrulukla saptayabilir (Zhang et al., 2021). Yine evrişimli sinir ağları, kötü amaçlı yazılımları %95 hassasiyetle belirleyebilir (Javaid et al., 2016). Yapay zekâ ayrıca kimlik doğrulama ve erişim kontrolünü geliştirebilir. Pekiştirmeli öğrenme, şüpheli oturum açma

girişimlerini tespit edip hesapları otomatik olarak kilitleyebilir (Ferdowsi & Saad, 2018). Böylece dinamik kimlik doğrulaması sağlanır. Federe öğrenme, cihazlardan veri paylaşımı olmadan model eğitimine izin vererek gizliliği korur. Böylece güvenli veri analitiği yapılabilir. Ancak yapay zekâ sistemleri de saldırılara açıktır. Dolayısıyla bu sistemlerin güvenliği ve dayanıklılığı kritik öneme sahiptir. Ayrıca veri gizliliği konusundaki endişeler de göz ardı edilmemelidir.

Yapay sinir ağları (ANN), beyin nöronlarının bağlantılarını taklit ederek hiyerarşik özellik temsillerini öğrenir. Çok katmanlı algılayıcılar (MLP'ler), karmaşık desen tanıma kullanarak zararlı yazılım tespiti ve ağ anormalliklerinin izlenmesi için uygulanan beslemeli derin sinir ağlarıdır (Meidan et al., 2017). Evrişimli sinir ağları (CNN'ler), tehditleri sınıflandırmadan önce özellik çıkarma ve veri indirgeme için ardışık evrişim katmanlarından oluşur. IoT'ye özgü CNN'ler, enfekte trafiği tespit etmede yüksek doğruluk göstermektedir (Kim et al., 2019). Döngüsel sinir ağları (RNN'ler), durum bilgilerini koruyan döngüsel bağlantılar kullanarak dinamik zamansal davranış modellemesini sağlar. Uzun kısa vadeli hafıza (LSTM) RNN'ler, IoT zararlı yazılım dizilerini ve zaman tabanlı enjeksiyon saldırılarını etkili bir şekilde tespit eder (Javaid et al., 2016). Derin inanç ağları (DBN'ler), büyük verilerden yararlanan kısıtlı Boltzmann makineleri (RBM) yığınından oluşan olasılıksal üretici modellerdir. DBN'ler, zararlı yazılım tespit oranlarını %97'nin üzerinde gösterir (Yin et al., 2017). Otokodlayıcılar (autoencoders, AE), girişleri boyut indirgeme yoluyla yeniden yapılandırarak sıkıştırılmış gizli özellik temsillerini öğrenir. Eksik AE'ler, normal IoT trafiğini modelleyerek anormallikleri gösterebilmektedir (Javaid et al., 2016).

IoT güvenliğinde yapay zekâ benimsenmesi, akıllı savunma mekanizmaları geliştirmeye yardımcı olabilir. Gerçek dünya verileriyle eğitim, insan-makine etkileşimi ve saldırılara karşı dayanıklılık hayati önem taşımaktadır. Özellikle IoT ağlarının güvenliğini sağlamada, yapay zekâ, değişen tehditlere ve durumlara hızla uyum sağlama ve gelişmiş güvenlik çözümleri sunma

kapasitesine sahiptir. Bu nedenle, IoT dünyasında yapay zekânın kullanımı, sadece bir seçenek değil, zorunluluk haline gelmektedir.

4.1.IoT Ağ Güvenliğinde Yapay Zekâ ve Makine Öğrenmesinin Rolü

4.1.1. Tehdit Önleme

IoT cihazlarında, yapay zekâ tabanlı test araçları kullanarak proaktif tehdit önleme stratejileri uygulanabilir. Bu araçlar, güvenlik açıklarını belirlemek için saldırıları taklit eden bulanıklaştırma tekniklerini ve yüksek risk taşıyan alanları önceden tespit eden tahmine dayalı kod analizörlerini içermektedir (Kolias et al., 2017). Bulanıklaştırma, uygulamalara geçersiz veya beklenmeyen rastgele veriler göndererek çökmeleri tetikler ve hataları açığa çıkarır. Öte yandan, tahmine dayalı kod incelemesi, makine öğrenimini kullanarak kodlama hatalarını ve güvenlik zayıflıklarını dağıtımdan önce belirlemektedir. Bu tür yapay zekâ destekli testler, IoT cihazlarının donanım ve yazılım güvenliğini, yalnızca reaktif önlemlere güvenmek yerine, geliştirme sürecinin her aşamasında proaktif bir şekilde güçlendirmektedir. Bu yaklaşım, güvenlik açıklarını erken aşamada tespit edip düzeltmek suretiyle IoT sistemlerinin daha güvenli hale gelmesine katkıda bulunmaktadır.

4.1.2. Anomali Tespiti

Denetimsiz makine öğrenimi teknikleri, IoT ağlarının normal trafik ve davranış modellerini belirlemek için kullanılmaktadır. Yalıtım ormanları, yerel aykırı değer faktörleri ve otomatik kodlayıcılar gibi yöntemler (Meidan et al., 2018), IoT ağlarının standart profillerini oluşturarak, herhangi bir sapmayı hızla tespit edebilir. Bu sapmalar, olası kötü amaçlı yazılım, yetkisiz erişim girişimleri veya bilgisayar korsanlığı faaliyetleri gibi güvenlik tehditlerine işaret edebilir ve bu durumlar, daha detaylı inceleme için uyarılar oluşturur. Ayrıca, kümeleme algoritmaları IoT olaylarını ve veri akışlarını gruplayarak (Alsoufi et al., 2021), normal dışı durumları belirleyebilir. Bu tür denetimsiz öğrenme modelleri, yeni ve daha önce görülmemiş veri modelleri arasında anormallikleri

bulmak için başlangıçta çok az veri eğitimi gerektirir. Bu yaklaşım, IoT ağlarının sürekli izlenmesi ve güvenliğinin sağlanması için etkili bir yöntem sunar, böylece potansiyel tehditler erkenden tespit edilip müdahale edilebilir.

4.1.3. Kullanıcı/Cihaz Kimlik Doğrulaması

Takviyeli öğrenme, makine öğrenmesinin bir dalı olup, ödül ve ceza mekanizması ile bir ajan deneyimlerinden öğrenmesini sağlamaktadır. IoT ağlarında şüpheli erişim denemelerini ve kimlik avı saldırılarını sürekli geribildirim sağlayarak uyarlanabilir kullanıcı kimlik doğrulaması için takviyeli öğrenme ajanları uygulanabilir (Ferdowsi & Saad, 2018). Bu ajanlar, saldırı desenlerini ortaya çıkarmak ve zaman içinde kimlik doğrulama politikalarını iyileştirmek için öğrenirler. Ek doğrulama faktörleri isteme, geçici kilitler vb. eylemler şüphe puanlarına göre alınır. Bu, dinamik kimlik doğrulamayı güçlendirir (Miettinen et al., 2017). Örneğin, yanlış parola girişi sayısını sınırlayarak veya coğrafi konum doğrulaması gibi ek adımlar ekleyerek saldırganları caydırabilir ve gerçek kullanıcılar için kullanım kolaylığını koruyabilirler. Takviyeli öğrenme ajanları, ödül ve ceza geribildirimine dayalı olarak optimum kimlik doğrulama politikalarını keşfeder. Bu sayede, değişen IoT tehdit ortamına uyum sağlarlar. Ayrıca insan müdahalesi olmadan çalıştıklarından ölçeklenebilirlik sağlarlar. Son çalışmalar, takviyeli öğrenmenin IoT'ler için kişiselleştirilmiş ve uyarlanabilir kimlik doğrulama çözümleri sunabileceğini göstermektedir (Diro & Chilamkurti, 2018). Takviyeli öğrenmenin IoT güvenliğinde benimsenmesi, dinamik tehdit ortamına uyum sağlayan akıllı kimlik doğrulama sistemlerinin geliştirilmesine yardımcı olabilir. Ancak gerçek dünya verileriyle eğitilmeleri ve insan-makine etkileşimlerini anlamaları kritiktir. Ayrıca, kötü niyetli saldırganların ajanları manipüle etmesini önlemek için tasarım ve testler çok önemlidir.

4.1.4. Veri Güvenliğini Sağlama

IoT sensörler ve IoT cihazlardan elde edilen verilerin hacmi ve çeşitliliği hızla artmaktadır. Ancak bu veriler kişisel bilgiler

içerebildiğinden, analiz edilirken gizlilik büyük önem taşımaktadır (Lu et al., 2014). Geleneksel merkezi veri analizi yaklaşımları gizliliği korumada yetersiz kalabilir. Birleşik öğrenme (federe öğrenme), cihazlardaki verileri merkezi bir yere göndermeden, dağıtık bir şekilde makine öğrenimi modellerini eğitmeye olanak tanımaktadır (Li et al., 2020). Her cihaz, kendi verilerini kullanarak yerel bir model oluşturur. Daha sonra bu yerel modellerin güncellemeleri, genel küresel modele katkıda bulunmak için bir araya getirilir. Böylece veriler asla cihazlardan dış ağa çıkmaz. Sağlık hizmetlerindeki giyilebilir cihazlardan gelen fitness verileri, yerel olarak işlenip modele katkıda bulunabilir. Akıllı şehirlerdeki trafik sensörlerinden elde edilen veriler, trafik tıkanıklığını öngörmek için kullanılabilir. Her iki durumda da veriler asla cihazlardan dışarıya çıkmaz (Lim et al., 2020). IoT tabanlı dolandırıcılık tespiti için de federe öğrenmeden faydalanılabilir. Ödeme işlem verilerini kullanarak yerel modeller eğitilir ve bu modeller birleştirilerek daha güçlü global bir model elde edilebilir (Zhang et al., 2021). Federe öğrenmenin getirdiği gizlilik ve güvenlik avantajları, IoT analitiğini ve karar verme süreçlerini geliştirme potansiyeline sahiptir. Ancak, verimlilik, iletişim maliyetleri, model birleştirme gibi zorluklar halen araştırma konusudur (Kairouz et al., 2021).

4.2. Denetimli, Denetimsiz ve Takviyeli Öğrenmeyi Kullanma

Denetimli öğrenme algoritmaları, etiketlenmiş veri setlerinden yararlanarak yeni verileri sınıflandırmak için tahmin modelleri oluşturur. Sinir ağları, rastgele ormanlar ve destek vektör makineleri (SVM) gibi yöntemler, desen tanıma tekniklerini kullanarak IoT ağlara yönelik bilinen zararlı yazılımları ve saldırı türlerini belirleyebilir (Meidan et al., 2018). Diğer yandan, denetimsiz öğrenme, etiketlenmemiş verilerdeki gizli desenleri ve yapıları ortaya çıkarmaktadır. Bu, kümeleme, anomali tespiti ve ilişkilendirme kuralı öğrenme gibi tekniklerle gerçekleştirilir. Bu yaklaşım, daha önce bilinmeyen yeni IoT tehditlerini ve sıfırıncı gün saldırılarını tanımlama yeteneğine sahiptir.

Takviyeli öğrenme, sürekli ödül geri bildirimi kullanarak denetimsiz ortamlarda hareket eden ajanları geliştirir. IoT güvenlik ajanları, düşman tehditlere karşı bile zamanla optimal politikaları öğrenebilir (Ferdowsi & Saad, 2018). Katmanlı AI, denetimli tehdit tespiti, denetimsiz anormallik uyarıları ve takviyeli kimlik doğrulama sunarak IoT saldırı direncini güçlendirir. Bu, IoT cihazlarımızın güvenliğini sağlamak için yapay zekânın farklı yönlerini nasıl kullanabileceğimizi gösteriyor. Denetimli öğrenme, bilinen tehditleri tanımak için; denetimsiz öğrenme, daha önce bilinmeyen saldırıları keşfetmek için ve takviye öğrenme, sürekli değişen tehditlere karşı etkili politikalar geliştirmek için kullanılabilir. Bu çok yönlü yaklaşım, IoT güvenliğini daha da güçlendirerek, cihazlarımızı ve ağlarımızı siber saldırılara karşı daha dirençli hale getirir. Bu, IoT teknolojisini daha akıllı ve güvenli bir geleceğe taşıyan önemli bir adımdır.

4.3.Deneysel araştırma örnekleri ve sonuçları

Denetimli SVM tek sınıf sınıflandırıcısı ve denetimsiz k-ortalama kümeleme kombinasyonundan oluşan bir sızma tespit sistemi (IDS), NSL-KDD veri setini kullanarak ağ tehditlerini %99'dan fazla doğrulukla tespit etti (Vinayakumar et al., 2019). LSTM RNN modeli, MLP, SVM, en yakın komşular (k-NN) ve Naive Bayes sınıflandırıcılarını geride bırakarak IoT zararlı yazılım trafiğini %98 doğrulukla tespit etti (Meidan et al., 2018). Federatif öğrenme (FL) çerçevesi, doğrudan veri paylaşımı olmadan cihazlar arasında IoT sızma tespit modellerinin işbirlikçi eğitimini sağlayarak test hatalarını azalttı (Lu et al., 2014).

Yapay zekâ çiplerindeki, kenar bilişimdeki ve veri analitiğindeki devam eden ilerlemeler, artan IoT siber tehditlerle mücadelede makine öğrenimini benimsemeyi teşvik edecek. Ancak teknikler, sensörler, giyilebilir teknolojiler vb. donanım kısıtlamaları göz önüne alındığında performans yüklerini en aza indirmeli ve algoritmaların kendilerine yönelik düşmanca saldırılara karşı dayanıklılığı sağlamalıdır. Bu araştırmalar, yapay zekânın IoT güvenliği alanında ne kadar etkili olabileceğini gösteriyor. Özellikle,

sızma tespiti ve zararlı yazılım analizi gibi konularda, yapay zekânın kullanımı, IoT ağlarını daha güvenli hale getirme potansiyeline sahip. Bu teknolojik ilerlemeler, IoT cihazlarını korumak için yeni ve yenilikçi yollar sunuyor ve bu alanda sürekli gelişmeye işaret ediyor. Yapay zekânın bu uygulamaları, IoT teknolojilerinin güvenliğini artırmada önemli bir rol oynayacak gibi görünmektedir.

5. SONUÇ

İnternet Nesneleri (IoT) cihazlarının ve sistemlerinin hızlı yayılımı, hemen hemen her alanda yaşam kalitesini büyük ölçüde artırmıştır. Bu araştırmanın sonucunda, gizlilik, saldırılar, tehditler ve güvenlik açıklarıyla ilgili güvenlik risklerinin de aynı hızda arttığı tespit edilmiştir. IoT ağları, kritik altyapıyı kapsayacak şekilde genişledikçe ve karmaşıklık kazandıkça, katı siber güvenlik önlemleri zorunlu hale gelmektedir. ZigBee, Bluetooth, WiFi ve diğerleri gibi başlıca IoT ağ protokollerini, çekirdek güvenlik mekanizmalarını ve saldırganların sahtecilik, tahrifat, inkâr, bilgi ifşası, hizmet reddi ve ayrıcalık yükseltme tipi saldırılar başlatmak için kullanabileceği potansiyel sınırlamalarını değerlendirilmiştir. Bulgular, şifreleme, erişim kontrolleri ve yetkilendirme tekniklerinin uygulanmasına rağmen, karmaşık IoT ekosistemlerinde uçtan uca güvenliğin sistematik bir sorun olarak kaldığını göstermektedir.

Blockchain, kuantum sonrası kriptografi ve hafif şifreleme gibi yükselen teknolojiler, merkezi olmayan kimlik yönetimi ve geleceğe yönelik koruma sağlayabilir. Özellikle kaynak kısıtlı IoT cihazları için optimize edilmiş güvenlik yüklerini ele almak bu teknolojilerin önemli bir işlevi olacaktır. Yapay zekâ ve makine öğrenimi de genişleyen ve hızla gelişen IoT saldırı yüzeylerinde otomatik, gerçek zamanlı tehdit tespiti, analizi ve yanıtı için kritik öneme sahiptir.

Bu çalışmada IoT ağlarındaki temel güvenlik tehditleri ve saldırı türleri ayrıntılı olarak incelenmiş; kötü amaçlı yazılımlar, ağ tabanlı saldırılar, büyük ölçekli gerçek saldırı örnekleri ele alınmıştır. Ayrıca IoT haberleşme protokollerinin güvenlik

mekanizmaları ve zaafiyetleri karşılaştırmalı olarak değerlendirilmiştir.

Elde edilen bulgular, IoT ekosisteminde kapsamlı bir güvenlik stratejisinin hayati olduğuna işaret etmektedir. Çok katmanlı savunma mekanizmaları, açık standartlar, en iyi uygulamalar, düzenlemeler gibi çeşitli önlemlerin bir arada ele alınması gereklidir. Özellikle yapay zekâ, blokzincir gibi yeni teknolojilerin sunduğu imkanlar, IoT güvenliğini önemli ölçüde geliştirme potansiyeline sahiptir.

IoT güvenliği, proaktif ve işbirlikçi bir yaklaşımla ele alındığı takdirde hem bireysel kullanıcılar hem de kurumlar için büyük fayda sağlayacak akıllı ve güvenilir çözümler geliştirilebilir. Bu alandaki araştırma ve yenilik çabalarına hız kesmeden devam edilmesi büyük önem taşımaktadır.

Sonuç olarak, IoT cihazlarının ve ağlarının hızla yaygınlaşması, beraberinde ciddi güvenlik risklerini de getirmektedir. Kullanıcı mahremiyetinin ve hassas verilerin korunması, siber saldırıların önlenmesi, kritik altyapıların güvenliğinin sağlanması gibi konular, IoT dünyasında hayati öneme sahiptir. Farklı teknolojileri, özel platformları, eski sistemleri ve yaygın tedarik zincirlerini kapsayan bütünsel güvenlik, devasa bir zorluk olmaya devam etmektedir. Cihaz üreticilerinden bulut platform sağlayıcılarına kadar IoT paydaşları arasındaki iş birliği: açık standartları, en iyi uygulamaları ve sertifika programlarını teşvikleri açısından hayati önem taşımaktadır. Enerji, sağlık ve ulaşım gibi güvenlik açısından kritik sektörlerde IoT benimsenmesi yoğunlaştıkça IoT güvenlik politikaları, uyum zorunlulukları ve raporlama çerçevelerini yürürlüğe koymada hükümetler ve düzenleyiciler kritik bir rol oynamaktadır. Bu açıdan bakıldığında Nesnelerin İnterneti kavramının sadece dijital ortamdan ibaret olmadığı, hayatımızı tamamen çevreleyen ve günlük yaşamımızı şekillendirip değiştiren bir teknoloji olduğunu görmekteyiz.

KAYNAKÇA

Abomhara, M., & M. Køien, G. (2015). Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. *Journal of Cyber Security and Mobility*, *4*(1), 65–88. https://doi.org/10.13052/jcsm2245-1439.414

Adame, T., Bel, A., Bellalta, B., Barcelo, J., & Oliver, M. (2014). IEEE 802.11 AH: the WiFi approach for M2M communications. *IEEE Wireless Communications*, *21*(6), 144-152.

Ahmed, M., Altamimi, A. B., Khan, W., Alsaffar, M., Ahmad, A., Khan, Z. H., & Alreshidi, A. (2023). PhishCatcher: Client-Side Defense Against Web Spoofing Attacks Using Machine Learning. *Ieee Access*, *11*, 61249-61263. https://doi.org/10.1109/ACCESS.2023.3287226

Allakany, A., Saber, A., Mostafa, S. M., Alsabaan, M., Ibrahem, M. I., & Elwahsh, H. (2023). Enhancing Security in ZigBee Wireless Sensor Networks: A New Approach and Mutual Authentication Scheme for D2D Communication. *Sensors*, *23*(12).

Alrubayyi, H., Goteng, G., & Jaber, M. (2023). AIS for Malware Detection in a Realistic IoT System: Challenges and Opportunities. *Network*, *3*(4), 522-537.

Alrubayyi, H., Goteng, G., Jaber, M., & Kelly, J. (2021). Challenges of malware detection in the IoT and a review of artificial immune system approaches. *Journal of Sensor and Actuator Networks*, *10*(4), 61.

Alsoufi, M. A., Razak, S., Siraj, M. M., Nafea, I., Ghaleb, F. A., Saeed, F., & Nasser, M. (2021). Anomaly-Based Intrusion Detection Systems in IoT Using Deep Learning: A Systematic Literature Review. *Applied Sciences*, *11*(18), 8383. https://www.mdpi.com/2076-3417/11/18/8383

Aranuwa, F., Olubodun, F., & Akinwumi, D. (2022). Hybridized Model for Data Security Based on Security Hash Analysis (SHA 512) and Salting Techniques. *International Journal*

of Network Security & Its Applications, *14*(2), 31-39. https://doi.org/https://doi.org/10.5121/ijnsa.2022.14203

Bhale, P., Ray, D., Biswas, S., & Nandi, S. (2023, 23-25 June 2023). WOMN: WOrMhole Attack DetectioN and Mitigation Using Lightweight Distributed IDS in IoT Network. 2023 IEEE Guwahati Subsection Conference (GCON),

Bhardwaj, M., Kumari, U., Kumar, S., & Choudhary, S. (2023). An Efficient User Authentication and Key Agreement Scheme Wireless Sensor Network and IOT Using Various Security Approaches. *SN Computer Science*, *4*(5), 574. https://doi.org/10.1007/s42979-023-01964-1

Bout, E., Loscri, V., & Gallais, A. (2022). Evolution of IoT Security: The Era of Smart Attacks. *IEEE Internet of Things Magazine*, *5*(1), 108-113. https://doi.org/10.1109/IOTM.001.2100183

Chen, R., Xiao, Y., Chen, Y., Xu, H., Yu, P., Peng, Q., Li, X., Guo, X., Huang, J., Li, N., Hu, X., Ou, R., Liu, W., Chen, B., Zhang, W., Xin, X., Zhao, B., & Chen, Z. (2022, 20-26 Feb. 2022). A 6.5-to-10GHz IEEE 802.15.4/4z-Compliant 1T3R UWB Transceiver. 2022 IEEE International Solid-State Circuits Conference (ISSCC),

Chi, H. R., Tsang, K. F., Wu, C. K., & Hung, F. H. (2016). ZigBee based wireless sensor network in smart metering. IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society,

Cintas-Canto, A., Mozaffari-Kermani, M., & Azarderakhsh, R. (2023, 31 Oct.-1 Nov. 2023). Reliable Code-Based Post-Quantum Cryptographic Algorithms through Fault Detection on FPGA. 2023 IEEE Nordic Circuits and Systems Conference (NorCAS),

Classen, J. (2020). *Security and Privacy for IoT Ecosystems* Technische Universität]. Darmstadt. http://tuprints.ulb.tu-darmstadt.de/11422/

Cost of cyber security. (n.d.). Purples. https://purplesec.us/resources/cyber-securitystatistics/#CyberCrime

Darius, P., Rangelov, D., Lämmel, P., & Tcholtchev, N. (2023, 28-30 Nov. 2023). An OMNeT++-Based Approach to Narrowband-IoT Traffic Generation for Machine Learning-Based Anomaly Detection. 2023 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS),

De Donno, M., Dragoni, N., Giaretta, A., & Spognardi, A. (2018). DDoS-Capable IoT Malwares: Comparative Analysis and Mirai Investigation. *Security and Communication Networks, 2018*, 7178164. https://doi.org/10.1155/2018/7178164

Diro, A., & Chilamkurti, N. (2018). Leveraging LSTM networks for attack detection in fog-to-things communications. *IEEE Communications Magazine, 56*(9), 124-130.

Duy Tan, N., Nguyen, D.-N., Hoang, H.-N., & Le, T.-T.-H. (2023). EEGT: Energy Efficient Grid-Based Routing Protocol in Wireless Sensor Networks for IoT Applications. *Computers, 12*(5).

Ekpenyong, M. E., Asuquo, D. E., Udo, I. J., Robinson, S. A., & Ijebu, F. F. (2022). IPv6 Routing Protocol Enhancements over Low-power and Lossy Networks for IoT Applications: A Systematic Review. *New Review of Information Networking*, *27*(1), 30-68. https://doi.org/10.1080/13614576.2022.2078396

Fargas, B. C., & Petersen, M. N. (2017). GPS-free geolocation using LoRa in low-power WANs. 2017 global internet of things summit (Giots),

Fazea, Y., Mohammed, F., & Alsamman, M. (2023, 10-11 Oct. 2023). Side-Channel Vulnerabilities in Discrete Ziggurat Sampler in Post-Quantum Cryptography. 2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA),

Ferdowsi, A., & Saad, W. (2018). Deep learning for signal authentication and security in massive internet-of-things systems. *IEEE Transactions on Communications*, *67*(2), 1371-1387.

Ferrag, M. A., Derdour, M., Mukherjee, M., Derhab, A., Maglaras, L., & Janicke, H. (2018). Blockchain technologies for the internet of things: Research issues and challenges. *IEEE Internet of Things Journal*, *6*(2), 2188-2204.

Gabriel, P.-E., Butt, S. A., Francisco, E.-O., Alejandro, C.-P., & Maleh, Y. (2022). Performance analysis of 6LoWPAN protocol for a flood monitoring system. *EURASIP Journal on Wireless Communications and Networking*, *2022*(1), 1-18.

Garcia-Morchon, O., & Wehrle, K. (2010). *Modular context-aware access control for medical sensor networks* Proceedings of the 15th ACM symposium on Access control models and technologies, Pittsburgh, Pennsylvania, USA. https://doi.org/10.1145/1809842.1809864

Gaur, R., Prakash, S., & Barik, R. K. (2021, 16-18 Dec. 2021). Analysis of detection and prevention mechanism for 6LowPAN Protocol Header in IoT assisted Cloud Environments. 2021 19th OITS International Conference on Information Technology (OCIT),

Giaretta, A., Pepe, S., & Dragoni, N. (2019, 2019//). UniquID: A Quest to Reconcile Identity Access Management and the IoT. Software Technology: Methods and Tools, Cham.

Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, *12*(9), 11734-11753.

Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, *12*(9), 11734–11753.

Granjal, J., Monteiro, E., & Silva, J. S. (2015). Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys & Tutorials*, *17*(3), 1294-1312. https://doi.org/10.1109/COMST.2015.2388550

Granjal, J., Monteiro, E., & Silva, J. S. (2015). Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey. *Ad Hoc Networks*, *24*, 264-287.

Harper, S., Mehrnezhad, M., & Leach, M. (2022, 6-10 June 2022). Are Our Animals Leaking Information About Us? Security and Privacy Evaluation of Animal-related Apps. 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW),

Haxhibeqiri, J., De Poorter, E., Moerman, I., & Hoebeke, J. (2018). A survey of LoRaWAN for IoT: From technology to application. *Sensors*, *18*(11), 3995.

