NESNELERİN İNTERNETİ

TEMELLER, TEKNOLOJİLER VE AKILLI UYGULAMALAR

EDİTÖRLER

DOÇ. DR. EMREHAN YAVŞAN

DR. ÖĞR. ÜYESİ MEHMET ALİ ŞİMŞEK

DR. ERHAN KAHYA

EĞİTİM yayınevi

NESNELERİN İNTERNETİ: TEMELLER, TEKNOLOJİLER VE AKILLI UYGULAMALAR

Editörler: Doç. Dr. Emrehan Yavşan, Dr. Öğr. Üyesi Mehmet Ali Şimşek, Dr. Erhan Kahya

Yayınevi Grubu Genel Başkanı: Yusuf Ziya Aydoğan (yza@egitimyayinevi.com)
Genel Yayın Yönetmeni: Yusuf Yavuz (yusufyavuz@egitimyayinevi.com)
Sayfa Tasarımı: Kübra Konca Nam
Kapak Tasarımı: Ceyda Duran

T.C. Kültür ve Turizm Bakanlığı
Yayıncı Sertifika No: 76780

E-ISBN: 978-625-5014-77-1
1. Baskı, Ekim 2026

Kütüphane Kimlik Kartı

NESNELERİN İNTERNETİ: TEMELLER, TEKNOLOJİLER VE AKILLI UYGULAMALAR
Editörler: Doç. Dr. Emrehan Yavşan, Dr. Öğr. Üyesi Mehmet Ali Şimşek, Dr. Erhan Kahya
E-ISBN: 978-625-5014-77-1
VI+271 s., 160x240 mm
Kaynakça var, dizin yok.

Copyright © Bu kitabın Türkiye'deki her türlü yayın hakkı Eğitim Yayınevi'ne aittir. Bütün hakları saklıdır. Kitabın tamamı veya bir kısmı 5846 sayılı yasanın hükümlerine göre kitabı yayımlayan firmanın ve yazarlarının önceden izni olmadan elektronik/mekanik yolla, fotokopi yoluyla ya da herhangi bir kayıt sistemi ile çoğaltılamaz, yayımlanamaz.

Yayınevi Türkiye Ofis:
Konya: Eğitim Yayınevi Tic. Ltd. Şti., Fevzi Çakmak Mah. 10721 Sok. B Blok, No: 16/B, Safakent, Karatay, Konya, Türkiye
İstanbul: Salon Yayınları, Atakent mah., Yasemen sok., No: 4/B, Ümraniye, İstanbul, Türkiye
Santral: +90 332 351 92 85
Editör hatları: +90 533 151 50 42, +90 507 151 50 43
bilgi@egitimyayinevi.com

Yayınevi Amerika Ofis: New York: Egitim Publishing Group, Inc.
P.O. Box 768/Armonk, New York, 10504-0768, United States of America
americaoffice@egitimyayinevi.com

Lojistik ve Sevkiyat Merkezi: Kitapmatik Lojistik ve Sevkiyat Merkezi, Fevzi Çakmak Mah. 10721 Sok. B Blok, No: 16/B, Safakent, Karatay, Konya, Türkiye
İnternet Satış: www.kitapmatik.com.tr
Whatsapp hattı: +90 553 950 50 37
bilgi@kitapmatik.com.tr

Kitabevi Şubesi: Eğitim Kitabevi, Şükran mah. Rampalı 121, Meram, Konya, Türkiye
Whatsapp hattı: +90 501 651 92 85
bilgi@egitimkitabevi.com

İÇİNDEKİLER

ÖNSÖZ IV

NESNELERİN İNTERNETİNE GİRİŞ VE TEMEL KAVRAMLAR 1

Yücel Yılmaz

IOT SİSTEM MİMARİLERİ 17

Ali Topal, Muzaffer Tatlı

SENSÖRLER, AKTÜATÖRLER VE GÖMÜLÜ SİSTEMLER 42

Ayhan Arısoy

OTONOM ARAÇLARDA DDOS SALDIRILARIN ENGELLENMESİ İÇİN YAPAY ZEKA DESTEKLİ SALDIRI TESPİT SİSTEMİN GELİŞTİRİLMESİ 83

İsa Tuncer, Ahmet Ali Süzen

IOT VERİLERİNİN TOPLANMASI, SAKLANMASI VE YÖNETİMİ..... 107

Oğuzhan Kilim, Cem Deniz Kumral

BULUT DESTEKLİ NESNELERİN İNTERNETİ SİSTEMLERİ 131

Enes Açıkgözoğlu, Serkan Keskin

EDGE VE FOG COMPUTING İLE GERÇEK ZAMANLI İŞLEME 157

Serkan Keskin, Enes Açıkgözoğlu

IOT GÜVENLİĞİ VE MAHREMİYET 179

Recep Çolak, Cem Deniz Kumral

IOT' DE YAPAY ZEKA VE VERİ ANALİTİĞİ 210

Ali Topal, İbrahim Arda Çankaya

AKILLI IOT UYGULAMALARI VE GELECEK EĞİLİMLERİ..... 241

Muzaffer Tatlı, İbrahim Arda Çankaya

ÖNSÖZ

Dijital dönüşümün giderek hız kazandığı günümüzde, fiziksel sistemler ile dijital teknolojiler arasındaki etkileşim yeni bir teknolojik ekosistemin oluşmasına zemin hazırlamaktadır. Bu dönüşümün temel bileşenlerinden biri olan Nesnelerin İnterneti (Internet of Things – IoT); sensörler, gömülü sistemler, haberleşme teknolojileri, bulut ve uç bilişim, veri analitiği, siber güvenlik ve yapay zekâ gibi farklı disiplinleri bütünleşik bir yapı içerisinde bir araya getirmektedir. IoT teknolojilerinin sağladığı sürekli veri üretimi, bağlantı ve akıllı karar verme yetenekleri; endüstriden tarıma, sağlıktan enerjiye, ulaşımdan akıllı şehirlere kadar pek çok alanda yenilikçi uygulamaların geliştirilmesine olanak sağlamaktadır.

IoT kavramı, yalnızca fiziksel cihazların internet üzerinden birbirine bağlanmasıyla sınırlı değildir. Modern IoT sistemleri; çevreden verinin algılanması, güvenilir biçimde iletilmesi, depolanması ve işlenmesinin yanı sıra elde edilen bilgilerin analiz edilerek karar destek ve otonom kontrol süreçlerine dönüştürülmesini kapsayan çok katmanlı bir yapıya sahiptir. Bu nedenle IoT'nin anlaşılması; donanım, yazılım, haberleşme, veri yönetimi ve güvenlik bileşenlerinin birbirlerinden bağımsız olarak değil, bütüncül bir sistem yaklaşımı içerisinde değerlendirilmesini gerektirmektedir.

Bu kitap, söz konusu gereksinimden hareketle Nesnelerin İnternetinin temel ilkelerinden güncel teknolojilerine ve uygulama alanlarına uzanan kapsamlı bir çerçeve sunmak amacıyla hazırlanmıştır. Eserin temel yaklaşımı, IoT sistemlerinin kuramsal altyapısını açıklamanın yanı sıra bu sistemleri oluşturan teknolojik bileşenler arasındaki ilişkileri ortaya koymak ve okuyucuya güncel gelişmeleri değerlendirebileceği sistematik bir bakış açısı kazandırmaktır.

Kitabın ilk bölümlerinde Nesnelerin İnternetinin kavramsal temelleri, tarihsel gelişimi ve temel sistem mimarileri ele alınmaktadır. Ardından sensörler, aktüatörler ve gömülü sistemler üzerinden IoT'nin fiziksel katmanını oluşturan teknolojiler incelenmektedir. Cihazlar ve sistemler arasındaki veri alışverişinin temelini oluşturan haberleşme altyapısı kapsamında Wi-Fi, Bluetooth, ZigBee, LoRaWAN ve NB-IoT gibi teknolojilerin yanı sıra MQTT, CoAP, HTTP ve TCP/IP gibi iletişim protokolleri değerlendirilmektedir.

IoT ekosisteminde üretilen verinin hacmi, çeşitliliği ve üretim hızı, etkin veri yönetimi ve ölçeklenebilir bilişim altyapılarını önemli bir gereksinim

hâline getirmiştir. Bu doğrultuda kitapta, IoT verilerinin toplanması, saklanması, işlenmesi ve yönetilmesi bulut bilişim yaklaşımlarıyla birlikte ele alınmaktadır. Bunun yanında, verinin kaynağına daha yakın noktalarda işlenmesini sağlayan edge ve fog computing yaklaşımları; düşük gecikme, gerçek zamanlı karar verme, bant genişliğinin etkin kullanımı ve sistem verimliliği açısından incelenmektedir.

IoT sistemlerinin yaygınlaşmasıyla birlikte güvenlik, mahremiyet ve veri bütünlüğü de sistem tasarımının ayrılmaz unsurları hâline gelmiştir. Çok sayıda heterojen cihazın aynı ağ içerisinde iletişim kurduğu yapılarda ortaya çıkabilecek güvenlik açıkları; yalnızca bilgi sistemlerini değil, fiziksel süreçleri de doğrudan etkileyebilmektedir. Bu çerçevede eserde kimlik doğrulama, şifreleme, cihaz ve ağ güvenliği, veri gizliliği ve güvenli yazılım geliştirme yaklaşımlarına yer verilerek güvenilir IoT sistemlerinin oluşturulmasına ilişkin temel ilkeler ele alınmaktadır.

Kitabın ilerleyen bölümlerinde ise IoT teknolojilerinin yapay zekâ ve veri analitiğiyle bütünleşmesi üzerinde durulmaktadır. Sensör ve görüntü verilerinin analizi, anomali tespiti, sınıflandırma ve tahminleme gibi yöntemlerin yanı sıra federated learning gibi güncel yaklaşımlar, IoT sistemlerinin veri üreten yapılardan öğrenebilen, uyarlanabilen ve karar verebilen akıllı sistemlere dönüşümü bağlamında değerlendirilmektedir. Böylece IoT'nin yalnızca bağlantılı cihazlardan oluşan bir teknoloji alanı olmadığı; veri odaklı ve akıllı sistemlerin geliştirilmesinde temel bir altyapı niteliği taşıdığı ortaya konulmaktadır.

Eserin uygulama odaklı bölümlerinde ise IoT'nin akıllı şehirler, sağlık, tarım, Endüstri 4.0, enerji ve ulaşım gibi farklı alanlardaki kullanım biçimleri ele alınmaktadır. Ayrıca dijital ikizler, sürdürülebilirlik ve etik gibi geleceğin IoT ekosistemini şekillendirmesi beklenen konulara yer verilerek teknolojinin yalnızca teknik boyutuyla değil, toplumsal ve çevresel etkileriyle birlikte değerlendirilmesi amaçlanmaktadır.

Bu kapsamıyla kitap; başta bilgisayar mühendisliği, yazılım mühendisliği, elektrik-elektronik mühendisliği ve ilgili disiplinlerde öğrenim gören lisans ve lisansüstü öğrenciler olmak üzere akademisyenler, araştırmacılar, mühendisler ve IoT alanında çalışan uygulayıcılar için akademik bir başvuru kaynağı olmayı hedeflemektedir. Farklı uzmanlık alanlarından araştırmacıların katkılarıyla oluşturulan bölümlerin, okuyuculara IoT teknolojilerinin temel bileşenlerini kavrama, bu bileşenler arasındaki ilişkileri değerlendirme ve güncel teknolojik gelişmeleri

disiplinler arası bir perspektifle yorumlama konusunda katkı sağlayacağı düşünülmektedir.

Teknolojik gelişmelerin hızla devam ettiği ve bağlantılı akıllı sistemlerin günlük yaşamdan endüstriyel süreçlere kadar giderek daha geniş bir alanda yer aldığı günümüzde, IoT alanındaki bilgi birikiminin sistematik ve güncel bir çerçevede sunulması önem taşımaktadır. Bu eserin, mevcut bilgi birikiminin aktarılmasının yanı sıra yeni araştırma sorularının geliştirilmesine, disiplinler arası çalışmaların teşvik edilmesine ve yenilikçi IoT uygulamalarının ortaya çıkmasına katkı sağlamasını temenni ediyoruz.

