BİLGİSAYAR MÜHENDİSLİĞİ ALANINDA ULUSLARARASI ÇALIŞMALAR

EDİTÖR

PROF. DR. SELAHATTİN BARDAK

Mart 2024

SERÜVEN YAYINEVİ

Genel Yayın Yönetmeni / Editor in Chief • C. Cansın Selin Temana

Kapak & İç Tasarım / Cover & Interior Design • Serüven Yayınevi

Birinci Basım / First Edition • © Mart 2024

ISBN • 978-625-6319-04-2

Serüven Yayınevi / Serüven Publishing
Türkiye Adres / Turkey Address: Kızılay Mah. Fevzi Çakmak 1. Sokak
Ümit Apt No: 22/A Çankaya/ANKARA
Telefon / Phone: 05437675765
web: www.seruvenyayinevi.com
e-mail: seruvenyayinevi@gmail.com

Baskı & Cilt / Printing & Volume
Sertifika / Certificate No: 47083

BİLGİSAYAR MÜHENDİSLİĞİ ALANINDA ULUSLARARASI ÇALIŞMALAR

Mart 2024

Editör

Prof. Dr. Selahattin BARDAK

İÇİNDEKİLER

BÖLÜM 1

AĞ GÜVENLİĞİNDE GÜVENLİK DUVARININ ROLÜ

Furkan ATICI[1]
Eren POLAT[2]
Aybars YÜCEL[3]

1 Mehmet Akif Ersoy Üniversitesi, Gölhisar Uygulamalı Bilimler Yüksekokulu, Bilişim Sistemleri ve Teknolojileri, Burdur, Türkiye, https://orcid.org/0009-0004-5461-9842
2 Mehmet Akif Ersoy Üniversitesi, Gölhisar Uygulamalı Bilimler Yüksekokulu, Bilişim Sistemleri ve Teknolojileri, Burdur, Türkiye, https://orcid.org/0009-0009-6221-338X
3 Mehmet Akif Ersoy Üniversitesi, Gölhisar Uygulamalı Bilimler Yüksekokulu, Bilişim Sistemleri ve Teknolojileri, Burdur, Türkiye, https://orcid.org/0009-0008-9554-7194

GİRİŞ

Teknoloji, gerek kişisel hayatımızda gerek iş dünyasında temel bir ihtiyaç haline gelmiştir. En başta bilgisayarların ve telefonların yaygınlaşması ve bu cihazların birbiri ile haberleşmesi ihtiyacından doğan ağ kavramı, haberleşme süreçlerini hızlandırmış olsa da güvenlik problemlerini de beraberinde getirmiştir. Bu güvenlik problemlerinden dolayı "Ağ Güvenliği" kavramı ortaya çıkmıştır. Ağ güvenliği bir şirketin veya bireyin bilgilerini, sistemlerini ve ağlarını, veri kaybı, yetkisiz ve izinsiz erişim gibi kötü niyetli saldırganlara karşı koruma sürecidir. Bu bağlamda ağ güvenliğinde en önemli unsurlardan biri olan güvenlik duvarları, güvenliği sağlamak adına önemli bir rol oynamaktadır.

Güvenlik duvarları temel olarak, ağ trafiğini denetler, iç ve dış ağlar arasında iletişimi filtreler ve izler. Bu iletişimde herhangi olağandışı bir durum tespit etmesi halinde müdahale edebilir. Bu süreçlerde güvenlik duvarları farklı türlerde ve işlevlerde olabilir. Türü ne olursa olsun güvenlik duvarının temel amacı, iç ağ ile dış ağ arasında bir bariyer görevi görüp, dışarıdan gelebilecek tehditlere karşı iç ağı korumaktır. Bu çalışmanın devamında, güvenlik duvarlarının çeşitleri, işlevleri, yapılandırılması, yönetilmesi, saldırı türleri ve bu saldırılara karşı güvenlik duvarında ne gibi önlemler alınabileceği, güncel tehdit istihbaratlarının kullanımı ve güvenlik duvarlarını daha etkili kullanmak için önemli konular ele alınacaktır.

Bu çalışmada, ağ güvenliğinin ne olduğu ve önemi hakkındaki bilgilendirmelerin yanı sıra, güvenlik duvarlarının günümüz dijital dünyasında bu kompleks yapının içindeki rolü ve önemine değinilecektir. Ağ güvenliği, bir organizasyonun bilişim sistemlerini koruma sürecine odaklanır. Bu koruma, kullanıcıları siber tehditlerden ve diğer kötü amaçlı eylemlerden kaynaklanan riskleri önlemeyi hedefler. İnternetin artık hayatımızın her yerinde olduğu günümüzde iş sürekliliği, müşteri güvenliği ve memnuniyeti gibi pek çok konuda ağ güvenliğinin önemi kilit bir önem taşır.

Güvenlik duvarları, ağ güvenliği stratejilerinin vazgeçilmez unsurlarından biridir. Bu çalışmanın ikinci bölümünde güvenlik duvarının temel tanımından başlayarak, güvenlik duvarı türleri, işlevleri ve ağ güvenliğinde güvenlik duvarından maksimum verimi alabilmek için güvenlik duvarı yapılandırılması ve yönetiminde dikkat edilmesi gereken kritik hususlara değinilecektir.

Çalışmanın üçüncü bölümünde, ağ güvenliğinde güvenlik duvarının rolü ve önemi daha detaylı incelenecektir. Popüler saldırı türleri ve bu saldırılara karşı güvenlik duvarının ürettiği çözümler ve güvenlik duvarı olguları anlatılacaktır. Ayrıca, güncel tehdit istihbaratının etkili bir şekilde kullanımının, güvenlik duvarındaki kullanımı ve güvenlik duvarında başarılı işlevine değinilecektir.

Bu çalışma, ağ güvenliği ve güvenlik duvarlarının karmaşıklığına ışık tutarak organizasyonların dijital varlıklarını koruma ve güvenlik stratejilerine katkıda bulunmayı amaçlanmıştır.

1. AĞ GÜVENLİĞİ NEDİR ?

Ağ güvenliği, cihaz, veri, sistemler ve kaynaklar üzerinde yetkisiz erişimleri, veri sızıntılarını ve diğer birçok tehditleri engellemeye çalışan geniş kapsamlı bir kurallar bütünüdür[1]. Yazılım ve donanım teknolojilerinden faydalanarak bilgisayar ağlarının ve verilerin bütünlük, gizlilik ve erişilebilirliğini korumak için tasarlanmış bir konfigürasyonlar dizisidir. Boyutu ve altyapısı ne olursa olsun, her kurum günümüzde artan siber tehditlerden korunabilmek için ağ güvenliğine gerek duymaktadır.

Günümüzde ki ağ mimarisi karmaşık ve sürekli değişmekte olan bir tehdit ortamıyla karşı karşıyadır; saldırganlar güvenlik zafiyetleri bulmaya ve bu güvenlik zafiyetlerine saldırmaya çalışmaktadır. Bu güvenlik açıkları, cihazlar, veriler, uygulamalar ve sistemler dahil olmak üzere pek çok farklı alanda olabilir.

Ağ güvenliği, bilgisayar ağlarını siber suçlulara karşı korumak ve güvende tutmak için çeşitli teknikler, protokoller ve yöntemler kullanılmaktadır. Bu yöntemler; kimlik doğrulama, şifreleme, güvenlik duvarları, güvenlik yazılımı ve düzenli güvenlik kontrolleri gibi çeşitli yöntemleri kapsamaktadır.

1.1 AĞ GÜVENLİĞİNİN ÖNEMİ

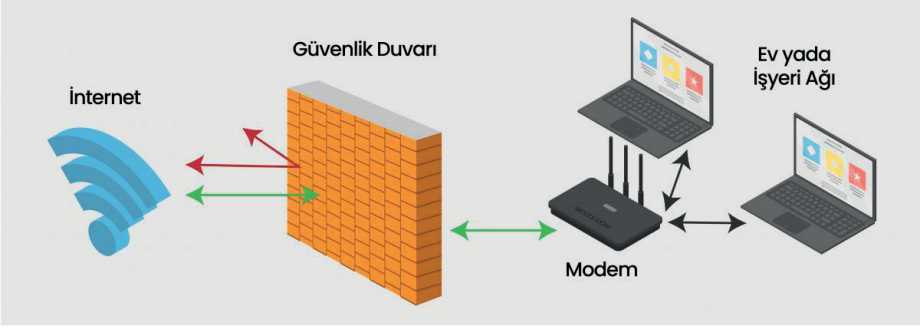
Bilgisayar ağları, bilgi alışverişinin hızlı bir şekilde gerçekleştiği ve bilgiye kolay ulaşım sağlayan bir köprü niteliğindedir. Bu ortamı oluşturan ve önemli verileri içerisinde barındıran ağ güvenliğinin önemi de her geçen gün artmaktadır. Dev bir bilgisayar ağı ve bunun sonucu oluşan internet herkes için vazgeçilmez bir bilgi havuzudur. Bütün mesleklerde bilgisayar kullanılması, internete ulaşmanın çok kolay ve ucuz bir hâle gelmesi, istisnasız her bilgisayarın bir bilgisayar ağına bağlı olması anlamına gelmektedir. Bilgisayar ağlarının bu denli önemli hâle gelmesi ile birlikte ağ güvenliğini sağlama konusunda bilgi sahibi olma ihtiyacı da artmıştır. Bilgisayar sistemlerine ve ağlarına yönelik saldırılar ciddi miktarda kaynak, zaman, hizmet sürekliliği, itibar ve değerli bilgi kaybına neden olabilir.[2] Özellikle kişisel bilgilere, finansal verilere ve ticari sırlara yetkisiz erişilmesi, saklanması ve paylaşılması ağ güvenliği önlemleri ihmal edildiğinde büyük riskler taşır. Ağ güvenliğinin sağlanması bu verilerin korunması konusunda önemli bir rol almaktadır.

2. GÜVENLİK DUVARI(FİREWALL)

Güvenlik duvarı, ağ trafiğini belirli filtrelerden geçiren ve gelen ağ trafiği içindeki zararlı eylemleri durdurmayı amaçlayan bir araçtır[3]. Güvenlik duvarı, belirlenmiş kurallara uymayan trafiği engelleyen bir mekanizmadır.

Bilgisayarlar, bir ağa bağlandıklarında birçok virüs ve saldırı riskiyle karşılaşırlar. Güvenlik duvarları, bu tür tehditleri engellemek için devreye girer ve iç ağ ile dış ağ arasındaki gelen ve giden ağ trafiğini kontrol eder[4]. Bu şekilde yetkisiz erişimler ve dış ağdan gelebilecek potansiyel siber tehditler minimuma indirilir.

Güvenlik duvarlarının temel amacı, zararlı olmayan trafiğe izin vermek ve tehlikeli trafiği içeri almamak, dışarıda tutmaktır. Güvenlik duvarları, önceden belirlenmiş kurallar ve politikalara göre hareket ederek ağı daha iyi korur. Bu kurallar doğrultusunda trafiği ilgili yere yönlendirir; uygun davranışlara izin verirken, tehdit olarak algılanan hareketleri engeller. Bu nedenle güvenlik duvarları, bir kuruluşun güvenlik savunmasında kritik bir rol oynar. Büyük ve çok kullanıcılı işletmeler genellikle güvenlik duvarı cihazlarını kullanmak durumundadır.



Şekil 1: *Güvenlik duvarı modellemesi[1].*

2.1 GÜVENLİK DUVARI TÜRLERİ

- **Proxy Firewall**

Proxy Güvenlik Duvarı, ağ kaynaklarını korumak için tasarlanmış olan bir güvenlik duvarı türüdür ve uygulama katmanındaki iletileri filtreleyerek en güvenli ve kusursuz korumayı sağlar[3]. Proxy güvenlik duvarı, bir ağın desteklediği uygulamalara sınırlar koyarak güvenliği artırır; ancak bu sınırlamaların uygulanması, sistem işlevselliğini etkileyebilir.

Proxy güvenlik duvarı, diğer güvenlik duvarlarından farklı bir çalışma prensibine sahiptir. İşletmelerin uygulama protokollerinin tehdit seviyelerini öngörmelerine, saldırıları algılamalarına, geçerlilik denetimi yapmalarına ve hata algılama işlevlerini devreye sokmalarına yardımcı olan özel bir noktadır.

- **Birleşik Tehdit Yönetimi Güvenlik Duvarı (Unified Threat Management (UTM) Firewall)**

Birleşik Tehdit Yönetimi, virüs koruması, kötü amaçlı yazılımdan koruma, içerik filtreleme gibi çeşitli güvenlik önlemlerini içeren kapsamlı bir çözümün parçasıdır[5]. Bu güvenlik duvarını tercih etmenin temel avantajı,

şirketlerin tehditleri yönetmek için sadece tek bir çözümle uğraşmalarını sağlayarak maliyet ve bakım açısından tasarruf etmeleridir.

- **Durum Denetimi Güvenlik Duvarı(Stateful Inspection Firewall)**

Durum bilgisi olan bir güvenlik duvarı, aktif ağ bağlantılarının durumunu sürekli izlerken aynı zamanda gelen trafiği analiz eden bir tür güvenlik duvarıdır, potansiyel riskleri ve tehditleri tespit etmeye odaklanır. Durum bilgisi olan güvenlik duvarları, OSI (Açık Sistemler Arası Bağlantı) modelinin 3. ve 4. katmanlarında konumlanmışlardır.

Durum bilgisi olan paket denetimi, güvenlik duvarından geçebilecek paketleri ve geçemeyecek paketleri belirlemek için durum bilgisi olan güvenlik duvarları tarafından kullanılır.

- **Yeni Nesil Güvenlik Duvarı(Next-generation Firewall (NGFW))**

Yeni nesil güvenlik duvarları (NGFW), geleneksel güvenlik duvarlarının işlevlerini birleştirerek oluşturulan bir güvenlik duvarı türüdür[5]. NGFW'ler, izinsiz giriş önleme sistemleri ve derin paket denetimi gibi özelliklere sahip üçüncü nesil güvenlik duvarlarıdır.

NGFW'ler, diğer güvenlik duvarlarına kıyasla daha detaylı bir denetim mekanizması kullanma eğilimindedir; paket yüklerini değerlendirir ve kötü amaçlı yazılım gibi zararlı unsurlar için imzaları eşleştirir. NGFW'ler, yöneticilere daha iyi farkındalık sağlar ve bireysel uygulamalar üzerinde daha derinlemesine inceleme ve kontrol yetenekleri sunar.

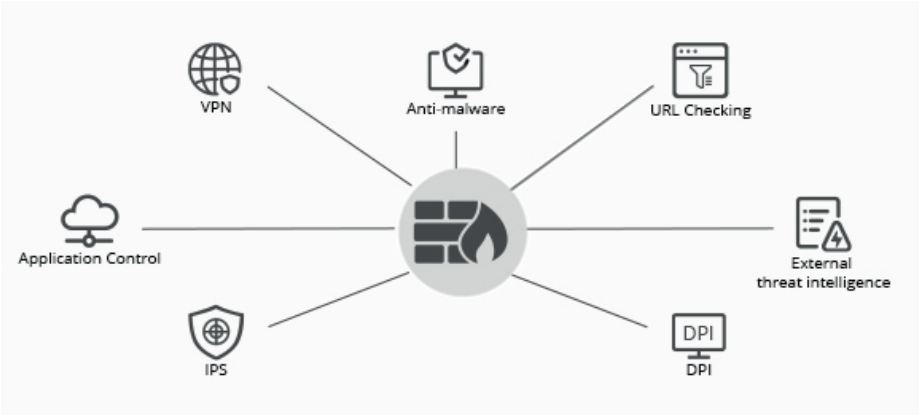
Tehdit odaklı NGFW'ler, bilinen veya bilinmeyen tüm tehditleri ve riskleri belirlemek, analiz etmek, engellemek, izlemek ve bu tehditlere karşı direnme yeteneği için tasarlanmıştır.

- **Sanal Güvenlik Duvarı(Virtual Firewall)**

Sanal güvenlik duvarı, aynı zamanda bulut güvenlik duvarı olarak da adlandırılır. Bu tür bir güvenlik duvarı, donanım güvenlik duvarlarının dağıtımının zor olduğu veya genel ve özel bulut ortamlarında SDN (Software Defined Networking) gibi seçeneklerin bulunmadığı durumlar için tasarlanmıştır. Sanal güvenlik duvarları aynı zamanda NGFW'lerin sanallaştırılmış örnekleri olarak kurulabilir.