Jamal Rashid, S., Alkababji, A., & Khidhir, A. M. (2021). Communication and Network Technologies of IoT in Smart Building: A Survey. *NTU Journal of Engineering and Technology*, *1*(1), 1-18. https://doi.org/10.56286/ntujet.v1i1.50

Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A deep learning approach for network intrusion detection system. Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS),

Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., & Qiu, D. (2014). Security of the Internet of Things: perspectives and challenges. *Wireless Networks*, *20*, 2481-2501.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., & Cummings, R. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, *14*(1–2), 1-210.

Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. *Future generation computer systems*, *82*, 395-411.

Khanji, S., Iqbal, F., & Hung, P. (2019). ZigBee security vulnerabilities: Exploration and evaluating. 2019 10th international conference on information and communication systems (ICICS),

Kim, J., Shin, Y., & Choi, E. (2019). An intrusion detection model based on a convolutional neural network. *Journal of Multimedia Information System*, *6*(4), 165-172.

Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *Computer*, *50*(7), 80-84.

Kollipara, V. N. H., Kalakota, S. K., Chamarthi, S., Ramani, S., Malik, P., & Karuppiah, M. (2023). Timestamp Based OTP and Enhanced RSA Key Exchange Scheme with SIT Encryption to Secure IoT Devices. *Journal of Cyber Security and Mobility*, *12*(01), 77–102. https://doi.org/10.13052/jcsm2245-1439.1214

Kumar, R. (2023, 25-27 Aug. 2023). Establishment of Telemedicine Architecture in IoT Based Smart Home Security System for Health Monitoring System. 2023 3rd Asian Conference on Innovation in Technology (ASIANCON),

Kundu, S., Karmakar, A., & Verbauwhede, I. (2023). On the Masking-Friendly Designs for Post-Quantum Cryptography. *arXiv preprint arXiv:2311.08040*.

Lee, R. M., Assante, M. J., & Conway, T. (2016). Defense Use Case: Analysis of the cyber attack on the Ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, *388*(3), 1-29.

Li, J., Li, Y., Ding, C., Yu, J., & Ren, Y. (2022, 25-27 Nov. 2022). Identity-based Secure and Efficient Intelligent Inference Framework for IoT-Cloud System. 2022 IEEE 13th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP),

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, *37*(3), 50-60.

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., & Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *22*(3), 2031-2063.

Lu, R., Zhu, H., Liu, X., Liu, J. K., & Shao, J. (2014). Toward efficient and privacy-preserving computing in big data era. *IEEE Network*, *28*(4), 46-50.

Malhotra, A. (2023, 27-29 Jan. 2023). Blend CAC: Integration for the Blockchain for Distributed Potential Network Access for the Internet of Things. 2023 International Conference on Artificial Intelligence and Smart Communication (AISC),

Mazhar, N., Salleh, R., Zeeshan, M., & Hameed, M. M. (2021). Role of Device Identification and Manufacturer Usage Description in IoT Security: A Survey. *Ieee Access*, *9*, 41757-41786. https://doi.org/10.1109/ACCESS.2021.3065123

Mazlan, M., Zakaria, N. A., & Abidin, Z. Z. (2020). SECURITY CHALLENGES IN 6LOWPAN PROTOCOL FOR INTERNET OF THINGS: A REVIEW.

Mehmood, Y., Ahmad, F., Yaqoob, I., Adnane, A., Imran, M., & Guizani, S. (2017). Internet-of-things-based smart cities: Recent advances and challenges. *IEEE Communications Magazine*, *55*(9), 16-24.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, *17*(3), 12-22.

Meidan, Y., Bohadana, M., Shabtai, A., Ochoa, M., Tippenhauer, N. O., Guarnizo, J. D., & Elovici, Y. (2017). Detection of unauthorized IoT devices using machine learning techniques. *arXiv preprint arXiv:1709.04647*.

Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT express*, *5*(1), 1-7.

Meng, Y., Zhu, H., & Shen, X. (2023). Wireless Signal Based Two-Factor Authentication at Voice Interface Layer. In *Security in Smart Home Networks* (pp. 77-106). Springer.

Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.-R., & Tarkoma, S. (2017). Iot sentinel: Automated device-type identification for security enforcement in iot. 2017 IEEE 37th international conference on distributed computing systems (ICDCS),

Naik, N. (2017, 11-13 Oct. 2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. 2017 IEEE International Systems Engineering Symposium (ISSE),

Ngouen, M., Rahman, M. A., Prabakar, N., Uluagac, S., & Njilla, L. (2023, 23-25 Oct. 2023). Q-SECURE: A Quantum Resistant Security for Resource Constrained IoT Device Encryption. 2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS),

Ojo, M. O., Giordano, S., Procissi, G., & Seitanidis, I. N. (2018). A Review of Low-End, Middle-End, and High-End Iot Devices. *Ieee Access*, *6*, 70528-70554. https://doi.org/10.1109/ACCESS.2018.2879615

Otoum, Y., Liu, D., & Nayak, A. (2022). DL-IDS: a deep learning–based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, *33*(3), e3803. https://doi.org/https://doi.org/10.1002/ett.3803

Padgette, J., Bahr, J., Batra, M., Holtmann, M., Smithbey, R., & Chen, L. (2017). Guide to Bluetooth Security (NIST Special Publication 800-121 Revision 2). *National Institute of Standards and Technology, Gaithersburg, MD*.

Qiu, H., Dong, T., Zhang, T., Lu, J., Memmi, G., & Qiu, M. (2021). Adversarial Attacks Against Network Intrusion Detection in IoT Systems. *IEEE Internet of Things Journal*, *8*(13), 10327-10335. https://doi.org/10.1109/JIOT.2020.3048038

Raza, S., Shafagh, H., Hewage, K., Hummen, R., & Voigt, T. (2013). Lithe: Lightweight secure CoAP for the internet of things. *IEEE Sensors Journal*, *13*(10), 3711-3720.

Raza, U., Kulkarni, P., & Sooriyabandara, M. (2017). Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, *19*(2), 855-873.

Samaila, M. G., Neto, M., Fernandes, D. A. B., Freire, M. M., & Inácio, P. R. M. (2018). Challenges of securing Internet of Things devices: A survey. *Security and Privacy*, *1*(2), e20. https://doi.org/https://doi.org/10.1002/spy2.20

Shafiq, M. Z., Ji, L., Liu, A. X., Pang, J., & Wang, J. (2013). Large-scale measurement and characterization of cellular machine-to-machine traffic. *IEEE/ACM transactions on Networking*, *21*(6), 1960-1973.

Shelby, Z., Chakrabarti, S., Nordmark, E., & Bormann, C. (2012). *Neighbor discovery optimization for IPv6 over low-power wireless personal area networks (6LoWPANs)* (2070-1721).

Sicari, S., Rizzardi, A., Miorandi, D., Cappiello, C., & Coen-Porisini, A. (2016). Security policy enforcement for networked smart objects. *Computer Networks*, *108*, 133-147.

Singh, S., Hosen, A. S. M. S., & Yoon, B. (2021). Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network. *Ieee Access*, *9*, 13938-13959. https://doi.org/10.1109/ACCESS.2021.3051602

Tschirschnitz, M. v., Peuckert, L., Franzen, F., & Grossklags, J. (2021, 24-27 May 2021). Method Confusion Attack on Bluetooth Pairing. 2021 IEEE Symposium on Security and Privacy (SP)

Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *Ieee Access*, *7*, 41525-41550.

Weng, D. (2023, 12-14 Oct. 2023). Performance and Energy Evaluation of Lightweight Cryptography for Small IoT Devices. 2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON),

Xu, R., Chen, Y., Blasch, E., & Chen, G. (2018). BlendCAC: A Smart Contract Enabled Decentralized Capability-Based Access Control Mechanism for the IoT. *Computers*, *7*(3).

Yang, Y.-S., Lee, S.-H., Wang, J.-M., Yang, C.-S., Huang, Y.-M., & Hou, T.-W. (2023). Lightweight Authentication Mechanism for Industrial IoT Environment Combining Elliptic Curve Cryptography and Trusted Token. *Sensors*, *23*(10).

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, *5*, 21954-21961.

Zhang, C., Yuan, X., Zhang, Q., Zhu, G., Cheng, L., & Zhang, N. (2021, 20-22 Oct. 2021). Privacy-Preserving Neural Architecture Search Across Federated IoT Devices. 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom),

Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent manufacturing in the context of industry 4.0: a review. *Engineering*, *3*(5), 616-630.

Zhou, X., Liang, W., Li, W., Yan, K., Shimizu, S., & Wang, K. I. K. (2022). Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System. *IEEE Internet of Things Journal*, *9*(12), 9310-9319. https://doi.org/10.1109/JIOT.2021.3130434

BÖLÜM XII

Rastgele Sayı Üreteç Testleri

Abdullah SEVİN[1]
İhsan Eren DELİBAŞ[2]
Ethem Belka ŞAHİN[3]
Kerem Can ÖZKUL[4]

Giriş

Rastgele sayı üreteçleri, belirli bir düzen veya örüntü olmaksızın, rastgele sayılar üreten matematiksel algoritmalar veya cihazlardır. Bu sayılar genellikle istatistiksel olarak bağımsız ve eşit olasılıkla dağılmıştır. Rastgele sayı üreteçleri, birçok bilim, mühendislik ve bilgi işlem alanında çeşitli uygulamalara sahiptir. Genel olarak, rastgele sayı üreteçleri, istatistiksel analizler, finans

[1] Dr. Öğr. ÜYESİ, Sakarya Üniversitesi, asevin@sakarya.edu.tr
[2] Bilgisayar Müh. Yüksek Lisans, Sakarya Üniversitesi, eren.delibas@ogr.sakarya.edu.tr
[3] Bilgisayar Müh. Yüksek Lisans, Sakarya Üniversitesi, ethem.sahin1@ogr.sakarya.edu.tr
[4] Bilgisayar Müh. Yüksek Lisans, Sakarya Üniversitesi, kerem.ozkul@ogr.sakarya.edu.tr

uygulamaları, simülasyonlar, güvenlik uygulamaları ve birçok bilgisayar bilimi alanında (bilgisayar oyunları, kriptografi, bilgisayar grafikleri, otomatik test sistemleri) yaygın olarak kullanılmaktadır. Bir rastgele sayı üretecinin test edilmesi için istatistiksel testler, düzgünlük testleri, bağımsızlık testleri ve spektral testler gibi testler uygulanmaktadır. Bu testler, bir rastgele sayı üretecinin ne kadar rastgele olduğunu değerlendirmek için kullanılan yaygın yöntemlerdir.

Rastgele bir bit dizisi, her bir yüzünün "0" ve "1" olarak etiketlendiği ve eşit olasılıkla "0" veya "1" üretebilen adil bir madeni paranın atışlarının sonucu olarak ifade edilebilir. Ayrıca, para atışları birbirinden bağımsızdır: herhangi bir madeni para atışının sonucu, sonra gerçekleşecek madeni para atışlarını etkilemez. Bu adil olasılıklı madeni para, "0" ve "1" değerlerinin rastgele dağıldığı ([0,1] arasında düzgün dağılıma sahip) mükemmel bir rastgele bit akışı üretecidir. Dizi elemanlarının tümü birbirinden bağımsız olarak üretilir ve bir sonraki elemanın değeri önceden tahmin edilemez. Fakat, kriptografik amaçlar için adil paraların kullanımı (gerçek rastgele sayı üreteci) pratik değildir. Bunun yerine gerçek rastgele sayı üreteçlerin istatistiksel özelliklerini barındıran sözde rastgele sayı üreteçleri kullanmak pratik uygulamalarda tercih edilmektedir. Bir diziyi gerçekten rastgele bir diziyle karşılaştırmak ve değerlendirmek için çeşitli istatistiksel testler uygulanmaktadır. Bir sayı dizisinin rastgele olması, olasılıksal bir özelliktir; yani, bir rastgele dizinin özellikleri olasılık terimleriyle karakterize edilir ve açıklanır. Sonsuz sayıda olası istatistiksel test bulunmaktadır. Bir dizinin rastgele olup olmadığını değerlendirmek için bu kadar çok test olduğundan, sınırlı bir test kümesi tam kapsamlı bir test kümesi olarak tanımlanamaz. Ayrıca, istatistiksel testlerin sonuçları, belirli bir üreteç hakkında yanlış sonuçlardan kaçınmak için dikkatlice ve tedbirli bir şekilde yorumlanmalıdır. Bir istatistiksel test, belirli bir nihai hipotezi ($H0$) test etmek üzere formüle edilir. Test edilen nihai hipotez, test edilen dizinin rastgele olduğudur. Bu nihai hipoteze ilişkin olarak, alternatif hipotez ($H1$) ise dizinin rastgele olmadığıdır. Her uygulanan test için, üretilen diziyi temel alan nihai hipotezi

kabul etme veya reddetme kararı veya sonucu türetilir, yani, üretecin rastgele değerler üretip üretmediğine dair hipotez kabul edilir veya reddedilir. Her bir test sırasında, veri üzerinde (test edilen dizi) bir test istatistik değeri hesaplanır. Bu test istatistik değeri, kritik değerle karşılaştırılır. Eğer test istatistik değeri kritik değeri aşarsa, rastgelelik için nihai hipotez reddedilir. Aksi takdirde, nihai hipotez (rastgelelik hipotezi) reddedilmez yani, hipotez kabul edilir (NIST-800-22, 2001).

İstatistiksel Testler ve Uygulamaları

Rastgele sayılar, belirli istatistiksel özelliklere sahip olmalıdır. Bunları test etmek için belirli testler tanımlanmıştır. Bu test örneklerini incelerken bazı kısa veri girdi örnekleri kullanılacaktır. Kapsamlı bazı testler için kullanılmak üzere Visual Studio 2022 kullanılarak C# nesneye dayalı programlama dilinde *random* fonksiyonu ile üretilen ondalık sayı dizisi ikilik taban formatına dönüştürülerek burada kullanılacak veri dizisi (ε) oluşturulmuştur. Tablo 1'de üretilen bit dizisi verilmiştir. Fakat bazı testler için gerekli bit sayısı yeterli olmadığından (ör: en az 1,000,000 sayı gerektiren testler) belirli testlerde farklı sayı dizileri de kullanılmıştır.

Tablo 3. Rastgele Oluşturulmuş Sayı Dizisi (ε)

Ondalık	12 97 27 12 40 17 14 78 49 83 26 86 35 79 56 35 54 40 67 2
İkilik	01100001100110001000100100000111101111011111111101111001001011001110000010010000110100010011011111111000111110100010000000010000

Bloktaki En Uzun Birlerin Akış Testi

Bu testin amacı bit setinin ($M - bit$) içerisindeki en uzun birler (1) ya da en uzun sıfırlar (0) serisinin ölçülmesidir (Anant ve Stavros, 1994). Test edilmesi istenilen veri setinin içerisindeki birler ya da sıfırlar serilerinin, beklenilen seri uzunluğu ile karşılaştırılıp uyumluluğunu test etmektir. Eğer birler serisinde bir uyumsuzluk

varsa aynı zamanda sıfırlar serisinde de uyumsuzluk olduğu anlamına geldiği için bu testte sadece birler serisi incelenecektir.

Veri dizisindeki her bir parçanın bit uzunluğu (M) olsun. Tablo 4'de M değerine göre test için gerekli minimum bit sayısı (n) verilmiştir.

Tablo 4. M değerine göre gerekli minimum n değeri

Gerekli minimum n bit:	128	6272	750,000
M:	8	128	10^4

Dizi M-bit bloklarına ayırılır. Bölünen her bir bloktaki en uzun birler serisi kategorilere ayırılır. Bu kategoriler v_i olarak adlandırılır. Her bir v_i kategorisinde M değerlerine göre en uzun 0 veya 1 akışını temsil eder (Tablo 3).

Tablo 5. M-bitlerine göre v_i hücreleri

v_i	$M = 8$	$M = 128$	$M = 10^4$
v_0	≤1	≤4	≤10
v_1	2	5	11
v_2	3	6	12
v_3	≥4	7	13
v_4		8	14
v_5		≥9	15
v_6			≥16

Ki-kare istatistiği Denklem 1'e göre hesaplanır. Örnek veri dizisi 128 bit ve M değeri 8 bit olduğundan $K = 3$ ve $N = 16$ alınır. Tablo 4'te $K = 3$ ve $N = 16$ için teorik olasılık değerleri verilmiştir.

$$X^2(obs) = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i} \quad (1)$$

Tablo 6. v sınıfları ve teorik olasılıklar ($K = 3, M = 8, N = 16$)

Sınıf	Teorik olasılık
v_0	$\pi_0 = 0.2148$
v_1	$\pi_1 = 0.3672$
v_2	$\pi_2 = 0.2305$
v_3	$\pi_3 = 0.1875$

$$P\text{-}değeri = igamc\left(\frac{K}{2}, \frac{X^2(obs)}{2}\right) \qquad (2)$$

Denklem 2'de gösterildiği gibi $P\text{-}değeri$ tamamlanmamış gama fonksiyonuna $(igamc)$, K değeri ve Ki-kare istatistiği parametre olarak girildiğinde 0.0993047 hesaplanır. P-değeri ≥ 0.01 ise dizi rastgelelik özelliklerine sahip anlamındadır. Bu sonuç doğrultusunda test edilen rastgele sayı dizisi testi geçmiştir.

İkili Matris Sıralama Testi

Testin odak noktası, tüm dizinin ayrık alt-matrislerinin sıralanmasıdır. Bu testin amacı, orijinal dizinin sabit uzunluktaki alt dizileri arasındaki doğrusal bağımlılığı kontrol etmektir (Kovalenko, 1972). Rastgele sayı dizisi sıralı olarak M•Q bitlik ayrık bloklara bölünür; $N = \left|\frac{n}{M*Q}\right|$ adet blok oluşur. Her matristeki satır sayısı (M), her matristeki sütun sayısı (Q) ile temsil edilir. Matrisin her satırı, orijinal dizi ε'nin ardışık Q-bit blokları ile doldurulur.

Örneğin: n = 20, M = Q = 3, ε= 01011001001010101101 olsun $N = \left|\frac{n}{3*3}\right| = 2$ buna göre oluşan matris: {0 1 0, 1 1 0, 0 1 0} ve {0 1 0, 1 0 1, 0 1 1}. İlk matris, birinci satırdaki ilk üç biti, ikinci satırdaki ikinci üç biti ve üçüncü satırdaki üçüncü üç biti içerir. İkinci matris de benzer şekilde, dizideki sonraki dokuz biti kullanarak oluşturulur.

Her matrisin ikili sıralaması (R_i) belirlenir. F_M , $R_i = M$ olan matris sayısı, F_{M-1}, $R_i = M - 1$ olan matris sayısı, $(N - F_M - F_{M-1})$ ise geriye kalan matris sayısıdır. Ki-kare istatistiği Denklem 3'teki eşitlik ile hesaplanır.

$$X^2(obs) = \frac{(F_M - 0.2888\mathrm{N})^2}{0.2888\mathrm{N}} + \frac{(F_{M-1} - 0.5776\mathrm{N})^2}{0.5776\mathrm{N}} + \frac{(\mathrm{N} - F_M - F_{M-1} - 0.1336)^2}{0.1336\mathrm{N}} \quad (3)$$

P-değeri ise Denklem 4'e göre hesaplanır ve eğer 0.01 den küçük çıkarsa rastgele sayı dizisi testi geçememektedir.

$$\text{P-değeri} = \mathrm{e}^{-x^2(obs)/2} \quad (4)$$

Veri seti olarak e'nin açılımındaki ilk 100.000 ikili basamak kullanılırsa uygulanan test sonucunda P-değeri = 0.532069 olarak bulunur. Bu sonuç doğrultusunda test edilen rastgele sayı dizisi uyumlu rastgele sayı üretmiştir kabul edilir.

Ayrık Fourier Dönüşüm Testi

Bu testin amacı, test edilen dizideki periyodik özellikleri (birbirine yakın tekrarlayan desenleri) tespit etmektir. Amaç, %95 eşiğini aşan tepe noktalarının sayısının %5'ten önemli ölçüde farklı olup olmadığını tespit etmektir (Bracewell, 1986). İlk olarak ε (girdi dizisi) içerisindeki 0'lar -1'e dönüştürülür.

Örneğin ε = 0011010010 ise X = -1, -1, 1, 1,0, 1, -1, -1, 1, -1 olarak oluşturulur.

Oluşan X dizisine Ayrık Fourier Dönüşüm uygulanır ve $S = AFD(X)$ üretilir. Bu, farklı frekanstaki bit dizisinin periyodik bileşenlerini temsil eden bir dizi kompleks değişkeni oluşturur. $M = modulus(S') \equiv |S'|$ hesaplanır. Burada S', S dizisinde ilk $n/2$ elemandan oluşan alt dizidir ve modül fonksiyonu bir dizi tepe yüksekliği üretir.

Tepe yüksekliği eşik değeri (T) Denklem 5'teki gibi hesaplanır. Testten elde edilen değerlerin 95%'i T'yi aşmamalıdır.

$$T = \sqrt{(log\frac{1}{0.05})n} \qquad (5)$$

T değerinin altında beklenen teorik tepe sayısı $N_0 = 0.95\text{n}/2$ işlemi ile hesaplanır. M içinde T'den küçük olan gerçek gözlemlenen tepe sayısı (N_1) hesaplanır. Sonra normalleştirilmiş fark Denklem 6'daki gibi hesaplanır (d).

$$d = \frac{(N_1 - N_0)}{\sqrt{n(0.95)(0.05)/4}} \qquad (6)$$

Son olarak P-değeri Denklem 7'de gösterildiği gibi tamamlayıcı hata fonksiyonu $(erfc)$ ile elde edilmektedir.

$$\text{P-değeri} = erfc(\frac{|\text{d}|}{\sqrt{2}}) \qquad (7)$$

P-değeri eğer 0.01 den küçükse sayı dizisi testi geçmemiş sayılır. Belirtilen veri seti kullanarak uygulanan test sonucunda P-değeri 0.3303900488487933 olarak bulunmuştur. Bu sonuç doğrultusunda test edilen rastgele sayı dizisi testi geçmiştir.

Seri Testi

Bu testin amacı, 2^m m-bitlik örtüşen desenlerin sayısının rastgele bir dizilim için beklenenle yaklaşık olarak aynı olup olmadığını belirlemektir. Rastgele dizilimler düzgün dağılıma sahiptir; yani, her m-bit deseninin diğer her m-bit desen gibi görünme olasılığı aynı oranda çıkması beklenmektedir (Good, 1953).

Test edilecek ε dizisinin başından m-1 adet bit dizinin sonuna eklenir. Örneğin ε=10100110 ve m=3 olduğunu varsayarsak ε' = 1010011010 şeklinde yeni bir ε dizisi oluşturulur. Tüm örtüşen m-bit blokların frekans sayımı yapılır. m-1 bit blokları için $i_1 \ldots i_m$, m-2 bit blokları için $i_1 \ldots i_{m-1}$ frekansları oluşturulur.

Örneğin 3-bit (m) bloklar için $v_{000} = 0, v_{001} = 1, v_{010} = 2, v_{011} = 1, v_{100} = 1, v_{101} = 2,\ v_{110} = 1, v_{111} = 0$ olarak hesaplanır. 2-bit $(m-1)$ bloklar için $v_{00} = 1, v_{01} = 3, v_{10} = 4, v_{11} = 1$ olarak hesaplanır . 1-bit $(m-2)$ bloklar için $v_0 = 5, v_1 = 5$ değerleri bulunur.

Frekansların eşleşme sonuçları Denklem 8'de gösterildiği gibi hesaplanır. $m, m-1, m-2$ şeklinde 3 adet hesaplama yapılır.

$$\psi_m^2 = \frac{2^m}{n} \sum_{i_1 \ldots i_m} (v_{i_1 \ldots i_m} - \frac{n}{2^m})^2 \qquad (8)$$

Frekansların eşleşme sonuçlarının değişimleri Denklem 9 ve 10'daki gibi hesaplanır.

$$\nabla\psi_m^2 = \psi_m^2 - \psi_{m-1}^2 \qquad (9)$$

$$\nabla^2\psi_m^2 = \psi_m^2 - 2\psi_{m-1}^2 + \psi_{m-2}^2 \qquad (10)$$

P-değerleri ise tamamlanmamış gama fonksiyonu yardımıyla hesaplanır (Denklem 11 ve 12).

$$\text{P-değeri1} = \text{igamc}(2^{m-2}, \nabla\psi_m^2) \qquad (11)$$

$$\text{P-değeri2} = \text{igamc}(2^{m-3}, \nabla^2\psi_m^2) \qquad (12)$$

P-değeri eğer 0.01 den küçükse testin başarısız olduğu sonucuna varılmaktadır. Tablo 1'de belirtilen veri seti kullanarak uygulanan test sonucunda P-değeri1 = 0.4989610874592239, P-değeri2 = 0.49853075529672125 olarak bulunmuştur. Bu sonuç doğrultusunda test edilen rastgele sayı dizisinin testi başarılı olarak geçtiği sonucuna varılmaktadır.

Yaklaşık Entropi Testi

Yaklaşık entropi karakteristiği (Pincus ve Singer, 1996), iki ardışık/bitişik uzunluktaki (m ve m+1) üst üste binen blokların sıklığını tanımlar ve rastgele bir dizi için beklenen sonuçla karşılaştırılarak test edilir. Bu testte ilk blok uzunluğu m, ikinci blok uzunluğu ise m+1 olarak ifade edilir. Test edilecek olan ε bit dizisi,

n uzunluk değeri ile örtüşen m-bit diziler oluşturmak için veri genişletilir. Bu işlem ile dizinin başından m-1 bit kopyalanıp verinin sonuna eklenir. Örneğin: m=3, ε = 1011000111 n=10 ise yeni ε = 101100011110 şeklinde oluşur.

N üzerinden örtüşen blokların frekans sayımı yapılır. Örneğin ε = 101100011110 bit verisi 101, 011, 110, 100, 000, 001, 011, 111, 111, 110 şeklinde bloklara bölünür. 2^m adet blok oluşur. Oluşan bloklardan aynı olanların sayıları n uzunluğuna bölünerek C_i^m elde edilir (Denklem 13).

$$C_i^m = \frac{i}{n} \qquad (13)$$

Elde edilen ondalıklı sayılarla Denklem 14'teki formülde gösterildiği gibi $\phi^{(m)}$ hesaplanır.

$$\phi^{(m)} = \sum_{i=0}^{2^m-1} \pi_i log \pi_i, \qquad \pi_i = C_j^3 \; ve \; j = \log_2 i \qquad (14)$$

Blok uzunluğu m+1 arttırılarak $\phi^{(m+1)}$ sonucu hesaplanır. Denklem 15 ile Ki-kare istatistik (X^2) değişkeni hesaplanır. P-değeri Denklem 16'daki tamamlanmamış gama fonksiyonu yardımı ile parametreler girilerek hesaplanır.

$$X^2 = 2n[log\ 2 - ApEn(m)], \qquad ApEn(m) = \phi^{(m)} - \phi^{(m+1)} \qquad (15)$$

$$P\text{-değeri} = \text{igamc}\left(2^{m-1}, \frac{X^2}{2}\right) \qquad (16)$$

P-değeri eğer 0.01 den küçükse test başarısız olmuş demektir. Belirtilen veri seti kullanarak uygulanan test sonucunda P-değeri 0,044528621589481954 olarak bulunmuştur. Bu sonuç doğrultusunda test edilen rastgele sayı dizisi ideal rastgele sayı özelliklerine uyumlu olduğu görülmektedir.