Kitabın hazırlanma sürecinde bilimsel bilgi, deneyim ve birikimleriyle katkıda bulunan tüm bölüm yazarlarına ve eserin yayıma hazırlanmasında emeği geçen herkese teşekkür ederiz.

Editörler

Doç. Dr. Emrehan YAVŞAN

Dr. Öğr. Üyesi Mehmet Ali ŞİMŞEK

Dr. Erhan KAHYA

NESNELERİN İNTERNETİNE GİRİŞ VE TEMEL KAVRAMLAR

Yücel Yılmaz*

NESNELERİN İNTERNETİNİN TANIMI

Nesnelerin İnterneti, fiziksel dünyada yer alan nesnelerin tanımlanabilir, izlenebilir, veri üretebilir ve ağ üzerinden diğer sistemlerle etkileşime girebilir hâle getirilmesini ifade eden dağıtık bir bilişim paradigmasıdır. Bu yaklaşımda nesneler yalnızca pasif fiziksel varlıklar olarak değerlendirilmez; algılama, veri üretme, iletişim kurma ve belirli ölçüde karar süreçlerine katkı sağlama yetenekleriyle dijital sistemlerin etkin bileşenleri hâline gelir. Literatürde IoT kavramı, tanımlama teknolojileri, algılama mekanizmaları, haberleşme altyapıları, veri işleme yaklaşımları ve uygulama servislerinin bütünleşik biçimde ele alındığı çok katmanlı bir sistem yaklaşımı olarak açıklanmaktadır (Atzori vd., 2010; Miorandi vd., 2012).

Teknik açıdan IoT'nin temel ayırt edici yönü, fiziksel dünya ile sayısal sistemler arasında sürekli bir veri ve kontrol ilişkisi kurmasıdır. Geleneksel bilgi sistemlerinde veri çoğunlukla kullanıcı girdileri veya kurumsal işlem kayıtları üzerinden oluşurken, IoT sistemlerinde veri fiziksel çevredeki durumların ölçülmesiyle üretilir. Bu nedenle IoT; algılama, sayısallaştırma, ağ üzerinden aktarım, işleme, karar üretme ve geri bildirim adımlarından oluşan uçtan uca bir sistem davranışı sergiler. Gubbi vd. (2013), IoT'nin gelişimini kablosuz sensör ağları, gömülü sistemler ve dağıtık bilişim altyapılarının yakınsamasıyla açıklamakta; bu yakınsamanın çevresel durumların ölçülmesi, yorumlanması ve hizmete dönüştürülmesi açısından belirleyici olduğunu vurgulamaktadır.

Standartlaştırma literatüründe de IoT, yalnızca cihazların internete bağlanması olarak değil, fiziksel ve sanal nesnelerin bilgi ve iletişim teknolojileri aracılığıyla bütünleştirildiği geniş kapsamlı bir altyapı olarak ele alınmaktadır. ITU-T Y.2060/Y.4000 önerisi, IoT'nin kavram, kapsam, temel karakteristikler, üst düzey gereksinimler ve referans model düzeyinde tanımlanması gerektiğini belirtmektedir. ISO/IEC 30141:2024 ise IoT

* Süleyman Demirel Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, Isparta, Türkiye, Email: yucelyilmaz@sdu.edu.tr, Orcid Id: https://orcid.org/0000-0002-3975-2598

referans mimarisi için ortak terminoloji, yeniden kullanılabilir tasarım yaklaşımları ve endüstri uygulamalarına dayalı genel bir çerçeve sunmaktadır. Bu standartlaştırma yaklaşımı, IoT'nin teknoloji-bağımsız biçimde anlaşılması ve farklı sistemlerde ortak kavramlarla ele alınması açısından önemlidir.

Bu bağlamda IoT şu şekilde teknik olarak tanımlanabilir: Nesnelerin İnterneti, fiziksel ortamdaki nesnelerin benzersiz biçimde tanımlanabildiği, durum bilgisi üretebildiği, bu bilgiyi ağ üzerinden aktarabildiği, yazılım tabanlı sistemler tarafından işlenebildiği ve gerektiğinde fiziksel ortama geri bildirim sağlayabildiği dağıtık, heterojen ve veri odaklı bir bilişim ekosistemidir. Bu tanım, IoT'nin yalnızca bağlantı teknolojilerine indirgenemeyeceğini; nesne, veri, ağ, yazılım, karar ve kontrol bileşenlerinin birlikte ele alınması gerektiğini ortaya koymaktadır (Atzori vd., 2010; Whitmore vd., 2015, Eylence vd., 2022, Özsoy vd., 2020).

IOT'NİN TEMEL KAVRAMLARI VE TERMİNOLOJİSİ

IoT alanında kullanılan temel terminoloji, sistemin yalnızca donanım veya yalnızca yazılım boyutuyla açıklanmasını önler. Bu nedenle "nesne", "akıllı nesne", "algılama", "bağlanabilirlik", "kimliklendirme", "veri akışı", "olay", "bağlam", "otonomi" ve "geri bildirim" gibi kavramlar IoT'nin teknik dilini oluşturur. Atzori vd. (2010), IoT'nin üç ana perspektiften geliştiğini belirtmektedir: nesne odaklı yaklaşım, internet odaklı yaklaşım ve semantik odaklı yaklaşım. Bu üç perspektif, IoT terminolojisinin neden yalnızca cihaz bağlantısı üzerinden değil, veri anlamlandırma ve sistemler arası etkileşim üzerinden de tanımlanması gerektiğini göstermektedir.

IoT'de "nesne", fiziksel dünyada yer alan ve dijital sistem tarafından tanımlanabilen varlığı ifade eder. "Akıllı nesne" ise yalnızca tanımlanabilen değil, aynı zamanda veri üretebilen, sınırlı işlem yapabilen, ağ üzerinden etkileşime girebilen ve gerektiğinde fiziksel çevreye tepki verebilen nesnedir. Whitmore vd. (2015), IoT nesnelerini tanımlama, algılama, ağ bağlantısı ve işlem yapma yetenekleriyle donatılmış varlıklar olarak ele alır. Bu yaklaşım, IoT'nin nesneyi pasif bir fiziksel unsur olmaktan çıkararak sistem davranışına katılan aktif bir bileşen hâline getirdiğini göstermektedir.

"Algılama" kavramı, fiziksel çevredeki ölçülebilir durumların dijital veriye dönüştürülmesini ifade eder. Buna karşılık "aktüasyon", dijital sistem tarafından üretilen kararın fiziksel ortamda bir eyleme dönüşmesidir. Bu iki kavram birlikte ele alındığında IoT sistemlerinin yalnızca veri toplayan

izleme sistemleri olmadığı, aynı zamanda karar ve kontrol döngüsü içerebilen sistemler olduğu anlaşılır. Miorandi vd. (2012), IoT araştırmalarında algılama, haberleşme, veri işleme, servis üretimi ve güvenilir etkileşim kavramlarının birlikte değerlendirilmesi gerektiğini belirtmektedir.

IoT terminolojisinde "kimliklendirme" de temel bir kavramdır. Bir nesnenin sistem içinde izlenebilmesi, onun benzersiz biçimde tanımlanmasına bağlıdır. Kimliklendirme, nesnenin yalnızca fiziksel olarak var olmasını değil, dijital sistemde adreslenebilir ve yönetilebilir hâle gelmesini sağlar. Al-Fuqaha vd. (2015), IoT'nin RFID, akıllı sensörler, iletişim teknolojileri ve internet protokollerindeki gelişmelerle mümkün hâle geldiğini belirtirken, bu teknolojilerin temel işlevlerinden birinin nesnelerin tanımlanması ve sistem içinde etkileşime sokulması olduğunu vurgulamaktadır.

"Bağlam farkındalığı" ve "otonomi" kavramları, IoT sistemlerinin ileri düzey teknik anlamlandırma kapasitesini ifade eder. Bağlam farkındalığı, sistemin yalnızca ham veriyi değil, verinin üretildiği çevresel ve operasyonel koşulları da dikkate almasıdır. Otonomi ise sistemin belirli durumlarda insan müdahalesi olmadan karar üretebilmesi veya bir süreci tetikleyebilmesidir. Perera vd. (2014), bağlam farkındalığının IoT sistemlerinde sensör verisinin anlamlandırılması için kritik olduğunu ve bağlamın toplanması, modellenmesi, akıl yürütülmesi ve dağıtılması gibi süreçleri içerdiğini belirtmektedir.

IoT'NİN TEMEL BİLEŞENLERİ

Bir IoT sistemi, fiziksel çevreden veri elde edilmesinden bu verinin işlenerek bilgi veya eyleme dönüştürülmesine kadar farklı işlevlere sahip bileşenlerin birlikte çalışmasıyla oluşur. Sistemlerin kullanım amacı, ölçeği ve teknik özellikleri değişebilmekle birlikte temel bir IoT yapısında fiziksel nesneler, sensörler, aktüatörler, işlem birimleri, iletişim altyapısı, ağ geçitleri, veri işleme ortamları ve uygulama servisleri bulunabilir. Bu bileşenler birbirinden bağımsız yapılar değil, uçtan uca veri ve kontrol akışının farklı noktalarında görev yapan bütünleşik sistem unsurlarıdır.

Fiziksel nesne, IoT sistemi tarafından gözlemlenen, izlenen veya kontrol edilen gerçek dünya varlığıdır. Bu nesne bir makine, araç, üretim hattı, bina, tarımsal alan veya günlük kullanım nesnesi olabilir. Nesnenin fiziksel

durumunun dijital sistem tarafından anlaşılabilmesi için ölçülebilir özelliklerin veriye dönüştürülmesi gerekir.

Sensörler, sıcaklık, nem, basınç, hareket, konum, ışık, titreşim veya benzeri fiziksel büyüklükleri algılayarak dijital sistem tarafından işlenebilecek verilere dönüştürür. Bu nedenle sensörler fiziksel dünya ile dijital ortam arasındaki temel veri giriş noktalarından biridir.

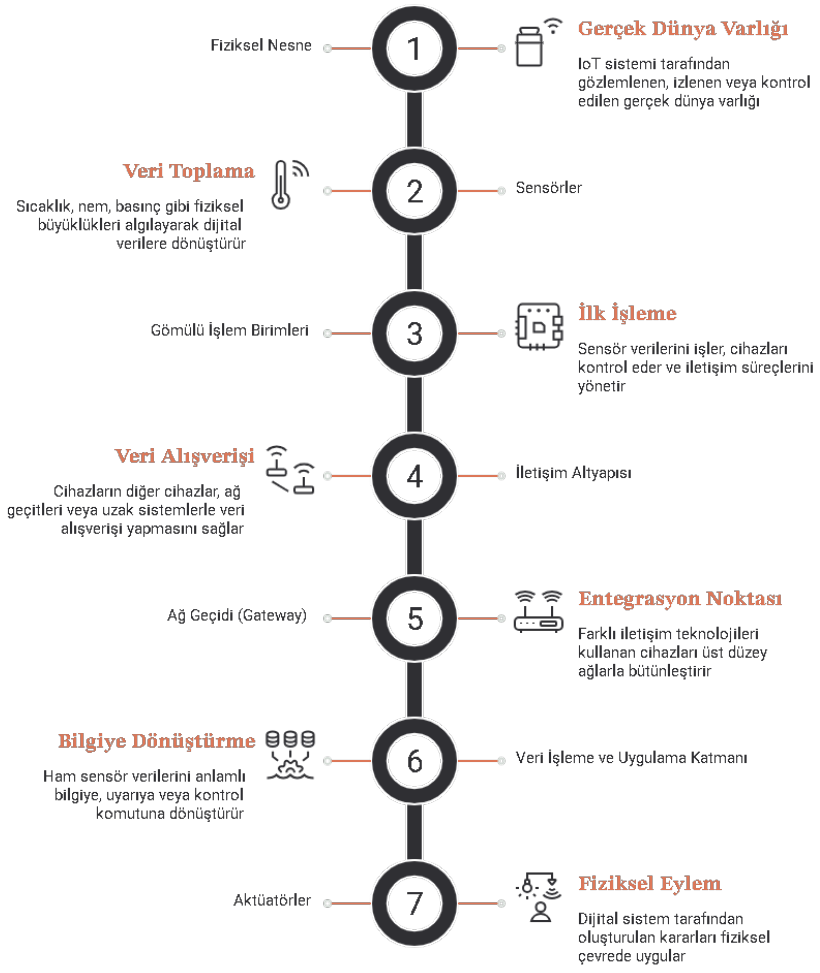
Aktüatörler, dijital sistem tarafından oluşturulan kararların fiziksel çevrede uygulanmasını sağlar. Motorların çalıştırılması, vanaların açılması, ışıkların kontrol edilmesi veya bir mekanik sistemin konumunun değiştirilmesi aktüasyon örnekleridir. Sensör ve aktüatörlerin birlikte kullanılması, IoT sistemlerinin yalnızca gözlem yapan yapılardan fiziksel çevre üzerinde etkide bulunabilen kontrol sistemlerine dönüşmesini mümkün kılar.