2.2 Güvenlik Duvarlarının İşlevleri

Artık günümüzde internet herkes için elzem bir ihtiyaç olmuştur. Ancak İnternet kullanımının bu kadar yaygınlaşması, bilişim unsurlarının güvenliği konusunda daha fazla hassasiyet gösterilmesi yükümlülüğünü de kendisiyle birlikte getirmiştir. Güvenlik duvarları, bu bağlamda bilişim sistemlerinin önemli bir savunma mekanizması olarak öne çıkmaktadır.



Şekil 2: *Güvenlik duvarı işlevleri modellemesi[2].*

Güvenlik duvarlarının temel işlevleri özelinde inceleme:

- **Erişim Kontrolü:**

Güvenlik duvarı erişim kontrolü, ağ üzerindeki giriş çıkış trafiğini denetler. İzinsiz ve zararlı kullanıcıların erişimlerini olanaksız kılar. Bu durum sistemler üzerinde sadece yetkili kullanıcıların erişim sağlamasına olanak tanır ve bilgi sistemlerini korumaya yardımcı olur.

- **Paket Filtreleme:**

Güvenlik duvarları, ağ üzerindeki veri paketlerini inceler ve belirlenmiş kurallar bütününü baz alır. İnceleme sonucunda paketlere izin verme veya engelleme aksiyonu alır. Alınan aksiyon nihayetinde zararlı içeriklerin engellenmesi ve saldırı girişimleri kontrol altına alınmış olur.

- **Port ve Protokol Kontrolü:**

Bilgisayarlar, aralarındaki iletişimi sağlamak amacıyla belirli bağlantı noktalarını (portlar) ve belirli protokolleri kullanır. Güvenlik duvarları bu noktada hangi portlar üzerinde trafiğin sağlanacağını denetleyip belirleyerek potansiyel güvenlik zaaflarını en aza indirir.

- **Proxy Hizmetleri:**

Proxy hizmetleri güvenlik duvarı içerisinde önemli bir unsurdur. İstemcilerin doğrudan dış ağlardaki kaynaklara erişmesini engelleyerek kullanıcıları potansiyel tehditlere karşı korumuş olur[6]. Dış kaynaklar ve kullanıcılar arasındaki etkileşimi yönetmesi, kuruluşların güvenlik politikalarını uygulamaları hususunda da katkıda bulunur.

2.3 Ağ Güvenliği:

Günümüzde ağlar giderek komplike bir hal almakta ve bu durum ağları çeşitli siber tehditlere karşı savunmasız hale getirmektedir. Bu noktada güvenlik duvarlarının ağ trafiğini denetleme, izleme ve gerektiğinde müdahale etme refleksini göstermesi gibi yetenekleri ağ düzeyinde koruma sağlar.

- **Günlük (Log) tutma ve izleme:**

Güvenlik duvarlarını ağdaki trafiği sürekli izler ve bu izleme verilerini günlük dosyalarına kaydedip ağ yöneticisine ağdaki güvenlik olaylarını anlamak ve analiz etmesi konusunda yardımcı olur[7].

- **Virtual Private Network (VPN) Kontrolü:**

VPN'ler, organizasyonların uzaktan erişim sağlamalarına ve bu erişimlerin güvenli halde gerçekleşmesinde rol oynar. Güvenlik duvarları, VPN bağlantılarına erişim sağlayacak istemcilerin kimlik doğrulama ve erişilebilirliğini kontrol etme işlemlerinin sağlanması yanı sıra trafiği de izleyerek güvenlik tehditlerinin tespiti ve müdahalesine de olanak sağlar.

1.4 **Güvenlik Politikalarını Uygulama:**

Çoğu organizasyon için güvenlik politikalarının uygulanması ve dijital varlıklarını güvenli bir şekilde yönetmek hayati bir önem taşır. Bu noktada güvenlik duvarları, belirlenmiş güvenlik politikalarını uygulayarak organizasyonların güvenlik standartlarını korumasına yardımcı olur.

- **Güvenlik Duvarının Yapılandırılması ve Yönetiminde Dikkat Edilmesi Gereken Temel Hususlar**

Güvenlik duvarları, bilişim sistemlerini dış tehditlere karşı koruma noktasında kilit bir rol oynar. Güvenlik duvarlarından maksimum verimi alabilmek için doğru bir şekilde yapılandırmak ve düzenli yönetmek gerekir. Güvenlik duvarlarının başarılı yapılandırılması ve yönetilmesi aşağıdaki belirtilen temel adımlardan oluşur.

- **İhtiyaç Analizi:**

Güvenlik duvarı yapılandırılması için ilk göz önünde bulundurulması gereken şey organizasyonun ihtiyaçlarını belirlemek ve bu doğrultuda planlamadır. Hangi servislere erişim sağlanacağı ve/veya sağlanmayacağı, hangi protokoller ve portların kullanacağı gibi temel unsurlar belirlenmelidir.

- **Güvenlik Politikalarının Belirlenmesi:**

Güvenlik duvarı politikaları belirlenirken organizasyonun güvenlik standartlarına uyumuna dikkat edilmelidir. Bu politikalar, kullanıcı erişimini, ağ trafiğini ve güvenlik protokollerini içermenin yanı sıra bu sistemlerin etkin ve güvenli bir şekilde çalışabilmesi için temel bir adımdır.

2.5 Paket Filtreleme Kurallarının Oluşturulması:

Paket filtreleme kurallarını belirli bir güvenlik politikasına uygun olmalıdır. Kurallar akan trafiği kontrol etmeli ve bunun için gerekli yetenekler tanımlanmalıdır. İzin verme ve reddetme kriterleri için eksiksiz ve net kurallar oluşturulmaya özen gösterilmelidir[7]. Kuralların genel ilkeleri belirlendikten sonra (varsa) istisnalar da tanımlanmalıdır. Bu, ağda esnek bir yapı oluşturulmasına olanak sağlar. Dikkatli ve özenli hazırlanmış kurallar ağ güvenliğini ve performans optimizasyonunu arttırmak için kilit bir rol oynar.

- **Port ve Protokol Yönetimi:**

Spesifik ihtiyaçlar belirlenip ve sınıflandırıldıktan sonra buna paralel olarak hangi port ve protokollerin kullanılacağı netleştirilmelidir. Güvenlik duvarında kullanılmayan portlar kapatılmalıdır. Güvenlik açıklarını minimuma indirmek için bu portlar periyodik olarak sürekli izlenip kontrol altında tutulmalıdır. HTTPS(443) gibi şifreli protokoller öncelik olarak tercih edilmeli ve zararlı olma potansiyeli taşıyan protokollerin engellenmesi gereklidir.

- **Güncellemeleri İzleme:**

Değişen ve gelişen teknoloji ve buna paralel olarak sürekli evrim geçiren siber tehditler göz önünde bulundurulmalıdır. Bu noktada, güvenlik duvarını güncel tutmak çok kritiktir. Güvenlik duvarı üreticisinin yayınladığı güvenlik yamaları, güncellemeler, güvenilir kurum veya kuruluşlar tarafından yayınlanan güncel uyarılar güvenlik duvarına eklenmelidir.

- **Yedekleme ve Kurtarma Planları:**

Güvenlik duvarının yapılandırılması sürecinde yedekleme ve kurtarma planları düşünülmelidir. Sistemde olası aksaklık veya saldırı durumunda sistemin eski haline getirilebilmesi için yedekleme ve kurtarma planlarının uygulanması önemlidir.

- **Kullanıcı Eğitimi:**

Güvenlik duvarı etkinliği kullanıcıların bilinç düzeyi ve yetkinliği ile doğru orantılıdır. Bu bağlamda kullanıcıların, güvenlik duvarının ne işe yaradığı ve güvenlik duvarının hangi durumlarda müdahale etme kabiliyetinin olduğunu anlamaları için eğitilmeleri önemlidir. Bilinçli kullanıcılar, sistemin güvenlik açıklarının azaltılması noktasında değerlidir.

3. Ağ Güvenliğinde Güvenlik Duvarının Rolü

Gelişen teknoloji ile birlikte dijital dünya istikrarlı bir şekilde genişleyen ve değişen siber tehditler ile karşı karşıyadır. İnternet kullanımının yaygınlaşması ve bilgisayar ağlarının genişlemesiyle veri güvenliği çok daha önemli hale gelmiştir. Bu noktada verileri korumak ve yetkisiz girişleri engellemek

için yapılan çalışmalarda güvenlik duvarları çok kritik bir rol oynamaktadır.

Güvenlik duvarı, yerel ağdaki internet trafiğini, gelen ve giden verileri denetleyen ve filtreleyen bir güvenlik önlemidir[8]. İnternet ile yerel ağ arasında bir kalkan gibi yerel ağı dış internetteki tehditlere karşı korur. Temel işlevi zararlı içerikleri ve istenmeyen erişimleri engellemektir.

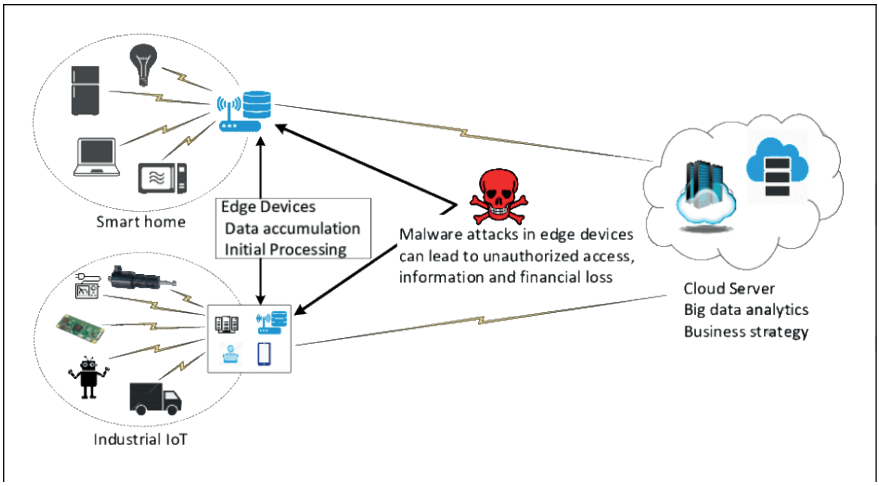
Erişim kontrolü ve filtreleme özellikleri sayesinde, ağa yetkisiz kişilerin girişini engeller ve dış ağdan gelebilecek zararlı yazılımlara karşı iç ağdaki bilgisayarları ve sistemleri korur. İçerik filtreleme ve denetleme özellikleri sayesinde ağdaki kullanıcıların hangi içeriklere erişebileceğini belirleyerek, ağdaki verimliliği arttırılabilir.

3.1. Saldırı Türleri ve Güvenlik Duvarlarındaki Çözümleri

Bilgisayarlar ve internet günümüzde birçok işimizi kolaylaştırmaktadır. Ancak bu teknolojiler ile birlikte gelen birçok tehdit bulunmaktadır. Bilgisayar korsanları ağa zararlı veya zararsız saldırılar yapabilirler. Zararlı saldırganların amacı, organizasyon içi bilgileri veya kullanıcı verilerini değiştirmek, silmek veya yönlendirmek olabilir. Ağ trafiğini manipüle edebilirler veya ağ kaynaklarını kullanılmaz hale getirebilirler[9]. Zararsız saldırganlar ise iç ağa sızıp verilere veya herhangi bir sisteme zarar vermeden çıkarlar. Tek amaçları kendilerini tatmin etmektir. Güvenlik duvarının asıl amaçlarından biri de bu tarz saldırıları engellemek veya önüne geçmektir. Aşağıda yaygın olarak yapılan saldırılar verilmiştir.

- **Malware (Malicious Software):**

Malware, Türkçe karşılığı kötü amaçlı yazılımlardır. Bu kötü amaçlı yazılımların amacı, kullanıcının izni dahilinde olmadan sistemlere sızarak çeşitli zararlar vermektir. Kötü amaçlı yazılımların birçok farklı türü vardır. Bunlara bilgisayar virüsleri, solucanlar, trojanlar, fidye yazılımları ve adware (advertising-supported software) örnek verilebilmektedir[10]. Malware, kullanıcılar ve organizasyonlar için ciddi tehditler oluşturabilir. Güvenli web sitelerinden indirme yaparak, güncel antivirüs programları kullanarak, şüpheli e-posta veya bağlantılardan uzak durarak koruma sağlamak önemlidir.



Şekil 3: *Malware saldırı modellemesi[3].*

- **Malware Saldırısına Karşı Güvenlik Duvarında Çözümler:**

Trafiği Filtreleme ve İzleme:

Güvenlik duvarlarının ağ trafiğinden geçen paketleri inceleme özelliği vardır. Dış ağdan gelen paketleri inceler ve zararlı bir paket tespit ederse bunu engelleyebilir. Filtreleme yaparak zararlı bir paketi güvenlik duvarına ulaşamadan engellemek mümkündür.

Güncel Veritabanları ve İmza Tabanlı Tespit:

Güvenlik duvarları güncel veritabanları sayesinde yaygın olan ve bilinen Malware türlerini tanır. Veritabanında da bulunan bu yazılımların ağa girişini engeller. Bu imza tabanlı sistem, belirli Malware'leri tespit etmek ve engellemek için kullanılır.

Davranışsal Analiz:

Güvenlik duvarları kullanılan uygulamaların veya yazılımların normal davranışlarını analiz eder. Herhangi bir saldırı sonrası şüpheli yazılım tespit edilir, uygulama ve yazılımların davranışlarında bir anormallik tespit edilirse, bu zararlı yazılımlar Sandbox ortamında çalıştırılır. Sandbox ortamı, cihaza zarar vermeden sanal bir ortam oluşturarak zararlı yazılımı orada çalıştırır ve potansiyel bir tehdit var ise bunları belirleyip izole eder.

Güncelleme ve Zafiyet İzleme:

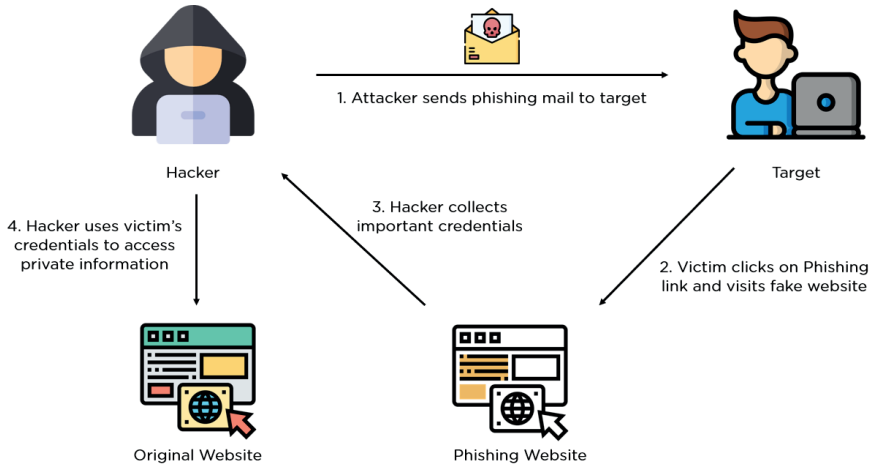
Güvenlik duvarları, sistemdeki güvenlik açıklarını izleme özelliğine sahiptir. Bir güvenlik açığı varsa ve bu açığı kullanan zararlı bir yazılım tespit ederse zararlı yazılımlara karşı güncellemeleri uygulayabilir[11].

Geçmiş Kayıtları Analiz:

Güvenlik duvarları sürekli akan ağ trafiğini kayıt altına alır ve bu kayıt verilerini analiz eder. Bu kayıtlar, geçmişte gerçekleşmiş saldırıları analiz etmek ve gelecekte yaşanabilecek saldırılar için bir strateji oluşturmak için önem arz etmektedir.

- **Phishing (Bilgi Hırsızlığı):**

Türkçe'de oltalama olarak da adlandırılan bu saldırı tipi, sahte e-postalar, bağlantılar veya mesajlar aracılığı ile gerçekleştirilebilir. Saldırının asıl amacı kullanıcıyı aldatmak ve kullanıcı cihazına sızmaktır. Kullanıcı, e-posta veya mesaj aracılığı ile gelen herhangi bir bağlantı veya dosyayı (resim, pdf., uygulama, döküman vb.) açtığında saldırgan kullanıcının bilgilerine erişebilir[12]. Saldırgan kullanıcıya bu mesajı veya e-postayı gönderirken ilk amacı, kullanıcıyı ikna edip mesaj içeriğine tıklamasını sağlamaktır.