Frekans Testi

Rastgele sayı dizilerinin rastgelelik testlerinden birisi olan frekans testi, adından da anlaşılacağı üzere dizideki sıfırların ve birlerin frekans değerlendirmesini yaparak rastgelelik analizi yapmamıza yardımcı olan bir test türüdür (Chung, 1979). Rastgele sayı dizilerinin rastgelelik özelliklerini değerlendirmemize yardımcı olan ve istatistiksel bir test türü olan frekans testi, bir dizideki sıfırların ve birlerin frekanslarını inceleyerek dizideki elemanların dağılımının düzgün olmasını test eder.

S_{obs}: Dizideki X_i (burada, $X_i = 2\varepsilon - 1 = \pm 1$) değerlerinin toplamının mutlak değeri, dizinin uzunluğunun karekköküne bölünerek hesaplanır (Denklem 17). Her bit değeri ($X_i = 2\varepsilon - 1$) formül gereği 1 bitleri 1 olarak 0 bitleri -1 olarak hesaplanır ve ideal değer olarak toplamının 0 çıkması gerekmektedir.

$$S_{obs} = \frac{|S_n|}{\sqrt{n}} \qquad (17)$$

$$P\text{-değeri} = erfc\left(\frac{S_{obs}}{\sqrt{2}}\right) \qquad (18)$$

Denklem 18'de görüldüğü gibi P-değeri tamamlayıcı hata fonksiyonu ile hesaplanmaktadır. Buradaki P-değeri 0.01'den küçük ise dizideki 0 ve 1 bitlerinin dağılımı rastgele olarak kabul edilmez. Fakat P-değeri 0.01'den büyük veya eşit ise dizideki 0 ve 1 bitlerinin dağılımı rastgele olarak kabul edilir. Tablo1'deki bit dizisi belirtilen formüllere göre frekans testi gerçekleştirildiğinde P-değeri 0.4795001221869535 olarak bulunmaktadır (n=128). Bu değer 0.01'den büyük olduğu için bu bit dizisindeki 0 ve 1 bitlerinin dağılımı rastgele olarak kabul edilir.

Blok Frekans Testi

Bu test, frekans testine benzer bir test türüdür. Frekans testi, rastgele sayı dizilerinin belirli bir dağılıma sahip olup olmadığını değerlendirmek için kullanılan bir istatiksel yöntemdir. Blok frekans testi ise, rastgele sayı dizilerini belirli uzunluktaki bloklara bölerek

bu m-bit bloklar içindeki bir bitlerinin oranını inceler (Abramowitz ve Stegun, 1964).

Blok terimi, rastgele sayı dizisinin belirli bir uzunluktaki kısmının belirli bir bölümünü temsil eder. Örneğin 16 bitlik bir blok, rastgele sayı dizisindeki her 16 biti ifade etmektedir. Bu blokların sahip olduğu bitlerin istatistiksel olarak beklenen özelliklere sahip olup olmadığı incelenir. Genel frekans dağılımının m-bit bloklar içinde ideal dağılıma uygun olması beklenmektedir.

Örneğin: n = 32, M = 4, ε= 11010101 00111010 10101010 01101100 olsun. Toplamda 8 adet blok oluşturulur (N = 8). Bu diziyi 4 bitlik bloklara ayıracak olursak;

- Blok 1: 1101
- Blok 2: 0101
- Blok 3: 0011
- Blok 4: 1010
- Blok 5: 1010
- Blok 6: 1010
- …

Şeklinde devam eder. Bu blokların içindeki bitlerin dağılımı incelenir. Bit dizisini bloklara böldükten sonra, her bloktaki birlerin oranı Denklem 19'a göre hesaplanır. Bu formülde $1 \leq i \leq n$ olmalıdır.

$$\pi_i = \frac{\sum_{j=l}^{M} \varepsilon_{(i-l)M+j}}{M} \quad (19)$$

Daha sonra Ki-kare istatistiği (χ^2) Denklem 20'ye göre hesaplanmaktadır.

$$\chi^2(obs) = 4M \sum_{i=1}^{N} \left(\pi_i - \frac{1}{2}\right)^2 \quad (20)$$

Rastgelelik ölçütü olan P-değeri tamamlanmamış gama fonksiyonu ile Denklem 21'deki gibi hesaplanır. P-değeri 0.01'den küçük olması durumunda verilen dizi rastgelelik testinde başarısız olarak kabul edilir. Eğer değer 0.01'den büyük veya eşit olursa verilen dizi rastgelelik testinde başarılı olarak kabul edilir.

$$P\text{-değeri} = \text{igamc}\,(N/2, \chi^2(obs)/2) \qquad (21)$$

Tablo 1'deki bit dizisi belirtilen formüllere göre blok frekans testine tabi tutulduğunda P-değeri 0.4795001221869535 olarak bulunmaktadır. Bu değer 0.01'den büyük olduğu için dizi rastgelelik testinde başarılı olarak kabul edilir.

Akış Testi

Akış testi (Bradley, 1960), sayı dizisindeki 0 ve 1'lerin arasındaki dalgalanmaların kontrol edilmesini amaçlar. Akış testi, sıfırlar ve birler arasındaki dalgalanmanın çok hızlı ya da çok yavaş olup olmadığını tanımlar. Dizideki toplam akış sayısı, aynı bitlerin kesintisiz bir dizisidir. Uzunluğu k olan bir akış tam olarak k tane aynı bitten oluşur ve öncesi ve sonrası zıt değerde olan bir bit ile sınırlanır. Akış testinin amacı, birlerin ve sıfırların çeşitli uzunluklardaki akış sayısının, rastgele bir dizi için beklendiği gibi olup olmadığını belirlemektir. Akış testi için ön koşul olarak bir frekans testinin gerçekleştirilmesi gerekmektedir.

Giriş dizisindeki birlerin ön test oranı (π) Denklem 22'ye göre hesaplanır.

$$\pi = \frac{\sum_j \varepsilon_j}{n} \qquad (22)$$

Örneğin, eğer ε = 0011101011 ise, o zaman n=10 ve $\pi = 6/10 = 3/5$ olur.

Akış testinin yapılabilmesi için önceden gerçekleşen frekans testinden geçmesi gerekmektedir. Eğer $|\pi - 1/2| \geq \tau$ koşulu doğru ise akış testi yapılmaya gerek yoktur. Eğer test uygulanabilir değilse,

P-değeri 0.0000 olarak ayarlanır. Örneğin, $\tau = 2/\sqrt{10} \approx 0.63246$ $|\pi - 1/2| = |3/5 - 1/2| = 0.1 < \tau$ ise test gerçekleştirilmez.

Gözlemlenen test istatistiği Denklem 23'e göre hesaplanır. Burada $\varepsilon_k = \varepsilon_{k+1}$ ise r(k)=0, aksi takdirde r(k)=1'dir. Yani mevcut bit ile sonra gelen bit aynı ise r(k)=0, bitler değişiyor ise r(k)=1 alınır.

$$V_n(obs) = \sum_{k=1}^{n-1} r(k) + 1 \qquad (23)$$

Örneğin ε = 0110101101 olduğunda $V_n(obs)$ = (1+0+1+1+1+1+0+1+1) + 1 = 8 olur. P-değeri Denklem 24'te görüldüğü gibi tamamlayıcı hata fonksiyonu ile hesaplanır.

$$P\text{-değeri} = erfc\left(\frac{|V_n(obs) - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right) \qquad (24)$$

Hesaplanan P-değeri < 0,01 ise, rastgele sayı dizisi akış testinden başarısız kabul edilir. Aksi takdirde, akış testini başarılı bir şekilde geçti kabul edilir. Tablo 1'deki bit dizisi belirtilen formüllere göre akış testine tabi tutulduğunda P-değeri 0.02365161665 olarak bulunmaktadır. Bu değer 0.01'den büyük olduğu için dizi akış testinde başarılı olarak kabul edilir.

Örtüşmeyen Şablon Eşleştirme Testi

Bu test, önceden belirlenmiş hedef dizilerin oluşum sayılarını inceler (Barbour ve ark., 1992). Bu testin amacı, belirli bir periyodu olmayan desenin üretiminin incelenmesini içerir. Bu test için, belirli bir m-bit deseni aramak için bir m-bit pencere kullanılır. Eğer desen bulunamazsa, pencere bir bit kaydırılır. Eğer desen bulunursa, pencere bulunan desenden sonraki bit konumuna sıfırlanır ve arama devam eder.

Dizi *M* uzunluğunda *N* bağımsız bloğa bölünür. Örneğin, ε = 10100100101110010110 ise, n = 20'dir. Eğer *N* = 2 ve *M* = 10 ise, o zaman bloklar; {1010010010} ve {1110010110} olacaktır.

W_j $(j = 1, \ldots, N)$, B'nin (şablon) j. bloğu içerişinde kaç kez geçtiğini gösterir. İndis elemanı j = 1,...,N, eşleşme araması için, dizi üzerinde m bitlik bir pencere oluşturarak ilerler, o pencere içindeki bitleri şablonla karşılaştırır. Eşleşme yoksa, pencere bir bit kayar. Örneğin, m = 3 ise ve mevcut pencere 3. ve 5. bit konumlarına bakıyorsa ve eşleşme yoksa sonraki pencere 4. ve 6. bit konumlarına bakacaktır. Eğer bir eşleşme varsa, pencere m-bit ötelenir. Örneğin mevcut (başarılı) pencere 3. ve 5. bit konumlarında eşleşiyor ise W_j bir arttırılır ve bir sonraki pencere 6. ve 8. bit konumlarına bakacaktır. Tablo 5'te örnek verilen veri dizisinin ayrılan blokları ve aranan şablon ile örtüşen kısımları gösterilmiştir.

Tablo 5. Örtüşmeyen Şablon Testi (m=3, B=001)

Bit konumları	Blok 1		Blok 2	
	Bitler	W_1	Bitler	W_2
1-3	101	0	111	0
2-4	010	0	110	0
3-5	100	0	100	0
4-6	001 (eşleşti)	1	001 (eşleşti)	1
5-7	atlandı		atlandı	
6-8	atlandı		atlandı	
7-9	001 (eşleşti)	2	011	1
8-10	010	2	110	1

Böylece, $W_1 = 2$ ve $W_2 = 1$ olur. Girdi dizisinin μ teorik ortalaması ve σ^2 varyansı Denklem 25'e göre hesaplanır.

$$\mu = \frac{M - m + 1}{2^m} \quad ve \quad \sigma^2 = M\left(\frac{1}{2^m} - \frac{2m - 1}{2^{2m}}\right) \qquad (25)$$

Sonuç olarak $\mu = (10 - 3 + 1)/2^3 = 1$ ve $\sigma^2 = 10 \cdot \left(\frac{1}{2^3} - \frac{2 \cdot 3 - 1}{2^{2 \cdot 3}}\right) = 0.46875$ olur. Ki-kare istatistiğini hesaplamak için Denklem 26 kullanılır.

$$\chi^2(obs) = \sum_{j=1}^{N} \frac{\left(W_j - \mu\right)^2}{\sigma^2} \qquad (26)$$

Yukarıda verilen örnek için $\chi^2(obs) = \frac{(2-1)^2 + (1-1)^2}{0.46875} = \frac{1+0}{0.46875} = 2.133333$ olarak hesaplanır. P-değeri ise tamamlanmamış gama fonksiyonu ile Denklem 27'deki gibi hesaplanır.

$$P\text{-}değeri = \left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right) \qquad (27)$$

Yukarıda verilen örnek için P-değeri = igamc $\left(\frac{2}{2}, \frac{2.133333}{2}\right) =$ 0.344154 olarak hesaplanır ve değer 0.01'den büyük çıktığı için testi geçer. Tablo 1'de verdiğimiz veri dizisi için test uygulanacak olursa P-değeri 8.830342779064649e-05 çıkmaktadır ve değer 0.01'in altında olduğu için testi geçememektedir.

Örtüşen Şablon Eşleştirme Testi

Örtüşen şablon eşleştirme testi, belirli bir bit desenini tespit etmek için kullanılan bir yöntemdir. Bu test, önceden belirlenmiş hedef dizelerinin ne kadar sıklıkla ortaya çıktığını analiz etmektedir. Burada, belirli bir bit desenini aramak için bir bit penceresi kullanılır. Eğer istenen desen bulunamazsa, pencere bir bit konum kaydırılarak arama işlemi sürdürülür. Desen bulunduğunda ise, pencere sadece bir bit kaydırılır ve arama işlemine devam edilir. Bu yöntem, bit desenlerinin bulunma sıklığını ve dağılımını analiz etmek için kullanılmaktadır (Chrysaphinou ve Papastavridis, 1988).

Dizi M uzunluğunda N bağımsız bloğa ayrılır. Örneğin, eğer girdi dizisi olarak ε=10111011110010110100011100101110111110000101101001

seçilir ise, n=50 olur. Eğer K=2, M=10 ve N=5 ise, beş blok oluşmaktadır: 1011101111, 0010110100, 0111001011, 1011111000 ve 0101101001. Her bir N bloğunda B'nin kaç kez oluştuğu hesaplanır. Eşleşmeleri arama işlemi, dizide m-bitlik bir pencere oluşturulur ve bu pencere içindeki bitlerin B (şablon) ile karşılaştırılması sonucunda eşleştirme olursa sayaç artırarak ilerler. Her blokta B'nin kaç kez oluştuğu, bir dizide $v_i(burada\ i = 0 \dots 5)$ tutulur. Örnek olarak B'nin hiç oluşmaması durumunda v_0 artırılır, B'nin bir kez oluşması durumunda v_1 artırılır… Yukarıdaki örnekte, eğer m=2 ve B=11 ise, ilk bloğun (1011101111) incelenmesi Tablo 6'daki gibi gerçekleştirilir.

Tablo 6. Örtüşen Şablon Testi

Bit konumları	Bitler	Tekrar sayısı
1-2	10	0
2-3	01	0
3-4	11 (eşleşti)	1
4-5	11 (eşleşti)	2
5-6	10	2
6-7	01	2
7-8	11 (eşleşti)	3
8-9	11 (eşleşti)	4
9-10	11 (eşleşti)	5

Ki-kare istatistiğinin hesaplanması için Denklem 28 kullanılır. Hamano ve Kaneko (2007) çalışmasında π_i değerlerinin detaylı hesaplamasına yer vermiştir ($\pi_0 = 0.364091, \pi_1 = 0.185959, \pi_2 = 0.139381, \pi_3 = 0.100571, \pi_4 = 0.070432,\ \pi_4 = 0.139865$).

$$\chi^2(obs) = \sum_{i=0}^{5} \frac{(v_i - N\pi_i)^2}{N\pi_i} \qquad (28)$$

P-değerini hesaplamak için Denklem 29 kullanılır ve tamamlanmamış gama fonksiyonu ile elde edilir. Yukarıdaki değerler için Ki-kare istatistiği $\chi^2(obs) = 3.167729$ ve P-değeri 0.274932 çıkar ve sonuç 0.01'den büyük çıktığı için testi geçmiştir.

$$P\text{-değeri} = igamc\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right) \qquad (29)$$

Örtüşen şablon eşleştirme testi için 10^6 bitten daha çok bit kullanılması önerilmektedir. Tablo 1'de verdiğimiz veri dizisi 128 bitten oluştuğu için onun yerine *e* sayısının ilk 1,000,000 biti için test uygulanacak olursa P-değeri 0.110434 çıkmaktadır ve değer 0.01'in üstünde olduğu için testi geçmektedir.

Maurer'in "Evrensel İstatistik" Testi

Maurer'in "Evrensel İstatistik" testi (Maurer, 1992), eşleşen desenler arasındaki bitlerin sayısını ölçen bir testtir. Testin amacı, bilgi kaybı olmadan dizinin sıkıştırılıp sıkıştırılamayacağını tespit etmektir. Önemli ölçüde sıkıştırılabilir bir dizinin rastgele olmadığı prensibine dayanmaktadır. n bitlik dizi (ε) iki bölüme ayrılır. Başlangıç segmenti Q adet L-bit örtüşmeyen bloklardan oluşur. İkinci kısım yani test segmenti K adet L-bit örtüşmeyen bloklardan oluşur. Dizinin sonunda kalan ve tam bir L-bit blok oluşturmayan bitler atılır.

Örneğin, ε = 01011010011101010111 ise, n = 20 olur. L = 2 ve Q = 4 ise, K = [n/L] - Q = [20/2]- 4 = 6 olur. Başlangıç segmenti 01011010; test segmenti 011101010111 şeklinde hesaplanır (Yılmaz, 2010). L-bit bloklar Tablo 7'de aşağıdaki tabloda gösterilmiştir.

Tablo 7. L-bit Bloklar

Bloklar	Tip	İçerik
1	Başlangıç segmenti	01
2		01
3		10
4		10
5	Test segmenti	01
6		11
7		01
8		01
9		01
10		11

Başlangıç segmenti kullanılarak, her olası L-bit değeri için bir tablo oluşturulur (yani, L-bit değeri tabloda bir indeks olarak kullanılır). Her L-bit bloğun son oluşumunun blok numarası tabloda belirtilir (yani, 1'den Q değerine kadar, $T_j = i$, burada j ise i'ninci L-bit bloğun ondalık gösterimidir). Aynı L-bit bloğunun tekrar oluşumları arasında hesaplanan mesafeyi, K bloklarında tespit edilen tüm farkların kümülatif $\log_2$ toplamına eklenmesi ile f_n değerleri oluşturulur (Denklem 30).

$$f_n = \frac{1}{K} \sum_{i=Q+1}^{Q+K} \log_2(i - T_j) \qquad (30)$$

Bu işlemlerden sonra P-değeri tamamlayıcı hata fonksiyonu ile bulunur (Denklem 31).

$$\text{P-değeri} = erfc\left(\left|\frac{f_n - beklenenDeğer(L)}{\sqrt{2}\sigma}\right|\right) \qquad (31)$$

Yukarıda örnekte verilen değerler ile Maurer'in "Evrensel İstatistik" testi çalıştırıldığında P-değeri $= erfc\left(\left|\frac{1.1949875-1.5374383}{\sqrt{2}\sqrt{1.338}}\right|\right) = 0.767189$ olarak hesaplanır ve P-değeri 0.01'den büyük çıktığı için verilen girdi dizisi testi geçmiş olarak kabul edilmektedir.

Doğrusal Karmaşıklık Testi

Doğrusal karmaşıklık testi, bir rastgele sayı üretecinin çıktılarının yeterince karmaşık olup olmadığını değerlendirmek amacıyla kullanılır. İlk olarak girdi dizisi her biri M uzunlukta N tane bloğa ayrılır (n = MN). Daha sonra her blokta Berlekamp Massey algoritması (Menezes ve ark., 1997) uygulanır. Her bir veri bloğu için T_i değeri Denklem 32'ye göre hesaplanır.

$$T_i = (-1)^M \cdot (L_i - \mu) + \frac{2}{9} \qquad (32)$$

Teorik ortalama değeri ise Denklem 33'teki gibi hesaplanmaktadır.

$$\mu = \frac{M}{2} + \frac{(9 + (-1)^{M+1})}{36} - \frac{\left(\frac{M}{3} + \frac{2}{9}\right)}{2^M} \qquad (33)$$

Denklem 34 ile Ki-kare istatistiği (X^2) hesaplanır.

$$\chi^2(obs) = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i} \qquad (34)$$

Ki-kare istatistik değeri bulunduktan sonra gama fonksiyonunu kullanarak Denklem 35'e göre P-değeri elde edilir.

$$P\text{-değeri} = igamc\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right) \qquad (35)$$

Elde edilen P-değeri 0.01'den küçükse verilen bit dizisi doğrusal karmaşıklık testinden başarısız olarak kabul edilir. Fakat P-değeri 0.01'den büyük veya eşit ise verilen bit dizisi doğrusal karmaşıklık testinden başarılı kabul edilir. Doğrusal karmaşıklık testi için 10^6 bitten daha fazla bit kullanılması önerilmektedir. Tablo 1'de verdiğimiz veri dizisi 128 bitten oluştuğu için onun yerine e sayısının ilk 1,000,000 biti için (M=1000) test uygulanacak olursa P-değeri 0.845406 çıkmaktadır ve değer 0.01'in üstünde olduğu için testi geçmektedir.

Kümülatif Toplam Testi

Kümülatif toplam testi, girdi dizisindeki normalizasyon işlemi sonucu oluşan (-1, +1) rakamların kümülatif toplamı ile tanımlanan rastgele yürüyüşün sıfırdan sapmasını inceler. Testin amacı, test edilen dizide meydana gelen kısmi dizilerin kümülatif toplamının, beklenen davranışına göre çok büyük veya çok küçük olup olmadığını belirlemektir. Rastgele bir dizi için, rastgele yürüyüşün gezileri sıfıra yakın olmalıdır. Rastgele olmayan dizilerin belirli türleri için, bu rastgele yürüyüşün sıfırdan sapmaları büyük olması beklenmektedir.

İlk olarak girdi dizisi normalize edilir, girdi dizisindeki (ε) sıfır bitleri ve bir bitleri $X_i = 2\,\varepsilon_i - 1$ eşitliği kullanılarak X_i değerleri (-1) ve (+1) olarak dönüştürülür. Örneğin, ε=1011010111 ise X = 1, (-1), 1, 1, (-1), 1, (-1), 1, 1, 1 olarak normalleştirilir. Bu işlemden sonra her biri X_i (eğer mod = 0) veya X_n (eğer mod = 1) ile başlayan, ardışık olarak daha büyük alt dizilerin kısmi toplamları olan S_i Tablo 8' deki gibi hesaplanır.

Tablo 8. Kısmi Toplamlar

Mod=0 (İleri)	Mod=1 (Geri)
$S_1 = X_1$	$S_1 = X_n$
$S_2 = X_1 + X_2$	$S_2 = X_n + X_{n-1}$
$S_3 = X_1 + X_2 + X_3$	$S_3 = X_n + X_{n-1} + X_{n-2}$
.	.
$S_k = X_1 + X_2 + X_3 + \cdots + X_k$	$S_k = X_n + X_{n-1} + X_{n-2} + \cdots + X_{n-k+1}$
.	.
.	.
$S_n = X_1 + X_2 + X_3 + \cdots + X_k \ldots + X_n$	$S_n = X_n + X_{n-1} + X_{n-2} + \cdots + X_{k-1} \ldots + X_1$

Daha sonra P-değerini bulmak için kullanılacak olan $z = max_{1\le k\le n}|S_k|$ (kısmi toplamların mutlak değerlerinin en büyüğü) değeri hesaplanır. Denklem 36'ya göre P-değeri elde edilir ve testin başarısı değerlendirilir.

$$P\text{-değeri} = 1 - \sum_{k=\left(\frac{-n}{z}+1\right)/4}^{\left(\frac{n}{z}-1\right)/4} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right)\right] + \sum_{k=\left(\frac{-n}{z}-3\right)/4}^{\left(\frac{n}{z}-1\right)/4} \left[\Phi\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right)\right]$$

Hesaplanan P-değeri sonucu 0.01'den küçük ise test başarısız olur fakat değer 0.01'den büyük veya eşit ise girdi dizisi testi başarılı kabul edilir. Tablo 1'deki bit dizisi belirtilen formüllere göre kümülatif toplam testine tabi tutulduğunda P-değeri ileri yönde 0.654760507613053 ve geri yönde 0.431439127632 olarak bulunur. Bu değer 0.01'den büyük olduğu için bu bit dizisi testten geçmiştir.

Rastgele Gezinimler Testi

Bu test, kümülatif toplam rastgele geziniminde belirli bir durumun ziyaret edildiği (yani meydana geldiği) toplam sayıyı inceler (Spitzer, 1964). Kümülatif toplamlı rastgele gezinim, (0,1)'lerden oluşan dizinin (-1,+1) olacak şekilde normalize edilmesinden sonra kısmi toplamlarından elde edilir. Bir rastgele gezinim döngüsü, rastgele kabul edilen bir yerden başlar tam bir döngü olana kadar belli bir uzunluktaki adım dizisinden oluşmaktadır. Bu testin amacı, bir döngü içinde belirli bir duruma yapılan ziyaretlerin sayısının rastgele bir dizi için beklenenden sapıp sapmadığını belirlemektir. Toplamda sekiz adet test ve çıkarımından oluşan bir test yapısıdır.

İlk olarak girdi dizisi normalize edilir, girdi dizisindeki (ε) sıfır bitleri ve bir bitleri $X_i = 2\,\varepsilon_i - 1$ kullanılarak X_i değerleri (-1) ve (+1) olarak dönüştürülür. Örneğin, ε=0110110101 ise X = -1, 1, 1, -1, 1, 1, -1, 1, -1, 1 olarak normalleştirilir. Bu işlemden sonra her biri X_1 ile başlayan, alt dizilerin kısmi toplamları (S_i) hesaplanır. S kümesinin öncesine ve sonrasına sıfırlar ekleyerek yeni bir dizi (S') oluşturulur. Burada $S' = 0, s_1, s_2, \ldots, s_n$,0 şeklinde oluşur. Yukarıdaki örneğe göre, $S' = 0, -1,0,1,0,1,2,1,2,1,2,0$ olacaktır.

J, S'dizisindeki sıfır geçişlerinin toplam sayısı olsun. Burada sıfır geçişleri, başlangıçtaki sıfırdan sonra S' dizisinde ortaya çıkan sıfırların sayısıdır. Yukarıdaki örnek için, eğer S' = {0, –1, 0 1, 0, 1, 2, 1, 2, 1, 2, 0} ve J= 3'tür. J = 3 olduğundan {0, -1, 0}, {0, 1, 0} ve {0, 1, 2, 1, 2, 1, 2, 0}'dan oluşan 3 çevrim vardır. Her çevrim için ve –4 ≤x ≤-1 ve 1 ≤x ≤4 değerlerine sahip sıfır olmayan her durum değeri x için, her çevrim içindeki x durum değerlerinin frekansı hesaplanır (Tablo 9).