Gömülü işlem birimleri, sensörlerden elde edilen verilerin ilk aşamada işlenmesi, cihazların kontrol edilmesi ve iletişim süreçlerinin yönetilmesi gibi görevleri gerçekleştirir. Mikrodenetleyiciler ve benzeri gömülü işlem platformları, özellikle kaynak kısıtlı IoT cihazlarında bu işlevlerin yerine getirilmesini sağlar.

İletişim altyapısı, cihazların diğer cihazlar, ağ geçitleri veya uzak bilgi sistemleriyle veri alışverişi yapmasını sağlar. IoT sistemlerinde iletişim teknolojisinin seçimi; kapsama alanı, veri miktarı, enerji tüketimi, gecikme, güvenilirlik ve maliyet gibi gereksinimlere bağlı olarak değişebilir.

Ağ geçidi (gateway), özellikle farklı iletişim teknolojileri veya protokoller kullanan cihazların üst düzey ağlar ve yazılım sistemleriyle bütünleştirilmesinde ara bileşen olarak kullanılabilir. Ağ geçitleri veri toplama, dönüştürme, filtreleme, protokol uyarlama ve bazı durumlarda yerel veri işleme görevlerini yerine getirebilir.

Veri işleme ve uygulama katmanı ise cihazlardan elde edilen verilerin saklanması, işlenmesi, yorumlanması ve kullanıcıya veya başka sistemlere hizmet olarak sunulmasını sağlar. Bu aşamada ham sensör verileri anlamlı bilgiye, uyarıya, karar destek çıktısına veya kontrol komutuna dönüştürülebilir.



Şekil 1. IoT sistemlerinde temel katmanlı mimari

Bu bileşenler birlikte değerlendirildiğinde IoT sisteminin yalnızca sensörlerin internete bağlandığı bir yapı olmadığı görülmektedir. IoT; fiziksel nesne, algılama, hesaplama, iletişim, veri işleme ve gerektiğinde aktüasyon süreçlerinin bütünleşik biçimde çalıştığı uçtan uca bir sistem yaklaşımıdır.

IOT'NİN TEKNOLOJİK GELİŞİM SÜRECİ

IoT'nin teknolojik gelişimi tek bir buluşa veya tek bir donanım sınıfına indirgenemez. Bu gelişim; gömülü sistemler, otomatik tanımlama teknolojileri, kablosuz sensör ağları, internet altyapısı, mobil iletişim, dağıtık bilişim ve veri işleme teknolojilerinin kademeli olarak yakınsamasıyla ortaya çıkmıştır. Atzori vd. (2010), IoT'nin temelinde farklı iletişim çözümleri ve tanımlama teknolojilerinin entegrasyonunun bulunduğunu belirtmektedir. Bu nedenle IoT, belirli bir cihaz türünden çok, farklı teknolojik katmanların birlikte çalışmasıyla oluşan sistemsel bir dönüşüm olarak değerlendirilmelidir.

IoT'nin erken gelişim sürecinde nesnelerin dijital olarak tanımlanabilmesi önemli bir eşik oluşturmuştur. RFID gibi otomatik tanımlama yaklaşımları, fiziksel nesnelerin dijital sistemlerde izlenebilir hâle gelmesine katkı sağlamıştır. Daha sonraki aşamada kablosuz sensör ağları, yalnızca nesnelerin tanımlanmasını değil, çevresel verilerin dağıtık biçimde toplanmasını da mümkün kılmıştır. Gubbi vd. (2013), RFID, yakın alan iletişimi, gömülü sensör düğümleri ve kablosuz sensör ağlarını IoT'nin gelişimini hızlandıran temel teknolojiler arasında değerlendirmektedir.

İnternet altyapısının yaygınlaşması, IoT'nin yerel ölçekte çalışan algılama sistemlerinden küresel ölçekte erişilebilen bağlantılı sistemlere dönüşmesinde belirleyici olmuştur. İnternet protokolleri, mobil iletişim altyapıları ve düşük güçlü cihaz teknolojileri, nesnelerin farklı ağ koşullarında veri üretebilmesini ve bu veriyi yazılım sistemlerine aktarabilmesini mümkün kılmıştır. Al-Fuqaha vd. (2015), IoT'nin RFID, akıllı sensörler, iletişim teknolojileri ve internet protokollerindeki gelişmelerle desteklendiğini ve makineden makineye iletişim yaklaşımlarının IoT'nin erken evresini oluşturduğunu belirtmektedir.

IoT'nin olgunlaşma sürecinde yalnızca bağlantı değil, verinin anlamlandırılması ve sistem davranışına dönüştürülmesi de önem kazanmıştır. Bu bağlamda bulut bilişim, büyük ölçekli veri işleme ve bağlam farkındalığı gibi yaklaşımlar, IoT sistemlerinin yalnızca ölçüm yapan sistemler olmaktan çıkıp veri odaklı karar destek ve otomasyon sistemlerine dönüşmesine katkı sağlamıştır. Miorandi vd. (2012), IoT'nin uygulamalar, etkinleştirici teknolojiler ve araştırma zorlukları açısından parçalı bir alan olmaktan çıkarılıp bütüncül bir bakışla ele alınması gerektiğini vurgulamaktadır.

IOT SİSTEMLERİNİN SOYUT ÇALIŞMA MODELİ

IoT sistemlerinin soyut çalışma modeli, fiziksel dünyadan başlayan ve dijital karar süreçlerine uzanan bir veri döngüsü üzerinden açıklanabilir. Bu döngüde ilk aşama fiziksel ortamda ölçülebilir veya izlenebilir bir durumun ortaya çıkmasıdır. İkinci aşamada bu durum dijital veriye dönüştürülür. Üçüncü aşamada veri, sistem içinde anlamlandırılabilecek bir temsile kavuşur. Daha sonra veri ilgili yazılım bileşenlerine aktarılır, işlenir ve sistem çıktısı üretilir. Bu çıktı kimi zaman kullanıcıya bilgi sağlama, kimi zaman uyarı üretme, kimi zaman da fiziksel ortama geri müdahale etme biçiminde ortaya çıkar. Bu genel döngü, IoT'nin fiziksel çevre ile dijital sistemler arasındaki etkileşim mantığını göstermektedir (Gubbi vd., 2013; Atzori vd., 2010).

Bu soyut model şu şekilde ifade edilebilir: fiziksel durum → algılama → dijital temsil → veri aktarımı → işleme → karar → geri bildirim. Buradaki "dijital temsil" kavramı önemlidir; çünkü IoT sisteminde fiziksel nesne veya olay, doğrudan değil, ölçüm verileri, zaman damgaları, kimlik bilgileri ve durum göstergeleri üzerinden temsil edilir. ISO/IEC 30141:2024 standardının ortak terminoloji ve referans mimari yaklaşımı da IoT sistemlerinin belirli ürün veya uygulamalardan bağımsız olarak kavramsal, işlevsel ve mimari düzeyde modellenmesi gerektiğini göstermektedir.

Soyut çalışma modelinde veri akışı tek yönlü bir süreç olarak görülmemelidir. Basit izleme sistemlerinde veri fiziksel ortamdan yazılım sistemine doğru ilerlerken, kontrol içeren sistemlerde karar çıktısı yeniden fiziksel ortama dönebilir. Bu nedenle IoT sistemleri çoğu zaman kapalı çevrimli bir davranış gösterir. Miorandi vd. (2012), IoT sistemlerinin nesneler, servisler ve kullanıcılar arasında etkileşim üreten bütüncül sistemler olarak ele alınması gerektiğini belirtmektedir. Bu yaklaşım, IoT'nin yalnızca veri toplama değil, veri üzerinden eylem üretme potansiyeline sahip olduğunu ortaya koymaktadır.

Bu modelde ayrıntılı mimari katmanlara veya belirli haberleşme protokollerine girilmeden, IoT sistemlerinin temel davranışı açıklanabilir. Örneğin bir ortam izleme sisteminde sıcaklık değişimi fiziksel bir durumdur; bu durum ölçülerek dijital veri hâline getirilir, veri belirli bir yazılım bileşeni tarafından değerlendirilir ve eşik aşımı durumunda sistem uyarı üretebilir. Bu örnek, IoT'nin temel işleyişini anlatmak için yeterlidir; sistemin hangi sensörle, hangi protokolle veya hangi veri tabanıyla çalıştığı bu bölümün kapsamı dışında bırakılmalıdır.

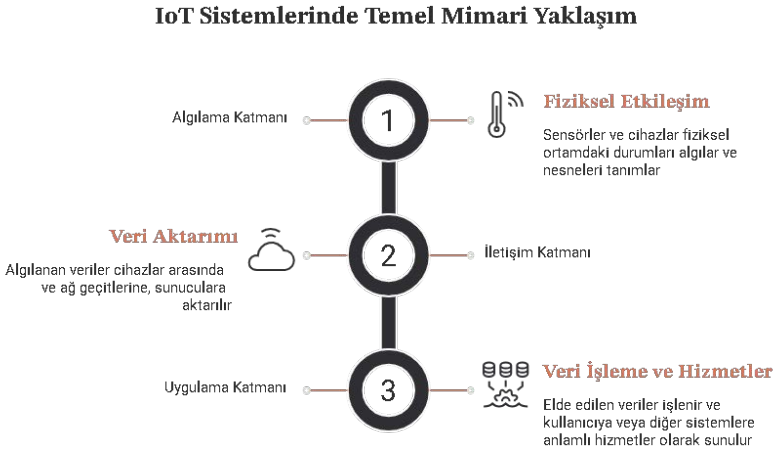
IOT SİSTEMLERİNDE TEMEL MİMARİ YAKLAŞIM

IoT sistemlerinde çok farklı mimari modeller kullanılabilmekle birlikte temel çalışma yapısı kavramsal olarak algılama, iletişim ve uygulama olmak üzere üç temel katman üzerinden açıklanabilir. Bu model, belirli bir teknoloji veya üreticiye bağlı olmadan IoT sisteminin temel işlevlerinin anlaşılmasını sağlar.

Algılama katmanı, fiziksel çevre ile dijital sistem arasındaki etkileşimin başladığı katmandır. Sensörler, RFID bileşenleri, gömülü cihazlar ve aktüatörler bu katmanda değerlendirilebilir. Katmanın temel görevi fiziksel ortamdaki durumları algılamak, nesneleri tanımlamak ve gerektiğinde fiziksel çevre üzerinde eylem gerçekleştirmektir.