Şekil 4: *Phishing saldırı modellemesi[4].*

- **Phishing Saldırısına Karşı Güvenlik Duvarında Çözümler:**

URL ve Domain Filtreleme:

Güvenlik Duvarları e-posta veya web sitelerindeki linkleri inceleyebilirler. Bu linkler tehlikeli ise tespit ettikten sonra engeller.

E-posta Filtreleme ve Tanıma:

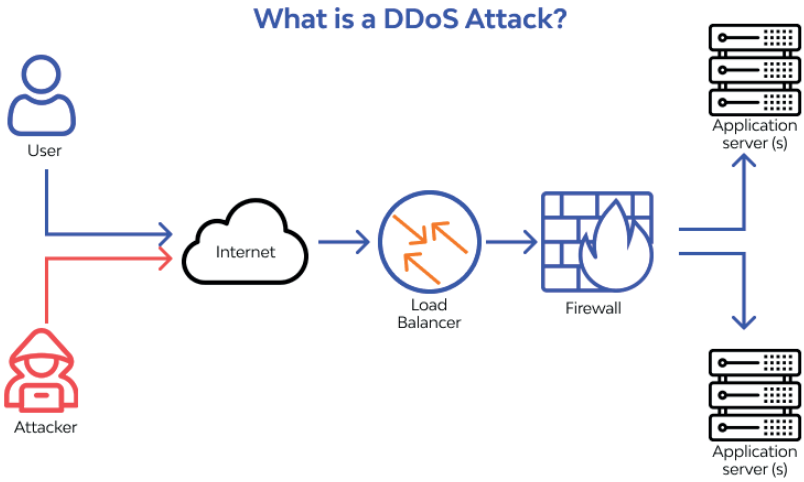
Güvenlik duvarları Phishing e-postalarını tespit edip önleyebilmek adına gönderen kişinin bilgilerini analiz eder. E-posta tanınmış bir Phishing şablonu içeriyorsa alıcıya ulaşmadan engellenir.

SSL(Secure Sockets Layer)/TLS(Transport Layer Security) İncelemesi ve Sertifika Kontrolü:

Güvenlik duvarları HTTPS (Secure Hyper Text Transfer Protocol) trafiğini izleyebilir. HTTPS trafiği içerisinde sahte veya geçersiz SSL/TLS sertifikaları tespit ederse web sitesini engeller[12].

- **DDoS(Distributed Denial of Service) Saldırıları :**

DDoS (Dağıtık Hizmet Engelleme), bir sistemi kapasitesinin üzerinde veri ile zorlayarak normal çalışmasını engelleme saldırısıdır. DDoS saldırıları genellikle bir kaynaktan değil farklı birçok kaynaktan gelir. Bu kaynaklar "botnet" adı verilen kötü amaçlı yazılımlar ile ele geçirilen milyonlarca cihazdan oluşur. Ele geçirilen bu cihazlar saldırı emirlerini dağınık şekilde çalışan kaynaklardan veya tek bir noktadan alabilir. DDoS saldırısı, hedeflenen sunucuyu veya ağı aşırı yükleyerek, belirli bir servisi ya da servisleri aşırı talep göndererek gerçekleştirilebilir. Önlem alınmadıysa ve saldırı amacına ulaştığında, hedef sistem talepleri işleyemez ve gerçek kullanıcılara hizmet veremez.



Şekil 5: *DDoS saldırı modellemesi[5].*

- **DDoS Saldırısına Karşı Güvenlik Duvarında Çözümler:**

Anormal Trafik Tespiti:

Güvenlik duvarı ağdaki anormal trafiği tespit edebilir. Normalden daha fazla bir trafik varsa ve bu aşırı yüklenmeye sebep oluyor ise bunun tespiti güvenlik duvarı tarafından yapılır.

Rate Limiting ve Trafik Yönetimi:

Güvenlik duvarları, belirli kaynaklardan gelen trafiği sınırlayabilir veya kesebilir. Bu sayede sistemde aşırı yüklenme olmadan hizmet vermeye devam edilir[13].

Geçici Bloklama ve Akıllı Engelleme:

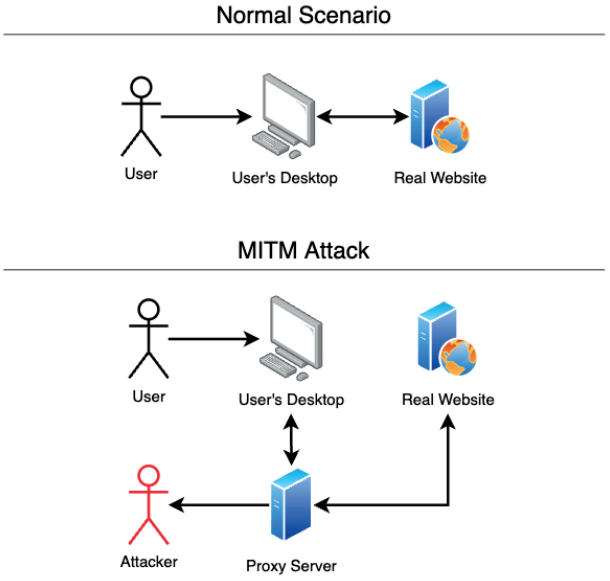
Güvenlik duvarları bir DDoS saldırısı esnasında saldırının yoğun olduğu kaynak IP'leri tespit eder. Tespit ettiği bu IP'leri geçici olarak engelleyebilir. Saldırı trafiğini analiz edebilir, analiz ettikten sonra gerçek trafiği tespit edebilir ve engelleme yapabilir.

Ağ Genişliği Yönetimi ve Yük Dağıtımı:

Güvenlik duvarlarının ağ genişliğini optimize etme özellikleri mevcuttur. Aşırı yük altındaki kaynakları korumak için yükü dağıtır.

- **MITM (Man-in-the-Middle) Saldırıları:**

Türkçe'ye "Ortadaki Adam" saldırısı olarak geçen bu saldırı türü, bir saldırganın iletişim kuran iki taraf arasına girerek iletişimi dinleme, değiştirme veya yönlendirme olarak tanımlanır[14]. Saldırı, iletişim halinde olan kullanıcıların haberi olmadan gerçekleşir ve saldırgan akan trafikteki bilgileri çalabilir. Saldırgan, halka açık düşük güvenlikli bir wifi ağına sızdıktan sonra ağda akan trafiği dinleyebilir. Kendilerini bir yönlendirici veya erişim noktası olarak tanıtarak veri trafiğini kontrol edebilir. Saldırgan ARP (Address Resolution Protocol) veya DNS (Domain Name System) gibi protokollerde hedef makineyi saldırganın kontrol ettiği bir makineye yönlendirebilir. Başarılı olursa tüm iletişimi kontrol edebilir. SSL/TLS gibi şifreleme protokollerini engelleyen saldırgan, güvenli iletişimi engeller ve verileri açık bir şekilde kontrol eder.



Şekil 6: *MITM saldırı modellemesi[6].*

- **MITM Saldırısına Karşı Güvenlik Duvarında Çözümler:**

Şifreli İletişim:

Güvenlik duvarları şifreli iletişimi destekler. İletişimin şifrelenmesini sağlayan protokellerin güvenli bir ortamda kullanılmasını sağlar. Bu sayede saldırganın verileri dinlemesi zorlaştırılır.

IDS(Intrusion Detection System) ve IPS(Intrusion Prevention System):

Gelişmiş Saldırı Tespit Sistemleri (IDS/IPS) sistemleri anormal ağ trafiğini izler ve saldırıları tespit eder. Güvenlik duvarları bu sistemler ile entegre çalışabilir.

3.2 Ağ Erişim Denetimi İhlalleri:

Bu saldırı yetkisiz erişimler ve zayıf güvenlikli şifreler üzerinden gerçekleşebilir. Saldırgan kolay tahmin edilebilir şifreleri tahmin ederek veya bazı yazılımlar aracılığı ile bu şifreleri deneyip bularak sisteme giriş yapabilir. Kolay tahmin edilir kısa şifrelerden uzak durmak ve aynı şifreyi birden fazla yerde kullanımından kaçınılmalıdır.

Ağ Erişim Denetimi İhlallerine Karşı Güvenlik Duvarında Çözümler:

Kimlik Doğrulama Kontrolleri:

Ağdaki kullanıcıların ve cihazların kimlik doğrulamasını yapan güvenlik duvarları, doğrulama sonrası yetkisiz erişimi engeller.

Yetkilendirme:

Kimlik doğrulamasının ardından kullanıcılar güvenlik duvarı tarafından verilen yetkiler ile ağda gezebilir. Yetkisi olmayan kullanıcı izin verilmeyen bir cihaza erişim sağlayamaz.

Segmentasyon ve Ağ Bölgeleri:

Güvenlik duvarları ağdaki farklı bölgeleri izole hale getirebilir. Bu bölgelere örnek olarak DMZ (Demilitarized Zone) verilebilir. Bu bölgelere karşıdan erişimi kısıtlayarak güvenliği arttırır[15].

Alarm Sistemleri:

Güvenlik duvarı ağda anormal bir durum tespit ettiğinde alarm durumuna geçer. Saldırı olması, yetkisiz bir erişim veya bağlantının kesilmesi durumunda yetkiliyi uyarabilir.

4. Gelecekteki Trendler ve Güvenlik Duvarları

4.1 Yapay Zeka ve Makine Öğrenimi ile Güvenlik Duvarları

Yapay zeka (YZ) ve makine öğrenimi (MO) teknolojileri, güvenlik duvarlarının gelecekteki evriminde önemli bir rol oynayacak. Bu teknolojiler, ağ

trafiğini analiz etme, tehditleri tanımlama ve anında tepki verme konusunda güvenlik duvarlarının etkinliğini artırabilir.

- **Yapay Zeka Destekli Tehdit İstihbaratı:**

Yapay zeka, büyük veri setlerini analiz ederek güvenlik duvarlarına gerçek zamanlı tehdit istihbaratı sağlayabilir. Tehdit tespit algoritmaları, geçmiş verileri kullanarak öğrenir ve bu sayede yeni ve karmaşık tehditlere karşı daha hassas hale gelir.

- **Davranış Tabanlı Analiz ve Öğrenme:**

Makine öğrenimi, ağ kullanıcılarının ve cihazlarının tipik davranışlarını anlayarak anormal aktiviteleri belirleyebilir. Bu, sıradışı veya potansiyel olarak zararlı aktiviteleri tespit etme yeteneğini artırarak güvenlik duvarlarını daha adaptif hale getirebilir.

- **Otomatik Tepki ve Güvenlik Olayları Yönetimi:**

Yapay zeka destekli güvenlik duvarları, tehdit algılandığında otomatik tepkilerle ağ güvenliğini güçlendirebilir. Bu, saldırılara anında müdahale edilmesini sağlayarak zararın minimize edilmesine yardımcı olabilir.

4.2 IoT (Nesnelerin İnterneti) ve Güvenlik Duvarları

Nesnelerin İnterneti (IoT), milyonlarca cihazın birbirine bağlı olduğu bir ekosistem oluşturuyor ve bu da yeni güvenlik zorluklarını beraberinde getiriyor.

- **Cihaz Tanıma ve Yetkilendirme:**

IoT güvenlik duvarları, ağa bağlanan her cihazı tanıyarak ve yetkilendirerek ağa güvenli bir şekilde entegre olmalarını sağlayabilir. Bu, yetkisiz cihazların ağa erişimini engeller.

- **Veri Şifreleme ve Güvenli İletişim:**

IoT cihazları arasındaki iletişim genellikle hassas veri içerir. Güvenlik duvarları, bu verileri şifreleyerek ve güvenli iletişim protokollerini kullanarak IoT ağlarını koruyabilir.

- **Sızma Tespit ve Tehdit Önleme:**

IoT cihazlarına yönelik potansiyel saldırıları önlemek için güvenlik duvarları, anomali tespiti ve sızma önleme yetenekleriyle donatılabilir.

4.3 Bulut Tabanlı Güvenlik Duvarları

Bulut tabanlı güvenlik duvarları, işletmelerin ve organizasyonların esneklik ve ölçeklenebilirlik ihtiyaçlarını karşılamak için geliştirilmiş çözümler sunar.

- **Uçtan Uca Şifreleme ve Veri Güvenliği:**

Bulut tabanlı güvenlik duvarları, kullanıcılar arasındaki iletişimi şifreleyerek veri güvenliğini sağlar. Bu, bulutta depolanan verilerin korunmasına yardımcı olur.

- **Otomatik Güncelleme ve Tehdit İstihbarat Paylaşımı:**

Bulut tabanlı güvenlik duvarları, otomatik güncelleme mekanizmaları ve geniş tehdit istihbarat ağlarına entegrasyon sayesinde sürekli olarak güncel kalabilirler.

- **Dağıtılmış Ağlarda Etkili Koruma:**

Bulut tabanlı güvenlik duvarları, coğrafi olarak dağılmış ağlarda da etkin bir koruma sağlayabilir. Bu, mobil çalışanlar, uzak ofisler ve bulut tabanlı altyapılarda güvenliği artırabilir.

Sonuç

Bu çalışmada, ağ güvenliği ve güvenlik duvarlarının önemi üzerine detaylı bir inceleme gerçekleştirilmiştir. Günümüzde, iş sürekliliği, müşteri güvenliği ve memnuniyeti gibi kritik konuların büyük ölçüde dijitalleşmesi ile birlikte, organizasyonların bilişim sistemlerini koruma süreci olan ağ güvenliği daha da önem kazanmıştır.

Güvenlik duvarları, ağ güvenliği stratejilerinin temel yapı taşlarından biridir. Ağlardaki güvenlik açıklarını en aza indirgemek ve siber saldırıları önlemek amacıyla kullanılan bu önemli cihazlar, birinci savunma hattını oluşturarak organizasyonları çeşitli tehditlere karşı korurlar. Ancak, güvenlik duvarlarının tek başına yeterli olmadığı bir gerçektir.

Çok katmanlı bir güvenlik stratejisi geliştirmek, organizasyonların daha etkili bir koruma sağlamalarına yardımcı olabilir. Bu strateji, güvenlik duvarlarını diğer güvenlik önlemleriyle entegre ederek, saldırılara karşı çok yönlü bir savunma sunar. Antivirüs yazılımları, düzenli güncellemeler, güvenli şifre kullanımı ve periyodik veri yedeklemeleri gibi önlemlerle birlikte güvenlik duvarları, bilişim altyapısını güven altına almanın bir parçasıdır.

Makalenin ilk bölümünde, ağ güvenliği kavramı geniş bir perspektifte ele alındı. İkinci bölümde güvenlik duvarlarının temel bileşenleri, türleri ve nasıl yapılandırıldığı detaylı bir şekilde incelendi. Üçüncü bölümde ise güvenlik duvarlarının ağ güvenliğindeki rolü daha ayrıntılı olarak tartışıldı. Popüler saldırı türleri örnekleriyle birlikte ele alındı ve bu saldırılara karşı güvenlik duvarlarının nasıl çözümler sunduğu açıklandı. Ayrıca, güncel tehdit istihbaratının etkili kullanımı ve güvenlik duvarlarının sürekli güncel tutulmasının önemi vurgulandı.

Bu çalışma, organizasyonların dijital varlıklarını koruma ve güvenlik stratejilerine katkıda bulunma amacını taşıyor. Güvenlik duvarları, bilişim sistemlerini koruma stratejisinin önemli bir parçasıdır, ancak bunlar tek başına yeterli değildir. Sürekli gelişen tehdit ortamında, organizasyonlar çok katmanlı bir güvenlik yaklaşımı benimsemeli ve güvenlik duvarlarını diğer güvenlik önlemleriyle entegre ederek güçlendirmelidir. Bu sayede, ağ güvenliği daha etkili bir şekilde sağlanabilir ve organizasyonlar dijital tehditlere karşı daha dirençli hale gelir.

KAYNAKÇA

[1] Erinç, B. (2002). *Internette Güvenlik Sorunu Eğitimi ve Firewall Kullanılarak Güvenliğin Sağlanması* (Doctoral dissertation, Marmara Universitesi (Turkey)).