Tablo 9. Çevrimlerdeki x durum değerlerinin frekansları

x Durumu	Çevrim 1 {0, -1, 0}	Çevrim 2 {0, 1, 0}	Çevrim 3 {0, 1, 2, 1, 2, 1, 2, 0}
-4	0	0	0
-3	0	0	0
-2	0	0	0
-1	1	0	0
1	0	1	3
2	0	0	3
3	0	0	0
4	0	0	0

Daha sonra sekiz x durumu için, $v_k(x)$ (durumun tüm çevrimler arasında tam olarak k kez meydana geldiği çevrimlerim toplam sayısı) değerleri hesaplanır. Burada k = 0, 1, ..., 5'tir ve $\sum_{k=0}^{5} v_k(x) = J$ olmalıdır.

Ki-kare istatistiği Denklem 37'ye göre hesaplanır. Formülde $\pi_k(x)$, x durumunun rastgele bir dağılımda k kez ortaya çıkma olasılığıdır.

$$\chi^2(obs) = \sum_{k=0}^{5} \frac{\left(v_k(x) - J\pi_k(x)\right)^2}{J\pi_k(x)} \quad (37)$$

Ki-kare istatistik değeri bulunduktan sonra gama fonksiyonunu kullanarak Denklem 38'e göre P-değeri elde edilir.

$$P\text{-değeri} = igamc\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right) \quad (38)$$

Elde edilen P-değeri 0.01'den küçükse verilen bit dizisi rastgele gezinimler testinden başarısız olarak kabul edilir. Fakat P-değeri 0.01'den büyük veya eşit ise verilen bit dizisi rastgele gezinimler testinden başarılı kabul edilir. Yukarıdaki örnek için Ki-kare istatistiği (χ^2) 4.333033 ve P-değeri 0.502529 olarak bulunur.

Test uygulaması olarak Tablo 1'deki bit dizisi belirtilen formüllere göre rastgele gezinimler testine tabi tutulursa P-değeri 0.30621891841327875 olarak bulunmaktadır. Bu değer 0.01'den büyük olduğu için girdi bit dizisi testi geçmektedir.

Rastgele Gezinimler Değişken Testi

Rastgele gezinimler değişken testi, bir kümülatif toplam rastgele geziniminde, özel durumların ziyaret edilme sayısını ölçer. Bu testte amaç, bir rastgele gezinimde özel durumların beklenen ziyaret sayısındaki sapmalarının belirlenmesidir. Toplamda 19 adet test (-9 ile +9 arasındaki durum değerleri için) ve çıkarımından oluşan bir test yapısıdır. İlk olarak girdi dizisi normalize edilir, girdi dizisindeki (ε) sıfır bitleri ve bir bitleri $X_i = 2\,\varepsilon_i - 1$ kullanılarak X_i değerleri (-1) ve (+1) olarak dönüştürülür. Örneğin, ε=0110110101 ise X = -1, 1, 1, -1, 1, 1, -1, 1, -1, 1 olarak normalleştirilir. Bu işlemden sonra her biri X_i ile başlayan, alt dizilerin kısmi toplamları (S_i) hesaplanır. S kümesinin öncesine ve sonrasına sıfırlar ekleyerek yeni bir dizi S' oluşturulur. $S' = 0, s_1, s_2, \dots, s_n$, 0 şeklinde oluşur. Yukarıdaki örneğe göre, $S' = 0, -1,0,1,0,1,2,1,2,1,2,0$ olacaktır (Yakut, 2019).

J, S'dizisindeki sıfır geçişlerinin toplam sayısı olsun, burada sıfır geçişleri, başlangıçtaki sıfırdan sonra S' dizisinde ortaya çıkan sıfırların sayısıdır. Burada X'in sıfır olmayan on sekiz durumunun her biri için, ξ(x), yani x durumunun tüm J çevrimleri boyunca meydana geldiği toplam sayı hesaplanır. ξ(-1) = 1, ξ(1) = 4, ξ(2) = 3 ve diğer ξ(*x*) değerleri 0 olarak elde edilir.

Sonrasında Denklem 39'a göre P-değeri bulunur. Yukarıdaki örnek için P-değeri 0.683091 çıkmakta ve değer 0.01'den büyük çıktığı için testi geçmektedir.

$$P\text{-değeri } = erfc\left(\frac{|\xi(x) - J|}{\sqrt{2J(4|x| - 2)}}\right) \quad (39)$$

Test uygulaması olarak Tablo 1'deki bit dizisi belirtilen formüllere göre rastgele gezinimler değişken testine tabi tutulursa P-değeri 0.6170750774 olarak bulunmaktadır. Bu değer 0.01'den büyük olduğu için bu bit dizisi testi geçmektedir.

Sonuç

Sonuç olarak, bu makale rastgele sayı üreteçleri için istatistiksel test yöntemlerinin kapsamlı bir analizini sunmaktadır. Frekans testi, akış testi ve yaklaşık entropi testi gibi çeşitli istatistiksel testler incelenerek, bu testlerin üretilen sayıların rastgeleliğini ve kalitesini nasıl etkili bir şekilde değerlendirebildiği gösterilmiştir. Örnek çözümlerin dahil edilmesi, bu test tekniklerinin pratikliğini ve gerçek dünyadaki uygulamasını daha da vurgulamaktadır.

Bu araştırma sayesinde, rastgele sayı üreteçlerinin güvenilirliğini ve rastgeleliğini doğrulamak için sıkı istatistiksel testlerin gerekli olduğu ortaya çıkmıştır. Araştırmacılar ve uygulayıcılar, rastgele sayı üreteçlerinin kalitesini geliştirerek çeşitli istatistiksel simülasyonların, kriptografi sistemlerinin ve büyük ölçüde rastgeleliğe dayanan diğer uygulamaların doğruluğunu ve geçerliliğini sağlayabilirler.

Ayrıca bu makale, rastgele sayı üreteçlerini kapsamlı bir şekilde değerlendirmek için çeşitli istatistiksel testlerin seçilmesinin ve kullanılmasının önemini vurgulamaktadır. Çoklu test yöntemleri kullanılarak, bireysel testlerin doğasında olan sınırlamalar ve sapmalar azaltılabilir, böylece daha sağlam bir rastgelelik değerlendirmesi sağlanabilir.

Sonuç olarak, rastgele sayı üretimi alanındaki araştırmacılar, geliştiriciler ve uygulayıcılar, bu makalede tartışılan istatistiksel test tekniklerini değerlendirme süreçlerinin ayrılmaz bir parçası olarak düşünmelidir. Bu yöntemleri benimseyerek rastgele sayı

üreteçlerinin kalitesini ve güvenilirliğini artırmak, sonuçta çeşitli çalışma ve uygulama alanlarında daha doğru ve güvenilir sonuçlara ulaşmak mümkündür.

KAYNAKÇA

Abramowitz, M. & Stegun, I. (1964) *Handbook of Mathematical Functions, Applied Mathematics Series. Vol. 55,* Washington: National Bureau of Standards.

Anant, P. G. & Stavros, G. P. (1994), *Runs and Patterns in Probability: Selected Papers*. Dordrecht: Kluwer Academic.

Barbour, A. D., Holst, L. & Janson, S. (1992) *Poisson Approximation*. Oxford: Clarendon Press.

Bracewell, R. N. (1986) *The Fourier Transform and Its Applications*. New York: McGraw-Hill.

Bradley, J. V. (1960). *Distribution-free statistical tests,* 60 (661). United States Air Force.

Chrysaphinou, O. & Papastavridis, S. (1988) A Limit Theorem on the Number of Overlapping Appearances of a Pattern in a Sequence of Independent Trials. *Probability Theory and Related Fields, Vol. 79,* 129-143.

Chung, K. L. (1979) *Elementary Probability Theory with Stochastic Processes*. New York: Springer-Verlag.

Good, I. J. (1953), The serial test for sampling numbers and other tests for randomness, *Proc. Cambridge Philos. Soc.. 47,* 276-284.

Hamano, K. & Kaneko, T. (2007) The Correction of the Overlapping Template Matching Test Included in NIST Randomness Test Suite, *IEICE Transactions of Electronics, Communications and Computer Sciences*, E90-A(9), 1788-1792.

Kovalenko, I. N. (1972), "Distribution of the linear rank of a random matrix, *Theory of Probability and its Applications*. *17,* 342-346.

Marsaglia, G. *DIEHARD Statistical Tests* (1995) https://ani.stat.fsu.edu/diehard/. (Erişim Tarihi: 17.12.2023)

Maurer, U. M. (1992). A universal statistical test for random bit generators. *Journal of cryptology, 5*, 89-105.

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1997). *Handbook of applied cryptography*. CRC press.

NIST-800-22, *A statistical test suite for random and pseudo RNGs for cryptographic applications*. (2001) National institute of stand. and tech., http://csrc.nist.gov/publications/ nistpubs/800-22/sp-800-22-051501.pdf, (Erişim Tarihi: 01.12.2023).

Pincus, S. & Singer, B. H. (1996) Randomness and degrees of irregularity, *Proc. Natl. Acad. Sci. USA. Vol. 93*, 2083-2088

Spitzer, F. (1964) *Principles of Random Walk*. Princeton: Van Nostrand.

Yakut, S. (2019). Gerçek Rastgele Sayı Üreteçlerinin Tasarlanması ve Analizi, Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü.

Yılmaz, R. (2010). Kriptolojik Uygulamalarda Bazı İstatistik Testler, Yüksek Lisans Tezi, Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

.

BÖLÜM XIII

Beyin Tümörü Görüntülerinde Veri Büyütme için Derin Öğrenme Tabanlı Stil Aktarım Yaklaşımları ve Uygulamaları

Birkan BÜYÜKARIKAN[1]

Giriş

Beyin tümörü, beyinde düzensiz ve anormal hücre gelişimi gösteren, yetişkin ve çocuklar arasında görülen tehlikeli ve ölümcül bir hastalıktır. Bu hastalığın erken teşhisi ve tedavisinin planlanması hastanın sağlığına kavuşması açısından kritik öneme sahiptir (Garg & Sahu, 2023). İnsan hayatını tehdit eden bu hastalığa etkili bir teşhis konulabilmesi için tıbbi görüntüleme teknolojilerinden bilgisayarlı tomografi (Computed Tomography, CT) ve manyetik rezonans görüntüleme (Magnetic Resonance Imaging, MRI) kullanılır. Ancak bu teknolojiler yardımıyla elde edilen görüntülerde

[1] Dr. Öğr. Üyesi, Isparta Uygulamaları Bilimler Üniversitesi, Uluborlu Selahattin Karasoy Meslek Yüksekokulu Bilgisayar Teknolojileri Bölümü, https://orcid.org/0000-0002-9703-9678, birkanbuyukarikan@isparta.edu.tr

tümör boyutunun küçük olması veya tümörün kan damarlarına benzemesi uzmanlar tarafından gerçekleştirilecek tümör tanısının konulmasını zorlaştırır. Ayrıca uzmanların manuel olarak bu görüntüleri incelenmesi çok zaman alıcı ve çaba da gerektirir (Albright & ark., 1993; Zhang & ark., 2021). Nitekim beyin tümörünün ciddiyetini göz önünde bulunduran ve karşılaşılan bu gibi sorunları çözmeye çalışan birçok araştırmacı, bilgisayar tabanlı teşhis sistemlerine yönelmiştir (Tariq & ark., 2022). Özellikle son yıllarda bilgisayar tabanlı teşhis uygulamalarında beyin tümörü görüntülerinin işlenmesi ve analizi için derin öğrenme modellerinin kullanımını ve geliştirilmesini içeren çalışmalar da popülerlik kazanmıştır (Cai, Gao & Zhao, 2020).

Derin öğrenme, görüntülerden özellikleri otomatik olarak çıkarılmasına olanak tanıyan, çeşitli sorunları çözebilecek kadar esnek ve güçlü denetimli bir öğrenme aracıdır (LeCun, Bengio & Hinton, 2015). Derin öğrenmede bir model eğitirken model hem mevcut verilerle daha iyi sonuçlar sağlamak hem de fazla uydurmayı önlemek için büyük miktarda veriye ihtiyaç duyar. Nitekim birçok tıbbi görüntüleme alanında veri eksikliğinin bulunması, iyi ve etiketli tıbbi görüntülerin elde edilmesinin pahalı ve zaman alıcı olması gibi nedenler derin öğrenme modellerinin performansını kısıtlayabilir. Bu sorunları çözebilmek için genellikle çeşitli yöntemler yardımıyla veriler çoğaltılır (Goodfellow, Bengio & Courville, 2016; Wang & Perez, 2017).

Veri büyütme, veri setindeki görüntülerin yapay olarak çoğaltıldığı ve modellerin daha iyi performans göstermesine yardımcı olan bir yöntemdir (Yang & ark., 2022). Bu yöntem, geleneksel ve derin öğrenme tabanlı tekniklerle kategorize edilebilir (Chlap & ark., 2021). Geleneksel veri büyütme tekniklerinden; renk değişimi, döndürme, yansıtma, kırpma ve öteleme popüler olarak veri setinin büyütülmesinde kullanılır (Tmenova, Martin & Duong, 2019). Görüntülerin döndürülmesi, yansıtılması, kırpılması gibi işlemlerle çok fazla sayıda görüntünün üretilmesi sağlanılabilir. Ancak geleneksel bu tekniklerle oluşturulan görüntüler çeşitlilikten yoksundur. Dolayısıyla veri büyütmede görüntülerin

çeşitlendirilmesi için alternatif yöntemlerin araştırılması gerekmektedir (Sajjad & ark., 2019). Üstelik tıbbi görüntüler standarttır ve katı düzenlemelere de uyar (Mahajan, 2023).

Tüm bu bilgiler ışığında tıbbi görüntüleme alanında orijinal görüntülerden tamamen farklı görüntülerin üretilmesi, modelin daha fazla görüntüyü tanımasına olanak tanır (Kiani Kalejahi, Meshgini & Danishvar, 2023). Diğer bir ifadeyle yüksek kaliteli ve çok fazla sayıda görüntüyü içeren bir veri setinin olması, derin öğrenme modellerinin dikkate değer bir performans sergilemesini sağlar (Alomar, Aysel & Cai, 2023; Lee & Ma, 2022; Tariq & ark., 2022; Zhang & ark., 2019). Bu yüzden görüntü çeşitliliğinin arttırılmasına olanak tanıyan derin öğrenmeyle stil aktarımı, özellikle tıbbi görüntü analizi uygulamalarında önemli bir alan bulmaktadır. Stil aktarımı, bir görüntünün stilini veya özelliklerini başka bir görüntüye uygulayarak yukarıda belirtilen dezavantajları azaltabilir (Cai & ark., 2023). Ayrıca stil aktarımı, sınıflandırıcı modelin genelleştirilebilirliğini artırmak ve modelin aşırı uyumunu azaltmak için mevcut görüntülerin özelliklerini öğrenerek gerçekçi görüntüler üretebilir (Chlap & ark., 2021).

Bugün tıbbi görüntü analizi uygulamalarında genellikle üretken çekişmeli ağlar (Generative Adversarial Network, GAN), stil aktarımını sağlayan ve gerçekçi görüntü üretebilen özgün ve yaratıcı popüler bir yöntem olarak kullanılmaktadır (Garcea & ark., 2022). Ayrıca son zamanlarda yaşanan gelişmeler derin evrişimli üretken rakipsel ağ (Deep Convolutional Generative Adversarial Network, DCGAN), döngüsel üretken çekişmeli ağ (Cycle Generative Adversarial Network, CycleGAN) ve Wasserstein GAN gibi yeni GAN tabanlı modellerin oluşturulmasına ortam sağlamıştır (Salimans & ark., 2016).

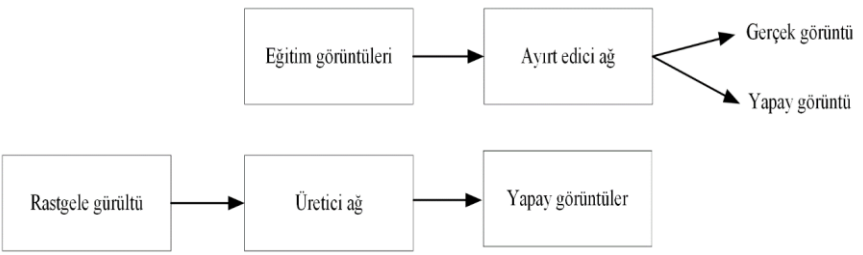
Bu çalışmada, beyin tümör görüntülerinde derin öğrenmeyle stil aktarım yaklaşımlarından GAN, DCGAN, CcyleGAN ve Wasserstein GAN modelleri tanıtılmış ve bu modellerle 2020-2023 yılları arasında gerçekleştirilen çalışmalar sunulmuştur. Çalışmanın diğer bölümleri de aşağıdaki gibi organize edilmiştir.

Bir sonraki bölümde derin öğrenme tabanlı bazı stil aktarım yaklaşımları kısaca açıklanmış, ardından beyin tümör görüntülerinde bu yaklaşımlarla gerçekleştirilen uygulamalar incelenmiş ve son olarak çalışmanın sonuçlarından bahsedilmiştir.

Derin Öğrenme Tabanlı Stil Aktarım Yaklaşımları

Derin öğrenme tabanlı stil aktarım yaklaşımlarının ortaya çıkışı tıbbi görüntü analizi uygulamalarında dikkate değer ilerlemeler kaydetmiştir. Bu uygulamalarda stil aktarımını gerçekleştirmek için genellikle GAN modeli kullanılmıştır (Giger, 2018; Liu & ark., 2019). Bu bölümde GAN ve GAN tabanlı modellerin yapıları ve bu modellerin etkinlikleri incelenmiştir.

GAN'lar 2014 yılından beri orijinal veri setine benzer nitelikleri koruyarak gerçekçi bir şekilde yüksek kaliteli görüntü oluşturma uygulamalarında kullanılan popüler yaklaşımlardan biridir (Alomar, Aysel & Cai, 2023; Lee & Ma, 2022; Tariq & ark., 2022). GAN, yapay görüntü üretmek için Şekil 1'deki temel çalışma diyagramını kullanır. Şekil 1'de de görüldüğü gibi GAN ağ yapısında üretici ve ayırt edici modellerini içerir. Bu ağ yapılarından üretici, modelin belirli bir veri setine benzeyen yeni yapay görüntüler üretmesi işlevinden sorumludur. Bu modelin ardından kullanılan ayırt edici ağ ise orijinal görüntülerle yapay görüntülerin ayırt edilmeye çalışıldığı bir çerçeve sağlar. Sağlanan bu sürekli rekabet, üretici modelin giderek daha gerçekçi veriler üretmesine olanak tanır (Tavse & ark., 2022).



Şekil 1. Çekişmeli Üretici Ağ Yapısının Diyagramı

Derin öğrenme alanı ilerlemeye devam ettikçe stil aktarım yöntemleri de giderek artacaktır. DCGAN, CycleGAN ve Wasserstein GAN görüntü üretmek için geliştirilen modellerden bazılarıdır (Cai & ark., 2023).

GAN'ın alt modellerinden biri olan DCGAN, evrişimli sinir ağ (Convolutional Neural Network, CNN) mimarisinin özellik çıkarma avantajlarını kullanarak sınıflandırıcı modelin görüntü analizi ve işleme yeteneğinin geliştirilmesine katkı sağlar (Zhou & ark., 2017). Bu model, CNN ve GAN modelinin birleştirilmesinden oluşur. Diğer bir ifadeyle DCGAN, üretici ve ayırt edici ağlar ile evrişim katmanı, batch normalizasyonu, Leaky/ReLU aktivasyon fonksiyonlarını ve kayıp fonksiyon yapılarını içerir. DCGAN bu yapıları bir araya getirerek öğrenme sürecinde daha kararlı yapay görüntüler üretilebilir (Radford, Metz & Chintala, 2015).

GAN tabanlı diğer bir model olan CycleGAN eşleştirilmiş herhangi bir veri setinin olmasına gerek kalmadan görüntüden görüntüye çeviri yapan bir yaklaşımdır. Bu yaklaşımda oluşturulan görüntüler orijinal görüntü alanlarına geri döndürülebilmesi için döngü tutarlılığı kaybını kullanır. Bu döngüsel yapı, her iki küme arasında çift yönlü çeviri yapabilen bir model oluşturur. Diğer bir ifadeyle CycleGAN iki adet üretici ve iki adet ayırt edici model yapılarını içerir. Ayrıca CycleGAN sınırlı eğitim verileri ile çalışabilme yeteneğine de sahiptir (Zhu & ark., 2017).

Wasserstein GAN, GAN modelinin eğitimini daha tutarlı hale getirmek için hem üretici hem de ayırt edici modelin kayıp fonksiyonu olarak Wasserstein mesafesini (Dünya Taşıyıcı Mesafesi olarak da bilinen) kullanır. Ayrıca bu yaklaşımda ayırt edici modeli yanıltmak için üretici model, yapay görüntüleri üretmek üzere eğitilir. Böylece iki model arasındaki Wasserstein mesafesinin yakınsama hızı da artar (Gulrajani & ark., 2017).

İncelemeler için yukarıda açıklanan GAN tabanlı modellerin her birinin farklı ağ yapıları bulunmakta ve bu modeller belirli uygulama alanlarına yönelik çeşitli avantajlar sunmaktadır. Söz konusu incelenen modeller değerlendirildiğinde, DCGAN modeli, özellikle

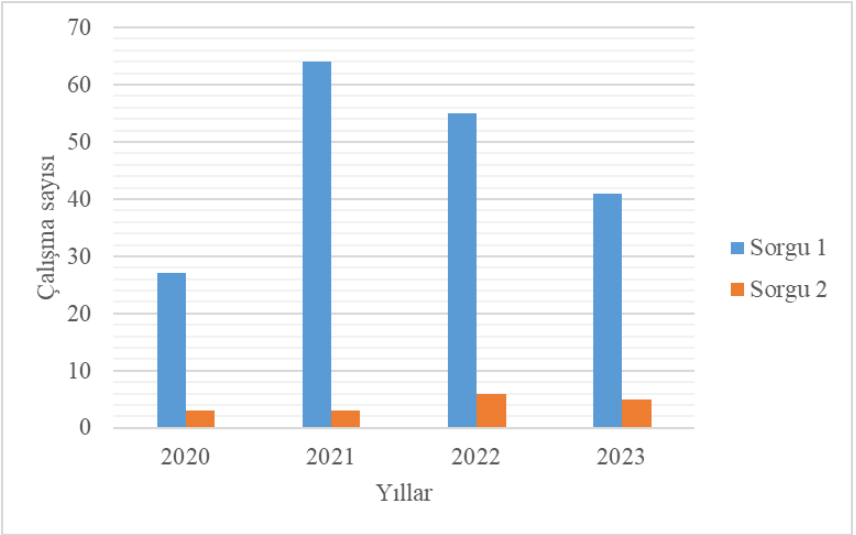
CNN kullanarak daha iyi sonuçlar elde etme avantajına sahiptir. CycleGAN modelinde ise farklı veri kümeleri arasında çift yönlü çeviri yapma yeteneği ile görüntülerin üretilmesine dikkat çekmektedir. Wasserstein GAN ise diğer GAN modellerinde yaygın olarak kullanılan kayıp fonksiyonu yerine dünya taşıyıcı mesafesini kullanmaktadır. Ayrıca Wasserstein GAN modeli eğitimde daha stabil bir davranış göstererek bazı GAN modellerinin eğitim zorluklarına çözüm de sağlamaktadır. Şunu belirtmekte fayda var ki, her GAN modelinin farklı avantaj ve sınırlılıklarının olması gelecekte yeni modellerin geliştirilmesine yol açacaktır.

Beyin Tümör Görüntülerinde Derin Öğrenme Tabanlı Stil Aktarım Uygulamaları

Bu bölümde beyin tümör görüntülerinde derin öğrenme tabanlı stil aktarım uygulamaları incelenmektedir. İncelenen çalışmalar Web of Science (WOS) veri tabanından elde edilmiştir. Bu veri tabanında araştırma yapmak için iki adet sorgu oluşturulmuştur.

İlk sorguda tıbbi görüntülerde derin öğrenme ve stil aktarımıyla yapılan tüm çalışmaları listelemek için ‘derin öğrenme’ ve ‘tıbbi görüntülerde veri büyütme’ ve (‘stil aktarımı’ veya ‘GAN’ veya ‘DCGAN’ veya ‘CycleGAN’ veya ‘Wasserstein GAN’) anahtar kelimeleri kullanılmıştır. Sorgu 1’e göre 187 çalışma (makale, konferans ve inceleme makalesi dahil) ortaya çıkmıştır. Elde edilen sonuçlara göre tıbbi görüntü analizi çeşitli alt alanlarda birçok araştırmayla literatüre katkı sağlamıştır.

İkinci sorguda ise beyin tümörü görüntülerinde derin öğrenme ve stil aktarımıyla yapılan çalışmalar, ‘derin öğrenme’ ve ‘beyin tümörü görüntülerinde veri büyütme’ ve (‘stil aktarımı veya ‘GAN’ veya ‘DCGAN’ veya ‘CycleGAN’ veya ‘Wasserstein GAN’)’ı içeren anahtar kelimelere göre tarama işlemi gerçekleştirilmiştir. Bu sorguya göre 17 çalışma (makale, konferans ve inceleme makalesi dahil) elde edilmiştir. Sorguların yıllara göre sonuçları Şekil 2’de görülmektedir.



Şekil 2. Tıbbi ve Beyin Tümör Görüntülerinde Veri Büyütme Uygulamalarının Sonuçları

Beyin tümör görüntülerinden yapay görüntü üretmek için çeşitli GAN ve GAN tabanlı modellerle son dört yılda stil aktarımını gerçekleştiren çalışmalar Tablo 1'de listelenmiştir. Burada Sorgu 2'deki konuyla ilgili birincil çalışmaları seçmek için öncelikle başlıklar incelenmiş ve daha sonra da çalışmaların içerikleri ele alınmıştır.

Araştırmalarda genellikle GAN modeli kullanılarak veri büyütme uygulamalarının gerçekleştirildiği belirlenmiştir. Ge & ark., (2020), Asiri & ark., (2023) ve Biswas & ark., (2023) gerçekçi beyin görüntüleri üretmek için GAN'ı kullanmışlardır. Gupta & Bibhu (2023) ise GAN kullanılarak tıbbi görüntülerden şüpheli bölgenin çıkartılması için bir tümör segmentasyon modülü oluşturmuşlardır. Kim, Kim & Park (2021) önerdikleri GAN modelinin veri büyütmedeki diğer GAN tabanlı yöntemlere kıyasla tümör segmentasyonunda önemli iyileştirmeler sağladığını belirtmişlerdir.