İletişim veya ağ katmanı, algılama katmanında oluşturulan verilerin ilgili işlem ve uygulama bileşenlerine aktarılmasını sağlar. Bu katman, cihazlar arasındaki iletişimin yanı sıra cihazlardan ağ geçitlerine, sunuculara veya diğer bilgi sistemlerine gerçekleştirilen veri aktarımını da kapsar. Farklı IoT uygulamalarında kullanılan iletişim teknolojileri; veri miktarı, menzil, enerji tüketimi, gecikme ve güvenilirlik gereksinimlerine göre değişiklik gösterebilir.

Uygulama ve servis katmanı, elde edilen verilerin belirli bir kullanım amacı doğrultusunda işlendiği ve kullanıcıya veya başka sistemlere anlamlı hizmetler olarak sunulduğu katmandır. İzleme panelleri, uyarı mekanizmaları, karar destek sistemleri, otomasyon uygulamaları ve diğer yazılım servisleri bu katmanda değerlendirilebilir.



Şekil 2. IoT sistemlerinde temel katmanlı mimari

Güncel IoT sistemlerinde bu temel model, verinin nerede işlendiğini açıklamak amacıyla cihaz, uç, bulut ve uygulama düzeyleriyle de ilişkilendirilebilir. Verinin tamamının uzak bir bilgi işlem ortamına gönderilmesi her uygulamada gerekli veya uygun olmayabilir. Gecikme, bant genişliği, güvenilirlik ve mahremiyet gereksinimleri nedeniyle bazı işlemler veri kaynağına daha yakın noktalarda gerçekleştirilebilir. Bu nedenle modern IoT tasarımında merkezi ve dağıtık veri işleme yaklaşımlarının birlikte kullanılması mümkündür.

IOT'Yİ GELENEKSEL BİLGİ SİSTEMLERİNDEN AYIRAN TEKNİK ÖZELLİKLER

IoT sistemleri geleneksel bilgi sistemlerinden öncelikle veri kaynağı bakımından ayrılır. Geleneksel bilgi sistemlerinde veri çoğunlukla kullanıcı işlemleri, kurumsal kayıtlar veya yazılım uygulamaları tarafından üretilirken, IoT sistemlerinde veri doğrudan fiziksel çevreden elde edilir. Bu durum, IoT sistemlerini fiziksel dünya ile doğrudan ilişkili, ölçüm temelli ve çoğu zaman sürekli veri üreten sistemler hâline getirir. Whitmore vd. (2015), IoT'nin gündelik nesnelerin tanımlama, algılama, ağ bağlantısı ve işlem yetenekleri kazanmasıyla ortaya çıkan bir paradigma olduğunu belirtmektedir.

İkinci temel fark, sistem yapısının dağıtık ve heterojen olmasıdır. Geleneksel bilgi sistemleri genellikle belirli istemci-sunucu yapıları veya kurumsal yazılım bileşenleri etrafında örgütlenirken, IoT sistemleri farklı donanım yeteneklerine, farklı enerji sınırlamalarına, farklı bağlantı koşullarına ve farklı veri üretim biçimlerine sahip çok sayıda bileşenden oluşabilir. Miorandi vd. (2012), IoT araştırmalarının parçalı teknoloji alanlarından oluştuğunu ve bu nedenle bütüncül sistem bakışının gerekli olduğunu belirtmektedir. Bu tespit, IoT'nin heterojen yapısının yalnızca teknik çeşitlilik değil, aynı zamanda sistem tasarımı açısından temel bir karmaşıklık kaynağı olduğunu göstermektedir.

Üçüncü fark, veri üretim biçimidir. Geleneksel bilgi sistemlerinde veri çoğunlukla işlem tabanlıdır; IoT sistemlerinde ise veri sürekli, periyodik veya olay tabanlı biçimde üretilebilir. Bu nedenle IoT sistemleri zaman damgalı veriler, akış verileri, çevresel değişkenlik ve veri kalitesi problemleriyle daha sık karşılaşır. Perera vd. (2014), IoT ortamlarında sensör verisinin anlamlı hâle getirilebilmesi için bağlamın toplanması, modellenmesi ve yorumlanmasının kritik olduğunu ifade etmektedir. Bu durum, IoT verisinin yalnızca niceliksel olarak fazla değil, aynı zamanda

bağlama duyarlı biçimde değerlendirilmesi gereken bir veri türü olduğunu ortaya koymaktadır.

Dördüncü fark, kaynak kısıtları ve hata modelleridir. Geleneksel bilgi sistemlerinde hata çoğunlukla yazılım, kullanıcı veya sunucu altyapısı bağlamında ele alınırken, IoT sistemlerinde donanım arızaları, çevresel koşullar, bağlantı kopmaları, enerji sınırlamaları ve ölçüm hataları da sistem davranışını doğrudan etkiler. NISTIR 8228 raporu, IoT cihazlarının geleneksel BT cihazlarından farklı siber güvenlik ve mahremiyet riskleri doğurabileceğini ve bu risklerin cihaz yaşam döngüsü boyunca yönetilmesi gerektiğini belirtmektedir. Bu yaklaşım, IoT'nin klasik bilgi sistemlerinden yalnızca işlevsel olarak değil, risk ve yönetim boyutları açısından da ayrıldığını göstermektedir.

Tablo 1. Geleneksel bilgi sistemleri ile IoT sistemlerinin temel farklılıkları

Özellik	Geleneksel Bilgi Sistemleri	IoT Sistemleri
Temel veri kaynağı	Kullanıcı ve kurumsal işlemler	Fiziksel çevre ve sensörler
Veri üretimi	Çoğunlukla işlem tabanlı	Sürekli, periyodik veya olay tabanlı
Sistem yapısı	Görece homojen	Dağıtık ve heterojen
Fiziksel çevre ile ilişki	Dolaylı	Doğrudan
Kaynak kısıtları	Görece sınırlı	Enerji, işlem ve haberleşme kısıtları yaygın
Hata kaynakları	Yazılım, kullanıcı, sunucu	Donanım, sensör, bağlantı, çevresel koşullar ve yazılım
Gerçek zamanlılık	Uygulamaya bağlı	Birçok uygulamada kritik
Aktüasyon	Genellikle bulunmaz	Sisteme bağlı olarak bulunabilir
Ölçeklenebilirlik	Kullanıcı ve işlem sayısı ağırlıklı	Cihaz, veri, bağlantı ve olay sayısı
Yaşam döngüsü yönetimi	Yazılım ve BT altyapısı ağırlıklı	Cihaz + ağ + yazılım + veri altyapısı

IOT SİSTEMLERİNİN TEKNİK SINIFLANDIRILMASI

IoT sistemleri yalnızca uygulama alanlarına göre değil, teknik özelliklerine göre de sınıflandırılmalıdır. Bu yaklaşım, sistemlerin hangi alanda kullanıldığından bağımsız olarak nasıl çalıştığını, hangi veri

davranışına sahip olduğunu, ne düzeyde otonomi içerdiğini ve hangi mühendislik gereksinimlerini doğurduğunu anlamayı sağlar. Miorandi vd. (2012), IoT alanının farklı uygulama ve teknoloji kümeleri etrafında parçalanma eğilimi gösterdiğini belirtmektedir. Bu nedenle teknik sınıflandırma, IoT sistemlerini uygulama örnekleri üzerinden değil, sistem davranışı üzerinden değerlendirmek için daha uygun bir çerçeve sunar.

İşlev temelli sınıflandırmada IoT sistemleri izleme, kontrol ve karar destek sistemleri olarak ele alınabilir. İzleme sistemleri fiziksel çevreden veri toplar ve bu veriyi kullanıcıya veya yazılım sistemine sunar. Kontrol sistemleri, toplanan veriye bağlı olarak fiziksel bir süreci etkileyen çıktı üretir. Karar destek sistemleri ise veriyi yorumlayarak kullanıcıya veya başka bir yazılım bileşenine karar sürecinde kullanılabilecek bilgi sağlar. Gubbi vd. (2013), IoT'nin çevresel göstergelerin ölçülmesi, anlaşılması ve ortak bir operasyonel görünüm oluşturulması açısından önemli olduğunu vurgulamaktadır. Bu yaklaşım, izleme ve karar desteğinin IoT sistem davranışının merkezinde yer aldığını göstermektedir.

Veri üretim biçimine göre IoT sistemleri sürekli veri üreten, periyodik veri üreten ve olay tabanlı veri üreten sistemler olarak sınıflandırılabilir. Sürekli veri üreten sistemlerde fiziksel çevreye ilişkin ölçümler düzenli akış hâlinde sisteme aktarılır. Periyodik sistemlerde veri belirli zaman aralıklarıyla üretilir. Olay tabanlı sistemlerde ise veri yalnızca anlamlı bir durum değişikliği ortaya çıktığında üretilir. Perera vd. (2014), sensör verisinin bağlam içinde anlamlandırılmasının IoT sistemleri için kritik olduğunu vurgulamakta; bu durum veri üretim biçiminin yalnızca teknik bir ayrıntı değil, sistemin anlamlandırma kapasitesini etkileyen temel bir unsur olduğunu göstermektedir.

Ölçek, hareketlilik, kritik seviye ve otonomi düzeyi de IoT sistemlerinin teknik sınıflandırmasında kullanılabilir. Ölçeğe göre sistemler kişisel, yerel, kurumsal veya geniş ölçekli olabilir. Hareketliliğe göre sabit, mobil veya karma yapı gösterebilir. Kritikliğe göre hata veya gecikmenin etkisi düşük, görev-kritik veya güvenlik-kritik düzeyde değerlendirilebilir. Otonomi düzeyine göre ise insan kontrollü, yarı otonom veya otonom sistemlerden söz edilebilir. Bu sınıflandırma, IoT sistemlerinin yalnızca "nerede kullanıldığı" sorusuna değil, "hangi teknik davranışa sahip olduğu" sorusuna yanıt verir.

Tablo 2. IoT sistemlerinin temel teknik sınıflandırma ölçütleri

Sınıflandırma ölçütü	Temel sınıflar	Açıklama
İşlev	İzleme / Kontrol / Karar destek	Sistemin temel görevini belirtir
Veri üretimi	Sürekli / Periyodik / Olay tabanlı	Verinin hangi koşullarda üretildiğini gösterir
Ölçek	Kişisel / Yerel / Kurumsal / Geniş ölçekli	Sistemin kapsadığı cihaz ve kullanıcı ölçeğini ifade eder
Hareketlilik	Sabit / Mobil / Karma	IoT bileşenlerinin hareket durumunu belirtir
Kritiklik	Düşük / Görev-kritik / Güvenlik-kritik	Hata veya gecikmenin oluşturacağı etkinin seviyesini gösterir
Otonomi	İnsan kontrollü / Yarı otonom / Otonom	Sistemin insan müdahalesinden bağımsız karar kapasitesini ifade eder

IOT'NİN BİLGİSAYAR MÜHENDİSLİĞİ AÇISINDAN KONUMLANDIRILMASI

IoT, bilgisayar mühendisliği açısından tek bir alt alanla sınırlı değildir; gömülü sistemler, bilgisayar ağları, yazılım mühendisliği, veri mühendisliği, dağıtık sistemler, siber güvenlik ve yapay zekâ gibi alanların kesişiminde yer alır. Bu nedenle IoT, bilgisayar mühendisliği öğrencileri ve araştırmacıları için yalnızca cihaz bağlantısı konusu değil, uçtan uca sistem tasarımı problemidir. Khodadadi vd. (2017), IoT'nin tanımlama, iletişim, kaynak keşfi ve servis entegrasyonu gibi farklı bilgi ve araştırma alanlarının bütünleşmesini gerektirdiğini belirtmektedir.