[2] Akbaş, D., & Gümüşkaya, H. Bir Kurumsal Ağın ve Güvenlik Yapılarının Modellenmesi Modeling and Analysis of an Enterprise Network and Its Security Structures.

[3] ERDEM, O. A., & Kocaoğlu, R. (2014). YENİ BİR AĞ GÜVENLİĞİ YAKLAŞIMI: DİNAMİK ZEKİ GÜVENLİK DUVARI MİMARİSİ. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, *29*(4).

[4] FIRLAR, T. (2003). AĞ GÜVENLİĞİ. *Sakarya University Journal of Science*, *7*(1), 9-16.

[5] URL-1. Güvenlik duvarı nedir nasıl çalışır

https://www.avansas.com/blog/firewall-guvenlik-duvari-nedir-nasil-calisir (Erişim tarihi: 15.12.2023)

[6] Wang, N., Chen, X., Song, G., & Parsaei, H. (2015). Using Node-HTTP-Proxy for Remote Experiment Data Transmission Traversing Firewall. *International Journal of Online Engineering*, *11*(2).

[7] Karaarslan, E. (2003). Ağ güvenlik duvarı çözümü oluşturulurken dikkat edilmesi gereken hususlar.

[8] Alkan, G. (2009). *Sanallaştirilmiş Ağ Topolojisi Üzerinde Güvenlik Duvari ve Tehdit Gözetleme Sistemlerinin Otomatize Test Edilmesi* (Doctoral dissertation, Sakarya Universitesi (Turkey)).

[9] KILINÇ, F., & EYÜPOĞLU, C. (2023). AĞ ORTAMINDAKİ SALDIRI TÜRLERİ: SALDIRI SENARYO ÖRNEKLERİ. *İstanbul Ticaret Üniversitesi Teknoloji ve Uygulamalı Bilimler Dergisi*, *6*(1), 99-109.

[10] Baykara, M., Daş, R., & Karadoğan, İ. (2013, May). Bilgi güvenliği sistemlerinde kullanılan araçların incelenmesi. In *1st International Symposium on Digital Forensics and Security (ISDFS'13)* (Vol. 20, p. 21).

[11] Şahinaslan, Ö. (2013). *Siber saldırılara karşı kurumsal ağlarda oluşan güvenlik sorunu ve çözümü üzerine bir çalışma* (Master's thesis, Trakya Üniversitesi Fen Bilimleri Enstitüsü).

[12] ARSLAN, M. E. SİBER GÜVENLİK VE SİBER SALDIRI TÜRLERİ.

[13] KARAARSLAN, E., AKIN, G., & FETAH, V. KURUMSAL AĞLARDA ZARARLI YAZILIMLARLA MÜCADELE KILAVUZU.

[14] Yüksel, M., & ÖZTÜRK, N. (2017). SIP Saldırıları ve Güvenlik Yöntemleri. *Bilişim Teknolojileri Dergisi*, *10*(3), 301-310.

[15] OTTEKİN, M. F. (2017). GÜVENLİK DUVARI ETKİNLİK ÖLÇÜMÜ. *Uluslararası Bilgi Güvenliği Mühendisliği Dergisi*, *3*(2), 10-15.

ŞEKİL KAYNAKLARI

[1] URL-1. Güvenlik duvarı modellemesi.

https://miro.medium.com/v2/resize:fit:720/format:webp/1*4ij9zgDXeHOgmbyvia_tHA.jpeg (Erişim tarihi: 15.12.2023)

[2] URL-2. Güvenlik duvarı işlevleri modellemesi.

https://media.fs.com/images/community/upload/kindEditor/202110/20/what-is-next-generation-firewall-1634694043-t1DBJ4bCUo.jpg (Erişim tarihi: 18.12.2023)

[3] URL-3. Malware saldırı modellemesi.

https://www.researchgate.net/profile/Mahbub-E-Khoda/publication/353914338/figure/fig1/AS:1057285103185920@1629087820546/Malware-attack-in-edge-devices.png (Erişim tarihi: 28.12.2023)

[4] URL-4. Phishing saldırı modellemesi.

https://cyberartspro.com/wp-content/uploads/2022/03/10-768x423.png (Erişim tarihi: 01.01.2024)

[5] URL-5. DDoS saldırı modellemesi.

https://www.resimupload.org/images/2022/04/02/609bbffb353a9d0077180cad_what-is-ddos-attack.png (Erişim tarihi: 05.01.2024)

[6] URL-6. MITM saldırı modellemesi.

https://www.netskope.com/wp-content/uploads/2022/04/MITM-Phishing-1.png (Erişim tarihi: 15.01.2024)

BÖLÜM 2

DİZGİ EŞLEŞTİRME ALGORİTMALARI

Şerife Esra DİNÇER [1]

1 Dr Öğretim Üyesi Şerife Esra Dinçer
Gedik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü

Metin işleme alanında sıklıkla yararlanılan dizgi eşleştirme algoritmaları (string matching algorithms) bir veya daha fazla dizgiyi (pattern) daha büyük boyutta bir metin (text) veya dizgi (string) içinde aramak amacıyla kullanılır. Bilgisayar bilimleri alanında oldukça geniş bir kullanım yelpazesine sahip bir algoritma grubudur. Algoritmalar içinde "arama algoritmaları" üst başlığında gruplanabilir. Dizgi eşleştirme algoritmaları eşitlik kontrollü (exact string matching) ve yaklaşık benzerlik kontrollü (approximate string matching) olarak iki temel gruba ayrılır. İlkinde aranan verinin aynısının bulunması amaçlanırken, diğerinde benzerlik ve yaklaşık değer koşulu ile adımlar gerçekleştirilir. Diğer dizi eşleştirme algoritmaları bu iki üst grubun altında uyguladıkları yöntemlere göre kendi içlerinde alt gruplar oluştururlar.

Yöntemin ayrıntılarına girmeden önce kullanılan terimlerin özelliklerini bilmek önemlidir. Metin (text) ve dizgi (string) aynı tipte alfasayısal bilgiler içeren veri tipleridir. Aralarındaki temel fark içerebilecekleri karakter sayısı ve bu karakterlerin işlenme yöntemlerdir. Yazılım kodlaması açısından, metin veya dizgi alfasayısal karakterlerden oluşan birer dizi veya dizgi veri tipleridir. Birçok konuda her iki terim aynı bilgiyi ifade etmek için kullanılır. Dizgi kelimesi bazı kaynaklarda örüntü (pattern) olarak adlandırılmaktadır.

Ortaya çıkma ve temel kullanım alanı alfa sayısal bilgiler olsa da bu algoritmalar çeşitli sayısal, simgesel veriler ve görüntü işleme dâhil birçok alana uygulanabilmektedir. Bilgisayar işletim sistemlerinden, internet sitelerindeki arama motorlarına kadar birçok yazılımın içinde çalıştırılmaktadır. Genel seviyede örnek uygulamalar şöyle sıralanabilir; bir belge içinde veya metin dosyası içinde kelime veya dizgi arama işlemi, moleküler biyolojideki gibi belirli bir sırada yerleştirilmiş veriler içinde arama işlemleri, verilerin sıralı ve metin dosyalarında tutulduğu uygulamalarda kullanılır [1].

ÇALIŞMA ŞEKLİ

Dizgi eşleştirme algoritmaları yapılan belirli bir işi yapan bir üst algoritmanın içinden çağrılarak çalıştırılır. Diğer arama algoritmaları gibi aranan veriyi içinde arama yapılan veri kümesi içinde tarayarak eşleştirme yapar. Aranan veri bir veya daha fazla dizgi olabilir, arama yapılan veri kümesi büyük boyutta bir metin dosyası veya başka bir dizgi olabilir. Eşleştirme sonucunda dizgi, aranan veri kümesi içinde belirli bir doğruluk oranında bulunabilir veya hiç bulunamayabilir. Algoritmanın girdisi, aranan dizgi ve içinde arama yapılacak veri, çıktısı ile dizginin veri içinde bulunduğu adresler veya dizginin bulunmadığı bilgisidir. Elde edilen sonuç, çalışmanın amacına bağlı olarak değerlendirilmek üzere bir üst algoritmaya aktarılır.

Dizgi eşleştirme algoritmalarında aranan verinin tamamen aynısının bulunması amaçlanabilir, bu durumda birebir eşitlik kontrolü yapılarak eşitlik kontrollü (exact string matching) arama yöntemleri kullanılır. Bunun dışında, diğer bir yöntem ise yaklaşık benzerlik (approximate string matching)

aramasıdır. Bu yöntemde aranan dizginin benzeri bir yakınlık oranına göre taranır. Benzer dizgilerin aranması (regular expression searching) dizgi arama kullanılan karmaşık bir yöntemdir. Belirli karakterler "bulunabilir ancak zorunlu değil" koşulu ile aranır.

Her iki arama yönteminde de içinde aranan veride, tek veya çoklu dizgi arama işlemi yapılabilir. Tekli dizgi eşleştirme (single pattern search) işleminde ilk eşlenen veri bulunduktan sonra işleme son verilir, çoklu arama (multiple pattern search) işleminde ise arama işlemi veri setinin sonuna ulaşana kadar devam eder[2].

Örnek;

Dizgi arama algoritması basit bir şekilde şöyle örneklenebilir;

"*Kitap 120 sayfadan oluşur, kitap yazarlarından biri çok tanınmış bir kitap yazarıdır*"

Yukarıdaki metnin içine eşitlik kontrollü ve çoklu bir arama yaparak, **kitap** dizgisi bulunmak istenirse, arama sonucunda, **kitap** kelimesinin metin içerisinde bulunduğu, 1., 28. ve 70. indislerden başlayarak 3 kez tekrarlandığı bilgileri elde edilir.

DİZGİ ARAMA ÖLÇÜTLERİ

Dizgi arama işlemleri ile eşitlik veya benzerlik koşullarına göre tarama yapılırken aşağıdaki bilgiler elde edilebilir;

- Aranan dizginin, arama yapılan verinin içinde var olup olmadığı,
- Kaç kez tekrarlandığı,
- Bulunduğu indisler,
- Başka karakterlerle bitişik bulunduğu durumlar,
- Benzerlerinin varlık kontrolü.

Arama işleminde yapılan işlemin amacına göre istenen sonuçları elde etmek için çeşitli arama ölçütleri tanımlanabilir. Metin, simge ve görüntü aramalarında, eldeki verinin özeliklerine göre ölçütler farklılık göstermektedir. Aşağıda metin içinde arama işleminde kullanılabilecek belli başlı koşullar listelenmiştir;

- Büyük küçük harf farklılığı,
- Aranan dizginin kısaltmasının varlık kontrolü,
- Öncesi ve/veya sonrasında başka karakterlerin varlığı (önek, sonek türetme vb),
- Aranan dizginin boşluk içermesi,

- Yeni satır atlama, sekme gibi aralıklama karakterleri,
- Kesme işaretleri,
- Noktalı harflerin noktasız yazılma olasılıkları,
- Uzun metinlerde liste numaraları, ek not referansları,
- Farklı yazılıp aynı anlama gelen kelimeler (anne, ana gibi).

Yukarıda listelenen kontroller, dizgi arama algoritmalarını farklılaştıran ve karmaşıklık seviyelerini belirleyen özelliklerdir. Tek bir algoritma tüm istenen kontrolleri sağlayamayabilir. Bu durumda istenen sonuca ulaşmak için birden fazla dizgi algoritması peş peşe çalıştırılabilir [3].

TEMEL DİZGİ EŞLEŞTİRME ALGORİTMALARI

Dizgi eşleştirme alanında birkaç temel algoritmadan yararlanılarak birçok değişik algoritma geliştirilmiş ve bu sayede temel algoritmaların daha verimli ve etkili şekle dönüştürülmesi amaçlanmıştır. Algoritmalar içlerinde basit sıralı aramadan, ön işleme ile oluşturulan indeks tablosu yardımıyla arama gibi daha karmaşık yöntemler içerir.

Geliştirilmiş yöntemlerin verimlilikleri aşağıdaki ölçütlerle karşılaştırılabilir;

- Arama şekli,
- Arama yönü,
- Ön işlem gerektirmesi,
- Toplam arama maliyeti
- Karşılaştırma sayısı
- Bellek maliyeti

Arama yönüne göre algoritmalar solda sağa ve sağdan sola arama olarak 2 gruba ayrılır. Soldan sağa aramalar diğer yönde aramaya oranla daha basittir. Knuth-Morris-Pratt ve Rabin Karp Algoritmaları bu gruba örnek verilebilir.

Arama işlemi öncesinde bazı algoritmalar indeks tablosu veya benzeri durum tabloları oluşturmaktadır. İşlem sonucunun doğruluğunu sağlamak için geliştirilen bu adımlar işlem süresini ve kullanılan bellek alanını artırmaktadır.

Karşılaştırma sayısının düşüklüğü hızı olumlu etkileyen bir faktördür ancak işlem başarısını garanti altına almaz.

Yüksek bellek kullanımı, maliyeti artırdığı için algoritmaların performans değerlendirmesinde dikkate alınmaktadır [4] [5].

Birçok dizgi algoritmasının temel olarak aldığı, yaygın kullanılan dizgi eşleştirme algoritmalarından bazıları Boyer-Moore, Morris-Prat, Knuth-Morris-Pratt, ve Colussi olarak sıralanabilir. Belli başlı dizgi arama algoritmaları aşağıda detaylandırılmıştır.

Algoritma çalışma yöntemleri açıklanırken, içinde aranan dizgi m uzunluğunda, x=x[0..m-1] dizisi ile, içinde arama yapılan metin n uzunluğunda y=y[0..n-1] dizisiyle temsil edilmiştir. Birçok dizgi eşleştirme algoritması metin içinde m uzunluğunda bir pencere (window) kaydırarak tarama yapar. Pencerenin sol köşesi metnin ilk elemanı üzerine konumlandırılarak, dizginin karakterleri ile metnin karakterleri karşılaştırılır. Arama işlemi pencerenin sağa doğru kaydırılması ile devam eder. Metnin sağ taraftaki son karakteri ile pencerenin sağdaki son karakteri karşılaştırıldığında işlem tamamlanır. Bu tür taramaya *kayan pencere mekanizması* adı verilir.

Basit dizgi arama (Naive string search) algoritması:

En temel ve anlaşılması kolay bir dizgi arama algoritmasıdır. Arama algoritmaları grubu içinde "sıralı arama" kategorisine dahil edilebilir. Algoritmanın kullandığı yöntem şöyledir;

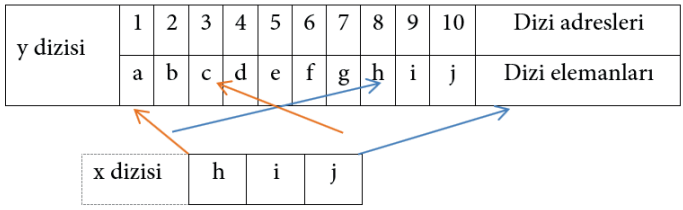
- Arama yapılacak metnin en başından başlanarak, aranan dizgi, sırayla metin elemanlarıyla karşılaştırılır.
- Karşılaştırma eldeki metnin sonuna kadar veya dizgi metnin içinde bulunana kadar devam eder.
- Eşleşen veri olması durumunda aranan dizgi metin içinde bulunmuştur.
- Çoklu arama yapılıyorsa içinde arama yapılan metnin sonuna kadar tarama yapılmaya devam edilir.
- Eşleşen verinin bulunduğu metin indisi veya pozisyonun sonuç olarak döndürülür.
- Metin sonuna gelindiğinde dizgi bulunamamışsa, metnin içinde olmadığı anlamına gelir.

Basit dizgi arama algoritması	
Arama şekli	Sıralı
Eşleştirme tipi	Tam
Ön işlem	Yok
Ek bellek	Gerektirmez
Tekli/çoklu arama	Tekli/çoklu
Karşılaştırma sayısı	(n-m+1)
Zaman karmaşıklığı	O(m x n)

Bu yöntem kayan pencere mekanizması (sliding window mechanism) olarak da adlandırılır. Pencere x dizisidir ve y dizisi üzerinde baştan başlayıp sağa doğru kaydırarak arama yapılır. Verilerin içinde yer aldığı dizinin (y) karakter sayısı n ve aranan dizginin (x) karakter sayısı m ise tam eşitlik aranması durumunda en fazla (n-m+1) kadar karşılaştırma yapılır.