Tablo 1. Stil Aktarım Yaklaşım Çalışmaları

Yaklaşım	Referans
GAN	Ge & ark., (2020), Asiri & ark., (2023), Biswas & ark., (2023), Gupta & Bibhu (2023) ve Kim, Kim & Park (2021)
DCGAN ve Wasserstein GAN	Mukherkjee & ark., (2022)
CycleGAN	Yapici, Karakis & Gurkahraman (2023)
DCGAN ve Vanilya GAN	Alrashedy & ark., (2022)
DCGAN ve tek GAN (SinGAN)	Alrumiah, Alrebdi & Ibrahim (2023)
StynMedGAN	Wali & ark., (2023)

Veri büyütmede GAN modellerinin kullanılması birçok avantaj sunmasına rağmen, modellerin bazı sınırlılıklarının üstesinden gelmek isteyen birçok araştırmacı hibrit modeller oluşturmak için farklı GAN modellerini birleştirmişlerdir (Meor Yahaya & Teo, 2023). Ayrıca araştırmalarda Vanilya GAN ve SinGAN gibi bir önceki bölümde açıklanan yaklaşımlardan farklı modellerinde kullanıldığı görülmektedir. Diğer bir ifadeyle stil aktarımın farklı modellerle uygulanması, aynı görüntünün farklı yorumlanmasını sağlayabilir.

Açıkçası bilgisayar tabanlı teşhis uygulamalarında görüntü kalitesinin yüksek olması tümör teşhisinin ayırt edilmesini kolaylaştırır. Beyin tümör görüntülerinde görüntü kalitesiyle ilgilenen bazı araştırmacılar GAN tabanlı model geliştirmişlerdir. Mukherkjee & ark., (2022) beyin tümörü görüntülerinden yapay görüntü üretmek için DCGAN ve Wasserstein GAN'ı birleştiren bir model önermişlerdir. Önerdikleri modeli iki veri setinde denemişler ve yapısal benzerlik indeksine göre bu modelle iyi kalitede görüntülerin üretildiğini belirtmişlerdir.

Öte yandan stil aktarımıyla beyin tümör görüntülerinde veri büyütme uygulamasını gerçekleştiren araştırmacılar genellikle sınıflandırıcı modelin genelleyebilirliğini geliştirmek için farklı GAN modelleri kullanmışlardır.

Yapici, Karakis & Gurkahraman (2023) CycleGAN modeliyle üretilen yapay görüntülerle derin öğrenmenin genelleme yeteneği

üzerindeki etkisini incelemişlerdir. Ürettikleri yapay görüntülerin literatürdeki ilgili diğer çalışmalara göre daha yüksek bir doğruluk değeri elde ettiğini belirtmişlerdir.

Alrumiah, Alrebdi & Ibrahim (2023) beyin MRI veri setindeki sınıf dengesizliği sorununu çözmek için DCGAN ve SinGAN modelleriyle veri büyütme uygulaması gerçekleştirmişlerdir. Onlar çalışmalarında orijinal veri setindeki ve DCGAN, SinGAN, DCGAN ile SinGAN modelleriyle üretilen veri setlerindeki görüntüleri VGG16 modeliyle sınıflandırmışlardır. SinGAN modeliyle üretilen görüntülerin DCGAN modeliyle üretilen görüntülere göre %4 daha fazla sınıflandırma doğruluğu gösterdiğini belirtmişlerdir.

Wali & ark., (2023) çalışmalarında görüntü üretmek için StynMedGAN adlı yeni bir GAN modeli önermişlerdir. Önerilen modelin sınıflandırmaya yönelik etkinliğini göstermek için CNN, DenseNet121 ve VGG-16 kullanmışlardır. StynMedGAN modeliyle üretilen görüntülerin orijinal görüntülere göre daha iyi sınıflandırma performansı elde ettiğini göstermişlerdir.

Alrashedy & ark., (2022) çalışmalarında DCGAN ve Vanilya GAN modelleriyle beyin MR görüntülerini üretmişler ve üretilen görüntüleri CNN, MobileNetV2 ve ResNet152V2 modelleriyle sınıflandırmışlardır. Deneysel sonuçlara göre en iyi performans, DCGAN modeliyle üretilen yapay görüntülerin ResNet152V2 modeliyle sınıflandırılmasıyla elde etmişlerdir. Bu modelle %99,09 doğruluk, %99,12 hassasiyet, %99,08 geri çağırma, %99,51 eğri altındaki alan (AUC) ve 0,196 kayıp değeri elde ettiklerini belirtmişlerdir.

Diğer taratan bazı araştırmacılar GAN tabanlı modelleri sentezleyerek gelecekteki çalışmalara yönelik stratejileri belirleme fırsatını sunmak için gözden geçirme çalışmalarını gerçekleştirmişlerdir. Sorin & ark., (2020) beyin tümör görüntülerinde ve Chlap & ark., (2021) tıbbi görüntülerde veri büyütme yaklaşımlarını sistematik olarak incelemişlerdir.

Mevcut literatürde daha gerçekçi ve etkili stil efektlerinin oluşturulmasını sağlayan GAN tabanlı modellerin daha fazla kullanılacağını göstermektedir. Ancak GAN tabanlı modellerle yüksek çözünürlüklü görüntülerin üretilmesi bilgisayarın daha fazla hesaplama ve zaman maliyetine sebep olacaktır (Cai & ark., 2023). Yine de bu araştırmaların yeni modellerin keşfedilmesine ve mevcut modellerin iyileştirilmesine katkı sağlayacağı bir gerçektir.

Sonuçlar

Beyin tümörünün neden olduğu ciddi sorunları tespit edebilmek için birçok araştırmacı, derin öğrenme algoritmalarıyla karar sistemleri geliştirmişlerdir. Ancak beyin tümörü görüntüleri için büyük veri setlerinin toplama zahmeti, kişisel verilerin korunamayacağı düşüncesi ve görüntülerin etiketlenme maliyetleri, bu algoritmalarla yapılacak uygulamaların gerçekleştirilmesinde hala daha büyük sorunlardandır. Bu sorunların üstesinden gelebilmek için veri büyütme iyi bir yöntemdir. Veri büyütmede geleneksel olarak kullanılan döndürme, kırma, parlaklık değiştirme gibi yöntemlerin yerine son yıllarda derin öğrenme ile oluşturulan veri büyütme uygulamaları büyük bir ivme kazanmıştır. Derin öğrenme yöntemleriyle oluşturulan veri büyütme, hastaların anatomik varyasyon görüntülerini simüle etmektedir. Böylece üretilen yapay görüntüleri içeren veri setleri gerçeğe daha yakın bir şekilde oluşturulabilmektedir.

Bu çalışmada beyin tümör görüntülerinde veri büyütme işlemi için derin öğrenme algoritmalarıyla stil aktarımına dayanan uygulamalar incelenmiştir. Çalışmada veri büyütme stratejilerinden GAN, DCGAN, CcyleGAN ve Wasserstein GAN modelleri tanıtılmış ve daha sonra bu modellerle 2020-2023 yılları arasında yapılan çalışmalar listelenmiştir. Listelenen çalışmalar WOS veri tabanından elde edilmiştir. WOS veri tabanından tıbbi hastalık ve beyin tümör görüntülerinde veri büyütmeye yönelik derin öğrenme ve stil aktarımıyla yapılan çalışmaları taramak için oluşturulan iki sorguda sırasıyla 187 ve 17 çalışma bulunmuştur. Beyin tümör görüntülerinde stil aktarımıyla veri büyütme çalışmalarından ilgili

birincil çalışmaları listelemek için öncelikle çalışmaların başlıkları incelenmiş ve ardından da ilgili çalışmaların içerikleri irdelenmiştir.

İnceleme sonuçları beyin tümör görüntülerini sınıflandırma çalışmalarında üretilen yapay görüntülerin sınıflandırma performansının arttırdığını göstermektedir. Ayrıca görüntü kalitesi çalışmalarında ise GAN tabanlı modeller görüntü kaliteni de geliştirdiği belirlenmiştir. Kısacası araştırmalarda stil aktarımı için GAN tabanlı modeller benzersiz yetenekler göstermiştir. Nitekim tüm bu uygulama sonuçlarının iyileştirilmesi, GAN tabanlı bilinen veya hibrit modellerin ürettikleri yapay görüntülere bağlıdır. Ancak sonuçlar görüntü büyütmenin etkinliğini vurgulamakta ve gelecekteki araştırmalar için de potansiyel yollar sunmaktadır.

KAYNAKÇA

Albright, A. L., Packer, R. J., Zimmerman, R., Rorke, L. B., Boyett, J. & Hammond, G. D. (1993). Magnetic resonance scans should replace biopsies for the diagnosis of diffuse brain stem gliomas: a report from the Children's Cancer Group. *Neurosurgery*, 33 (6), 1026-1030.

Alomar, K., Aysel, H. I. & Cai, X. (2023). Data augmentation in classification and segmentation: A survey and new strategies. *Journal of Imaging*, 9 (2), 46. Doi:10.3390/jimaging9020046

Alrashedy, H. H. N., Almansour, A. F., Ibrahim, D. M. & Hammoudeh, M. A. A. (2022). BrainGAN: brain MRI image generation and classification framework using GAN architectures and CNN models. *Sensors*, 22 (11), 4297. Doi: 10.3390/s22114297

Alrumiah, S. S., Alrebdi, N. & Ibrahim, D. M. (2023). Augmenting healthy brain magnetic resonance images using generative adversarial networks. *PeerJ Computer Science*, 9, e1318. Doi: 10.7717/peerj-cs.1318

Asiri, A. A., Shaf, A., Ali, T., Aamir, M., Usman, A., Irfan, M., Alshamrani, H. A., Mehdar, K. M., Alshehri, O. M. & Alqhtani, S. M. (2023). Multi-Level Deep Generative Adversarial Networks for Brain Tumor Classification on Magnetic Resonance Images. *Intelligent Automation & Soft Computing*, 36 (1), 127-143. Doi: 10.32604/iasc.2023.032391

Biswas, A., Bhattacharya, P., Maity, S. P. & Banik, R. (2023). Data augmentation for improved brain tumor segmentation. *IETE Journal of Research*, 69 (5), 2772-2782. Doi: 10.1080/03772063.2021.1905562

Cai, L., Gao, J. & Zhao, D. (2020). A review of the application of deep learning in medical image classification and segmentation. *Annals of translational medicine*, 8 (11), 713. Doi: 10.21037/atm.2020.02.44

Cai, Q., Ma, M., Wang, C. & Li, H. (2023). Image neural style transfer: A review. *Computers and Electrical Engineering*, 108, 108723. Doi: 10.1016/j.compeleceng.2023.108723

Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L. & Haworth, A. (2021). A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65 (5), 545-563. Doi: 10.1111/1754-9485.13261

Garcea, F., Serra, A., Lamberti, F. & Morra, L. (2022). Data augmentation for medical imaging: A systematic literature review. *Computers in Biology and Medicine*, 106391. Doi: 10.1016/j.compbiomed.2022.106391

Garg, S. & Sahu, S. (2023). Enhancing Brain Tumor Detection Through CNN and Data Augmentation: A Comprehensive Study. 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET), 14-15 Eylül 2023, ABESEC Ghaziabad, India, (pp. 536-543).

Ge, C., Gu, I. Y.-H., Jakola, A. S. & Yang, J. (2020). Enlarged training dataset by pairwise GANs for molecular-based brain tumor classification. *IEEE access*, 8, 22560-22570. Doi: 10.1109/ACCESS.2020.2969805

Giger, M. L. (2018). Machine learning in medical imaging. *Journal of the American College of Radiology*, 15 (3), 512-520. Doi: 10.1016/j.jacr.2017.12.028

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*: MIT press.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. C. (2017). Improved training of wasserstein gans. 30th Advances in neural information processing systems (NIPS 2017), 4-9 Aralık 2017, Long Beach, CA, USA, (pp. 1-11)

Gupta, V. & Bibhu, V. (2023). Deep residual network based brain tumor segmentation and detection with MRI using improved

invasive bat algorithm. *Multimedia Tools and Applications*, 82 (8), 12445-12467. Doi: 10.1007/s11042-022-13769-0

Kiani Kalejahi, B., Meshgini, S. & Danishvar, S. (2023). Brain tumor segmentation by auxiliary classifier generative adversarial network. *Signal, Image and Video Processing*, 1-7. Doi: 10.1007/s11760-023-02555-6

Kim, S., Kim, B. & Park, H. (2021). Synthesis of brain tumor multicontrast MR images for improved data augmentation. *Medical Physics*, 48 (5), 2185-2198. Doi: 10.1002/mp.14701

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. nature, 521 (7553), 436-444. Doi: 10.1038/nature14539

Lee, J.-S. & Ma, Y.-X. (2022). Stain style transfer for histological images using S3CGAN. *Sensors*, 22 (3), 1044. Doi: 10.3390/s22031044

Liu, S., Wang, Y., Yang, X., Lei, B., Liu, L., Li, S. X., Ni, D. & Wang, T. (2019). Deep learning in medical ultrasound analysis: a review. *Engineering*, 5 (2), 261-275. Doi: 10.1016/j.eng.2018.11.020

Mahajan, D. S. (2023). Generating MRI images using style transfer learning, Dublin, National College of Ireland.

Meor Yahaya, M. S. & Teo, J. (2023). Data augmentation using generative adversarial networks for images and biomarkers in medicine and neuroscience. *Frontiers in Applied Mathematics and Statistics*, 9, 1162760. Doi: 10.3389/fams.2023.1162760

Mukherkjee, D., Saha, P., Kaplun, D., Sinitca, A. & Sarkar, R. (2022). Brain tumor image generation using an aggregation of GAN models with style transfer. *Scientific reports*, 12 (1), 9141. Doi: 10.1038/s41598-022-12646-y.

Radford, A., Metz, L. & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Sajjad, M., Khan, S., Muhammad, K., Wu, W., Ullah, A. & Baik, S. W. (2019). Multi-grade brain tumor classification using deep CNN with extensive data augmentation. *Journal of computational science*, 30, 174-182. Doi: 10.1016/j.jocs.2018.12.003

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. & Chen, X. (2016). Improved techniques for training gans. 29th Advances in neural information processing systems (NIPS 2016), 5-10 Aralık 2016, Barcelona, Spain, (p. 901).

Sorin, V., Barash, Y., Konen, E. & Klang, E. (2020). Creating artificial images for radiology applications using generative adversarial networks (GANs)–a systematic review. *Academic radiology*, 27 (8), 1175-1185. Doi: 10.1016/j.acra.2019.12.024

Tariq, U., Qureshi, R., Zafar, A., Aftab, D., Wu, J., Alam, T., Shah, Z. & Ali, H. (2022). Brain Tumor Synthetic Data Generation with Adaptive StyleGANs. Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2022), 8-9 Aralık 2022, Munster, Ireland, (pp. 147-159).

Tavse, S., Varadarajan, V., Bachute, M., Gite, S. & Kotecha, K. (2022). A Systematic Literature Review on Applications of GAN-Synthesized Images for Brain MRI. *Future Internet*, 14 (12), 351. Doi: 10.3390/fi14120351

Tmenova, O., Martin, R. & Duong, L. (2019). CycleGAN for style transfer in X-ray angiography. *International journal of computer assisted radiology and surgery*, 14, 1785-1794. Doi: 10.1007/s11548-019-02022-z.

Wali, A., Ahmad, M., Naseer, A., Tamoor, M. & Gilani, S. (2023). StynMedGAN: Medical images augmentation using a new GAN model for improved diagnosis of diseases. *Journal of Intelligent & Fuzzy Systems*, 44 (6), 10027-10044. Doi: 10.3233/JIFS-223996

Wang, J. & Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, 11 (2017), 1-8.

Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J. & Shen, F. (2022). Image data augmentation for deep learning: A survey. arXiv preprint arXiv:2204.08610.

Yapici, M., Karakis, R. & Gurkahraman, K. (2023). Improving Brain Tumor Classification with Deep Learning Using Synthetic Data. *Computers, Materials and Continua*, 74 (3), 5049-5067. Doi: 10.32604/cmc.2023.035584

Zhang, Y.-D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K. & Wang, S.-H. (2019). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*, 78, 3613-3632. Doi: 10.1007/s11042-017-5243-3

Zhang, Y., Wang, S., Wu, H., Hu, K. & Ji, S. (2021). Brain tumors classification for MR images based on attention guided deep learning model. 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 1-5 Kasım 2021, Mexico, (pp. 3233-3236).

Zhou, Z., Wu, Q. J., Huang, F. & Sun, X. (2017). Fast and accurate near-duplicate image elimination for visual sensor networks. International Journal of Distributed Sensor Networks, 13 (2). Doi: 10.1177/155014771769417

Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017 IEEE International Conference on Computer Vision (ICCV), 22-29 Ekim 2017, Venice, Italy, (pp. 2223-2232).

BÖLÜM XIV

Yalın Yazılım Yapısı

Muammer AKÇAY[1]
Berna Ataş AKÇAY[2]

Yalın Üretim

Yalın üretim israfları çıkarıp sadeleştirmek, sadece müşterinin istediği ürün/hizmeti, müşterinin istediği zamanda, en az kaynak kullanarak karşılamayı hedefleyen düşünce ve teknikler bütünüdür. Womack ve Jones (Womack et al, 1996) tarafından rapor edilen yedi temel israf Şekil 1. de gösterilmiştir. Bunlar:

Fazla üretim: gereksiz fazla üretim yapmamak

Bekleme: üretim sürecinde gereksiz boşa beklemek.

Taşıma: üretim hattında gereksiz taşıma.

[1] Dr. Öğretim Üyesi, Kütahya Dumlupınar Üniversitesi
[2] 2 Yüksek Lisans Öğrencisi

Gereksiz İşlem: üretim sürecini etkilemeyen gereksiz işlerle uğraşmak.

Gereksiz hareket & Yürüme: üretim sürecine katkısı olmayan hareketler.

Tamir & Fireler: üretimi engelleyen tamir bakım, onarım için yapılan faaliyetler.

Stok: gereksiz fazla üretim tutmak.

Yalın üretim teknikleri ile bu israfların ortadan kaldırılması veya azaltılması hedeflemektedir (Sarı et al, 2018). Çalışmalar bu doğrultuda yoğunlaşmıştır.

Toyota yalın üretimi otomobil üretim hattında karmaşık üretim modelleri üretmede kullanılmıştır. Toyota hızlı ve mümkün olan düşük maliyette yazılım kullanarak gerçekleştirmiştir.

Şekil 1. *Yalın üretim bileşenleri (Womack et al, 1996)*

Yalın Yazılım

Yalın yazılımın prensipler şekil 2. de verilmiştir ve şunlardır (Karataş et al, 2014):

Bütünü en iyi yap

İsraftan Kurtul

Kalite oluştur

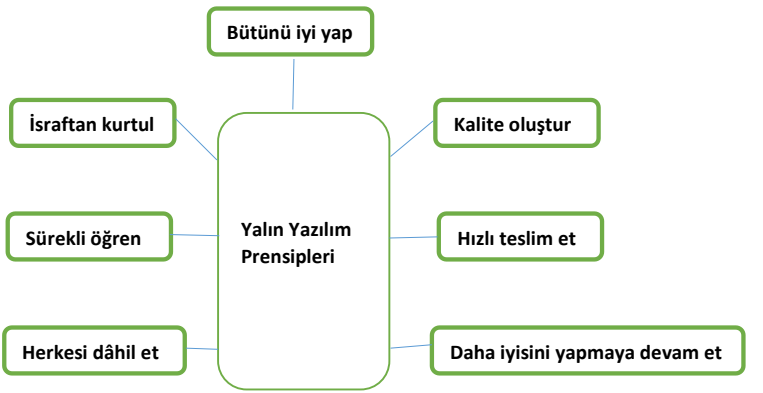
Sürekli öğren

Hızlı teslim et

Herkesi dâhil et

Daha iyisini yapmaya devam et

Bu çalışmada yalın yaklaşım ve değer akış haritalama yöntemi kullanılarak israflar belirlenmiş ve israfların ortadan kaldırılabilmesi için analizler ile süreç iyileştirmeleri yapılmıştır (Serdarasan et al,, 2021).



Şekil 2. Yalın Yazılım Bileşenleri

Yalın yazılım geliştirme özellikleri (Poppendieck et al, 2003) şunlardır:

Gereksiz ilave süreçleri, ilave özellikleri, beklemeleri, bozulan kısımları azaltmak gerekir.

Geri dönütler alarak ve iterasyonlarla öğrenmeyi arttırmak gerekir.

Mümkünse en son karar verin. Kritik kararları mümkün olduğu kadar geç vermek

Dağıtımı mümkün olduğunca en hızlı yapın. Yeni dağıtımları hızlı yapmak.

Takımı güçlü tutun. Yazılımı geliştiren ekibi güçlü tutmak.

Bütünlüğü sağlayın. Bütün sistemi uyumlu yapmak.

Bütünü (büyük resmi) görün. Büyük idealler için çalışmak.

Toyota, Intel, John Deere and Nike da yalın yazılım kullanılmıştır.

Bulgular

Çevik yazılım yöntemlerini ve yalın prensipleri kullanan Dropbox, Zalando, Wealthfront vaka çalışmaları incelenmiş kıyaslama yapılmıştır (Çubukçu et al, 2020).

Çevik ve yalın yazılım geliştirme metotları karşılaştırılmıştır. Çevik yazılımda ile birlikte geri besleme ve kullanıcı istekleri, sistemin dinamik yapısı dikkate alınarak yalım yazılım geliştirme modeli önerilmiştir (Ching et al., 2018).

Ölçeklenebilir çevik yazılım geliştirme yapısı (agile-to-lean) revize edilerek yalın yazılım geliştirme modeli geliştirilmiş ve yalın g-yazılım geliştirme modelinin faydaları, karşılaşılan fırsatlar ve ölçüm metotları sistematik olarak incelenmiştir. Geliştirilen yalın modelin faydası:

azaltılan zaman,

iyileştirilen akış,

sürekli iyileştirme,

hata onarma oranındaki iyileşme

olarak ölçülmüştür (Kiss et al. 2018).

İyileşme odaklı çevik ürün geliştirme süreci olan LAPIS'in (Logo Agile Process Improvement System) veriye dayalı makine öğrenmesi ile desteklenmiştir. Elde edilen sonuçlar incelenmiş ve değerlendirilmiştir (Tekbulut et al, 2020).

Geliştirilen Yalın Yazılım Modeli

Yalın yazılım geliştirme özellikleri mümkün olduğunca karşılamalıdır. Bunlar:

Gereksiz yapılan tüm ilave süreçleri, ilave eklenen özellikleri, boşa beklemeleri, bozulan / eskiyen kısımları azaltmak gerekir.

Geri dönütler alarak ve iterasyonlarla öğrenmeyi arttırmak gerekir. Bu aşamada öğrenme tekniklerinden yararlanabilir.

Mümkünse en son karar verin. Kritik kararları mümkün olduğu kadar geç vermek

Dağıtımı mümkün olduğunca en hızlı yapın. Gerektiği kadar hızlı yapmak.

Takımı güçlü tutun. Geliştiren ekibi güçlü ve canlı tutmak. Bunu başarmak için insan faktörünü dikkatli incelemek gerekir.

Bütünlüğü sağlayın. Bütün sistemi birlikte tutmak için çaba göstermek gerekir.

Bütünü (büyük resmi) görün. Büyük idealler için çalışmak

Sonuçlar

Sonuç olarak yazılımda bütünü en iyi yapmak, israftan kurtulmak, kaliteyi arttırmak, sürekli öğrenen, hızlı teslim edilen, herkesin dâhil edildiği bir yapının ortaya çıkması hedeflenmektedir.

Yalın yazılım geliştirme yapısının amacına ulaşması için geliştirme modeline öğrenme metotları da eklenerek çalışmalar devam etmektedir.

Sonuç olarak esnek, verimli, uyumlu, her bileşenin mutlu olduğu bir yazılım yapısı kurulmuş olacaktır.

KAYNAKÇA

Womack, J. P., ve Jones, D. T. (1996). Lean Thinking: Banish Waste and Create Wealth in Your Corporation. London: Simon and Schuster.

Sarı, E. B. (2018). Yalın üretim Uygulamaları ve Kazanımları. UİİİD-IJEAS, 2018 (17. UİK Özel Sayısı):585-600 ISSN 1307-9832.

Karataş, G., Çatal, Ç.(2014). Yalın Yazılım Geliştirme: Sistematik Eşleme Çalışması. VIII. Ulusal Yazılım Mühendisliği Sempozyumu 8-10 Eylül 2014, Güzelyurt, KKTC. Sayfa 607-612. https://ceur-ws.org/Vol-1221/58_Deneyim.pdf

Çubuçu, C., Yücel, U.O., (2020). Çevik Metodojilere karşı Yalın Prensipler: Hangi Teknoloji Girişim Şirketlerinin Büyümesi için Kullanılmalıdır. Bilişimde Güncel Uygulamalar Cilt I, Sayfa 25-43.

Serdarasan,,Ş., Ertek, E. (2021). Yazılım Geliştirme Sürecinde Değer Odaklı İyileştirme. Jornal of Industrial Engineering 32(1), 90-107. https://doi.org/10.46465/endustrimuhendisligi.809438

Poppendieck, M, Poppendieck T. (2003). Lean Software Development: An Agile Toolkit. Addison Wesley.

Ching, P.M., Mutuc,J.E. (2018). Evaluating Agile and Lean Software Development Methods from a System Dynamics Perspective. 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM).

Kiss, F. Rossi, B. (2018). Agile to Lean Software Development Transformation: a Systematic Literature Review. Proceedings of the Federated Conference on Computer Science and Information Systems ACSIS, Vol. 15, pp. 969–973 DOI: 10.15439/2018F5.

Tekbulut, T., Canbaz, N., Kaya, T.Ö (2020). LAPIS Çevik Yazılım Geliştirme Sürecinde Makine Öğrenmesi Uygulaması. 2020 Turkish National Software Engineering Symposium (UYMS).