Gömülü sistemler açısından IoT, fiziksel nesnelerin programlanabilir ve veri üretebilir hâle gelmesini sağlar. Bilgisayar ağları açısından IoT, cihazlar, servisler ve kullanıcılar arasında veri aktarımını mümkün kılan bağlantılı bir yapı oluşturur. Yazılım mühendisliği açısından ise IoT, cihazlardan gelen verilerin işlenmesi, servisler aracılığıyla sunulması, kullanıcı arayüzleriyle erişilebilir hâle getirilmesi ve sistem bileşenlerinin sürdürülebilir biçimde yönetilmesiyle ilgilidir. Al-Fuqaha vd. (2015), IoT'nin etkinleştirici teknolojiler, iletişim yaklaşımları ve uygulama sorunlarıyla birlikte değerlendirilmesi gerektiğini ortaya koymaktadır.

Veri mühendisliği ve dağıtık sistemler açısından IoT, sürekli ve heterojen veri üreten çok sayıda bileşenin birlikte çalıştığı bir sistem alanıdır. Bu tür sistemlerde veri yalnızca depolanacak bir çıktı değil, karar süreçlerinin temel girdisidir. Siow vd. (2018), IoT uygulamalarında analitiğin veriden bilgi ve eyleme dönüştürülebilir içgörü üretme açısından önemli olduğunu belirtmektedir. Ancak bu bölümde yapay zekâ veya analitik yöntem ayrıntılarına girilmemeli; IoT'nin veri odaklı sistem geliştirme açısından bilgisayar mühendisliğine nasıl bağlandığı vurgulanmalıdır.

Siber güvenlik açısından IoT, cihaz, veri, iletişim ve kullanıcı etkileşimi boyutlarının birlikte korunmasını gerektirir. Sicari vd. (2015), IoT sistemlerinde gizlilik, kimlik doğrulama, erişim kontrolü, güven ve güvenlik politikalarının önemli gereksinimler olduğunu belirtmektedir. Bununla birlikte bu bölümde güvenlik teknikleri ayrıntılandırılmamalı; yalnızca IoT'nin bilgisayar mühendisliği açısından güvenilir, ölçeklenebilir, sürdürülebilir ve yönetilebilir sistemler tasarlama sorumluluğu doğurduğu ifade edilmelidir.

IOT SİSTEMLERİNDE TEMEL TASARIM GEREKSİNİMLERİ

IoT sistemlerinde tasarım başarısı yalnızca cihazların veri üretebilmesine bağlı değildir. Sistemin ölçeklenebilir, güvenilir, yönetilebilir, birlikte çalışabilir, enerji açısından verimli ve veri kalitesi bakımından tutarlı olması gerekir. ITU-T Y.2060/Y.4000 önerisi, IoT'nin temel karakteristikleri ve üst düzey gereksinimleriyle birlikte ele alınması gerektiğini belirtmektedir. ISO/IEC 30141:2024 ise ortak terminoloji, yeniden kullanılabilir tasarımlar ve iyi uygulamalar üzerinden standartlaştırılmış bir referans mimari sunmaktadır. Bu iki yaklaşım, IoT tasarımında teknoloji seçiminin ötesinde sistem gereksinimlerinin açık biçimde tanımlanması gerektiğini göstermektedir.

Ölçeklenebilirlik, IoT sistemlerinin cihaz sayısı, kullanıcı sayısı, veri hacmi ve olay yoğunluğu arttığında işlevselliğini sürdürebilmesini ifade eder. Birlikte çalışabilirlik ise farklı donanım, yazılım ve iletişim yaklaşımlarına sahip bileşenlerin ortak bir sistem içinde uyumlu çalışabilmesidir. ISO/IEC 30141:2024 standardının ortak kelime dağarcığı ve yeniden kullanılabilir tasarım vurgusu, IoT sistemlerinde birlikte çalışabilirliğin yalnızca teknik bağlantı meselesi değil, ortak kavramsal modelleme meselesi olduğunu göstermektedir.

Güvenilirlik, süreklilik ve hata toleransı da IoT tasarımının temel gereksinimlerindendir. IoT sistemlerinde cihaz arızası, bağlantı kesintisi, veri kaybı, çevresel koşullar veya enerji sınırlamaları sistem davranışını etkileyebilir. Bu nedenle sistemin tekil bileşen hatalarında tamamen işlevsiz hâle gelmemesi, kritik durumlarda kabul edilebilir çalışma davranışını sürdürebilmesi ve izlenebilir olması gerekir. NISTIR 8228, IoT cihazlarının yaşam döngüsü boyunca siber güvenlik ve mahremiyet risklerinin yönetilmesi gerektiğini vurgulamakta; bu da tasarım gereksinimlerinin yalnızca kurulum aşamasında değil, işletim ve bakım aşamalarında da geçerli olduğunu göstermektedir.

Veri kalitesi, IoT sistemlerinde doğrudan karar kalitesini etkileyen bir gereksinimdir. IoT verisi eksik, gecikmiş, tutarsız, hatalı veya bağlamdan kopuk olduğunda sistemin üreteceği kararlar da güvenilir olmayabilir. Cichy ve Rass (2019), IoT'de veri kalitesiyle ilgili çalışmaları inceleyerek veri kalitesi boyutları, veri kalitesi problemleri ve ölçüm yöntemleri arasındaki ilişkilerin uygulayıcılar açısından önemli olduğunu belirtmektedir. Bu nedenle IoT tasarımında veri doğruluğu, tamlık, zamanlılık, tutarlılık ve izlenebilirlik gibi unsurlar teknik gereksinim olarak ele alınmalıdır.

Son olarak enerji verimliliği, bakım yapılabilirlik, gözlemlenebilirlik ve yaşam döngüsü yönetimi IoT sistemlerinin uzun süreli başarısı açısından belirleyicidir. Kaynak kısıtlı cihazların kullanıldığı sistemlerde enerji tüketimi, cihaz ömrü ve bakım maliyeti doğrudan sistem sürdürülebilirliğini etkiler. Gözlemlenebilirlik ise sistemin çalışma durumunun ölçümler, kayıtlar, olaylar ve performans göstergeleri üzerinden izlenebilmesini sağlar. Bu gereksinimler birlikte değerlendirildiğinde IoT tasarımı, yalnızca cihazların birbirine bağlanması değil, sistemin tüm yaşam döngüsü boyunca yönetilebilir ve güvenilir kalmasını sağlayacak mühendislik kararlarının bütünüdür.

IOT SİSTEMLERİNDE GÜVENLİK VE MAHREMİYETE GİRİŞ

IoT sistemlerinin fiziksel çevreyle doğrudan etkileşim hâlinde olması, güvenlik ve mahremiyet gereksinimlerini geleneksel bilgi sistemlerinden farklı boyutlara taşımaktadır. Bir IoT sisteminde yalnızca yazılım ve ağ altyapısının değil; cihazların, sensörlerin, kullanıcı kimliklerinin, iletişim kanallarının ve üretilen verilerin de korunması gerekir. Bu nedenle güvenlik, sonradan sisteme eklenen bağımsız bir özellikten çok sistem yaşam döngüsünün tamamında dikkate alınması gereken temel bir tasarım gereksinimi olarak değerlendirilmelidir.

IoT güvenliği genel olarak cihaz güvenliği, iletişim güvenliği, veri güvenliği, kimlik ve erişim yönetimi ile mahremiyet boyutlarında ele alınabilir. Cihaz güvenliği fiziksel veya yazılımsal müdahalelere karşı IoT düğümlerinin korunmasını; iletişim güvenliği cihazlar arasında aktarılan verilerin yetkisiz erişime ve değişikliğe karşı korunmasını; veri güvenliği ise üretilen, aktarılan ve depolanan bilgilerin gizlilik, bütünlük ve erişilebilirliğinin sürdürülmesini amaçlar.

Kimlik ve erişim yönetimi açısından sistemin hangi cihazın veya kullanıcının hangi kaynaklara erişebileceğini güvenilir biçimde belirlemesi gerekir. Özellikle çok sayıda cihazın bulunduğu IoT sistemlerinde cihaz kimliğinin doğrulanması ve yetkisiz bileşenlerin sisteme katılmasının engellenmesi kritik önem taşır.

Mahremiyet ise IoT cihazlarının fiziksel çevreden sürekli veri toplayabilmesi nedeniyle özel bir önem taşımaktadır. Konum, davranış, çevresel koşullar veya kullanıcı alışkanlıklarına ilişkin veriler, doğrudan kişisel veri niteliği taşımasa bile farklı veri kümeleriyle birleştirildiğinde kullanıcı hakkında hassas bilgiler ortaya çıkarabilir. Bu nedenle hangi verinin neden toplandığı, ne kadar süre saklandığı, kimlerin erişebildiği ve hangi amaçlarla kullanılabileceği sistem tasarımının başlangıcından itibaren değerlendirilmelidir.

NISTIR 8228, IoT cihazlarının geleneksel BT cihazlarından farklı siber güvenlik ve mahremiyet riskleri oluşturabileceğini ve bu risklerin cihaz yaşam döngüsü boyunca yönetilmesi gerektiğini vurgulamaktadır. Bu yaklaşım doğrultusunda IoT güvenliği yalnızca saldırıları engellemek değil; cihazların güvenilir biçimde yapılandırılması, güncellenmesi, izlenmesi, bakımının yapılması ve kullanım ömrünün sonunda güvenli biçimde devreden çıkarılması gibi süreçleri de kapsamaktadır.

Bu bölümün amacı IoT güvenlik mekanizmalarını ayrıntılı biçimde incelemek değil, güvenlik ve mahremiyetin IoT sistemlerinin temel mühendislik gereksinimlerinden biri olduğunu ortaya koymaktır. Kimlik doğrulama, şifreleme, güvenli iletişim, erişim kontrolü ve saldırı tespit yaklaşımları gibi ayrıntılı teknik konular sonraki bölümlerde ayrıca ele alınmaktadır.

KAYNAKÇA

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). *Internet of Things: A survey on enabling technologies, protocols, and applications*. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. DOI: 10.1109/COMST.2015.2444095.

Atzori, L., Iera, A., & Morabito, G. (2010). *The Internet of Things: A survey*. Computer Networks, 54(15), 2787–2805. DOI: 10.1016/j.comnet.2010.05.010.

Borgia, E. (2014). *The Internet of Things vision: Key features, applications and open issues*. Computer Communications, 54, 1–31. DOI: 10.1016/j.comcom.2014.09.008.

Cichy, C., & Rass, S. (2019). *Data quality and the Internet of Things*. Computing, 101, 790–809. DOI: 10.1007/s00607-019-00746-z.

Eylence M., Yücel M., Özmen M. M., Aksoy B., "Railway Security System Design Using Unmanned Aerial Vehicle Image Processing and Deep Learning Methods," Bingöl Doğa ve Fen Dergisi, vol. 11, no. 3, pp. 150–154, Sep. 2022.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). *Internet of Things (IoT): A vision, architectural elements, and future directions*. Future Generation Computer Systems, 29(7), 1645–1660. DOI: 10.1016/j.future.2013.01.010.

ISO/IEC. (2024). *ISO/IEC 30141:2024 Internet of Things (IoT) — Reference Architecture*.

ITU-T. (2012). *Recommendation ITU-T Y.2060/Y.4000: Overview of the Internet of Things*.

Khodadadi, F., Dastjerdi, A. V., & Buyya, R. (2017). *Internet of Things: An overview*. In Internet of Things: Principles and Paradigms. Elsevier.

Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). *Internet of Things: Vision, applications and research challenges*. Ad Hoc Networks, 10(7), 1497–1516. DOI: 10.1016/j.adhoc.2012.02.016.

NIST. (2019). *NISTIR 8228: Considerations for Managing Internet of Things Cybersecurity and Privacy Risks*.

Özsoy K., Aksoy B., Yücel M., "Design And Manufacture Of Continuous Automatic 3D Printing Device With Conveyor System By Image Processing Technology," Erzincan Üniversitesi Fen Bilimleri Enstitüsü Dergisi, vol. 13, no. 2, pp. 392–403, Aug. 2020.

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). *Context aware computing for the Internet of Things: A survey*. IEEE Communications Surveys & Tutorials, 16(1), 414–454. DOI: 10.1109/SURV.2013.042313.00197.