Örnek:

Aşağıdaki örnekte aranan 3 karakterli dizgi, 10 karakterli bir metnin içinde aranmak istendiğinde (10-3+1) 8 adımda işlem tamamlanmıştır. Eğer aranan dizgi "xyz" olsaydı yine 8 arama adımı gerçekleştirilirdi.



Arama işleminde sırasıyla yapılan karşılaştırmalar;

1. abc == hij
2. bcd == hij
3. cde == hij
4. def == hij
5. efg == hij
6. fgh == hij
7. ghi == hij
8. hij == hij

Boyer-Moore algoritması:

Yaygın kullanılan, değişik türevleri geliştirilen verimli bir dizgi algoritmasıdır. Metin editörlerinde arama ve yerine yazma işlemlerinde kullanılır.

Boyer-Moore algoritması	
Arama şekli	Dizgi sağdan sola eşlenir, sıralı
Eşleştirme tipi	Tam
Ön işlem	Sadece dizgi için var
Ek bellek	Gerektirmez
Tekli/çoklu arama	Tekli/çoklu
Karşılaştırma sayısı	3n
Zaman karmaşıklığı	O(m x n)

Geleneksel arama yönteminde, önce aranan dizginin ilk karakterinden (x[0]) başlanarak metin içinde karşılaştırma yapılır ve bulunması durumunda (if x[0]==y[j]), dizginin diğer karakterleri (if x[1]==y[j+1]) karşılaştırılır. Bu yöntemde son karakterden (x[m-1]) başlanarak (if x[m-1]==y[m-1]) karşılaştırma yapılır.

Algoritma aranan dizginin en son karakterlerinden başlayarak, sağdan sola tarama yapar. Dizginin bulunması veya bulunamaması durumlarında, pencereyi sağa kaydırmak için iki fonksiyon kullanılır. Bunlar *önek kaydırması* (matching shift/good suffix shift) ve *uyumsuz karakter kaydırması* (occurrence shift/bad character shift) olarak adlandırılır.

Uyumsuz karakter kaydırması

Eşleşmeyen karakter "uyumsuz karakter" (bad character) olarak isimlendirilebilir. Dizgi ile metin karşılaştırıldığında, karakter eşleşmez ise, karakterin dizgi içinde bulunma ve bulunmama durumuna göre belirli sayıda sağa kaydırılır.

Bu kaydırma şeklinde önce dizgi için bir arama tablosu (look up table) veya uyumsuz karakter tablosu (bad character table) oluşturulur. Bu tablo metin içinde sağa yapılacak kaydırma sayısını belirler. Dizginin her bir elemanı ve elemanların en sağdaki elemana veya dizgi sonuna uzaklıkları hesaplanarak tabloya yerleştirilir. Eğer tekrarlanan eleman varsa, en sağdaki elemanın uzaklığı yazılır.

Örnek;

Dizgi elemanları K R K B ise arama tablosu şöyledir;

K	R	B
1	2	0

Burada K harfi, 1. ve 3. gözde tekrarlanmaktadır ve 3. gözün sağa uzaklığı 1'dir.

Uyumsuz karakter kaydırmasının işlem adımları şöyledir;

1) Eğer dizginin en son elemanı ile metin elemanı birbirine eşit ise, dizginin sol karakterleri metin ile geriye doğru karşılaştırılır ve

a) Eğer hepsi eşit ise, 1 tane "bulundu" kaydı yapılarak, dizgi sağa kendi uzunluğu kadar kaydırılır,

b) Eğer metnin karşılaştırılan dizgi elemanına eşit olmayan bir karakteri varsa ve bu karakter dizginin arama tablosunda yoksa dizgi sağa kendi uzunluğu kadar kaydırılır,

c) Eğer metnin karşılaştırılan dizgi elemanına eşit olmayan bir karakteri varsa ve bu karakter dizginin arama tablosunda varsa, bu karakterin tab-

lodaki uzaklık değeri kadar dizgi sağa kaydırılır,

2) Eğer dizgi ile metin elemanı eşit değil ise ve

a) Metnin karşılaştırılan elemanı dizginin arama tablosunda yoksa dizgi sağa kendi uzunluğu kadar kaydırılır,

b) Metnin karşılaştırılan elemanı dizginin arama tablosunda varsa bu karakterin tablodaki uzaklık değeri kadar dizgi sağa kaydırılır.

Örnek:

İndis, j	1	2	3	4	5	6	7	8	9	10	11	12
Y dizisi (metin)	A	B	E	C	F	A	B	A	A	B	A	C
X dizisi (dizgi)	A	B	A	C								
				A	B	A	C					
						A	B	A	C			
							A	B	A	C		
								A	B	A	C	

A	1
B	2
C	0

Aranan dizgi, m uzunluğunda x dizisi ile, içinde arama yapılan metin, n uzunluğunda y dizisi ile temsil edilsin. Yöntem adımları;

1) x dizisi ile y dizisi 0. indisten başlayacak şekilde konumlandırılır, (x[0..m-1], y[0..m-1]), x dizisinin en son karakterine karşılık gelen y dizi elemanı ile karşılaştırılır, x[3]==y[3] olduğu için karşılaştırma, x dizisinin başına kadar veya eşleşmeyen karakter bulunana kadar geriye doğru yapılmaya devam eder. (while (i > =0 and x[i]==y[j]) i--;j--;)

2) İkinci karakterler eşit olmadığı için ve metindeki karakter dizgi içinde bulunmadığı için (x[2] != y[2] and x $\nexists$ y[2]) dizgi kendi uzunluğu (j=j+m) kadar kaydırılır ve j=7'den karşılaştırma devam eder.

3) (x[4] != y[7] and x $\exists$ y[7]) karakterler eşit olmadığı için ve y[7] arama tablosunda bulunan bir karakter olduğu için u=2'dir. Dizgi (j=j+u) kadar kaydırılır ve j=9'den karşılaştırma devam eder.

4) (x[4] != y[9] and x $\exists$ y[9]) karakterler eşit olmadığı için ve y[7] arama tablosunda bulunan bir karakter olduğu için u=1'dir. Dizgi (j=j+u) kadar kaydırılır ve j=10'dan karşılaştırma devam eder.

5) (x[4] != y[10] and x $\exists$ y[10]) karakterler eşit olmadığı için ve y[10] arama tablosunda bulunan bir karakter olduğu için u=2'dir. Dizgi (j=j+u) kadar

kaydırılır ve j=12'dan karşılaştırma devam eder.

6) Tüm elemanlar eşit olduğu için arama sonlanır. (while (i > =0 and x[i]==y[j]) i--;j--;))

Bu algoritma 2 durumda başarısız olur; birincisi en son karakterin arama tablosunda 0 değer alması nedeniyle kaydırma yapılamaması ve ikincisi dizgi işaretçisinin sağında eşit olmayan karakter bulunmasıdır. Bu sorunlar *önek* k*aydırması* (good suffix shift) ile çözülebilir [1].

KAYNAKÇA

[1] Charras, C., Lecroq, T., 2004, Handbook of Exact String_Matching Algorithms

[2] A. Chayapathi, S. Kumar , Survey and comparison string matching algorithms,

Turkish Journal of Computer and Mathematics Education, Vol.12 No.12 (2021), 1471-1491

[3] K. Alhendawi, A. Baharudin, et al, 2013, String Matching Algoritms (SMAs): Survey & Empirical Analysis, International Journal of Computer Science and Management Research, Vol 2 Issue 5 May 2013, ISSN 2278-733X

[4] S. Hakak, A. Kamsin et al, Exact String Matching Algorithms: Survey, Issues, and Future Research Directions, Specıal Sectıon On New Trends in Brain Signal Processing And Analysis

[5] L. Boytsov, Indexing methods for approximate dictionary searching: Comparative analysis, Journal of Experimental Algorithmics, 2011, 16(1): 1–91. doi:10.1145/1963190.1963191.

BÖLÜM 3

YAPAY ZEKA DESTEKLİ KİŞİSEL AKILLI ASİSTANIN GELİŞTİRİLMESİ

Şenay KOCAKOYUN AYDOĞAN[1]

1 Dr. Öğr. Üyesi Şenay KOCAKOYUN AYDOĞAN, İstanbul Gedik Üniversitesi, Bilişim Güvenliği Teknolojisi Programı senay.aydogan@gedik.edu.tr, ORCİD: https://orcid.org/0000-0002-3405-6497

Giriş

Yapay zeka (AI), insan zekasının insan gibi düşünmeye ve hareket etmeye programlanmış makinelerde simülasyonunu ifade etmektedir. Kişisel akıllı asistanlar bağlamında, yapay zeka teknolojisi, doğal dil komutlarını ve kullanıcılardan gelen sorguları anlayabilen ve bunlara yanıt verebilen sanal asistanlar geliştirmek için kullanılır. Yapay zeka destekli bu kişisel asistanlar, kullanıcılara randevu planlama, hatırlatıcı sağlama, soruları yanıtlama ve daha fazlası gibi çeşitli görevlerde yardımcı olmak için tasarlanmıştır (Khaokaew, Holcombe-James, Rahaman, Liono, Trippas, Spina, & Salim, 2022). Yapay zeka destekli kişisel akıllı asistanların gelişimi, doğal dil işleme (NLP) ve makine öğrenimi algoritmalarındaki gelişmeler sayesinde son yıllarda önemli ölçüde ilerlemiştir. Bu teknolojiler, kişisel asistanların karmaşık sorguları ve komutları anlamalarına ve bunlara yanıt vermelerine ve "öğrenme" olarak bilinen bir süreç aracılığıyla performanslarını zaman içinde iyileştirmelerine olanak tanımaktadır (Canbek, & Mutlu, 2016).

Yapay zeka destekli popüler kişisel akıllı asistanlara bazı örnekler arasında Apple'ın Siri'si, Amazon'un Alexa'sı ve Google'ın Asistanı yer almaktadır (Castro & New, 2016). Bu asistanlar, akıllı telefonlar, akıllı hoparlörler ve akıllı ev cihazları dahil olmak üzere çeşitli cihazlarda yerleşiktir. Basit sesli komutlar veya metin tabanlı sorgular yoluyla kullanıcılara sunulmaktadır. Yapay zeka destekli kişisel akıllı asistanlar, günlük görevlere yardımcı olmanın yanı sıra, kullanıcılar için artan üretkenlik, rahatlık ve erişilebilirlik gibi bir dizi başka avantaj da sağlayabilir (Göksel-Canbek, & Mutlu, 2016). Teknoloji ilerlemeye devam ettikçe, kişisel akıllı asistanların daha da sofistike hale gelmesi ve daha geniş bir görev ve zorluk yelpazesinin üstesinden gelme becerisine sahip olması muhtemeldir.

Yapay zeka teknolojileri insan davranışında zeka ile ilişkilendirdiğimiz özellikleri sergileyen bir alandır. Yapay zeka, insanların sağduyuları ve sezgileri sayesinde yeni durumlar karşısında yaratıcı çözümler üretebilme yeteneğine sahiptir, ancak insanların sahip olduğu bazı sınırlamalara da sahiptir. Örneğin, insanların sınırlı dikkat süreleri ve tek seferde sadece birkaç alternatifi analiz edebilme yetenekleri vardır. Bunun yanı sıra, insanların stres, yorgunluk ve zaman sıkıntısı gibi faktörler insanların hafızalarını etkileyebilir. Yapay zeka ise hızlı, dikkatli, kusursuz, açık ve nesnel olma özelliklerine sahip olabilir, ancak sağduyu ve yeni durumlarla başa çıkma yeteneklerinden yoksundur. Bu nedenle, insan ve yapay zeka giderek daha fazla işbirliği yapmakta ve birbirlerini tamamlamaktadır (Kişi, 2021). İnsanlar, yapay zekanın hızlı ve kusursuz çalışma özelliklerinden faydalanırken, yapay zeka da insanların yeni durumlarla başa çıkma yeteneklerinden yararlanabilir. Bu sayede, ikisi de birbirlerinin güçlerini tamamlayarak daha etkili hale gelebilirler.

Yapay Zeka

Yapay zeka teknolojisi bilişsel işlevleri yerine getirmeyi sağlayan bir teknolojik dalga olarak tanımlanabilir (Gür, Ayden, & Yücel, 2019). Yapay zeka, işletmelerin karmaşık sorunlarını çözmede önemli bir rol üstlenmektedir. Bu rolünü, algoritmalarda ilerleme, büyük hacimli veri setlerinin analizi, artan hesaplama gücü ve düşük maliyetli depolama gibi teknolojik gelişmeler sayesinde ortaya çıkarmaktadır. Ayrıca, yapay zeka tahmin faaliyetlerini uygun maliyetli hale getirerek, rutin ve tekrarlanabilir işlerin makineler aracılığıyla anında otomatikleştirilmesini sağlamaktadır. Bu sayede, işletmeler daha verimli ve kârlı hale gelebilir (Gures, Shayea, Ergen, Azmi, & El-Saleh, 2022).

Son yıllarda, AI hızla gelişmiş ve insanların yaşam tarzlarını değiştirmiştir (Huang, Cai, Xu, Xu, Gu, & Jiang, 2019). Yapay zekanın geliştirilmesi, ulusal rekabet gücünü artırarak ve güvenliği koruyarak dünya çapındaki ülkeler için önemli bir kalkınma stratejisi haline gelmiştir (Rajkomar, Oren, Chen, Dai, Hajaj, Hardt,... & Dean, 2018). Birçok ülke, yeni bir uluslararası rekabette liderliği ele geçirmek için tercihli politikalar uygulamaya koymuş ve kilit teknolojilerin ve yeteneklerin dağıtımını güçlendirmiştir (Xu, Tan, Zhen, & Shen, 2008). Bilim ve teknolojide; Google, Microsoft ve IBM gibi büyük şirketler kendini yapay zekaya adamıştır ve yapay zekayı giderek daha fazla alana uygulamaktadır (Shi, Huang, He, Xu, Liu, Qin,... & Zhao, 2007).

AI, biliş, makine öğrenimi , duygu tanıma, insan-bilgisayar etkileşimi, veri depolama ve karar vermeyi entegre etme yeteneğine sahip çok disiplinli bir teknolojidir (Lu, 2019). İlk olarak John McCarthy tarafından 20. yüzyılın ortalarında Dartmouth Konferansı'nda önerilmiştir. 1993'ten beri AI, bazı dönüm noktası sonuçları elde etmiştir. BP algoritmasının geniş uygulaması nedeniyle, sinir ağı hızla gelişmiştir. Geniş ölçekli bir ortamda, uzman sistemlerin kapsamlı kullanımı, endüstriyi çok fazla maliyetten kurtarmış ve endüstri verimliliğini artırmıştır (Cai-Ming, & Hao-Nan, 2020).

Yapay zeka, belirli görevleri yerine getirmek için insan zekasını taklit edebilen ve topladığı bilgileri tekrarlayarak kendi kendini geliştiren sistemlerdir. Yapay zekayı günümüz teknoloji sistemlerinden ayıran özellik insan zekasını taklit edebilme özelliğidir. Bu sistem, var olan durumları gözlemleyip daha önceden belirlenen parametreler ile bir tepki verir. Yapay zeka duruma ilişkin verileri hızlı, tekrarlamalı ve akıllı algoritmaları kullanarak işlem yapmaktadır.

Yapay zeka teknolojisinin gelişimi ile birlikte, insanların hayatlarında birçok farklı alanda kullandıkları teknolojik ürünlerde birçok gelişme gözlemlenmiştir. Yapay zeka, makine öğrenimi, nesne tanıma ve veri analitiği gibi çeşitli teknolojileri kullanarak, insanların hayatlarını kolaylaştıran ve hızlandıran birçok çözüm üretmiştir. Örneğin, yapay sinir ağları sayesinde, bilgisayarlar insan gibi düşünebilir ve kendi kendine öğrenebilirler. Bu saye-

de, bilgisayarlar insanların hayatını kolaylaştıran ve hızlandıran birçok uygulama geliştirilebilir (Öztürk, & Şahin, 2018). Doğal dil işleme teknolojisi, makine öğrenimi, nesne tanıma ve veri analitiği de benzer şekilde, insanların dilini anlayarak, yapay zeka sistemleri tarafından kullanılabilecek veri üretebilir.

Yapay zekanın ilerlemesi ve endüstrilerde makine öğrenimi ve derin öğrenme tabanlı yöntemlerin kullanılması, uygulamalarını Endüstri 4.0'ın bir parçası olacak şekilde güçlendirmektedir (Ahmed, Jeon, & Piccialli, 2022).