BÖLÜM XV

Dağıtık Sistemlerde Servislerin Bulut Sunuculara Yerleştirilmesi için Kullanılan Docker Konteyner Teknolojisinde İmaj Oluşturma ve Veri İşlemleri

Işıl KARABEY AKSAKALLI[1]

Giriş

Konteyner teknolojileri, yazılımın kodunu ve tüm bağımlılıklarını paketleyen standart bir yazılım birimidir [1]. Bu sayede uygulama, bir bilgisayar ortamından diğerine hızlı ve güvenilir bir şekilde aktarılarak çalıştırılabilmektedir. Yazılım konteynerleri, fiziksel dünyadan nakliye konteynerlerinin çalışma şeklinden esinlenilerek ortaya çıkarılmıştır. Nakliye konteynerlerinin çalışma şekli incelendiğinde, farklı taşıma modları kullanılarak bu konteynerler etkili bir şekilde hareket ettirilmekte, bir kamyonun bir limana taşınması ile başlamakta, daha sonra

[1] Dr. Öğr. Üyesi, Erzurum Teknik Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü

dünyanın diğer tarafına taşınan büyük bir konteyner gemisinde binlerce başka nakliye konteynerlerinin yanında toplu bir şekilde yığılmaktadır. Burada yolculuğun hiçbir noktasında o konteynerin içindekilerin herhangi bir şekilde değiştirilmesi veya yeniden ambalajlanması gerekmemektedir. Nakliye konteynerleri her yerde var olan, standart ve dünyanın her yerinde mevcut olup kullanımı son derece kolaydır: sadece açılır, kargonun içine koyulur ve paketlenir. Her bir konteynerin içeriği diğerlerinden ayrı tutulmaktadır. Farklı içeriğe sahip konteynerler herhangi bir problem olmadan yan yana bulundurulabilmektedir. Konteyner gemisinde bir yer ayırt edildikten sonra, tüm seyahat boyunca paketlenmiş kargolar için de bir yer ayrıldığından komşu konteynerin diğer bir konteynerin alanını işgal etmesi engellenmektedir.

Yazılım konteynerleri de geliştirilen uygulamada benzer rol oynamaktadır. Konteynerleri paketlemek, geliştirilecek olan uygulama için işletim sistemi, kütüphaneler, yapılandırma dosyaları, uygulama ikili (binary) dosyaları ve diğer teknolojiler gibi gerekli olan kavramları tanımlamayı içermektedir. Konteynerler, tüm mevcut kaynakların tam olarak kullanımına izin vererek çoğu aynı makine üzerinde çalışabilmektedir. Linux konteyner ve cgroups, konteynerler arasında çapraz bulaşım olmaması için veri dosyaları, kütüphaneler, portlar, ad alanları ve hafıza içeriklerinin tümünün birbirlerinden ayrı tutulması için kullanılmaktadır. Ayrıca bir konteynerin sistem kaynaklarını ne kadar tüketebileceğinin üst sınırlarını zorlarlar, böylece kritik bir uygulama, gürültülü konteynerler tarafından sıkıştırılamaz. Birbirleri ile etkileşim halinde olmayan ve birbirlerinin yardımına ihtiyacı olmayan somut taşıma konteynerlerının aksine yazılım konteynerlerinin iyi tanımlanmış arayüzler üzerinden birbirleri ile etkileşime girmesi, onları çok güçlü hale getirebilmektedir. Örneğin bir konteyner, başka bir konteynerde çalışan bir uygulamaya, kararlaştırılan bir bağlantı noktası üzerinden erişebildiği bir veritabanı servisi sağlamaktadır. En popüler konteyner teknolojilerinden biri olan Docker teknolojisinde Docker konteyneri görüntüsü, bir uygulamayı çalıştırmak için kod, çalışma zamanı, sistem araçları, sistem

kitaplıkları ve sistem ayarları gibi gereken her şeyi içeren hafif, bağımsız, yürütülebilir bir yazılım paketidir. Konteyner imajları Docker Engine üzerinde çalışırken Docker konteynerleri haline gelmektedir. Hem Linux hem de Windows tabanlı uygulamalar için mevcut olan konteynerli yazılımlar, altyapıdan bağımsız olarak her zaman aynı şekilde çalışmaktadır. Konteynerler, yazılımı ortamdan izole etmekte, ayrıca geliştirme ve aşamalandırma arasında farklılıklar olsa bile, yazılımın minimum hata ile çalışmasını sağlamaktadır. Docker Engine üzerinde çalışan Docker konteynerleri aşağıdaki özelliklere sahiptir [1]:

Standart: Docker, konteynerler için endüstri standardını oluşturduğundan dolayı konteynerler her yerde taşınabilir bir hale gelmektedir.

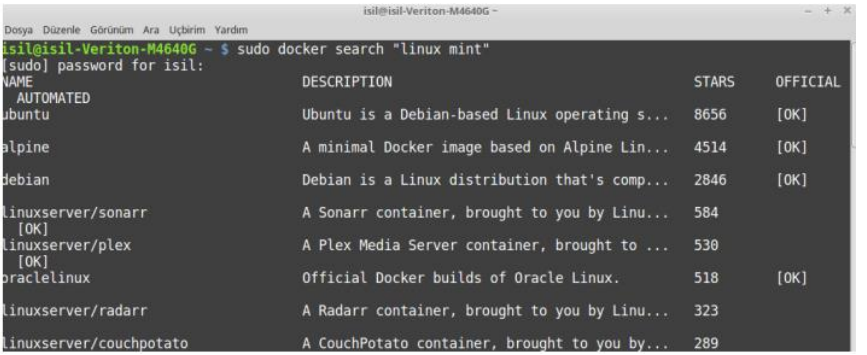
Hafif: Konteynerler, makinenin işletim sistemi çekirdeğini paylaşır ve bu nedenle uygulama başına bir işletim sistemi gerektirmez, daha yüksek sunucu verimliliği sağlayarak sunucu ve lisans maliyetlerini düşürmektedir. Güvenli: Uygulamalar konteynerlerde daha güvenli bir hale gelmektedir ve Docker, sektördeki varsayılan en güçlü yalıtım yeteneklerini sağlamaktadır.

Bu çalışmada docker konteynerlerin çalışma mantığı pratik uygulamalarla açıklanarak dockerfile ile docker imajı oluşturma, oluşturulan konteynerler arası veri paylaşım ve kopyalama işlemleri, docker ana bilgisayarda bulunan verileri konteynerler ile paylaşma, oluşturulan imajların Docker Dağıtıcı'ya (Docker Hub) aktarılması ve docker ağı gibi ileri düzey konteynerleştirme konularına uygulamalı olarak yer verilmiştir.

Docker Konteynerlerin Çalıştırılması

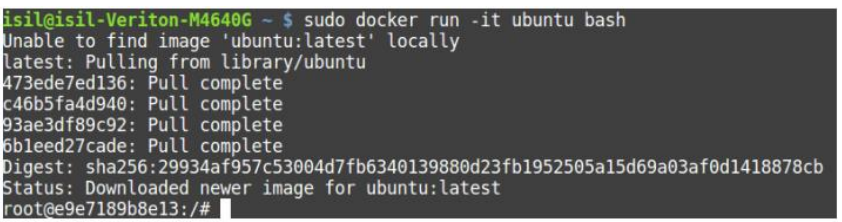
Docker, izole edilmiş bir ortam olan konteyner içerisinde uygulama yürütmeyi kolaylaştıran açık kaynaklı bir projedir. Kendine ait çekirdeği olan sanal makinenin (VM) aksine, bir konteyner ana bilgisayar işletim sisteminin çekirdeğine bağımlıdır [2]. Bu çalışmada Docker kullanılarak konteynerler oluşturulmuş ve

Docker'ın özellikleri uygulamalar ile test edilmiştir. Docker konteynerlerini çalıştırmak için kullanılan görüntüler (imaj) Docker Hub üzerinde barındırılmaktadır. Mevcut olan yüzlerce ya da binlerce görüntü içerisinden konteyneri çalıştıracak olan görüntü Docker'ın arama komutu kullanılarak bulunabilmektedir. Örneğin Linux Mint'in Docker Hub'da barındırılan bir görüntüsü olup olmadığı Şekil 1'de ifade edilen "docker search" komutu ile bulunmaktadır.



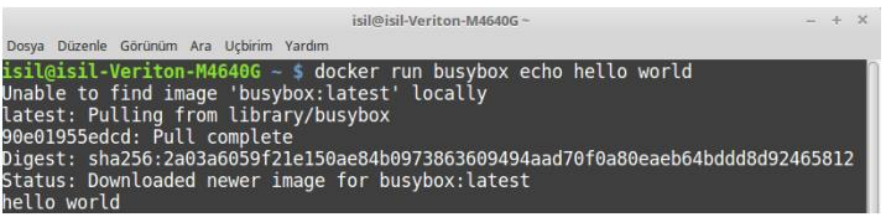
Şekil 1. Docker Hub üzerindeki görüntüler

Şekil 1'de görülen komut sonucunda listelenen verilere göre Linux Mint'in resmi bir görüntüsü olmadığı, fakat ubuntu, debian, alphine ve oracle linux için bir görüntüsü olduğu OFFICIAL-[OK] sütunundan tespit edilebilmektedir. Bu liste görüldükten sonra resmi ubuntu görüntü kullanılarak bir konteyner Şekil 2'deki komut kullanılarak çalıştırılmaktadır.



Şekil 2. Docker Hub'da yer alan ubuntu imajının çalıştırılması

Şekil 2'de görülen komut ile ubuntu görüntüsü indirildikten sonra, konteyner çalıştırılır ve çalışır halde tutulur. Bash kabuğu ile de konteyner içerisine erişim sağlanır. "@" işaretinden sonraki rakamlar konteynerin tekil id'sini ifade etmektedir. Böylece ana makine Linux Mint'i kullanırken kullanıcı bir ubuntu konteyner içerisinde çalışmaktadır. Komut satırı ile konteyner içerisine erişim yapıldıktan sonra sistem güncellemesi, konteyner içerisine herhangi bir yazılım yükleme, konteynerden istenildiği zaman çıkma (exit komutu ile) gibi işlemler yapılabilmektedir Konteynerler görüntü tabanlı olup bir görüntünün oluşması için "docker run" komutuna ihtiyacı vardır. Bir önceki örnekte veritabanında var olan bir ubuntu görüntüsü çalıştırıldığı için açık kayıt defterinden (public registry) görüntüyü çekmeye ihtiyaç duyulmamıştır. Fakat Şekil 3'te verilen örnekte sistemde bulunmayan busybox görüntüsü açık kayıt defterinden çekilmektedir. Bir kayıt defteri, Docker istemcisinin iletişime geçebileceği ve Docker görüntülerinin indirildiği bir katalog olarak ifade edilmektedir [3]. Görüntü kayıt defterinden çekildiği zaman Docker konteyneri başlatır ve "echo hello world" komutunu çalıştırarak komut satırı arayüzüne hello world yazısını yazdırır. Böylece konteynerin çalışıyor olduğu anlaşılmaktadır.



Şekil 3. Açık kayıt defterinden yeni görüntü oluşturulması

Dockerfile ile docker imajı oluşturma

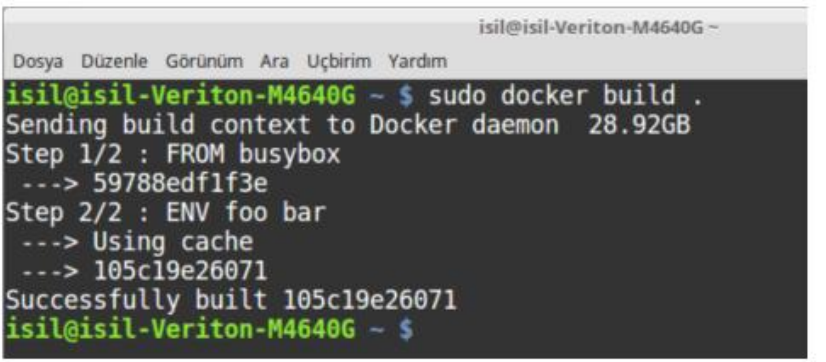
Geliştirici açık kaynak bir Docker kayıt defteri üzerinden görüntürleri indirmesi yerine kendi Docker imajını oluşturmak isterse Dockerfile isimli bir metin dosyasına bir imajı oluşturması gerekmektedir. Bunun için gerekli paketler, dizinler ve çevre

değişkenlerini tanımlama gibi Docker'ın ihtiyaç duyacağı bilgileri yazması gerekmektedir. Bulunulan dizinde Dockerfile oluşturmak için "touch Dockerfile" komutu girilmekte ve "gedit Dockerfile" komutu ile dosya açılmaktadır. Basit bir örnek olarak Dockerfile içerisine aşağıdaki metindeki gibi busybox adında bir konteyner oluşturularak "docker build" komutu ile yapılandırılmaktadır [3].

FROM busybox

ENV foo=bar

Daha sonra busybox2 isminde yeni bir imaj inşa etmek için Şekil 4'deki gibi "sudo docker build" komutu kullanılmaktadır. "sudo" komutu sayesinde güvenliği sağlamak amacıyla yönetici olarak işlem yapılmakta, aksi taktirde yeni bir imaj oluşturmamaktadır.

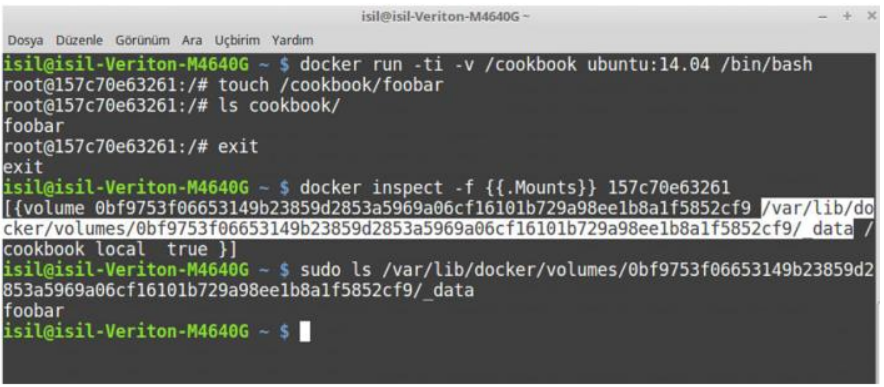


Şekil 4. Dockerfile içerisinde belirtilen imajı build komutu ile inşa etme

İmaj oluşturma işlemi tamamlandıktan sonra "docker images" komutu ile elde edilen çıktılarda busybox ve busybox2 imajı görülmektedir. Ortam değişkeni foo'nun bar olarak eşleştiğini görmek amacıyla çalıştırmak için yazılan "docker run busybox2 komutuna env | grep foo" metni eklenmektedir [3].

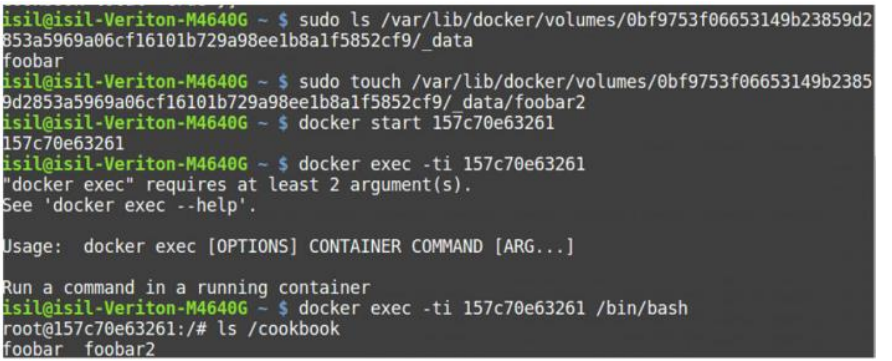
Konteynerler Arasında Veri Paylaşımı

Bir konteynerde tanımlanan bir disk bölümü, diğer konteynerler ile paylaşmak amacıyla bir sunucu biriminin ve bir konteynerdeki yolu belirtmek için, bu birimin bağlanacağı -v seçeneği kullanılmaktadır. Ana makine yolu yok olduğunda, bir veri konteyner oluşturulmaktadır. Belirtilen disk bölümü, konteyner imajı oluşturmak için kullanılan salt okunur katmanların üstüne katmanlı olmayan bir okuma-yazma dosya sistemi olarak konteyner içinde yaratılmaktadır. Docker bu dosya sistemini yönetmekte, fakat kullanıcı ancak ana bilgisayardan okuyabilir ve yazabilmektedir. Şekil 5'te konteynerler arasındaki bu veri paylaşımı gösterilmektedir.



Şekil 5. Konteynerler Arasında Dosya Kopyalama

Konteyner başlatıldığı zaman Docker /cookbook dizini oluşturmaktadır. Konteyner ile birlikte bu dizin okunabilmekte ve yazılabilmektedir. Konteynerden çıkıldıktan sonra ana makine üzerinden birimin nerede oluştuğunu öğrenmek için "inspect" komutu kullanılmaktadır. Docker bunu / var / lib / docker / volumes / altında oluşturmaktadır. Ana makine üzerinden bu dizin okunabilir ve yazılabilmektedir. Konteyner Şekil 6'daki gibi yeniden başlatıldığında değişiklikler devam etmektedir [3].

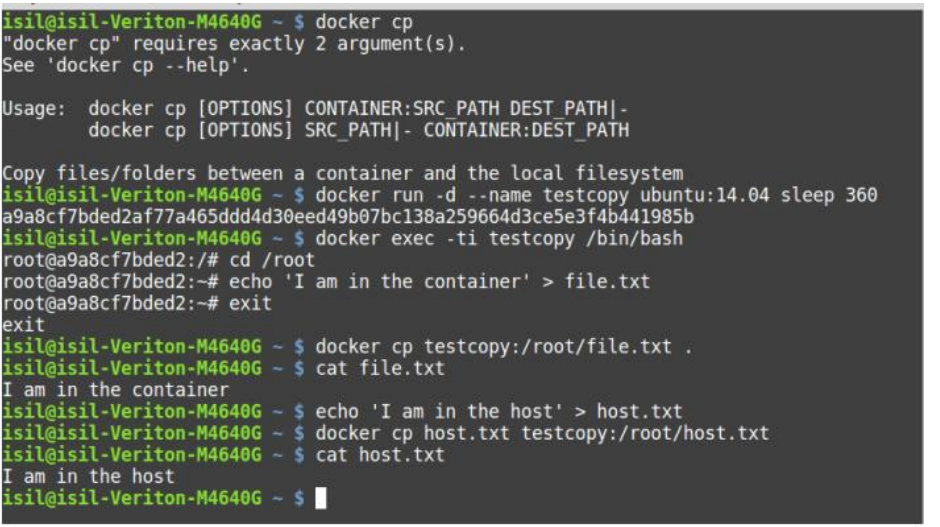


Şekil 6. Kopyalanan Dosyanın Konteynerde Görülmesi

Bu veri disk bölümünü diğer konteynerler ile paylaşmak için "–volumes-from" komutu kullanılmaktadır. Yeni bir konteyner ve oluşturulduktan sonra bu konteynerdeki kaynağı diğer konteynere paylaşımını sağlama işleminde konteyner çalışmıyor olsa da, –volumes-from (örneğin; docker run -ti –volumes-from data ubuntu:14.04 /bin/bash) ile disk bölümüne yerleştirilmektedir [3].

Konteynerler Arasında Veri Kopyalama

Herhangi bir disk bölümü yapılandırması olmaksızın çalışan bir konteynere dosya kopyalamak ya da konteynerden Docker ana bilgisayarına dosya kopyalamak için "docker cp" komutu kullanılmaktadır. Docker cp komutunun kullanımı Şekil 7'de görüldüğü gibidir. Uyuyan bir konteyner çalıştırılarak bu konteyner içerisinde file.txt dosyası oluşturulmakta ve içerisine "echo" komutu ile "I am in the container" yazılmaktadır. Daha sonra bu konteynerden çıkılarak ana bilgisayara konteyner içerisindeki file.txt dosyası kopyalanmakta ve çıktıda dosya içerisinde yazılı olan "I am in the container" yazısı ekrana yansımaktadır. Aynı şekilde ana bilgisayardan konteynere de host.txt dosyası kopyalanmaktadır.



Şekil 7. "docker cp" komutu ile veri kopyalama

Bir konteynerden başka bir konteynere veri kopyalamak için ise ana bilgisayardaki dosyaları geçici olarak kaydederek iki yöntem birleştirilmekte, örneğin aynı dosya her iki çalışan konteynerden kopyalanacaksa konteynerlere c1 ve c2 adları verilerek aşağıdaki gibi kopyalanabilmektedir.

$ docker cp c1:/root/file.txt

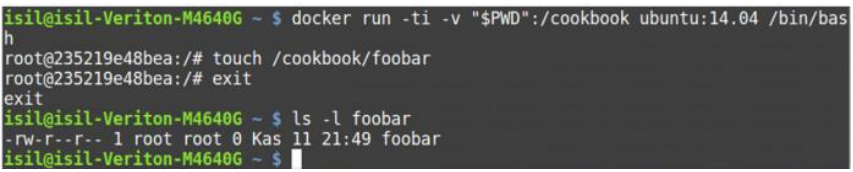
$ docker file.txt c2:/root/file.txt.

Docker Ana Bilgisayarda Bulunan Verileri Konteynerler ile Paylaşma

Bir ana bilgisayar birimini bir konteynere bağlamak için docker run -v seçeceği kullanılmaktadır. Örneğin ana makinenin çalışma dizinini bir konteyner içerisinde /cookbook dizinine aktarmak için aşağıdaki komut satırı kullanılmaktadır:

```
$ ls
data
$ docker run -ti -v "$PWD":/cookbook ubuntu:14.04 /bin/bash
root@11769701f6f7:/# ls /cookbook
data
```

Yukarıda verilen komut satırında, ana makinede çalışan dizin konteynerdeki cookbook dizinine bağlanmaktadır. Konteynerde dosya veya dizin oluşturulduğunda, değişiklikler Şekil 8'deki gibi doğrudan ana makine çalışma dizinine yazılmaktadır. Varsayılan olarak okuma ve yazma izni modundaki birimi sadece okunur hale getirmek için "-v "$PWD":/cookbook:ro" komutu kullanılmaktadır.



Şekil 8. Konteynerde oluşturulan yeni dizinin ana makinede listelenmesi

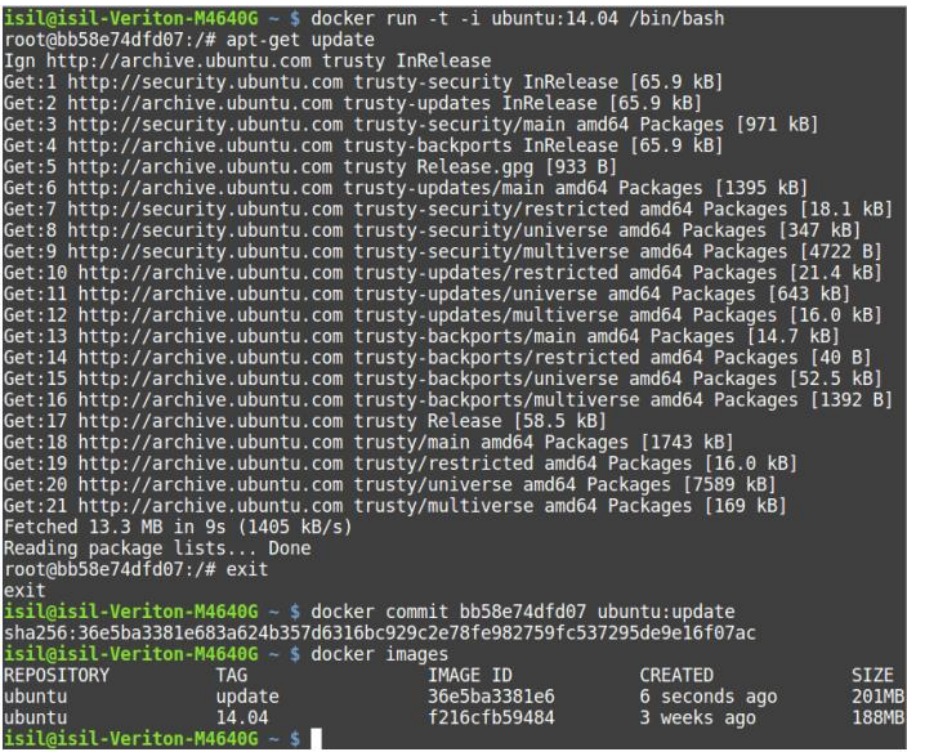
Docker İmajı Oluşturma Ve Paylaşma

Önceki bölümlerde Docker kullanarak temel prensipler keşfedilmiş, bu bölümde ise kullanıcının kendi konteyner imajını nasıl oluşturacağı hakkında bilgiler verilmektedir. Kullanıcı var olan bir uygulamasını paketlemek isteyebilir ya da Docker'dan yararlanarak sıfırdan bir konteyner imajı inşa etmek isteyebilir. Bu bölümde de konteyner imajı oluşturma ve bu imajların diğer konteynerler ile paylaşımı anlatılmaktadır. Dışa aktarma (export) ve içe aktarma (import) özelliğini kullanarak imaj paylaşma kolaylıkla yapılabilmektedir, ancak imajları başkalarıyla paylaşmak ve Docker Hub'ı sürekli bir tümleştirme (continuous integration) hattına entegre etmek için Docker Hub tercih edilebilmektedir. Docker Hub bir uygulama deposudur ve Docker üzerindeki imajlar açık bir şekilde paylaşılabilmekte, GitHub ve Bitbucket gibi kod barındırma hizmetleri ile entegre bir şekilde otomatik olarak inşa

edilebilmektedir. Kullanıcı Docker Hub kullanmak istemediğinde kendi Docker imaj kayıt defterini (registry) oluşturabilir ve otomatik olarak derleyebilmektedir [3].

Bir Konteynerde Yapılan Değişiklikleri Bir İmaja Teslim Ederek (Committing) Saklamak

Konteyner içerisinde değişiklikler yapıldıktan sonra bu değişiklikleri sistemi kapattıktan sonra veya konteyneri durduktan sonra da saklamak için "docker commit" komutu kullanılmaktadır. Şekil 9'da Docker commit komutunu kullanmadan önce etkileşimli bir bash kabuğuna sahip bir konteyner başlatılmakta ve içerisinde paketler güncellenmektedir [3].