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). *Security, privacy and trust in Internet of Things: The road ahead*. Computer Networks, 76, 146–164. DOI: 10.1016/j.comnet.2014.11.008.

Siow, E., Tiropanis, T., & Hall, W. (2018). *Analytics for the Internet of Things: A survey*. ACM Computing Surveys / arXiv preprint.

Whitmore, A., Agarwal, A., & Xu, L. D. (2015). *The Internet of Things—A survey of topics and trends*. Information Systems Frontiers, 17(2), 261–274. DOI: 10.1007/s10796-014-9489-2.

IOT SİSTEM MİMARİLERİ

Nesnelerin İnterneti (IoT), IoT Sistem Mimarileri, Edge Computing, Fog Computing, Cloud Computing

Ali Topal*, Muzaffer Tatlı**

GİRİŞ

Nesnelerin İnterneti (Internet of Things-IoT), fiziksel dünyadaki nesnelerin algılama, haberleşme, veri işleme ve kontrol yetenekleri kazanarak birbirleriyle ve farklı bilgi sistemleriyle iletişim kurduğu bütünleşik bir teknoloji ekosistemidir. Günümüzde IoT yalnızca sensörlerden oluşan bir haberleşme yapısı olmaktan çıkmış; cihazlar, haberleşme ağları, veri işleme altyapıları, yazılım platformları ve kullanıcı uygulamalarının birlikte çalıştığı çok bileşenli sistemlere dönüşmüştür. Bu dönüşüm, IoT sistemlerinin yalnızca teknik bileşenlerinin değil, bu bileşenlerin hangi mimari yapı içerisinde konumlandırıldığı ve birbirleriyle nasıl etkileşim kurduğu konusunun da önem kazanmasına neden olmuştur.

IoT sistemlerinin temel amacı fiziksel ortamdan anlamlı veriler toplamak, bu verileri uygun iletişim altyapıları üzerinden taşımak, gerekli durumlarda işlemek ve elde edilen sonuçları kullanıcıya veya fiziksel dünyadaki aktüatörlere aktarmaktır. Bu süreç; akıllı evlerden endüstriyel üretim tesislerine, sağlık hizmetlerinden akıllı tarıma, ulaşım sistemlerinden enerji altyapılarına kadar farklı uygulama alanlarında gerçekleştirilmektedir. Uygulama alanlarının çeşitlenmesiyle birlikte IoT sistemlerinin veri miktarı, cihaz sayısı, haberleşme çeşitliliği ve işlem gereksinimleri de artmaktadır.

Bu gelişmeler, IoT sistemlerinin merkezi ve tek tip bir yapı üzerinden tasarlanmasını zorlaştırmaktadır. Özellikle milyarlarca cihazın sisteme dahil olması, sürekli veri üretimi, düşük gecikme gerektiren uygulamaların yaygınlaşması ve cihazların farklı donanımsal özelliklere sahip olması mimari tasarımın temel sorunları arasında yer almaktadır. Bir IoT sisteminin başarısı yalnızca kullanılan sensör veya haberleşme teknolojisine bağlı

* Öğretim Görevlisi, Isparta Uygulamalı Bilimler Üniversitesi, Uzaktan Eğitim Meslek Yüksekokulu, Bilgisayar Teknolojileri, Isparta, alitopal@isparta.edu.tr, ORCID: orcid.org/0000-0001-5975-5932

** Öğretim Görevlisi, Isparta Uygulamalı Bilimler Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Bilgisayar Teknolojileri, Isparta, muzaffertatli@isparta.edu.tr, ORCID: orcid.org/ 0000-0001-9075-3791

değildir. Verinin hangi noktada işlendiği, hangi iletişim protokolünün tercih edildiği, işlem yükünün cihazlar arasında nasıl dağıtıldığı ve uygulama katmanının bu altyapıyı nasıl kullandığı da sistemin genel performansını doğrudan etkilemektedir.

Bu nedenle IoT sistem mimarileri, fiziksel dünyadan verinin elde edilmesinden kullanıcıya sunulan hizmete kadar uzanan bütüncül bir yapı içerisinde değerlendirilmelidir. Katmanlı mimariler, farklı teknolojilerin ve görevlerin sistem içerisindeki konumunu açıklamak açısından önemli bir model sunmaktadır. Bununla birlikte günümüzde yalnızca klasik katmanlı modellerin açıklanması yeterli görülmemekte; bulut, fog ve edge computing gibi dağıtık işlem yaklaşımlarının da IoT mimarilerinin önemli bir parçası haline geldiği görülmektedir.

Bu çalışmada IoT sistem mimarileri; temel katmanlı mimari yaklaşımlar, algılama katmanı, ağ katmanı, veri işleme katmanı ve uygulama katmanı kapsamında ele alınmaktadır. Bunun yanında cloud, fog ve edge computing yaklaşımlarının IoT mimarilerine etkisi incelenmekte, farklı mimari yaklaşımlar performans, gecikme, kaynak kullanımı ve uygulama gereksinimleri açısından karşılaştırılmaktadır. Son olarak yapay zekâ destekli IoT, 5G/6G tabanlı mimariler ve dijital ikiz gibi güncel yaklaşımların IoT sistem mimarilerinin gelecekteki gelişimine etkileri değerlendirilmektedir.

IOT SİSTEM MİMARİLERİNİN TEMEL YAPISI

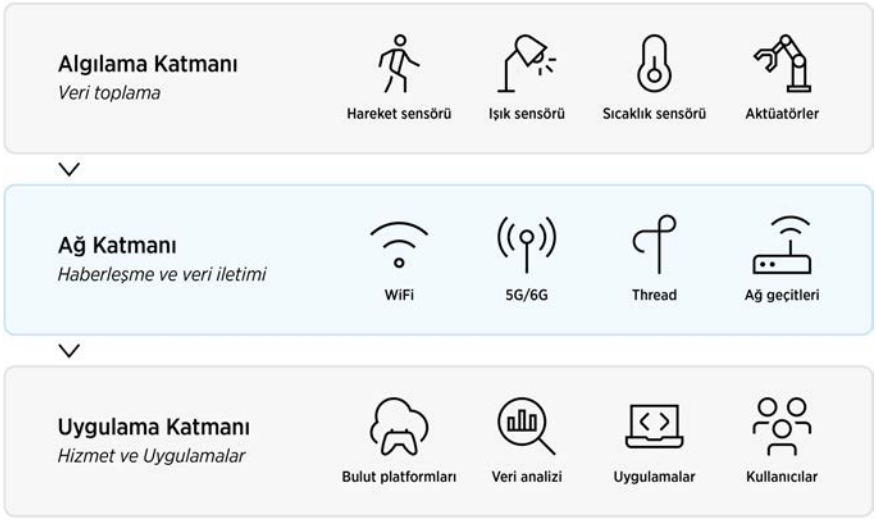
IoT sistemleri çok sayıda fiziksel ve sanal bileşenin birlikte çalıştığı, farklı teknolojilere sahip yapılardır. Bir sistem içerisinde sensörler, aktüatörler, mikrodenetleyiciler, ağ geçitleri, kablosuz haberleşme teknolojileri, veri tabanları, sunucular, bulut platformları ve kullanıcı uygulamaları aynı mimarinin farklı bölümlerini oluşturabilir. Bu nedenle IoT mimarisinin temel amacı, söz konusu bileşenler arasındaki görev dağılımını ve veri akışını sistematik biçimde tanımlamaktır.

IoT literatüründe tek bir mimari model bulunmamaktadır. Sistemin kullanım amacı, ölçeği, işlem gereksinimi ve haberleşme altyapısına bağlı olarak üç katmanlı, beş katmanlı ve daha ayrıntılı referans mimariler geliştirilmiştir. Güncel çalışmalar, özellikle uygulama mimarilerinin cloud, edge ve fog gibi farklı dağıtım modelleri üzerinden ele alındığını göstermektedir. IoT uygulama mimarileri; bulut, edge ve fog tabanlı yaklaşımlar açısından incelendiğinde, bu yapıların gecikme, ölçeklenebilirlik

ve kaynak verimliliği bakımından farklı özellikler taşıdığı görülmektedir (Dauda vd., 2024).

Üç Katmanlı IoT Mimarisi

Üç katmanlı IoT mimarisi, IoT sistemlerinin açıklanmasında kullanılan en temel modellerden biridir. Bu model algılama, ağ ve uygulama katmanlarından oluşmaktadır. Algılama katmanı fiziksel ortamdan verilerin elde edilmesini, ağ katmanı elde edilen verilerin iletilmesini, uygulama katmanı ise işlenen bilgilerin belirli bir hizmet veya uygulama kapsamında kullanılmasını temsil etmektedir. Üç katmanlı IoT sistem mimarisinin temel veri akışı Şekik 1'de gösterilmiştir.



Şekil 1. Üç katmanlı IoT sistem mimarisi

Algılama katmanında sensörler ve aktüatörler fiziksel dünya ile IoT sistemi arasında bağlantı oluşturur. Sıcaklık, nem, basınç, hareket, ışık, konum ve titreşim gibi fiziksel değişkenler sensörler aracılığıyla ölçülebilir. Ağ katmanı ise bu verilerin ağ geçitleri, kablosuz haberleşme sistemleri veya internet altyapısı üzerinden daha üst katmanlara aktarılmasını sağlar. Uygulama katmanında ise elde edilen veriler kullanıcıya sunulmakta veya belirli bir otomasyon sürecinin yürütülmesinde kullanılmaktadır.

Bu yaklaşımın en önemli avantajı IoT sistemini kavramsal olarak basit biçimde açıklamasıdır. Bununla birlikte gerçek sistemlerde veri işleme, güvenlik, servis yönetimi ve iş mantığı gibi işlevlerin yalnızca uygulama katmanı içerisinde değerlendirilmesi yeterli olmayabilir. Bu nedenle daha

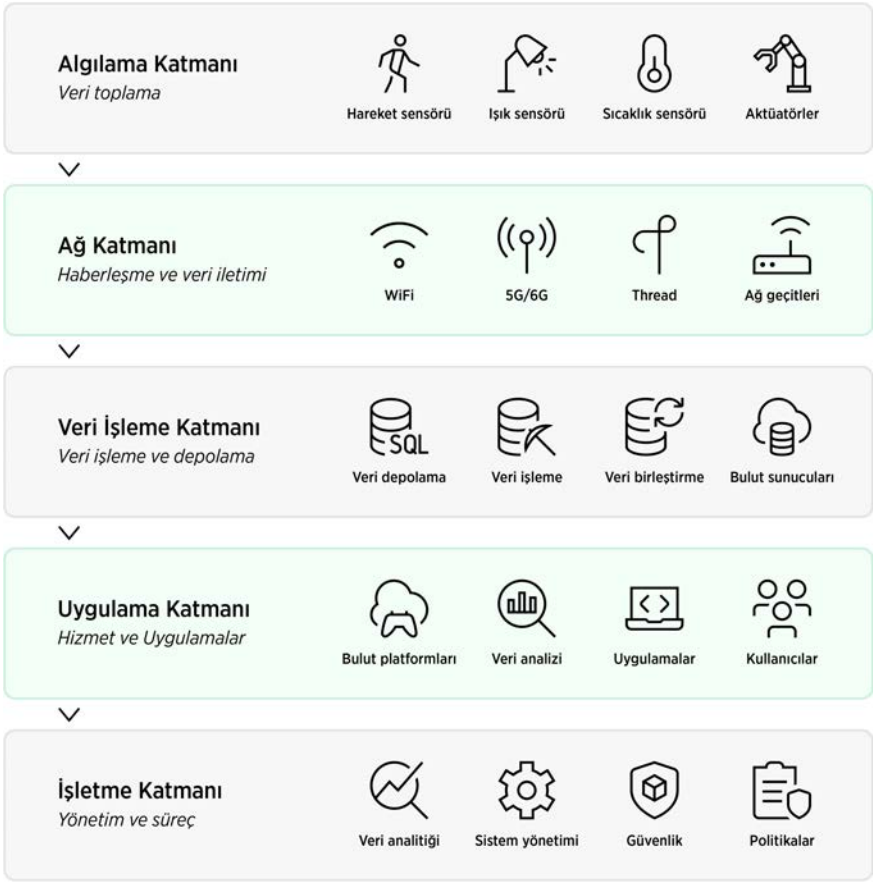
ayrıntılı mimariler geliştirilmiştir. IoT'nin sensör, işlem, haberleşme ve uygulama bileşenlerinin bütünleşik biçimde ele alınması gerektiği farklı incelemelerde de vurgulanmaktadır (Lombardi vd., 2021; Choudhary, 2024).