Makine Öğrenme

Makine Öğrenmesi (ML) bir bilgisayarın kendi kendine öğrenme yeteneğini kullanan bir yöntemdir. Bu yöntem sayesinde, bir bilgisayar veri kümesi üzerinden öğrenme yaparak, kendi kendine yeni bilgiler edinebilir. Makine öğreniminin temel fikri, veriden öğrenerek performansını artıran bir algoritmanın kullanılmasıdır (Nilsson, 1982). ML, veri üzerinden öğrenme yaparak, bir bilgisayarın bir görevi daha etkin ve hızlı bir şekilde yerine getirme yeteneğini geliştirir. Örneğin, bir ML modeli, veri kümesi üzerinden öğrenme yaparak, bir görüntü içerisinde nesneleri tanımlayabilir veya bir metin içerisinde anahtar kelimeleri tespit edebilir.

Makine öğreniminin, günümüzde mevcut olan veri zenginliğinden iş değeri elde etmeleri konusunda yüksek bir potansiyeli bulunmaktadır. ML, müşterilerin davranışlarını ve risk faktörlerini elde edebilmek için kullanılan en iyi AI teknolojilerinden biridir. ML, kodlama yapmadan halihazırda kodlanmış olan talimatlardan faydalanarak bilgisayar programlarını kendi başlarına öğrenmesi ve geliştirilmesi için kullanılmaktadır (Kumar, Srivastava, Bisht, 2019). Yapay zekânın ana uygulama yöntemlerinden biri olan makine öğrenimi, örneklerden öğrenebilen algoritmalar ile zaman geçtikçe daha fazla veri ile yüksek performans gösterebilmektedir (Oxborough, Cameron, and Rao, 2017). ML, hazır talimatları kullanarak çözüm üretmek yerine örneklerden öğrenerek, görüntü, resim ve ses tanıma gibi birçok zor probleme çözüm getirmektedir. Bilgisayar ortamında saklanan büyük boyuttaki verilerin analizi ve yorumlanması ML algoritmaları ile mümkündür (Tuba, & Delice,2019). Bilgisayarlara otonom olarak görevlerini nasıl gerçekleştireceklerini öğreten makine öğrenmesi algoritmaları geniş bir uygulama alanına sahiptir (Karakuş, 2021).

Makine öğrenimi tabanlı yöntemler, örneğin eğitim yoluyla sistem sonuçlarını otomatik olarak öğrenir ve geliştirir (Ang, Goh, Saldivar, & Li, 2017). Bu yöntemler, her tanınabilir model için nihai çıktıyı inceler ve bir çıktı sağlamak için tersine mühendislik yönlerini arar. Önceki deneyimlere dayalı olarak nasıl sonuçlara ve kararlara varılacağına dair bir sistem geliştirir (Bougdira, Akharraz, & Ahaitouf, (2020). Makine öğrenimi yoluyla çözülmesi gereken en önemli dört problem türü; tahmin, kümeleme, sınıf-

landırma ve boyut azaltmadır (Erhan, Courville, Bengio, & Vincent, (2010). Öğrenme yöntemlerinin sınıflandırılması göz önüne alındığında, makine öğrenmesi dört kategoriye ayrılabilir. Bunlar denetimli öğrenme, denetimsiz öğrenme , yarı denetimli öğrenme ve pekiştirmeli öğrenme olarak kategorize edilir (Bose, 2017).

Denetimli öğrenme, yeni verilerin türünü veya değerini tahmin etmek için etiketlenmiş verilerin eğitilmesi için kullanılması anlamına gelir. Farklı tahmin sonuçlarına göre, bu iki kategoriye ayrılabilir: sınıflandırma ve regresyon. Denetimli öğrenmenin tipik yöntemleri, SVM (Süper Vektör Makinesi) ve doğrusal ayrımdır (Morocho-Cayamcela, Lee, & Lim, 2019; Neyshabur, Bhojanapalli, McAllester, & Srebro, 2017). Regresyon problemi , sürekli değerlerin çıktısının tahminini ifade etmektedir. Konut fiyat verilerini analiz edebilir, örnek veri girişine göre sığdırabilir ve ardından konut fiyatlarını tahmin etmek için sürekli bir eğri elde edebilir (Bose, 2017).

Verilerin etiketi olmadığında, denetimsiz öğrenme veri madenciliği kullanılmaktadır. Denetimsiz öğrenme esas olarak kümelemede yansıtılır. Kısacası, veriler etiketsiz olarak farklı özelliklere göre sınıflandırılabilir. Tipik denetimsiz öğrenme yöntemleri, k-kümeleme ve temel bileşen analizini içerir. k-kümelemenin önemli öncülü, veriler arasındaki farkın Öklid mesafesi ile ölçülebilmesidir . Ölçülemiyorsa, kullanılabilir bir Öklid mesafesine dönüştürülmesi gerekir. Temel bileşen analizi istatistiksel bir yöntemdir. Ortogonal dönüşüm kullanarak, ilgili değişkenler ilişkisiz değişkenlere dönüştürülür; dönüştürülen değişkenlere temel bileşenler denir. Temel fikir, orijinal ilgili göstergeleri bir dizi bağımsız kapsamlı göstergeyle değiştirmektir (Baryannis, Validi, Dani, & Antoniou, 2019).

Yarı denetimli öğrenme, denetimli öğrenme ve denetimsiz öğrenmenin bir karışımı olarak ifade edilebilir. Aslında, etiketlenmiş veriler ve etiketlenmemiş veriler, öğrenme sürecinde karıştırılır. Normal şartlar altında, işaretlenmemiş veri miktarı, işaretlenmiş veri miktarından çok daha fazladır. Yarı denetimli öğrenme fikri idealdir, ancak pratik uygulamalarda pek kullanılmaz. Yaygın olarak kullanılan yarı denetimli öğrenme algoritmaları, kendi kendine eğitim, grafik tabanlı yarı denetimli öğrenme ve yarı denetimli destek vektör makinelerini (S3VM) içermektedir (Kızılkaya, & Oğuzlar, 2018).

Takviyeli öğrenme, çevre ile etkileşime girerek, eylemlerin kalitesini ödül seviyelerine göre değerlendirerek ve ardından modeli eğiterek ödüller elde etme yöntemidir. Takviyeli öğrenmede keşif ve geliştirmenin önemi zorlu bir konudur: daha iyi ödüller elde etmek için insanlar en yüksek ödülü alabilecek eylemi seçmelidir, ancak insanlar aynı zamanda bilinmeyen eylemleri de bulmalıdır (Chollet, 2017). Takviyeli öğrenmenin temeli davranışsal psikolojiden gelir. Başka bir deyişle, pekiştirmeli öğrenme, ödül davranışını iyileştirebilir ve ceza davranışını zayıflatabilir. En büyük getiriyi elde etmek için en

iyi işlemi ve davranışı bulmak için model deneme-yanılma mekanizmasıyla eğitilebilir. Bu, insanların veya hayvanların öğrenme modelini taklit eder ve belirli bir yönde öğrenmeleri için aracıları yönlendirmeye ihtiyaç duymaz (Li, Warfield, Guo, Guo, & Qi, 2007; Qi, Wu, Li, & Shu, 2007; Wang, Zou, Su, Li, & Chaudhry, 2013).

Derin Öğrenme

Derin Öğrenme (Deep Learning, DL) bir makine öğrenimi ML yöntemidir ve yapay sinir ağlarının (Artificial Neural Networks, ANNs) bir çeşididir (Ahmed, Jeon, & Piccialli, 2022). Derin öğrenme, makineleri karmaşık sorunları çözmek için gereken tekniklerle donatan makine öğreniminin bir alt kümesidir. Veri bilimi, eyleme geçirilebilir sonuçlara varmak için yapay zeka, makine öğrenimi ve derin öğrenmeyi uygulayan ayrı bir çalışma dalıdır (Goodell, Kumar, Lim, & Pattnaik, 2021). DL, birçok katmandan oluşan ağlar kullanarak, veri üzerinden öğrenme yapar. Bu katmanlar arasında veri aktarımı gerçekleştirilir ve bu sayede, bir bilgisayar veri kümesi üzerinden öğrenme yaparak, kendi kendine yeni bilgiler edinebilir. DL, genellikle çok büyük veri kümeleri üzerinden öğrenme yapar ve bu sayede, daha hassas sonuçlar elde edilebilir. Örneğin, bir DL modeli, çok sayıda görüntü verisi üzerinden öğrenme yaparak, görüntülerdeki nesneleri daha hassas bir şekilde tanımlayabilir. Ayrıca, DL, görüntülerdeki nesnelerin yerlerini, özelliklerini ve diğer detaylarını da tespit edebilir. Bir sistemi veya makineyi, bilgileri katmanlar aracılığıyla işlemesi, sınıflandırması, yorumlaması ve sonucu tahmin etmesi için yönlendirir. Temel olarak kullanılan bazı DL yaklaşımları, evrişimli sinir ağları (CNN), tekrarlayan sinir ağları (RNN) ve üretken sinir ağlarıdır (GNN) (Ahmed, Jeon, & Piccialli, (2022).

Makine öğrenimi, bir bilgisayarın veri ve öğrenme algoritmalarını kullanarak bir görevin çözümünü öğrenmesini ve geliştirmeyi amaçlar. Derin öğrenme, bu yöntemlerin bir alt kümesidir ve özellikle yüksek boyutlu veriler için etkili olan ağırlıklı bir ağ modelidir. Derin öğrenme modelleri, birçok katmanı olan ağlar oluşturur ve her katman, verileri işleme ve öğrenme için kullanılır. Bu sayede, derin öğrenme modelleri, veri içerisinde gizli kalıcı özellikleri keşfedebilir ve daha yüksek performanslı tahminler yapabilir. Makine öğrenimi, derin öğrenmeyle bilgilendirildiğinde, çeşitli görevler için dikkate değer ölçüde başarılı olduğu bulunmuştur (Dixon, Halperin, & Bilokon, 2020).

Doğal Dil İşleme

Doğal Dil İşleme (Natural Language Processing, NLP) ise, bir bilgisayarın insan dilini anlama ve kullanma yeteneğini geliştirmeyi amaçlayan bir yöntemdir. NLP, yapay zeka (YA) teknolojisi içinde yer alan bir alt dalıdır ve genellikle, metinleri analiz etme, dil öğrenme ve dil üreteme gibi görevleri yerine getirir. NLP, bir bilgisayarın insan dilini anlama ve kullanma yeteneği-

ni geliştirmeyi amaçlamaktadır (Seker, 2015). NLP, metinleri analiz etme, dil öğrenme ve dil üreteme gibi görevleri yerine getirir. NLP, birçok farklı alanda kullanılır ve insanların dilini anlayarak, yapay zeka sistemleri tarafından kullanılabilecek veri üretebilir (Kang, Cai, Tan, Huang, & Liu, 2020). Örneğin, bir NLP sistemi, bir metin içerisinde anahtar kelimeleri tespit ederek, bir arama motoru tarafından kullanılabilir. Ayrıca, NLP, bir chatbot tarafından kullanılarak, insanların dilini anlayarak, cevaplar verebilir veya bir çeviri sistemi tarafından kullanılarak, bir dilin diğer bir dile çevrilmesini sağlayabilir.

NLP, bilgisayar bilimi ve insan dilbilimi arasında disiplinler arası bir konu olan bilgisayarların insan metin dilini tanıma ve anlama yeteneğini ifade etmmektedir. Doğal dildeki en büyük fark , insan düşüncesinin dile dayalı olmasıdır, bu nedenle doğal dil işleme aynı zamanda AI'nın bir hedefini temsil etmektedir. Doğal dil işleme yedi yöne ayrılır: gramer ve semantik analiz, bilgi çıkarma, metin madenciliği, bilgi alma, makine çevirisi, soru cevaplama sistemi ve diyalog sistemi (Zhang, Xu, & Chen, 2020).

NLP, bilgisayarlarla iletişim kurmak için doğal dili kullanan bir teknolojidir. Doğal dili işlemenin anahtarı, bilgisayarların doğal dili "anlamasına" izin vermektir. bu nedenle hesaplamalı dilbilim olarak da adlandırılır. Dil bilgi işleme ve yapay zekanın kesiştiği noktada yer alır. Bir makine olarak önce ses sinyallerini toplar. Doğal dil işleme teknolojisi, ses sinyallerini metin sinyallerine ve metnin anlamına dönüştürür. Daha sonra makine sesleri kelimelere, kelimeleri de anlamlara dönüştürür. Bu iki işlemi tamamladıktan sonra makine duyabilir ve anlayabilir. Konuşma tanıma ve anlamsal anlama teknolojisi ile donatılmış makine, algoritmayı sürekli öğrenmede optimize eder, böylece makine sadece dinlemekle kalmaz, aynı zamanda anlayabilir ve hatta duyguları anlayabilir (Bostrom, & Yudkowsky, 2018).

Akıllı Asistanların Kullanım Alanları

Yapay zeka tabanlı dijital asistanlara yönelik araştırmaların, Joseph Weizenbaum'un 1966'daki ünlü ELIZA'sına kadar uzanan uzun bir geçmişi vardır. Buna paralel olarak, Microsoft, IBM, Google ve Amazon gibi küresel teknoloji şirketleri, yapay zekayı ilerletmek için on yıllardır yoğun bir şekilde çalışmaktadır (Maedche, Legner, Benlian, Berger, Gimpel, Hess, & Söllner, 2019). Bulut hizmeti altyapısı, doğal dil işleme, anlamsal akıl yürütme, ses tanıma ve ses sentezi gibi teknik gelişmelerin ortaya çıkışı, Apple'ın Siri'si, Microsoft'un Cortana'sı, Samsung'un Bixby'si, Amazon'un Alexa'sı, Google'ın Google Asistanı gibi modern akıllı asistanların yolunu açmıştır. Yapay zekadaki son gelişmelerden güç alan bu asistanlar, günlük hayatımızın bir parçası haline gelmiştir. Amazon Alexa gibi ses tabanlı asistanlar, Facebook Messenger'a gömülü olanlar gibi metin tabanlı (sohbet robotları) asistanlar gibi çeşitli dijital asistanların sürekli artan kullanımını gözlemliyoruz.

Yapay zeka tabanlı dijital asistanların, işin geleceğinde kilit bir unsur olacağı öngörülmektedir. Slack veya Microsoft Teams gibi günümüzün kurumsal iletişim platformları, işi artırmak için zaten birçok farklı bot türü sağlamaktadır. Bu akıllı hizmet sistemleri, kullanıcı ile doğal dil aracılığıyla etkileşime girerek birçok hizmet ve bilgi edinme olanağı sunarken, kullanıcıların günlük görevlerini bildirme ve karmaşıklığını azaltmak için yararlanılabilmektedir (Cowan, Pantidi, Coyle, Morrissey, Clarke, Al-Shehri, & Bandeira, 2017). Dijital platformlar üzerinde bugün yapabileceğiniz tüm işlemleri akıllı asistanlar ile yapabilirsiniz, web tarayıcınızda gerçekleştireceğiniz bir aramayı sesli olarak yapabilirsiniz veya akıllı bir eve sahipseniz evinizdeki teknolojik ürünlerin çoğunu akıllı asistanınız ile yönetebilirsiniz.

Günümüzde yapay zeka ve robotların her ne kadar insanların yerini alacağı düşünülse de yapay zeka ve robot çalışmaları insanların işlerini kolaylaştırmak, geliştirmek ve iş hızını artırmak için geliştirilmektedir Bugün direk olarak kendi kişisel asistanlarımız olmasa da herhangi bir şekilde hepimiz akıllı asistan çözümlerinden faydalanmaktayız. Buna şuan dünyada kullanıma geçmiş akıllı süpürgeler, banka, okul web siteleri, telefonlarımızdaki asistanlar, hatta çoğu web sitesinin kullandığı akıllı asistanları örnek olarak gösterebiliriz. Günün sonunda gelecekte birçok insanın evde, işte veya gittikleri yerlerde kullanacakları yapay zekaya sahip teknolojik aletlere sahip olmaları muhtemeldir, bu insanları daha verimli ve günlük işlerinde zaman kazandıracak bir ilerlemedir.