Şekil 9. "docker commit" komutu ile konteyner başlatma

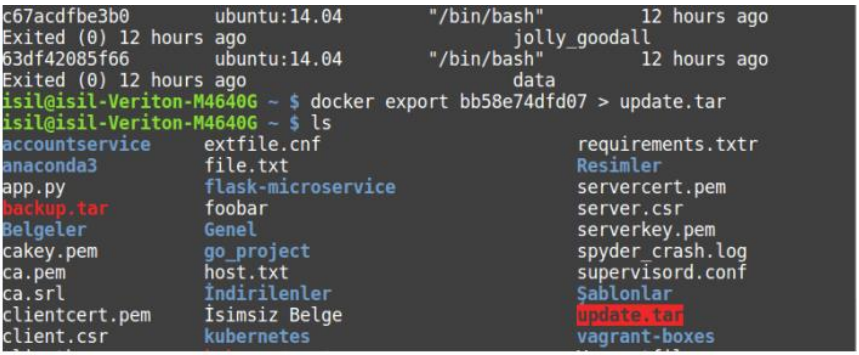
Paketler güncellendikten sonra konteynerden çıkıldığında, konteynerin çalışması durmakta fakat docker rm ile tamamen kaldırılana kadar ana bilgisayar içerisinde bulunmaktadır. Konteyner tamamen kaldırılmadan önce yapılan değişiklikleri kaydetmek ve "ubuntu:update" adında yeni bir imaj oluşturmak için docker commit CONTAINER ID ubuntu:update komutu ile bb58e74dfd07 konteynerindeki değişiklikler update etiketindeki ubuntu imajına kaydedilir. Bu işlemden sonra 14.04 etiketli güvenle kaldırılabilmekte ve "ubuntu:update" isimli yeni imaj üzerinden devam edilebilmektedir. Aynı zamanda konteynerde yapılan değişiklikler Şekil 10'da görüldüğü gibi "docker diff" komutu ile kontrol edilebilmektedir. Şekilde görülen harflerden A, listelenen dosya ya da dizinin eklendiği, C belirtilen dizinde bir değişiklik yapıldığı, D ise belirtilen dizinin silindiği anlamına gelmektedir.

```
isil@isil-Veriton-M4640G ~ $ docker diff bb58e74dfd07
C /root
A /root/.bash_history
C /tmp
C /var
C /var/cache
C /var/cache/apt
D /var/cache/apt/pkgcache.bin
D /var/cache/apt/srcpkgcache.bin
C /var/lib
C /var/lib/apt
C /var/lib/apt/lists
A /var/lib/apt/lists/archive.ubuntu.com_ubuntu_dists_trusty-backports_InRelease
A /var/lib/apt/lists/archive.ubuntu.com_ubuntu_dists_trusty-backports_main_binar
y-amd64_Packages.gz
A /var/lib/apt/lists/archive.ubuntu.com_ubuntu_dists_trusty-backports_multiverse
_binary-amd64_Packages.gz
A /var/lib/apt/lists/archive.ubuntu.com_ubuntu_dists_trusty-backports_restricted
_binary-amd64_Packages.gz
A /var/lib/apt/lists/archive.ubuntu.com_ubuntu_dists_trusty-backports_universe_b
inary-amd64_Packages.gz
A /var/lib/apt/lists/archive.ubuntu.com_ubuntu_dists_trusty-updates_InRelease
A /var/lib/apt/lists/archive.ubuntu.com_ubuntu_dists_trusty-updates_main_binary-
amd64_Packages.gz
```

Şekil 10. "docker diff" komutu ile yapılan değişikliklerin kontrolü

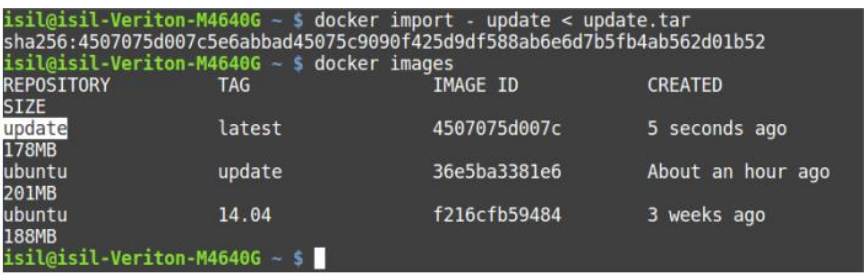
Paylaşım için Tar Dosyası Olarak Konteynerlerin ve İmajların Kaydedilmesi

Ortak çalışanlarla paylaşmak istenilen yada saklamak istenilen konteynerler veya imajlar için Docker CLI kullanılmaktadır. Önceden oluşturulmuş bir imajdan tarball oluşturmak için Docker CLI kaydetme ve yükleme komutları veya konteynerler için Docker CLI içe aktarma (import) ve dışa aktarma (export) komutları bulunmaktadır. Durdurulmuş bir konteyneri "export" komutu kullanarak tar dosyası haline getirmek için Şekil 11'de görüldüğü gibi "docker export CONTAINERID > dosyaismi.tar" komutu yazılmaktadır.



Şekil 11. Konteynerin tar dosyası haline getirilmesi

Bu konteyner yerel bir şekilde veya Şekil 12'deki gibi "docker import" komutu ile yeni bir imaj olarak oluşturulabilmektedir.



Şekil 12. Tar dosyasından yeni imaj oluşturma

Eğer oluşturulan bu imaj, başka kullanıcılar ile paylaşılmak istenirse bir web sunucuna tar dosyası yüklenebilir ve paylaşılan kullanıcının imajı indirmesine ve "import" komutu ile kendi Docker ana bilgisayarına eklemesine izin verilebilmektedir. Daha önce oluşturulan imajlar üzerinde çalışılmak istendiğinde ise Şekil 13'deki gibi "load" komutu kullanılarak Update imajı "docker rmi" komutu ile silinse bile update1.tar dosyası ile update imajı tekrar oluşmaktadır.

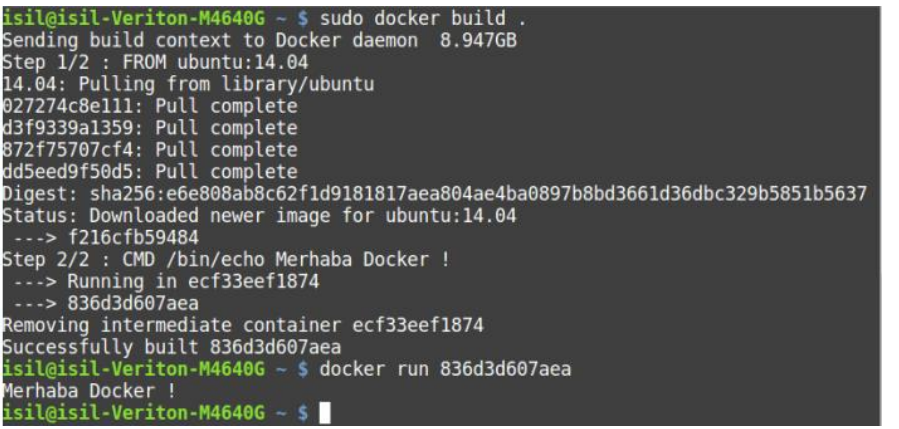
```
-rw------- 1 isil isil 187193344 Kas 12 12:57 update1.tar
-rw-rw-r-- 1 isil isil 187185152 Kas 12 12:56 update.tar
isil@isil-Veriton-M4640G ~ $ docker rmi update
Untagged: update:latest
Deleted: sha256:fc4103f5e858ebbc397dfa576f6e5c1f61757f56d13da6f356d38a22fb81b903
Deleted: sha256:de613a8c29c98ab327ca02bd339763baa75632c3237182e9b5dd97eb2b3854e9
isil@isil-Veriton-M4640G ~ $ docker load < update1.tar
de613a8c29c9: Loading layer  187.2MB/187.2MB
Loaded image: update:latest
isil@isil-Veriton-M4640G ~ $ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
update              latest              fc4103f5e858        40 seconds ago
178MB
ubuntu              update              36e5ba3381e6        3 hours ago
201MB
ubuntu              14.04               f216cfb59484        3 weeks ago
188MB
isil@isil-Veriton-M4640G ~ $
```

Şekil 13. Silinen imajın "load" komutu ile tekrar yüklenmesi

Dockerfile Oluşturma

Önceki çalışmalarda bir konteyneri etkileşimli modda çalıştırma, konteynerde değişiklik yapma ve sonra da yeni bir imaj oluşturmak için bu değişiklikleri işleme uygulamaları yapılmıştır. Bu başlıkta ise kullanıcının kendi imajını oluştumayı otomatikleştirmesi ve yapı adımlarını başkaları ile paylaşması için izlemesi gereken yollardan bahsedilmektedir. Bir Docker imajını oluşturmayı otomatikleştirmek için Dockerfile olarak adlandırılan Docker manifestosunda inşa etme aşamaları belirtilmelidir [4]. Bu metin dosyası; yeni konteynernın hangi temel imaja bağlı olduğunu, çeşitli bağımlılıkları ve uygulamaları indirmek için hangi adımlara ihtiyaç duyduğunu, hangi dosyaların imajda hazır bulunması gerektiğini, imajın nasıl konteynere uygun hale getirileceğini ve bir

konteyner başladığında hangi komutun çalıştırılacağını tanımlayan komut kümelerini kullanmaktadır. Basit bir Dockerfile dosyasının içeriği aşağıdaki gibidir: “FROM ubuntu:14.04 ENTRYPOINT ["/bin/echo", Merhaba Docker:)]” Dockerfile’daki “FROM” komutu yeni imajın hangi imaja temellendirileceğini belirtmektedir. Burada, Docker Dağıtıcı’daki (Docker Hub) Resmi Ubuntu (Official Ubuntu) deposundan ubuntu: 14.04 görüntüsü seçilmiştir. ENTRYPOINT komutu ise, bu görüntüye dayalı bir konteyner başlatıldığında hangi komutun çalıştırılması gerektiğini söylemektedir [5]. Bu dosya oluşturulduktan sonra Şekil 14’deki gibi “sudo docker build .” komutu ile istenilen imaj oluşturulmaktadır.



Şekil 14. Dockerfile dosyasına yazılan imajın oluşturulması

Build komutundan sonra konteyner artık çalışmaya hazırdır ve “docker run” komutu ile çalıştırılmaktadır. Konteyner çalıştırıldıktan sonra Dockerfile içerisindeki ENTRYPOINT yönergesiyle tanımlanan komut yürütülerek “Merhaba Docker:)” yazısı ekrana gelmektedir. Bu da konteynernın imaj kullanılarak oluşturulup başlatıldığı anlamına gelmektedir.

Flask Uygulamayı Konteyner İçerisinde Paketleme

Linux Ubuntu işletim sistemi üzerinde çalışan ve bir Python çerçevesi olan Flask ile birlikte bir web uygulaması, bir kapsayıcı içerisinde çalıştırılmak istendiğinde öncelikle hello.py dosyası içerisine aşağıdaki kod satırları yazılmaktadır [6].

```
#!/usr/bin/env python
from flask import Flask
app = Flask(__name__)
app.route(´/hi´)
def hello_world():
return ´Hello World´
if __name__ == ´__main__´:
app.run(host=´0.0.0.0´, port=5000)
```

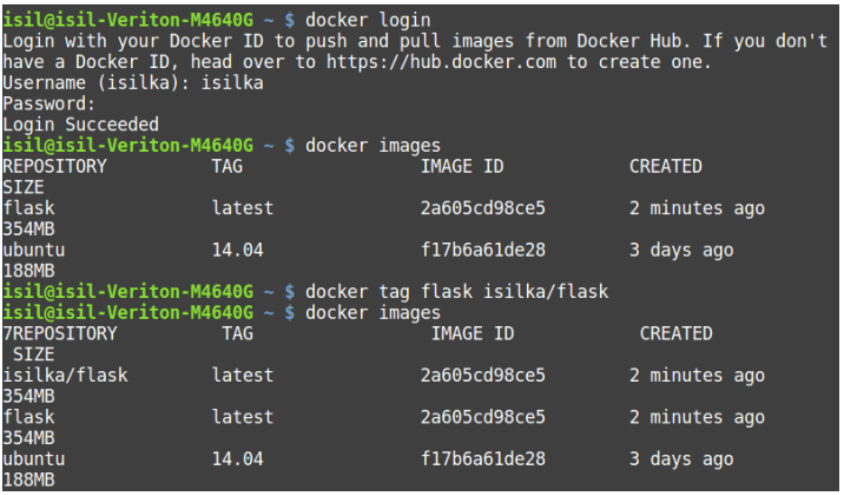
Bu dosyayı bir konteyner içerisinde çalıştırmak için bir Dockerfile oluşturarak bu dosyayı çalıştırmak için gerekli RUN komutları ve hangi portu kullanacağı EXPOSE komutu ile belirtilmektedir. Ayrıca ADD komutu kullanılarak da uygulamayı konteyner dosya sistemi içerisine taşımak gerekmektedir. Son olarak CMD komutu ile spesifik olarak kapsayıcının python /tmp/hello.py dosyasını çalışma zamanında çalıştıracağını belirtmektedir. Örnek bir Dockerfile dosyası içeriğine [3] nolu referanstan ulaşılabilir.

flask imajı oluşturduktan sonra konteyneri düzenleyen -d ayarı ve Docker ana bilgisayarda hangi portta çalışacağını belirtmek için -P seçeneği kullanılarak "docker run -d -P flask" komutu yazılarak konteyner çalıştırılmaktadır. Konteynerin çalışıyor olup olmadığı "docker ps" komutunda listelenme durumuna göre anlaşılmaktadır.

Oluşturulan İmajın Docker Dağıtıcıya (Docker Hub) Aktarılması

Docker Hub üzerinde, Dockerfile yazılarak oluşturulan imajı paylaşmak için öncelikle Docker Hub hesabı oluşturulmakta, sonrasında Docker host üzerinden huba giriş yapılmakta ve imajın hub'a aktarılması işlemi gerçekleşmektedir [7]. Şekil 15'te görüldüğü gibi foobar etiketi kullanılarak oluşturulan "isilka/foobar"

imajı önce ana bilgisayarda oluşturulup daha sonra “push” komutu ile Docker Hub’a aktarılmaktadır. Burada Docker Hub için oluşturulan kullanıcı adı (bu çalışma için:isilka) ile birlikte imajın etiketlenmesi işlemi yapılmaktadır.



Şekil 15. Docker Hub’da imaj etiketleme

isilka deposu içerisinde kullanılacak olan flask uygulaması “docker push isilka/flask” komutu ile Docker Hub’a aktarılmaktadır. Oluşturulan Flask imajının https://hub.docker.com/ adresinden Docker Hub deposuna yüklenmektedir.

Docker Ağı

Dağıtık uygulamalar inşa edildiği zaman servislerin birbirleri ile iletişiminin sağlanması gerekmektedir. Konteynerlerde çalışan bu servisler tek bir makinede veya birçok makinede, hatta veri merkezlerinde bulunmaktadır. Bu yüzden herhangi bir Docker tabanlı dağıtık uygulama için konteyner ağ iletişimi oldukça önemlidir [8]. Konteynerlerin haberleşmesi için kullanılan yöntemler sanal makineler için kullanılan yöntemlere benzemektedir. Ana bilgisayar üzerinde olan konteyner, yazılım anahtarına bağlanabilir ve ip tabloları, konteynerler arasındaki ağı

kontrol etmek ve ana bilgisayarın bağlantı noktaları üzerindeki konteynerde çalışan işlemleri tutmak için kullanılmaktadır. Bu benzerliğin yanısıra konteyner haberleşmesini sanal makineden ayıran önemli bir özellik vardır. Konteynerlerle, kullanılmakta olan ağ yığını seçilebilmektedir. Örneğin kullanıcı ana bilgisayarının ağ yığınını bir konteyner ile paylaşabilmekte, böylece konteynere aynı IP adresi verilebilmektedir. Bu durum ayrıca aynı ağ yığının konteynerler arasında paylaşımına da izin vermektedir.

Bir Konteynerin IP Adresini Bulma

Varsayılan bir Docker ağında bir konteynerin IP adresini bulmak için birçok yöntem bulunmaktadır. Bu yöntemlerden ilki "docker inspect" komutunu aşağıdaki formatta kullanmaktır. Aynı zamanda "docker exec" komutunu kullanarak konteyner hem çalıştırılmakta hem de IP adresi öğrenilebilmektedir [9].

```
$ docker run -d -name nginx nginx
$ docker inspect -format ' .NetworkSettings.IPAddress ' nginx
172.17.0.3
$ docker exec -ti nginx bash
root@fe3f887f7d0b:/# cat /etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6loopback
fe00::0 ip6localnet
ff00::0 ip6mcastprefix
ff02::1 ip6allnodes
ff02::2 ip6allrouters
172.17.0.3 fe3f887f7d0b
```

Ana Bilgisayar Üzerinde bir Bağlantı Noktası Atama

Docker, "docker run" komutunda -P seçeneğini kullanarak ana bilgisayar üzerindeki bir bağlantı noktasına konteynerdeki ağ bağlantısını dinamik olarak eşleştirmektedir. Kullanıcı yine -P ayarı ile istediği bağlantı noktasına eşleştirme yapabilmektedir. Bir örnek

üzerinden anlatılacak olursa, kullanıcının bir Python Flask uygulamasını koşturduğu bir imajı inşa ettiği düşünülmektedir. Bu imajı oluşturmak için yazılan Dockerfile dosyasının içeriği aşağıdaki gibidir:

Dockerfile yazıldıktan sonra imaj inşa edilip herhangi bir bağlantı noktası bayrağı kullanmadan konteyner çalıştırılmaktadır [10].

$ docker build t flask

$ docker run d -name foobar flask

Konteyner çalıştırıldıktan sonra Flask uygulamasına 5000.port üzerinden erişilebilmekte ve IP adresi görülebilmektedir:

$ docker inspect f '.NetworkSettings.IPAddress' foobar

172.17.0.4

$ curl http://172.17.0.4:5000/ Hello World!

Fakat bu uygulamaya ana bilgisayar dışından erişilememektedir. Bunu gerçekleştirmek için konteyner port eşleştirmesi kullanarak yeniden çalıştırılmaktadır:

$ docker kill foobar $ docker rm foobar

$ docker run -d -p 5000 –name foobar flask

$ docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

e9a7be6363ea flask "python /tmp/hello.py" 4 seconds ago Up 3 seconds 0.0.0.0:32768- >5000/tcp foobar

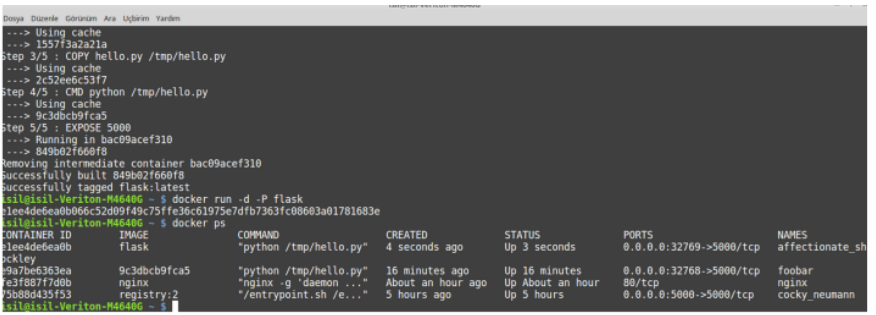
Yukarıda görüldüğü gibi "docker ps" komutundaki port sütununun artık konteynerin port 32768 ile 5000 arasında bir eşleşmeye döndüğü görülmektedir. Ana bilgisayar 0.0.0.0 arayüzündeki TCP 32768 bağlantı noktasını dinler ve istekleri konteynerin 5000 numaları bağlantı noktasına iletmektedir. Docker ana bilgisayarını 32768 numaralı bağlantı noktasında bükerek (try to curl) Flask

uygulamasına ulaşıldığı görülmektedir. “docker port” komutu kullanılarak da bir konteynernın port eşleştirmeleri listelenmektedir.

$ docker port foobar 5000

0.0.0.0:32768

Dockerfile dosyasına EXPOSE 5000 satırı eklenip “build” komutu çalıştırıldıktan sonra Docker otomaik olarak uygun eşleşmeyi sağlamaktadır. Bağlantı noktasını expose etmek için -P bayrağı kullanılmaktadır. Yani konteyneri çalıştırıken “docker run -d -P flask” komutu kullanıldıktan sonra “docker ps” listesinde eşleşmenin otomatik olarak yapıldığı Şekil 16’daki gibi görülmektedir.



Şekil 16. Docker port ile konteynere eşlenen portun listelenmesi

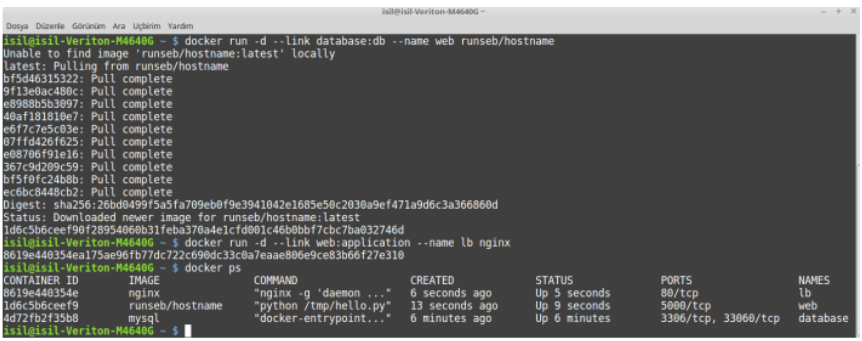
Kullanıcı aynı zamanda TCP ya da UDP protokollerini seçerek birden fazla konteyner bağlantı noktasını da gösterebilmektedir. Örneğin bağlantı noktası 5000’i TCP üzerinden, 53’ü ise UDP üzerinden göstermek istenildiğinde Şekilde uygulaması yapılan “docker run d p 5000/tcp p 53/udp flask” komutu yazılmaktadır. Bağlantı noktası eşleştirme işlemi iki mekanizma tarafından yapılmakta ve bunlardan ilki, varsayılan olarak Docker ana bilgisayarın IP tablosunu değiştirebilmektedir. Flask uygulaması çalıştırılırken IP tablosu kuralları kontrol edilirse Docker zincirinde yeni bir kural bulunabilmektedir. İkinci olarak Docker, dinamik olarak seçilen bağlantı noktasını kullanarak ana bilgisayar arabirimini dinleyen küçük bir vekil sunucu başlatmaktadır.

İşlemleri listelemek için “ps -ef | grep docker” komutu kullanılmaktadır.

Docker’da Konteynerleri Bağlamak

Birçok servisten oluşan dağıtık bir uygulama tasarlandığında bu servislerin nerede olduğunu, sistemin çeşitli bileşenlerinin birbirlerine nasıl ulaşabileceğini keşfetmenin bir yoluna ihtiyaç duyulmaktadır. Kullanıcı manuel olarak her bir servisin IP adresini çıkarabilmektedir fakat ölçekleme için kendi kendini keşfeden (self-discovery) bir sisteme ihtiyaç vardır [11]. Buna çözüm olarak konteynerler “–link” komutu ile birbirlerine bağlanmaktadır. Konteynerleri Şekil 17’deki gibi birbirlerine bağlamayı göstermek adına aşağıdaki komutlar kullanılarak veritabanı, web uygulaması ve yük dengeleyici olmak üzere üç katmanlı bir sistem tasarlanmaktadır. Veritabanından başlayarak web uygulamasına bağlanmaktadır. Bağlantıları silmek için ise her konteynere isim verilmelidir [3].

```
$ docker run d -name database e MYSQL_ROOT_PASSWORD=root mysql
$ docker run d -link database:db -name web runseb/hostname
$ docker run d -link web:application -name lb nginx
```



Şekil 17. Konteynerlerin Birbirlerine Bağlanması

Konteynerleri birbirine yukarıdaki gibi bağlamanın sonucunda uygulama konteyneri artık veritabanına işaret eden çevre değişkenlerini içermektedir. Benzer şekilde yük dengeleyici de uygulama konteynerine işaret eden çevre değişkenlerini içermektedir. Şekil 18'deki bu çevre değişkenleri hakkındaki detaylar konteynerlerin çalıştırılması ile listelenmektedir.

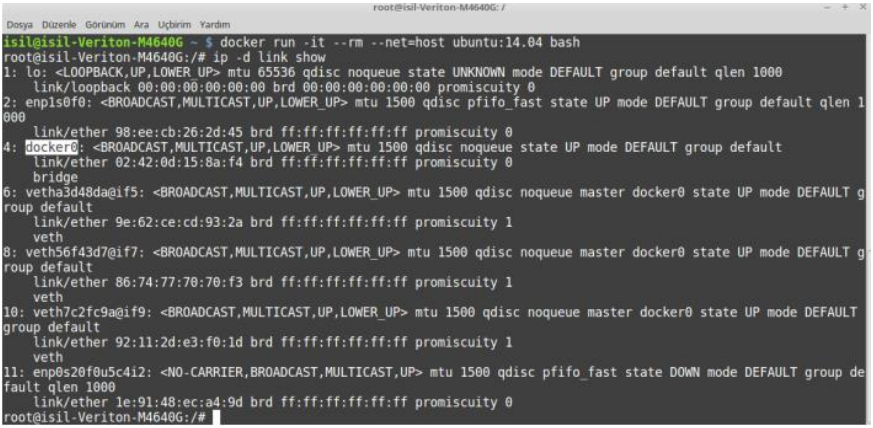
```
isil@isil-Veriton-M4640G ~ $ docker exec -ti web env | grep DB
DB_PORT=tcp://172.17.0.2:3306
DB_PORT_3306_TCP=tcp://172.17.0.2:3306
DB_PORT_3306_TCP_ADDR=172.17.0.2
DB_PORT_3306_TCP_PORT=3306
DB_PORT_3306_TCP_PROTO=tcp
DB_PORT_33060_TCP=tcp://172.17.0.2:33060
DB_PORT_33060_TCP_ADDR=172.17.0.2
DB_PORT_33060_TCP_PORT=33060
DB_PORT_33060_TCP_PROTO=tcp
DB_NAME=/web/db
DB_ENV_MYSQL_ROOT_PASSWORD=root
DB_ENV_GOSU_VERSION=1.7
DB_ENV_MYSQL_MAJOR=8.0
DB_ENV_MYSQL_VERSION=8.0.13-1debian9
isil@isil-Veriton-M4640G ~ $ docker exec -ti lb env | grep APPLICATION
APPLICATION_PORT=tcp://172.17.0.3:5000
APPLICATION_PORT_5000_TCP=tcp://172.17.0.3:5000
APPLICATION_PORT_5000_TCP_ADDR=172.17.0.3
APPLICATION_PORT_5000_TCP_PORT=5000
APPLICATION_PORT_5000_TCP_PROTO=tcp
APPLICATION_NAME=/lb/application
isil@isil-Veriton-M4640G ~ $
```

Şekil 18. Çevre değişkenlerinin listelenmesi

Konteyner Ağı Ad Alanı Seçme

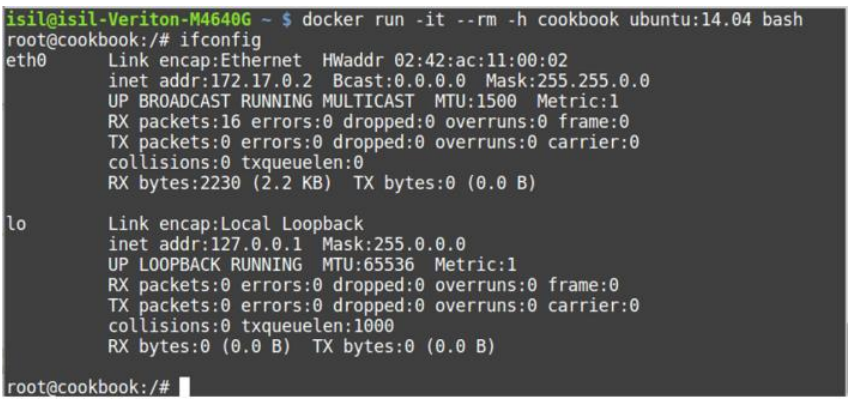
Bir konteyner başlatılırken özel bir ağ ad alanı seçmek istenebilmektedir. Konteynerlerde çalıştırılan belirli uygulamalar için, varsayılan köprü ağından farklı bir ağ kurulumu kullanmak gerekebilir veya hiç bir ağa ihtiyaç olmayabilir. "-net=none" komutu ile herhangi bir ad alanı olmadan Docker makine üzerinde bir konteyner koşturulabilmektedir. Şekilde görüldüğü gibi herhangi bir ağ ad alanı girilmeden çalıştırılan konteynerde ağ bağlantıları listelendiğinde yalnızca yerel adresler görülmektedir. Diğer ağ arayüzleri ve ağ rotaları bulunmamaktadır. -net=host kullanarak konteyner başlatıldığı zaman bu konteynerde yer alan ağ bağlantıları

listelendiğinde docker 0 köprüsü dahil olmak üzere ana bilgisayarda görülen tüm arayüzlerin aynısı Şekil 19'daki gibi görülmektedir.