Beş Katmanlı IoT Mimarisi

Beş katmanlı mimari, üç katmanlı modele göre veri işleme ve yönetim süreçlerini daha ayrıntılı biçimde açıklamaktadır. Bu modelde algılama ve ağ katmanlarına ek olarak işlem (processing), uygulama ve iş (business) katmanları bulunmaktadır. Böylece fiziksel verinin elde edilmesi ile son kullanıcıya sunulan hizmet arasında gerçekleşen veri işleme ve karar verme süreçleri daha açık biçimde gösterilebilmektedir.

İşlem katmanı, IoT cihazlarından ve ağ altyapısından gelen verilerin depolanması, filtrelenmesi, analiz edilmesi ve gerektiğinde karar üretmek amacıyla kullanılmasını sağlar. Bu katmanda bulut platformları, veri tabanları, büyük veri altyapıları ve çeşitli analiz mekanizmaları yer alabilir. İş katmanı ise IoT sisteminden elde edilen sonuçların organizasyonel süreçler, iş modelleri, politika yönetimi ve karar destek mekanizmaları ile ilişkilendirilmesini mümkün kılar. Beş katmanlı IoT sistem mimarisinin aşamaları Şekil 2 ‘de gösterilmiştir.

Beş katmanlı yaklaşımın önemli bir avantajı, IoT sisteminin yalnızca teknik veri aktarımı olarak değerlendirilmesini engellemesidir. Fiziksel verinin toplanması kadar bu verinin işlenmesi, hizmete dönüştürülmesi ve işletme düzeyinde kullanılabilmesi de mimarinin bir parçası haline gelmektedir. IoT referans mimarilerinin farklı sistemlerde değişebildiği ve tek bir mimari yapının bütün uygulamalar için yeterli olmadığı belirtilmektedir (Hamdan vd., 2020; Dauda vd., 2024).



Şekil 2. Beş katmanlı IoT sistem mimarisi

Katmanlı Mimarilerin Değerlendirilmesi

Katmanlı mimarilerin temel amacı, IoT sisteminin karmaşıklığını azaltarak farklı işlevleri belirli katmanlar altında toplamaktır. Bununla birlikte katmanların fiziksel olarak birbirinden tamamen bağımsız olduğu düşünülmemelidir. Modern IoT sistemlerinde veri işleme, ağ yönetimi ve uygulama hizmetleri arasında karşılıklı etkileşim bulunmaktadır.

Örneğin bir akıllı fabrika uygulamasında sensörlerden gelen titreşim verileri algılama katmanında elde edilirken, ağ katmanında bu veriler uygun haberleşme teknolojisi üzerinden taşınabilir. İşlem katmanında veriler analiz edilerek makine arızasına ilişkin bir tahmin oluşturulabilir. Uygulama katmanı sonucu operatöre sunarken, iş katmanı bakım zamanlamasını veya üretim planlamasını değiştirebilir. Bu nedenle katmanlar veri akışının farklı

aşamalarını temsil etmekle birlikte gerçek sistemlerde birbirleriyle sürekli etkileşim içerisindedir.

IoT mimarisinin seçilmesinde sistemin ölçeği, gecikme gereksinimi, cihazların kaynak kapasitesi, veri miktarı, haberleşme altyapısı, güvenilirlik ve uygulamanın kritikliği birlikte değerlendirilmelidir. Özellikle merkezi bulut mimarisinde cihazlardan gelen verilerin veri merkezine gönderilmesi, yüksek veri hacmine sahip sistemlerde gecikme ve bant genişliği sorunlarını artırmaktadır (Kong vd., 2022; Andriulo vd., 2024).

ALGILAMA KATMANI

Algılama katmanı, IoT mimarisinin fiziksel dünya ile doğrudan etkileşim kuran bölümüdür. Bu katmanda fiziksel ortamdan veri toplayan sensörler, RFID bileşenleri, kameralar, akıllı cihazlar, giyilebilir sistemler ve çeşitli ölçüm birimleri yer almaktadır. Aynı katmanda fiziksel ortama etki eden aktüatörler de bulunabilir.

Algılama katmanının temel görevi fiziksel olayları sayısal verilere dönüştürmek ve bu verileri sistemin diğer bölümlerinde kullanılabilecek biçime getirmektir. Bu nedenle algılama katmanı yalnızca sensörlerden oluşan pasif bir yapı olarak değerlendirilmemelidir. Güncel IoT cihazları ölçüm, ön işleme, haberleşme ve belirli seviyelerde karar verme yeteneklerini aynı donanım üzerinde gerçekleştirebilmektedir.

Sensörler, Aktüatörler ve IoT Cihazları

Sensörler IoT sistemlerinin temel veri üretim bileşenleridir. Fiziksel büyüklüklerin ölçülmesini sağlayan bu bileşenler uygulama alanına göre farklı özellikler göstermektedir. Akıllı tarım sistemlerinde toprak nemi, sıcaklık, nem ve ışık gibi değişkenler ölçülürken endüstriyel sistemlerde titreşim, basınç, sıcaklık ve makine çalışma parametreleri izlenebilmektedir.

Aktüatörler ise IoT sisteminin fiziksel dünyaya müdahale etmesini sağlar. Bir motorun çalıştırılması, sulama vanasının açılması, aydınlatmanın değiştirilmesi veya bir üretim hattındaki mekanik bileşenin kontrol edilmesi aktüatörler aracılığıyla gerçekleştirilir. Böylece IoT sistemleri yalnızca veri toplayan değil, algılama ve kontrol döngüsünü birlikte gerçekleştiren siber-fiziksel yapılara dönüşmektedir.

IoT cihazlarının önemli özelliklerinden biri kaynaklarının uygulamaya göre sınırlı olabilmesidir. Düşük maliyetli sensör düğümleri genellikle düşük işlem gücü, sınırlı bellek ve düşük enerji tüketimi hedeflenerek

tasarlanmaktadır. Bu durum cihaz üzerinde gerçekleştirilebilecek veri işleme miktarını doğrudan etkilemektedir. Özellikle batarya ile çalışan cihazlarda sürekli veri işleme veya yüksek hacimli haberleşme enerji tüketimini artırmaktadır (Lombardi vd., 2021; Shaikh vd., 2022).

Veri Toplama ve Ön İşleme

IoT sistemlerinde sensörlerden sürekli veri üretilmesi, ağ üzerinden aktarılacak veri miktarının hızla artmasına neden olabilir. Bu nedenle verinin daha üst katmanlara gönderilmeden önce belirli seviyelerde ön işleme tabi tutulması önemlidir. Ön işleme; filtreleme, veri temizleme, normalleştirme, örnekleme, sıkıştırma ve veri birleştirme gibi işlemleri içerebilir.

Örneğin sıcaklık sensörünün her saniye ölçüm yapması gereken bir uygulamada, değişimin olmadığı uzun dönemlerde aynı değerin tekrar tekrar buluta gönderilmesi gereksiz ağ trafiği oluşturabilir. Benzer biçimde bir makine sensöründen elde edilen yüksek frekanslı titreşim verilerinin tamamının merkezi sunucuya aktarılması yerine belirli özelliklerin uç noktada çıkarılması daha verimli olabilir.

Veri azaltma tekniklerinin amacı yalnızca ağ trafiğini azaltmak değildir. Aynı zamanda gecikmeyi düşürmek, enerji tüketimini azaltmak ve bulut altyapısındaki işlem yükünü sınırlamak da hedeflenmektedir. IoT ve fog ortamlarında işlenen veri miktarının azaltılmasında; filtreleme, sıkıştırma, veri birleştirme ve tahmine dayalı yöntemlerden yararlanılabilmektedir (Sadri vd., 2022).

Algılama Katmanının Kısıtları

Algılama katmanının karşılaştığı temel sorunlar cihazların fiziksel koşulları, enerji kapasitesi, işlem gücü, bellek miktarı, haberleşme menzili ve cihazların heterojen yapısından kaynaklanmaktadır.

Özellikle çok sayıda sensörün aynı ortamda kullanıldığı sistemlerde cihazların farklı üreticilere ait olması ve farklı iletişim teknolojilerini kullanması uyumluluk sorunlarına neden olabilir. Bu durum ağ geçitlerinin ve ara yazılım katmanlarının önemini artırmaktadır. IoT ağ geçitleri, farklı cihazlardan gelen verilerin toplanması, ön işlenmesi ve üst sistemlere aktarılması gibi işlevleri yerine getirerek cihazlar ve ağlar arasında bir ara katman görevi üstlenmektedir (Beniwal vd., 2022).

Algılama katmanının bir diğer problemi veri kalitesidir. Sensör hataları, ölçüm gürültüsü, bağlantı kesintileri ve kalibrasyon problemleri üst

katmanlarda gerçekleştirilecek analizlerin doğruluğunu etkileyebilir. Bu nedenle mimari tasarım sırasında yalnızca sensör sayısına değil, ölçüm doğruluğuna, veri güvenilirliğine ve sensörlerin çalışma koşullarına da dikkat edilmelidir.

AĞ KATMANI

Ağ katmanı, IoT cihazları tarafından üretilen verilerin farklı sistemler arasında taşınmasını sağlayan temel iletişim altyapısını temsil etmektedir. IoT sistemlerinin farklı özelliklere sahip olması nedeniyle bu katmanda tek bir haberleşme teknolojisinin kullanılması mümkün değildir. Uygulamanın menzili, enerji gereksinimi, veri hızı, gecikme toleransı ve maliyeti gibi faktörlere bağlı olarak farklı haberleşme teknolojileri tercih edilmektedir.

IoT Haberleşme Teknolojileri

IoT sistemlerinde Wi-Fi, Bluetooth, Zigbee, Z-Wave, Thread, LoRaWAN, NB-IoT, LTE-M ve 5G gibi farklı haberleşme teknolojileri kullanılmaktadır. Bu teknolojiler birbirlerinin doğrudan alternatifi değildir. Örneğin kısa mesafeli ve yüksek veri hızlı uygulamalarda Wi-Fi tercih edilebilirken, geniş bir coğrafi alana dağılmış ve düşük enerji tüketmesi gereken sensörlerde LoRaWAN veya NB-IoT daha uygun olabilir. Thread ise düşük güç tüketimine ihtiyaç duyan IoT cihazları için geliştirilmiş, IEEE 802.15.4 tabanlı ve IPv6 destekli bir kablosuz mesh ağ protokolüdür. Cihazların birbirleri üzerinden veri iletebildiği çok atlamalı yapısı sayesinde Thread, özellikle akıllı ev ve bina otomasyonu gibi düşük güç tüketiminin önemli olduğu IoT uygulamalarında kullanılmaktadır (Kim vd., 2019; Silva vd., 2024).

LPWAN (Low Power Wide Area Network) teknolojileri, özellikle düşük veri hacmine sahip cihazların uzun mesafelerde ve düşük enerji tüketimiyle haberleşmesini sağlamak amacıyla geliştirilmiştir. Bu özellikleri nedeniyle tarım, çevresel izleme, akıllı şehir ve altyapı izleme gibi uygulamalarda kullanılabilmektedir. 5G ve LPWAN teknolojileri, akıllı şehirler ve uzak bölgelerde IoT bağlantısının yaygınlaştırılması açısından farklı gereksinimlere cevap veren tamamlayıcı haberleşme yaklaşımları olarak değerlendirilmektedir (Ogbodo vd., 2022).