Akıllı Asistanlar, birçok farklı alanda kullanılabilir. Aşağıdaki bazı örnekler, Akıllı Asistanların kullanılabileceği alanları göstermektedir:

1. Ev Otomasyonu: Akıllı Asistanlar, ev otomasyon sistemleriyle entegre edilerek, evdeki ısıtma, aydınlatma ve ev cihazlarının kontrolü gibi görevleri yerine getirmeye yardımcı olabilir

2. Eğitim: Akıllı Asistanlar, öğrencilerin derslerini anlamalarına yardımcı olmak için kullanılabilir. Öğrenciler, Akıllı Asistanlar aracılığıyla derslerini tekrar etme ve öğrendiklerini test etme imkanına sahip olabilir.

3. Sağlık: Akıllı Asistanlar, hastaların sağlık durumlarını takip etmeye ve doktorlarına ulaşmaya yardımcı olabilir. Ayrıca, Akıllı Asistanlar, kişinin ilaç kullanımını hatırlatabilecek ve sağlık bilgileri hakkında bilgi verebilir.

4. İşletmeler: Akıllı Asistanlar, işletmelerin müşteri hizmetlerine yardımcı olabilir. Örneğin, Akıllı Asistanlar, müşterilerin sorularını yanıtlamaya ve ürünler hakkında bilgi vermeye yardımcı olabilir.

5. Günlük Yaşam: Akıllı Asistanlar, insanların günlük yaşamlarında yapacaklarını hatırlatmaya ve önemli tarihleri hatırlatmaya yardımcı olabilir. Ayrıca, Akıllı Asistanlar, insanların yemek tariflerini aramasına ve yol tarifleri vermeye yardımcı olabilir.

Bu tür asistanları akıllı evler (Benlian, Klumpe, & Hinz, 2019), akıllı arabalar (Mihale-Wilson, Zibuschka, & Hinz, 2019), robo-danışmanlık (Adam, Toutaoui, Pfeuffer, Hinz, 2019; Jung, Dorner, Glaser, & Morana, 2018), müşteri hizmetleri (Gnewuch, Morana, & Maedche, 2017). elektronik ticarette (Qiu, & Benbasat,2009), sağlık hizmetlerinde (Laranjo, Dunn, Tong, Kocaballi, Chen, Bashir, & Coiera, 2018), veya pedagojik ajanlar olarak (Fryer, Ainley, Thompson, Gibson, & Sherlock, 2017) dahil olmak üzere çok sayıda uygulama bulunmaktadır. Bu örnekler, Akıllı Asistanların kullanılabileceği alanların birkaçını göstermektedir. Akıllı Asistanlar, birçok farklı alanda kullanılabilecek ve insanların hayatlarını kolaylaştırmaya yardımcı olacak çeşitli görevler yerine getirebilir.

Yapay Zeka Destekli Kişisel Akıllı Asistanın Geliştirilmesi

Yapay Zeka Destekli Kişisel Akıllı Asistan geliştirilirken, işlemler için farklı, arayüz için farklı dil kullanılmıştır. Akıllı asistanın arka plan işlemleri için Python dili kullanılırken, kullanıcı arayüzü için C# dili kullanılmıştır. Bu sayede Python kütüphaneleri ile tasarlanabilen eski arayüzlere göre daha modern bir arayüz elde edilip, arka plan işlemlerinin gecikmesi daha aza indirilmektedir.

Yapay Zeka Destekli Kişisel Akıllı Asistan geliştirilmesi için Azure alt yapısı kullanılmaktadır. Azure, Microsoft tarafından sunulan bir bulut bilişim platformudur ve yapay zeka (YA) uygulamaları geliştirmek için kullanılmaktadır. Azure, yapay zeka uygulamaları geliştirme, derleme ve dağıtma süreçlerini kolaylaştıran birçok araç ve hizmet sunmaktadır. Bu araçlar ve hizmetler arasında, Azure Machine Learning, Azure Bot Service ve Azure Cognitive Services gibi özellikler bulunmaktadır.

Azure bağlantısı için internet bağlantısı gerektiğinden internet bağlantı kalitesine göre gecikme oranları değişiklik göstermektedir. Akıllı asistanın ilerleyen aşamalarında Azure ile bağlantısı tamamen kesilip, kendi ses TTS' ini kullanması ve internet bağlantısı gerekmeden kullanılması amaçlanabilmektedir. Bu sayede internet bağlantısına bağlı gecikmelerden, internet gerekliliğinden ve asistanın Azure ile yaptığı veri alışverişini kesmesi beklenmektedir. Tamamen kişiselleştirilmiş bir asistan yapısını oluşturmak zaman ilerledikçe ve kullanıcı beklentileri kayıt edildikçe ilerleyen bir süreçtir.

Azure'a Python ile bağlanmak için aşağıdaki adımları izleyebilirsiniz:

1. Azure Portal'ına giriş yapın.
2. Sol menüden "Tüm hizmetler"i seçin ve "API Anahtarları"nı bulun.
3. "Yeni anahtar oluştur" düğmesine tıklayın ve anahtar için bir ad girin.

4. Anahtar için bir geçerlilik süresi seçin ve "Oluştur" düğmesine tıklayın.

5. Oluşturulan anahtarınızı kopyalayın ve güvenli bir yerde saklayın.

6. Daha sonra, Python kodunuzda Azure API anahtarınızı kullanarak Azure hizmetlerine bağlanabilirsiniz.

Python'da Azure bağlantısını yapmak için kullanılan örnek kod Şekil 1'de gösterilmektedir.

```python
import azure.cognitiveservices.speech as speechsdk
from Config.c_key import *


speech_config = speechsdk.SpeechConfig(subscription=speech_key, region=service_region)
audio_config = speechsdk.audio.AudioConfig(use_default_microphone=True)

speech_config.speech_recognition_language="tr-TR"
speech_config.speech_synthesis_voice_name = "tr-TR-AhmetNeural" #Emel #Ahmet

speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config)
speech_recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config, audio_config=audio_config)
```

Şekil 1. *Python'da Azure Bağlantısını Yapmak İçin Kullanılan Kod.*

Asistanlar, mikrofondan aldıkları veriyi metin haline çevirir ve daha sonra bu metinleri komutlar olarak kullanarak işlemler gerçekleştirirler. Bu işlemler, insanların söylediklerine göre değişebilir ve çeşitli görevleri yerine getirebilirler. Örneğin, bir asistan, bir kişinin "sıcaklık ne kadar" diye sormasını anlayarak, o anki hava sıcaklığını söyleyebilir. Bu işlemleri gerçekleştirirken, asistanların ana Python dosyasında sonsuz bir while döngüsü içerisinde takeCommand() fonksiyonunu kullanarak mikrofonu sürekli dinlemesi sağlanmaktadır. Bu sayede, asistan her zaman komutları almaya hazır olur ve kullanıcıların isteklerini anında yerine getirir. Mikrofondan alınan veriyi metin haline çevirmek için aşağıdaki gibi bir kod kullanılabilir:

```
import speech_recognition as sr

def takeCommand():
    # Mikrofonunuzu tanımlayın
    mic = sr.Microphone()

    # SpeechRecognition sınıfını kullanarak bir dinleyici oluşturun
    r = sr.Recognizer()

    # Mikrofondan veri alın
    with mic as source:
        print("Listening...")
        audio = r.listen(source)

    # Veriyi metin haline çevirin
    try:
        text = r.recognize_google(audio)
        print(text)
    except sr.UnknownValueError:
        print("Sorry, I didn't understand what you said.")
        return None
    except sr.RequestError as e:
        print("Error occurred: {0}".format(e))
        return None

    return text

# Sonsuz bir döngü oluşturun
while True:
    # Komutu alın
    command = takeCommand()
    # Komutu işleyin
    processCommand(command)
```

Bu kod, sonsuz bir while döngüsü içerisinde takeCommand() fonksiyonunu kullanarak mikrofonu sürekli dinlemektedir.

Şekil 2'de bu proje için mikrofondan alınan veriyi metin haline çeviren kod bulunmaktadır.

```python
import azure.cognitiveservices.speech as speechsdk
from Config.s_config import *

def takeCommand():

    print("Dinleniyor...")
    query = speech_recognizer.recognize_once_async().get()
    if query.reason == speechsdk.ResultReason.RecognizedSpeech:
        print("Anlaşılan: {}".format(query.text))
    elif query.reason == speechsdk.ResultReason.NoMatch:
        print("{} Komutlarımda mevcut değil".format(query.no_match_details))
    elif query.reason == speechsdk.ResultReason.Canceled:
        cancellation_details = query.cancellation_details
        print("Konuşma tanıma iptal edildi: {}".format(cancellation_details.reason))
        if cancellation_details.reason == speechsdk.CancellationReason.Error:
            print("Hata Detayları: {}".format(cancellation_details.error_details))
            print("Azure key yada konum hatalı.")
    return query
```

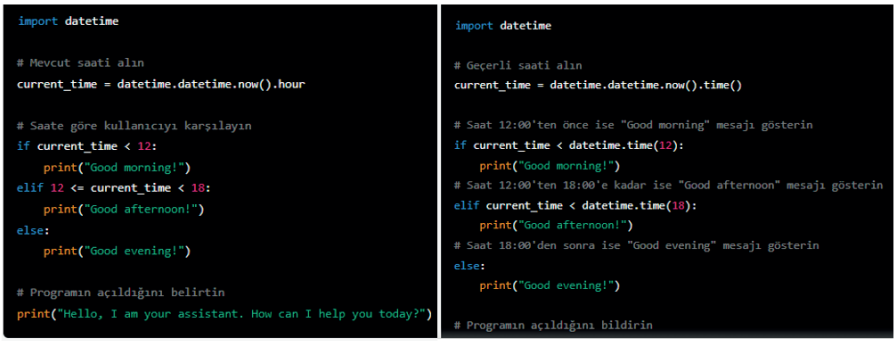
Şekil2. Mikrofondan Alınan Veriyi Metin Haline Çeviren Kod.

Şekil 3'te proje için geliştirilen sonsuz bir döngü içerisinde mikrofondan verinin alındığını gösteren kod örneği bulunmaktadır.

```python
import azure.cognitiveservices.speech as speechsdk
from Config.s_config import *
from Config.c_key import *
from Commands.OnHour import *
from Commands.takeCmd import *
from Commands.Ysearcher import *
from Commands.Gsearcher import *
from Commands.Wopen import *

if __name__ == '__main__':

    wishMe()

    while True:
        query = takeCommand()
```

Şekil 3. Sonsuz Bir Döngü İçerisinde Mikrofondan Verinin Alındığını Gösteren Kod.

Bir asistan programının açılışında, kullanıcıyı karşılamak ve programın açıldığını belirtmek için kullanılabilecek bir kod bloğu bulunmaktadır. Bu kod bloğu, mevcut saate göre kullanıcıyı karşılayabilir ve programın açıldığını belirtmek için kullanılmaktadır. Örneğin, bir asistan programı açılışında Şekil 4'teki gibi kod blokları kullanılabilir.



```
import datetime

# Mevcut saati alın
current_time = datetime.datetime.now().hour

# Saate göre kullanıcıyı karşılayın
if current_time < 12:
    print("Good morning!")
elif 12 <= current_time < 18:
    print("Good afternoon!")
else:
    print("Good evening!")

# Programın açıldığını belirtin
print("Hello, I am your assistant. How can I help you today?")
```

```
import datetime

# Geçerli saati alın
current_time = datetime.datetime.now().time()

# Saat 12:00'ten önce ise "Good morning" mesajı gösterin
if current_time < datetime.time(12):
    print("Good morning!")
# Saat 12:00'ten 18:00'e kadar ise "Good afternoon" mesajı gösterin
elif current_time < datetime.time(18):
    print("Good afternoon!")
# Saat 18:00'den sonra ise "Good evening" mesajı gösterin
else:
    print("Good evening!")

# Programın açıldığını bildirin
```

Şekil 4. Programı Açılışında Asistanın Kullanıcıyı Karşılaması.

```
import datetime
# Geçerli saati alın
current_time = datetime.datetime.now().time()
# Saat 12:00'ten önce ise "Good morning" mesajı gösterin
if current_time < datetime.time(12):
    print("Good morning!")
# Saat 12:00'ten 18:00'e kadar ise "Good afternoon" mesajı gösterin
elif current_time < datetime.time(18):
    print("Good afternoon!")
# Saat 18:00'den sonra ise "Good evening" mesajı gösterin
else:
    print("Good evening!")
# Programın açıldığını bildirin
print("Assistant is now ready to serve you.")
```

Bu kod blokları, mevcut saate göre kullanıcıyı karşılamakta ve programın açıldığını belirtmektedir. Bu sayede, kullanıcı asistan programı ile iletişim kurabilmektedir. Bu proje Yapay Zeka Destekli Kişisel Akıllı Asistan için yazılan ilgili kod Şekil 5'te gösterilmektedir.

```python
import azure.cognitiveservices.speech as speechsdk
import datetime
from Config.s_config import *
import socket

def wishMe():

    hour = int(datetime.datetime.now().hour)
    if hour >6 and hour <12:
        text ="Günaydın"
        speech_synthesizer.speak_text_async(text).get()
    elif hour >=12 and hour <18:
        text = "Tünaydın!"
        speech_synthesizer.speak_text_async(text).get()
    elif hour >=18 and hour < 23:
        text = "iyi Akşamlar"
        speech_synthesizer.speak_text_async(text).get()
    else:
        text = "İyi Geceler"
        speech_synthesizer.speak_text_async(text).get()
        print(text)
    text = "Merhaba,"+ socket.gethostname() + " Sana nasıl yardımcı olabilirim?"
    print(text)
    speech_synthesizer.speak_text_async(text).get()

```

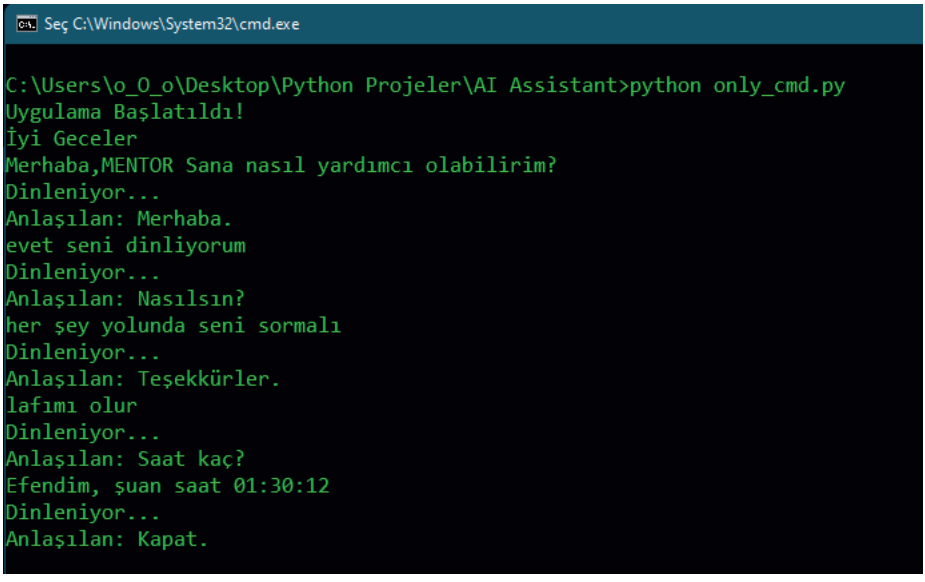
Şekil 5. Saate Göre Kullanıcıyı Karşılamak İçin Gerekli Kod.

Program sistem adını kullanarak kullanıcıyı karşılamakta ve mikrofondan veri almaya başlamaktadır. Örnek olarak, aşağıdaki kod kullanılabilir:

```
import platform
import speech_recognition as sr

# Sistem adını alın
system_name = platform.system()

def takeCommand():
    # Mikrofonunuzu tanımlayın
    mic = sr.Microphone()

    # SpeechRecognition sınıfını kullanarak bir dinleyici oluşturun
    r = sr.Recognizer()

    # Kullanıcıyı karşılayın
    print("Hello, I am your {} assistant.".format(system_name))

    # Mikrofondan veri alın
    with mic as source:
        print("Listening...")
        audio = r.listen(source)

    # Veriyi metin haline çevirin
    try:
        text = r.recognize_google(audio)
```

```
            print(text)
        except sr.UnknownValueError:
            print("Sorry, I didn't understand what you said.")
            return None
        except sr.RequestError as e:
            print("Error occurred: {0}".format(e))
            return None

        return text

    # Sonsuz bir döngü oluşturun
    while True:
        # Komutu alın
        command = takeCommand()
        # Komutu işleyin
        processCommand(command)
```

Şekil 6'da bu projedeki Yapay Zeka Destekli Kişisel Akıllı Asistanın terminal görüntüsü görüntülenmektedir.