Şekil 19. Ana bilgisayardaki arayüzlerin listelenmesi

Son olarak halihazırda çalışan bir diğer konteynerin ağ yığıtını kullanmak önerilmektedir [3]. Makine adı cookbook olan bir konteyner Şekil 20'deki gibi başlatılıp IP adresleri listelendiğinde komut satırında IP adresinin 172.17.0.2 olduğu ve makine isminin cookbook'a ayarlandığı görülmektedir.



Şekil 20. Çalışan bir konteynerin ağ yığıtını yeni bir konteynerde kullanma

Bir Örgü Ağında Konteynerleri Çalıştırma

Kullanıcı oluşturduğu konteynerler için otomatik IP adresi ataması yapan ve DNS aracılığıyla entegre edilmiş servis keşfi ile tek bir makineden binlerce makineden oluşan çoklu veri merkezlere ölçeklenen bir ağ yaratmak istediğinde Weave Net kullanabilir. Weave Net'i denemek amacı ile Ubuntu işletim sistemine sahip ve Docker indirilen iki Vagrant makinesi aşağıdaki komutlar ile oluşturulmuştur [12].

$ git clone https://github.com/how2dock/docbook.git
$ cd ch03/weavesimple
$ vagrant up ("vagrant up" komutundan sonra "172.17.8.101 weave-gs-01" ve "172.17.8.102 weave-gs-02" olmak üzere iki adet Vagrant makinesi oluşturulmuştur. Makineler oluşturulduktan sonra "launch" komutu kullanılarak her iki makine üzerinde de Weave Net çalştırılmaktadır.)
$ vagrant ssh weave-gs-01
$ weave launch
$ vagrant ssh weave-gs-02
$ weave launch 172.17.8.101

Bu işlemlerden sonra artık kullanıcı konteynerlere otomatik olarak IP adresi tahsis edeceği ve DNS ile servis keşfini sağlayabileceği bir ağ oluşturmuştur. Her bir vagrant makinede kolay bir şekilde konteyner çalıştırmak için "weave env" komutu kullanarak DOCKER_HOST çevresel değişken ayarlanmaktadır:

```
$ vagrant ssh weave-gs-01
eval $(weave env)
docker run -d -h lb.weave.local fintanr/myip-scratch
docker run -d -h lb.weave.local fintanr/myip-scratch
docker run -d -h hello.weave.local fintanr/weave-gs-simple-hw
$ vagrant ssh weave-gs-02 eval $(weave env)
docker run -d -h lb.weave.local fintanr/myip-scratch
docker run -d -h hello-host2.weave.local fintanr/weave-gs-simple-hw
```

Yukarıda çalıştırılan komutlarda öncelikle makineler üzerinde basit bir Hello World uygulaması çalıştırılmış, daha sonra konteynerlere karşı lb isminde bir yük dengeleyici servis oluşturmak için DNS kullanılmaktadır. Örgü ağınızda bir konteyner çalıştırıldıktan sonra başlatılan çeşitli konteynerlere bazı isteklerde bulunulabilmektedir [3]:

```
$ vagrant ssh weave-gs-01
$ eval $(weave env)
$ C=$(docker run -d -ti fintanr/weave-gs-ubuntu-curl)
$ docker attach $C
root@ad6b7c0b1c6e:/#
root@ad6b7c0b1c6e:/# curl lb
Welcome to Weave, you probably want /myip
root@ad6b7c0b1c6e:/# curl lb/myip
10.128.0.2
root@ad6b7c0b1c6e:/# curl lb/myip
10.160.0.1
root@ad6b7c0b1c6e:/# curl hello
"message" : "Hello World",
"date" : "2019-12-03 15:59:50"
```

Aynı zamanda "./launch-simple-demo.sh" komutu ile önceki komutlar için bir komut dosyası da sağlanarak istekleri yapmak için konteyner başlatılabilmektedir.

Docker Ağ ile Birden Fazla Makine Üzerinde Konteynerleri Haberleştirme

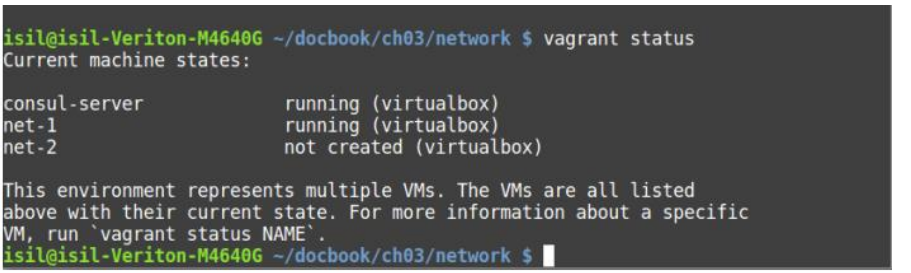
Kullanıcı Docker makineler arasında manuel olarak bir tünel inşa etmesine rağmen yeni bir Docker Ağ özelliğinin avantajını kullanmak istediğinde VXLAN katmanını kullanabilmektedir [8]. Bu katmanı kullanmak için öncelikle üç adet sanal makineyi çalıştıran bir Vagrantfile hazırlanmaktadır. Bunlardan biri bir Consul sunucu olarak davranacak, diğer ikisi ise Docker ana bilgisayarı olarak hareket edecektir. Depoyu "clone" komutu ile kopyalayıp dizini docbook/ch03/networks dizinine girdikten sonra vagrant Şekil

21'deki gibi iki adet Docker makine, 1 adet ise Consul sunucu oluşturmaktadır. "vagrant status" komutu ile oluşturulan bu makineler listelenmektedir [3].

$ git clone https://github.com/how2dock/docbook/

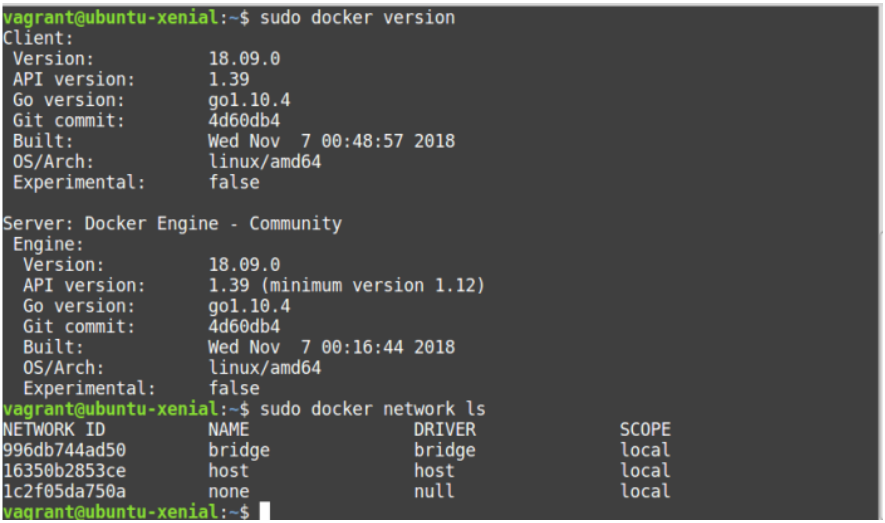
$ cd docbook/ch03/network

$ vagrant up



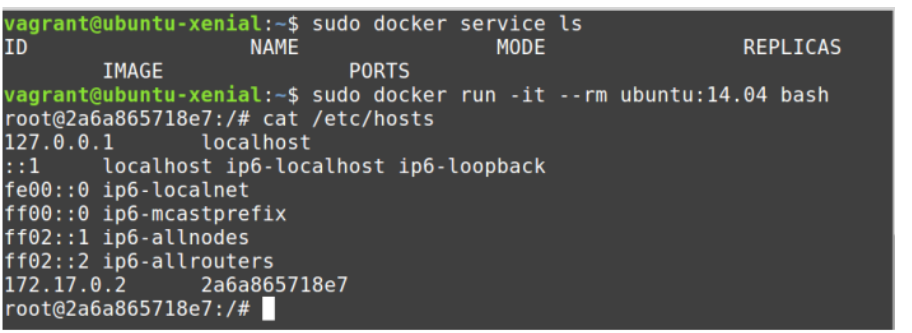
Şekil 21. vagrant ile Docker makine ve consul oluşturma

Bu makineleri kurduktan sonra artık Docker makineler "ssh" komutu ile çalıştırılabilir ve konteynerlar başlatılabilir. "vagrant ssh net-1" komutu ile net-1 makinesi çalıştırıldıktan sonra içerisine docker yüklenmekte ve "docker version" komutu ile istemci ve sunucunun docker versiyonları Şekil 22'deki gibi listelenmektedir. Benzer şekilde varsayılan ağlar listelenerek Docker Ağ'ın faaliyette olduğu kontrol edilebilmektedir.



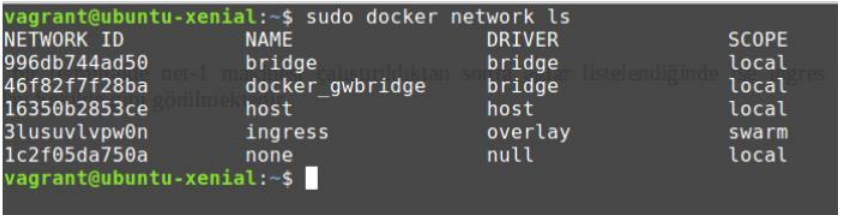
Şekil 22. Varsayılan ağların "docker network" komutu ile listelenmesi

Buraya kadar yapılan işlemlerde henüz hiçbir servis yayınlanmamıştır, bu yüzden "docker service ls" komutundan boş bir liste dönmektedir. Net-1 makinesinde bir ubuntu:14.04 konteyneri başlatılarak /etc/hosts içeriği Şekil 23'deki gibi listelenmektedir.



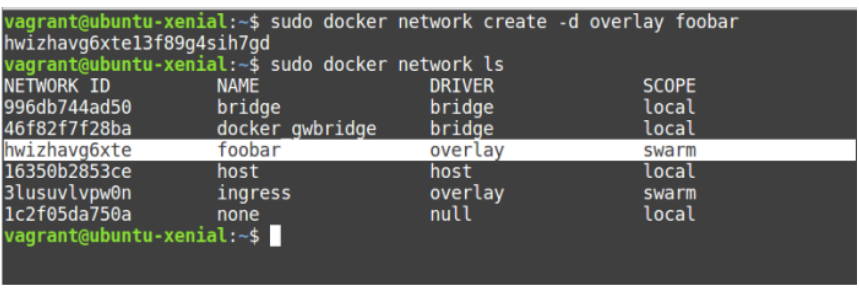
Şekil 23. net-1 makinesinde bir konteyner çalıştırma

Ayrı bir terminalde net-1 makinesi çalıştırıldıktan sonra ağlar listelendiğinde ise ingres isimli overlay Şekil 24'teki gibi görülmektedir.



Şekil 24. net-1 içerisindeki ağların listelenmesi

İngress isimli overlay ağı kullanıcının varsayılan ağıdır. Bu overlay, vagrantın kurulumu sırasında Docker daemon tarafından kurulmuştur. Ayarlanan seçenekleri görmek için /etc/ default/docker dizinine bakılabilmektedir. Farklı bir terminalde net-2 makinesindeki servis ve ağlar kontrol edildiğinde ağların aynı olduğu ve varsayılan ağın ise ingress isimli overlay olduğu görülmektedir. Fakat servis, net-1'de başlatılan konteyneri gösterecektir. Böylece sadece net-2 üzerinde çalıştırılan konteyneri görmekle kalmayıp daha önce net-1'de çalıştırılan konteynere ulaşılabilmektedir ve bu konteynerlerin her ikisine de ping atılabilmektedir. Konteynerleri haberleştirmek için önerilen çözümde varsayılan ağ kullanılmıştır fakat kullanıcı varsayılan olmayan bir overlay ağ da kullanabilmektedir. Böylece kullanıcı istediği kadar overlay oluşturarak her bir overlayde başlatılan konteynerların birbirlerinden izole olmasını sağlayacaktır. Varsayılanın dışında yeni bir ağ oluşturmak için herhangi bir Docker makine üzerinde Şekil 25'te görüldüğü gibi "docker network create -d overlay foobar" komutu yazılmaktadır.



Şekil 25. Varsayılanın dışında yeni bir ağ oluşturma

Docker Konfigürasyonu

Kullanıcı Docker Daemon'ı başlatma, durdurma ve yeniden başlatma işlemlerine ek olarak farklı bir ağ köprüsü (network bridge) ya da Docker binary'ye giden yolu değiştirmek gibi özel yollarla Docker daemon'ı yapılandırmak isteyebilir. Bu durumda docker init script kullanılarak Docker daemon yönetilebilir. Ubuntu/Debian tabanlı sistemlerin çoğunda bu script /etc/init.d/docker içerisindedir. Diğer init servisler gibi service komutu aracılığı ile yönetilmektedir. /etc/default/docker içerisinde bulunan yapılandırma dosyası aşağıdaki gibi görünmektedir [3]:

```
# Docker Upstart and SysVinit configuration file

# THIS FILE DOES NOT APPLY TO SYSTEMD

#

# Please see the documentation for "systemd drop-ins":

# https://docs.docker.com/engine/admin/systemd/

# Customize location of Docker binary (especially for development testing). #DOCKERD="/usr/local/bin/dockerd"

# Use DOCKER_OPTS to modify the daemon startup options.

#DOCKER_OPTS="–dns 8.8.8.8 -dns 8.8.4.4"

DOCKER_OPTS="-H tcp://isilVeritonM4640G:2376 -tlsverify -tlscacert=/etc/docker/ca.pem -tlscert=/etc/docker/servercert.pem -tlskey=/etc/docker/serverkey.pem" # If you need Docker to use an HTTP proxy, it can also be specified here. #export http_proxy="http://127.0.0.1:3128/" # This is also a handy place to tweak where Docker's temporary files go.

#export DOCKER_TMPDIR="/mnt/bigdrive/dockertmp"

DOCKER_OPTS="-iptables=false -ip-forward=false"

DOCKER_OPTS="-s overlay"

DOCKER_OPTS="-s overlay"
```

Örneğin uzaktan API erişimine izin vermek için bir TCP socket üzerinden daemon yapılandırmak istenirse yukarıdaki yapılandırma dosyasının içeriğine DOCKER_OPTS="- H tcp://127.0.0.1:2375" kodu eklenmektedir. Daha sonra Docker daemon sudo service docker restart ile çalıştırılmaktadır. Daha sonra TCP kullanarak erişilen bir ana bilgisayar belirtilerek Docker istemcisi Şekil 26'daki gibi kullanılabilmektedir:

```
isil@isil-Veriton-M4640G ~ $ docker -H tcp://127.0.0.1:2375 images
REPOSITORY              TAG                 IMAGE ID            CREATED
   SIZE
ubuntu                  14.04               f17b6a61de28        3 weeks ago
   188MB
busybox                 latest              59788edf1f3e        2 months ago
   1.15MB
localhost:5000/busy1    foobar              59788edf1f3e        2 months ago
   1.15MB
localhost:5000/busy1    latest              59788edf1f3e        2 months ago
   1.15MB
localhost:5000/busy     latest              59788edf1f3e        2 months ago
   1.15MB
isil@isil-Veriton-M4640G ~ $
```

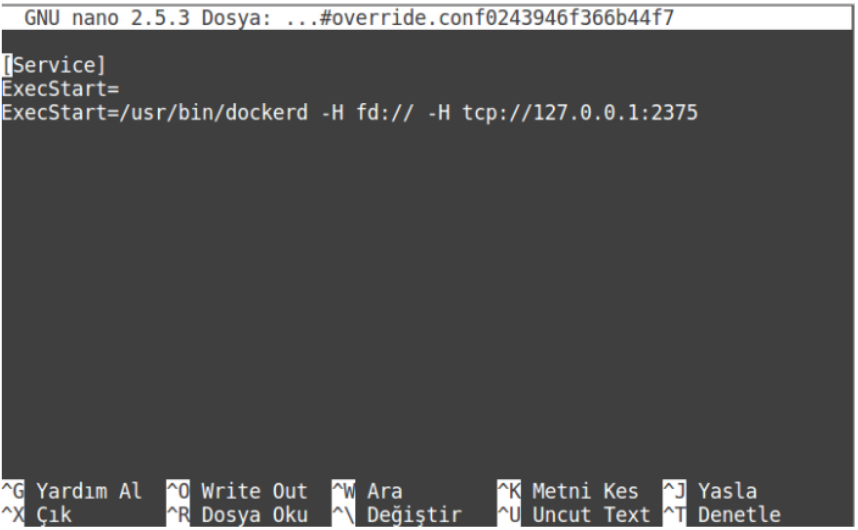
Şekil 26. TCP kullanılarak belirtilen ana bilgisayara erişmek

TCP üzerinden Docker daemon dinleme ile API çağrıları yapmak için Şekil 27'deki gibi curl komutu kullanılıp gelen cevap görülebilmektedir. Bu da Docker Remote API'yi öğrenmenin iyi bir yolu olarak belirtilmektedir.

```
isil@isil-Veriton-M4640G ~ $ curl -s http://127.0.0.1:2375/images/json | python -m json.to
ol
[
    {
        "Containers": -1,
        "Created": 1542662598,
        "Id": "sha256:f17b6a61de28594fb3ec53b1cca7164fba66357d1635b414eeed4d586744342e",
        "Labels": null,
        "ParentId": "",
        "RepoDigests": [
            "ubuntu@sha256:f961d3d101e66017fc6f0a63ecc0ff15d3e7b53b6a0ac500cd1619ded4771bd
6"
        ],
        "RepoTags": [
            "ubuntu:14.04"
        ],
        "SharedSize": -1,
        "Size": 188075713,
        "VirtualSize": 188075713
    },
    {
        "Containers": -1,
        "Created": 1538500774,
        "Id": "sha256:59788edf1f3e78cd0ebe6ce1446e9d10788225db3dedcfd1a59f764bad2b2690",
        "Labels": null,
        "ParentId": "",
```

Şekil 27. Docker Remote API kullanımı

Systemd Unit Dosyası ile Uzaktan Erişimi Yapılandırma: sudo systemctl edit docker.service komutu çalıştırıldıktan sonra docker.service dosyası içerisinde Şekil 28'deki gibi kullanıcının istediği kurallar koyularak tcp ile erişilen ana bilgisayar belirlenmektedir [13].



Şekil 28. TCP ile erişilen ana bilgisayarın docker servis dosyasına yazılması

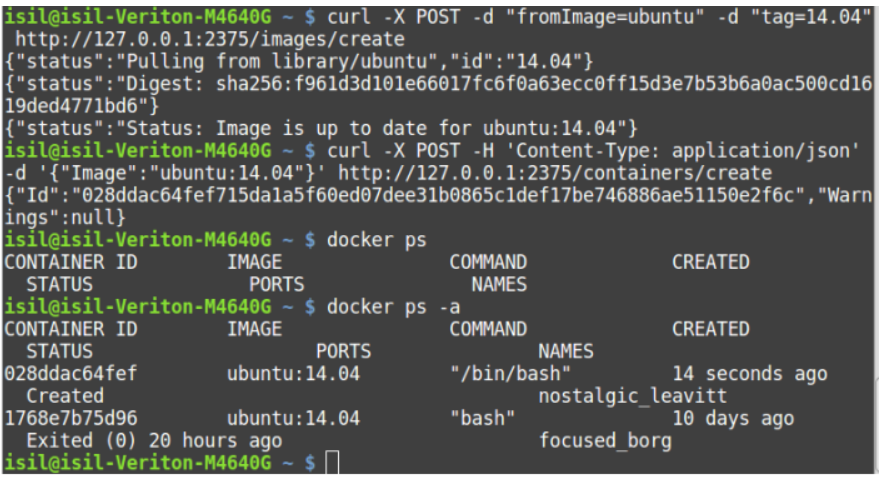
Daha sonra systemctl daemon-reload Docker daemon tekrar yüklenerek Docker servisi tekrar başlatılmaktadır. Son olarak Docker'ın yapılandırılmış bağlantı noktasını dinlediğini onaylamak için netstat'ın çıktısı incelenip değişikliğin yapıldığı Şekil 29'daki gibi görülmektedir [13].

Şekil 29. Docker serviste yapılan değişikliğin "netstat" komutu ile görülmesi

Docker Görevlerini Otomatikleştirmek için Docker Remote Kullanımı

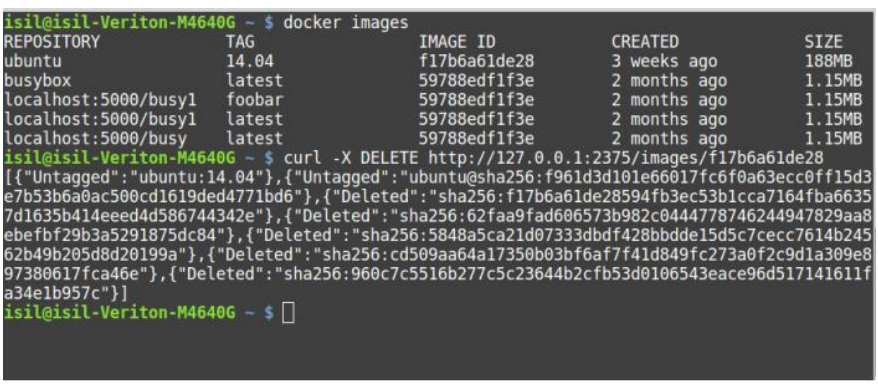
Docker daemon'a uzaktan erişilebildikten sonra program yazmak için Docker remote API kullanılarak Docker görevleri

otomatikleştirilebilmektedir [14]. Docker remote API'ye örnek olarak Docker Hub açık kayıt defteri üzerinden Ubuntu 14.04 imajı indirilerek bu imajdan bir konteyner Şekil 30'daki gibi oluşturulmakta ve başlatılmaktadır.



Şekil 30. Docker Hub'dan ubuntu imajının indirilmesi

"curl -X DELETE http://127.0.0.1:2375/containers/028ddac64fef" komutu ile oluşturulan konteyner silinmektedir. Daha sonra imajlar tekrar listelenerek ubuntu deposunun imaj ID'si bulunup Şekil 31'deki gibi silinmektedir.



Şekil 31. Oluşturulan imajın silinmesi

Sonuç

Konteyner teknolojileri standart bir uygulama programlama arayüzü (API) sağlayarak konteynerle birlikte paketlenen uygulamaların ve konteyner yaşam döngüsünün yönetimi için sunulmaktadır. Bu API heterojen dağıtıma homojen bir arayüz sunarak bulut sunucularda altyapının verimli kullanılması, uygulamaların güvenliğini sağlama, konteynerler arasında veri paylaşımı gibi birçok görevi yerine getirmektedir. Bu çalışmada Docker konteyner teknolojisi ile konteyner görüntüleri oluşturma ve paylaşma, tek veya birçok ana bilgisayarda ağ konteynerleri ve docker yapılandırması gibi gelişmiş konulara yer verilerek bir konteyner teknolojisinin ileri düzey kullanımı hakkında gelecek çalışmalara yol göstermek hedeflenmiştir. Dağıtık sistemlerde servislerin paketlenmesi, yapılandırılması ve birleştirilmesi, yapılandırma yönetimi ve orkestrasyonda yapılan birçok çalışmaya rağmen konteyner teknolojilerinin kullanım zorlukları devam etmektedir. Dağıtılmış bir uygulamayı geniş ölçekte ve hataya dayanıklı bir şekilde dağıtmak ve çalıştırmak zor bir problem olarak literatürde yer almaktadır.

KAYNAKÇA

[1] *Use containers to Build, Share and Run your applications*. 2023 [cited 2023 June, 2]; Available from: https://www.docker.com/resources/what-container/.

[2] *How to install Docker and run Docker containers on Linux Mint* 18/18.1. [cited 2023 May, 20]; Available from: https://linuxbsdos.com/2016/12/13/how-to-install-docker-and-run-docker-containers-on-linux-mint-1818-1/.

[3] Goasguen, S., *Docker Cookbook: Solutions and Examples for Building Distributed Applications*. 2015: " O'Reilly Media, Inc.".

[4] Nüst, D. and M.J.J.o.O.S.S. Hinz, containerit: *Generating Dockerfiles for reproducible research with R*. 2019. 4(40): p. 1603.

[5] Devisetty, U.K., et al., *Bringing your tools to CyVerse discovery environment using Docker*. 2016. 5.

[6] Jangla, K. and K.J.A.D.V.U.D.D.A.M. Jangla, *Docker compose*. 2018: p. 77-98.

[7] Scott, S., *Docker Hub and Image Repositories, in Oracle on Docker: Running Oracle Databases in Linux Containers*. 2023, Springer. p. 357-382.

[8] Suo, K., et al. *An analysis and empirical study of container networks*. in IEEE INFOCOM 2018-IEEE Conference on Computer Communications. 2018. IEEE.

[9] Mouat, A., Using Docker: Developing and deploying software with containers. 2015: " O'Reilly Media, Inc.".

[10] Fink, J.J.C.L.J., Docker: a software as a service, operating system-level virtualization framework. 2014(25).

[11] Smith, R., Docker orchestration. 2017: Packt Publishing Ltd.

[12] Kane, S.P. and K. Matthias, Docker: Up & Running. 2023: " O'Reilly Media, Inc.".

[13] Manage Docker as a non-root user. [cited 2023 April, 30]; Available from: https://docs.docker.com/engine/install/linux-postinstall/#upstart.
[14] Sarmiento, E.M. and E.M.J.T.S.S.D.s.G.t.D.C.A.D.w.I.L.-i. Sarmiento, Managing and Administering Containers. 2020: p. 99-131.


100
TÜRKİYE CUMHURİYETİ'NİN YÜZÜNCÜ YILI
BİDGE