Şekil 6. Yapay Zeka Destekli Kişisel Akıllı Asistanın Terminal Görüntüsü.

Resim 5'te, terminalde görüldüğü gibi asistan aldığı verileri kontrol edip komutlarına göre kullanıcıya sesli olarak dönüş yapmaktadır. Pythonun random kütüphanesinden faydalanılarak tek bir cümle ile kısıtlamayıp her ko-

mutta farklı tepkiler vermesi sağlanmaktdır. Azure bağlantısı için internet bağlantısı gerektiğinden internet bağlantı kalitesine göre gecikme oranları değişiklik göstermektedir.

7. Sonuç

Bugün direkt olarak kişisel asistanımız olmasa da herhangi bir şekilde hepimiz akıllı asistan çözümlerinden faydalanmaktayız. En çok tercih edilen yapay zeka asistanlarna Siri, Alexa, Cortana, Google Asistan, Elsa Speak gibi uygulamaları gösterebiliriz. Yapay zeka asistanlarının kişiselleştirilmesi, daha fazla veri toplandıkça ve kullanıcı beklentilerinin kaydedildiği sürece daha kolay hale gelebilir. Yapay zeka asistanları, makine öğrenimi ve doğal dil işleme gibi teknolojileri kullanarak kullanıcıların ihtiyaçlarına göre öğrenir ve daha ilgili ve yardımcı hale gelebilirler. Bu teknolojiler sayesinde, yapay zeka asistanları, gerçeğe yakın diyaloglar kurulmasına olanak sağlar ve kullanıcıların işlerini kolaylaştırır. Dijital platformlar üzerinde bugün yapabileceğiniz tüm işlemleri akıllı asistanlar ile yapabilirsiniz, web tarayıcınızda gerçekleştireceğiniz bir aramayı sesli olarak yapabilirsiniz veya akıllı bir eve sahipseniz evinizdeki teknolojik ürünlerin çoğunu akıllı asistanınız ile yönetebilirsiniz. Akıllı asistanlar, insan gibi düşünen, insan gibi davranan ve belirli bir mantık çerçevesinde kullanıcının komutlarına anında yanıt veren sistemlerdir. Bu sayede, makine-insan etkileşimi artmaktadır ve akıllı asistanlar, insan yaşamında etkin bir şekilde kullanılmaktadır.

Yapay zeka platformları makine öğrenme ve doğal dil işleme teknolojilerinden faydalanmaktadır. Bu teknolojileri geliştirebilmenin yolu daha fazla veriye sahip olmaktır, şuan Google, Cortana, Alexa, Siri gibi asistanlar arkasındaki veri arttıkça durmadan kendini geliştiren, geliştirilen asistanlardır. Yapay zeka alanlarına olan ilgi arttıkça, yapay zeka destekli akıllı asistanlarını geliştirilmesi merak edilen ve ilgi duyulan bir süreç haline gelmiştir. Geliştirilen uygulamaların yazılım geliştirme süreci büyük önem taşımaktadır. Bu süreçleri kontrol edebilmek ve en iyi sonuçları alabilmek için uygulama geliştirme aşamalarının detaylı bir şekilde anlatılması gerekmektedir. Bu alanda yapılan çalışmaların büyük çoğunluğu yazılım geliştirme süreçlerinin işleyişine ilişkin sonuçlar vermektedir.

Bu uygulamaların nasıl geliştirildiğini anlatan çalışmaların azlığı dikkat çekicidir. Ancak yapay zeka destekli kişisel asistan uygulama yazılımlarının gelişiminin anlatıldığı platformların zayıf olduğu ve bu konuda ilerleyişi ile ilgili kabul gören ifadelerin yeterli olmadığı bilinmektedir. Araştırma yöntemlerinde hala eksiklikler olduğuna dikkat çekmek ve aynı zamanda yapay zeka destekli akıllı asistan geliştirme sürecinde ortaya çıkabilecek endişeleri ve zorlukları belirlemek gerekmektedir. Yazılım geliştirme genellikle soyut öneriler getirdiği için yazılımın geliştirme sürecini olduğu gibi yansıtmak gerekir.

Bu çalışmada yapay zekâ tabanlı doğal dil işleme uygulamalarının (NLP Natural Language Processing) bir uzantısı olarak kişilerin özel hayatlarında destek sağlayabilecek bir öneri olarak akıllı asistan geliştirilmiştir. Bu çalışmada, tanınmış asistanlardan farklı tamamen kişiselleştirilmiş bir ürün ortaya çıkartılmıştır. Bir asistandan ziyade daha çok kullanıcı ile makine arasındaki bağı güçlendiren bir sistem geliştirilmiştir.

Yapay zeka destekli kişisel akıllı asistan geliştirilirken, işlemler için farklı, arayüz için farklı dil kullanılmıştır. Akıllı asistanın arka plan işlemleri için Python dili kullanılırken, kullanıcı arayüzü için C# dili kullanılmıştır. Bu sayede Python kütüphaneleri ile tasarlanabilen eski arayüzlere göre daha modern bir arayüz elde edilip, arka plan işlemlerinin gecikmesi daha aza indirilmiştir.

Yapay Zeka Destekli Kişisel Akıllı Asistan geliştirilmesi için Azure alt yapısı kullanılmıştır. Asistanın temelde yaptığı şey mikrofondan aldığı veriyi metin haline çevirmektir. Daha sonrasında komutlarda bu veri mevcut ise işlemi gerçekleştirip yeni komut için başa dönmektir. Sonsuz bir döngü içerisinde mikrofondan verinin alındığı gösterilmiştir. Programın açılışında kullanıcıyı karşılamak için ufak bir kod bloğu bulunmaktadır. Mevcut saate göre kullanıcıyı karşılamak ve programın açıldığını belirtmek için kullanılmıştır. Program sistem adını kullanarak kullanıcıyı karşılamakta ve mikrofondan veri almaya başlamaktadır. Terminalde, yapay zeka destekli akıllı asistan aldığı verileri kontrol edip komutlarına göre kullanıcıya sesli olarak dönüş yapmaktadır. Pythonun random kütüphanesinden faydalanılarak tek bir cümle ile kısıtlamayıp her komutta farklı tepkiler vermesi sağlanmaktadır. Azure bağlantısı için internet bağlantısı gerektiğinden internet bağlantı kalitesine göre gecikme oranları değişiklik göstermektedir. Yapay zeka destekli akıllı asistanlar üzerinde çalışan ve gelişitirmek isteyen yazılımcılar Azure ile bağlantısını tamamen kesip, kendi ses TTS' ini kullanması ve internet bağlantısı gerekmeden kullanılması yapabileceklerdir. Bu sayede internet bağlantısına bağlı gecikmelerden, internet gerekliliğinden ve asistanın Azure ile yaptığı veri alışverişini kesmesi beklenmektedir.

KAYNAKÇA

Adam M, Toutaoui J, Pfeuffer N, Hinz O (2019) Investment decisions with robo-advisors: the role of anthropomorphism and personalized anchors in recommendations. In: Proceedings of the 27th European Conference on Information Systems (ECIS 2019). https://aisel.aisnet.org/ecis2019_rp/33/

Ahmed, I., Jeon, G., & Piccialli, F. (2022). From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where. *IEEE Transactions on Industrial Informatics*, *18*(8), 5031-5042.

Ang, J. H., Goh, C., Saldivar, A. A. F., & Li, Y. (2017). Energy-efficient through-life smart design, manufacturing and operation of ships in an industry 4.0 environment. *Energies*, *10*(5), 610.

Baryannis, G., Validi, S., Dani, S., & Antoniou, G. (2019). Supply chain risk management and artificial intelligence: state of the art and future research directions. *International Journal of Production Research*, *57*(7), 2179-2202.

Benlian, A., Klumpe, J., & Hinz, O. (2020). Mitigating the intrusive effects of smart home assistants by using anthropomorphic design features: A multimethod investigation. *Information Systems Journal*, *30*(6), 1010-1042. https://doi.org/10.1111/isj.12243

Bose, B. K. (2017). Artificial intelligence techniques in smart grid and renewable energy systems—Some example applications. *Proceedings of the IEEE*, *105*(11), 2262-2273.

Bostrom, N., & Yudkowsky, E. (2018). The ethics of artificial intelligence. In *Artificial intelligence safety and security* (pp. 57-69). Chapman and Hall/CRC.

Bougdira, A., Akharraz, I., & Ahaitouf, A. (2020). A traceability proposal for industry 4.0. *Journal of Ambient Intelligence and Humanized Computing*, *11*(8), 3355-3369.

Cai-Ming, Z., & Hao-Nan, C. (2020, December). Preprocessing method of structured big data in human resource archives database. In *2020 IEEE International Conference on Industrial Application of Artificial Intelligence (IAAI)* (pp. 379-384). IEEE.

Canbek, N. G., & Mutlu, M. E. (2016). On the track of artificial intelligence: Learning with intelligent personal assistants. *Journal of Human Sciences*, *13*(1), 592-601.

Castro, D., & New, J. (2016, Ekim 10). https://datainnovation.org/2016/10/the-promise-of-artificialintelligence/. https://datainnovation.org/: https://datainnovation.org/2016/10/the-promise-of-artificialintelligence/, (Erişim tarihi: 7 Mart 2022)

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).

Cowan, B. R., Pantidi, N., Coyle, D., Morrissey, K., Clarke, P., Al-Shehri, S., ... & Bandeira, N. (2017, September). " What can i help you with?" infrequent users' experiences of intelligent personal assistants. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (pp. 1-12).

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). *Machine learning in Finance* (Vol. 1406). New York, NY, USA: Springer International Publishing.

Erhan, D., Courville, A., Bengio, Y., & Vincent, P. (2010, March). Why does unsupervised pre-training help deep learning?. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 201-208). JMLR Workshop and Conference Proceedings.

Fryer, L. K., Ainley, M., Thompson, A., Gibson, A., & Sherlock, Z. (2017). Stimulating and sustaining interest in a language course: An experimental comparison of Chatbot and Human task partners. *Computers in Human Behavior*, *75*, 461-468.

Gnewuch, U., Morana, S., & Maedche, A. (2017, December). Towards Designing Cooperative and Social Conversational Agents for Customer Service. In *ICIS*.

Goodell, J. W., Kumar, S., Lim, W. M., & Pattnaik, D. (2021). Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*, *32*, 100577.

Göksel-Canbek, N., & Mutlu, M. E. (2016). Sayısal gelecekte yeni adım: akıllı kişisel yardımcılar. *Açıköğretim Uygulamaları ve Araştırmaları Dergisi*, *2*(1), 114-129.

Gures, E., Shayea, I., Ergen, M., Azmi, M. H., & El-Saleh, A. A. (2022). Machine Learning Based Load Balancing Algorithms in Future Heterogeneous Networks: A Survey. *IEEE Access*.

Gür, Y. E., Ayden, C., & Yücel, A. (2019). Yapay zekâ alanindaki gelişmelerin insan kaynaklari yönetimine etkisi. *Fırat Üniversitesi Uluslararası İktisadi ve İdari Bilimler Dergisi*, *3*(2), 137-158.

Huang, C., Cai, H., Xu, L., Xu, B., Gu, Y., & Jiang, L. (2019). Data-driven ontology generation and evolution towards intelligent service in manufacturing systems. *Future Generation Computer Systems*, *101*, 197-207

Jung, D., Dorner, V., Glaser, F., & Morana, S. (2018). Robo-advisory: digitalization and automation of financial advisory. Business and Information Systems Engineering 60 (1): 81–86. *DOI: https://doi. org/10.1007/s12599-018-0521-9*.

Kang, Y., Cai, Z., Tan, C. W., Huang, Q., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics*, *7*(2), 139-172.

Karakuş, C. (2021). Makine Öğrenmesi Temelleri Ders Notu, Çalışma Notları, pp. 1–343, 2021, [Online]. Available: https://ckk.com.tr/ders/ML/ML 00 Makine Öğrenmesi Ders Notu.html

Khaokaew, Y., Holcombe-James, I., Rahaman, M. S., Liono, J., Trippas, J. R., Spina, D., ... & Salim, F. D. (2022). Imagining future digital assistants at work: A study of task management needs. *International Journal of Human-Computer Studies*, *168*, 102905.

Kızılkaya, Y. M., & Oğuzlar, A. (2018). Bazı Denetimli Öğrenme Algoritmalarının R Programlama Dili İle Kıyaslanması. *Karadeniz Uluslararası Bilimsel Dergi*, *37*(37), 90-98.

Kişi, N.(2021). Yapay Zeka Çağında Değişen Liderlik Anlayışı. Erişim Tarihi:15.12.2022 Erişim Adresi: https://sadabsempozyum.org/sadabantalya/wp-content/uploads/2021/07/Ornek-Bildiri-Tam-Metin.docx

Kumar, N., Srivastava, J.D., Bisht, H., 2019. Artificial Intelligence in Insurance Sector 21, 79–91.

Laranjo, L., Dunn, A. G., Tong, H. L., Kocaballi, A. B., Chen, J., Bashir, R., ... & Coiera, E. (2018). Conversational agents in healthcare: a systematic review. *Journal of the American Medical Informatics Association*, *25*(9), 1248-1258.

Li, L., Warfield, J., Guo, S. J., Guo, W. D., & Qi, J. Y. (2007). Introduction: Advances in intelligent information processing. *Information Systems*, *32*(7), 941-943.

Lu, Y. (2019). Artificial intelligence: a survey on evolution, models, applications and future trends. *Journal of Management Analytics*, *6*(1), 1-29.

Maedche, A., Legner, C., Benlian, A., Berger, B., Gimpel, H., Hess, T., ... & Söllner, M. (2019). AI tabanlı dijital asistanlar. *İşletme ve Bilgi Sistemleri Mühendisliği* , *61* (4), 535-544.

Mihale-Wilson, A. C., Zibuschka, J., & Hinz, O. (2019). User preferences and willingness to pay for in-vehicle assistance. *Electronic Markets*, *29*(1), 37-53.

Morocho-Cayamcela, M. E., Lee, H., & Lim, W. (2019). Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions. *IEEE access*, *7*, 137184-137206.

Neyshabur, B., Bhojanapalli, S., McAllester, D., & Srebro, N. (2017). Exploring generalization in deep learning. *Advances in neural information processing systems*, *30*.

Nilsson, N. J. (1982). *Principles of artificial intelligence*. Springer Science & Business Media.

Oxborough, C. Cameron, E. and Rao, A. (2017). Explainable AI Driving business value through greater understanding, PwC, pp. 1–26, 2017.

Öztürk, K., & Şahin, M. E. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. *Takvim-i Vekayi*, *6*(2), 25-36.

Qi, J., Wu, F., Li, L., & Shu, H. (2007). Artificial intelligence applications in the telecommunications industry. *Expert Systems*, *24*(4), 271-291.

Qiu, L., & Benbasat, I. (2009). Evaluating anthropomorphic product recommendation agents: A social relationship perspective to designing information systems. *Journal of management information systems*, *25*(4), 145-182.

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., ... & Dean, J. (2018). Scalable and accurate deep learning with electronic health records. *NPJ digital medicine*, *1*(1), 1-10.

Seker, S. E. (2015). Doğal Dil İşleme (Natural Language Processing). *YBS Ansiklopedi*, *2*(4), 14-31.

Shi, Z., Huang, Y., He, Q., Xu, L., Liu, S., Qin, L., ... & Zhao, L. (2007). MSMiner—a developing platform for OLAP. *Decision Support Systems*, *42*(4), 2016-2028.

Tuba, A. & Delice, E. K. (2019). A literature review on the use of machine learning algorithms in health. 4th Int. energy Eng. Congr., pp. 928–956, 2019.

Wang, L., Zou, H., Su, J., Li, L., & Chaudhry, S. (2013). An ARIMA- ANN hybrid model for time series forecasting. *Systems Research and Behavioral Science*, *30*(3), 244-259.

Xu, L., Tan, W., Zhen, H., & Shen, W. (2008). An approach to enterprise process dynamic modeling supporting enterprise process evolution. *Information Systems Frontiers*, *10*(5), 611-624.

Zhang, C., Xu, X., & Chen, H. (2020). Theoretical foundations and applications of cyber-physical systems: a literature review. *Library Hi Tech*.