IoT haberleşme teknolojilerinin seçiminde yalnızca teorik veri hızı dikkate alınmamalıdır. Enerji tüketimi, kapsama alanı, cihaz maliyeti, ağ yoğunluğu, gecikme, güvenilirlik ve uygulamanın hareketlilik gereksinimleri birlikte değerlendirilmelidir.

IoT İletişim Protokolleri

Haberleşme teknolojilerinin yanında IoT sistemlerinde uygulama düzeyinde farklı iletişim protokolleri ve standartları kullanılmaktadır. MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), HTTP, WebSocket ve Matter bu alanda kullanılan başlıca yaklaşımlar arasında yer almaktadır. MQTT ve CoAP daha çok cihazlar arasındaki veri alışverişini sağlarken, Matter farklı üreticilere ait IoT cihazlarının ortak bir uygulama modeli üzerinden birlikte çalışmasını amaçlayan bir uygulama katmanı standardı olarak öne çıkmaktadır.

MQTT, özellikle yayınla/abone ol (publish/subscribe) modeli üzerinden çalışan IoT uygulamalarında yaygın olarak kullanılmaktadır. Cihazlar doğrudan birbirleriyle iletişim kurmak yerine bir broker üzerinden mesaj alışverişi gerçekleştirebilmektedir. Bu yapı cihazların birbirine doğrudan bağlı olmasını gerektirmediğinden ölçeklenebilir IoT uygulamalarında avantaj sağlayabilmektedir.

CoAP ise sınırlı kaynaklara sahip cihazlar için tasarlanmış hafif bir protokoldür. HTTP'nin istek-cevap yaklaşımına benzer bir yapı sunmakla birlikte kısıtlı cihaz ve ağ koşullarına uygun olacak şekilde tasarlanmıştır. IoT uygulama katmanı protokollerinin uygulama gereksinimlerine göre uyarlanması ve protokol seçiminin ağ koşullarıyla birlikte değerlendirilmesi önem taşımaktadır (Donta vd., 2022).

Matter, farklı üreticilere ait IoT cihazlarının ortak bir iletişim modeli üzerinden birlikte çalışmasını amaçlayan IP tabanlı bir uygulama katmanı standardıdır. Matter belirli bir kablosuz haberleşme teknolojisine bağlı olmayıp Ethernet, Wi-Fi ve Thread gibi farklı ağ teknolojileri üzerinde çalışabilmektedir. Bu nedenle Matter, Wi-Fi, Thread veya LoRaWAN gibi doğrudan bir haberleşme teknolojisi olarak değil, farklı IoT cihazları ve ekosistemleri arasındaki uyumluluğu ve birlikte çalışabilirliği artırmaya yönelik bir uygulama katmanı standardı olarak değerlendirilmelidir. Özellikle akıllı ev ve bina otomasyonu uygulamalarında farklı üreticilere ait cihazların aynı IoT ekosistemi içerisinde kullanılmasını kolaylaştırması, Matter'ın güncel IoT mimarilerindeki önemini artırmaktadır (Belli vd., 2024; Mota vd., 2024).

Protokol seçiminde veri boyutu, mesajlaşma sıklığı, bağlantı sürekliliği, enerji tüketimi ve gecikme gibi ölçütler dikkate alınmalıdır. Bununla birlikte, tek bir protokolün tüm IoT uygulamaları için en uygun çözümü sunması

mümkün değildir. Protokol seçiminin cihazların kaynak kısıtları, ağ koşulları ve uygulamanın iletişim gereksinimleriyle birlikte değerlendirilmesi gerekmektedir (Sobin, 2020).

Ağ Performansı ve Güvenilirlik

IoT ağlarında performans değerlendirmesi veri hızı ile sınırlı değildir. Gecikme, paket kaybı, güvenilirlik, enerji tüketimi, kapsama alanı ve bağlantı sürekliliği sistem performansını belirleyen temel ölçütlerdir.

Özellikle gerçek zamanlı uygulamalarda gecikme kritik bir parametredir. Örneğin otonom araçlar, endüstriyel kontrol sistemleri veya kritik sağlık uygulamalarında verinin uzak bir sunucuya gönderilmesi ve kararın tekrar cihaza dönmesi kabul edilemeyecek gecikmeler oluşturabilir. Bu nedenle ağ mimarisinin yalnızca bağlantıyı sağlamak yerine uygulamanın zaman gereksinimlerini de karşılaması gerekir.

Heterojen IoT ortamlarında farklı kablosuz teknolojilerin aynı fiziksel ortamda bulunması da ağ performansını etkileyebilir. Farklı teknolojiler arasındaki girişim ve uyumluluk problemleri özellikle yoğun IoT ağlarında önemli hale gelmektedir. IoT ortamlarında farklı kablosuz teknolojiler arasındaki iletişim ve birlikte çalışabilirlik, bu nedenle önemli bir araştırma konusu olarak değerlendirilmektedir (He vd., 2022).

5G ve Yeni Nesil IoT Haberleşmesi

5G teknolojisi IoT sistemlerinin yüksek cihaz yoğunluğu, düşük gecikme ve yüksek veri aktarımı gereksinimlerine cevap verebilecek önemli bir haberleşme altyapısı olarak değerlendirilmektedir. Özellikle massive machine-type communications (mMTC) ve ultra-reliable low-latency communications (URLLC) gibi kullanım senaryoları IoT uygulamaları açısından önem taşımaktadır.

5G'nin IoT mimarilerine etkisi yalnızca daha hızlı bağlantı sağlamasıyla sınırlı değildir. 5G ağlarının edge computing ile kullanılması, verinin kullanıcıya veya cihazlara daha yakın işlem noktalarında değerlendirilmesine imkân sağlamaktadır. Böylece haberleşme ve işlem altyapısının birlikte tasarlanması mümkün olmaktadır.

Gelecekte 6G sistemlerinin IoT mimarilerindeki rolünün daha da artması beklenmektedir. Özellikle yapay zekâ destekli ağ yönetimi, yüksek cihaz yoğunluğu, gelişmiş konumlandırma ve hesaplama altyapılarının haberleşme sistemleriyle bütünleşmesi yeni nesil IoT mimarilerinin temel özellikleri

arasında değerlendirilmektedir. 6G ve massive IoT ortamlarında yapay zekâ, SDN ve NFV gibi teknolojilerin birlikte kullanılması da güncel araştırma konularındandır (Ogbodo vd., 2022; 6G tabanlı IoT araştırmaları).

Gelecekte 6G sistemlerinin IoT mimarilerindeki rolünün daha da artması beklenmektedir. Özellikle yapay zekâ destekli ağ yönetimi, yüksek cihaz yoğunluğu, gelişmiş konumlandırma ve hesaplama altyapılarının haberleşme sistemleriyle bütünleşmesi yeni nesil IoT mimarilerinin temel özellikleri arasında değerlendirilmektedir. 6G ve massive IoT ortamlarında yapay zekâ destekli ağ yönetimi, edge intelligence ve düşük gecikmeli haberleşme gibi yaklaşımların birlikte kullanılması da güncel araştırma konuları arasında yer almaktadır (Nguyen vd., 2021).

VERİ İŞLEME KATMANI

IoT sistemlerinde sensörlerden toplanan verilerin tek başına anlamlı bilgiye dönüşmesi için ham verilerin temizlenmesi, sınıflandırılması, depolanması, analiz edilmesi ve gerektiğinde karar mekanizmalarına aktarılması gerekir. Bu işlemler veri işleme katmanının temel görevlerini oluşturmaktadır.

Veri işleme mimarisi, IoT sisteminin performansı üzerinde doğrudan etkilidir. Geleneksel yaklaşımlarda veriler büyük ölçüde merkezi bulut sistemlerine gönderilerek işlenirken, günümüzde edge computing ve fog computing gibi yaklaşımlar veri işleme süreçlerinin veri kaynaklarına daha yakın konumlarda gerçekleştirilmesine olanak sağlamaktadır.

IoT Verilerinin İşlenmesi ve Yönetimi

IoT cihazları sürekli ve yüksek hacimli veri üretebilmektedir. Bu verilerin tamamının merkezi bir sunucuya aktarılması ağ trafiğinin artmasına, depolama ihtiyacının büyümesine ve özellikle gerçek zamanlı uygulamalarda gecikmenin artmasına neden olabilir.

Veri yönetiminin ilk aşamalarından biri gereksiz verinin azaltılmasıdır. Sensörlerden gelen veriler filtrelenebilir, benzer ölçümler birleştirilebilir veya belirli eşik değerleri dışında kalan veriler gönderilmeyebilir. Böylece ağ üzerinden taşınan veri miktarı azaltılabilir.

Bunun yanında IoT verilerinin farklı formatlarda üretilmesi de önemli bir problemdir. Farklı üreticilere ait cihazların farklı veri yapıları ve protokoller kullanması veri birleştirme süreçlerini zorlaştırmaktadır. Bu nedenle IoT platformlarında ara yazılım (middleware) ve ağ geçidi yapıları önemli bir rol

üstlenmektedir. Middleware sistemlerinin temel amacı farklı donanım ve yazılım bileşenlerini ortak bir platform altında birleştirmek ve uygulama geliştirmeyi kolaylaştırmaktır (Vikash vd., 2021).

Gerçek Zamanlı Veri İşleme

Gerçek zamanlı IoT uygulamalarında verinin yalnızca doğru olması yeterli değildir; kararın gerekli süre içerisinde üretilmesi de gerekir. Bu nedenle veri işleme mimarisinin gecikme, işlem süresi ve görev planlama gibi ölçütler açısından değerlendirilmesi gerekir.

Gerçek zamanlı uygulamalarda görevlerin hangi işlem düğümünde çalıştırılacağı önemli bir mimari karardır. Düşük karmaşıklıktaki işlemler doğrudan IoT cihazında gerçekleştirilebilirken daha yoğun işlemler edge, fog veya cloud altyapısına aktarılabilir.

IoT sistemlerinde gerçek zamanlı görev planlama üzerine yapılan çalışmalar sağlık, endüstri, akıllı şehir ve altyapı uygulamalarında gecikme, enerji tüketimi ve işlem süresi gibi ölçütlerin öne çıktığını göstermektedir (Abolhassani Khajeh vd., 2022).

Bu nedenle veri işleme katmanının tasarımında "veri nerede işlenmelidir?" sorusu temel bir mimari karar haline gelmiştir. Bu sorunun cevabı uygulamanın gecikme gereksinimi, veri hacmi, cihaz kaynakları, bağlantı kalitesi ve güvenilirlik gereksinimleri birlikte değerlendirilerek verilmelidir.

Dağıtık Veri İşleme ve Yapay Zekâ

IoT sistemlerinin gelişmesiyle birlikte veri işleme yalnızca merkezi sunucuların görevi olmaktan çıkmıştır. Edge ve fog sistemleri sayesinde verinin kaynağına daha yakın işlem yapılabilmektedir. Bunun yanında yapay zekâ algoritmalarının IoT sistemlerine dahil edilmesi, sensör verilerinden otomatik tahmin ve karar üretme imkânı sağlamaktadır.

Örneğin bir üretim hattında titreşim sensörlerinden elde edilen veriler kullanılarak makine arızası önceden tahmin edilebilir. Geleneksel cloud mimarisinde bu verilerin tamamı uzak sunucuya aktarılırken edge tabanlı bir sistemde ilk analiz doğrudan üretim hattına yakın bir işlem düğümünde gerçekleştirilebilir.

AIoT (Artificial Intelligence of Things), IoT ile yapay zekânın bir araya gelmesini ifade etmektedir. AIoT sistemlerinde cihazlar yalnızca veri üretmemekte, aynı zamanda belirli seviyelerde veri analizi ve karar verme

yeteneği de kazanmaktadır. AIoT yaklaşımı, mimari açıdan algılama, veri işleme ve akıllı karar verme süreçlerinin bir arada yürütülmesini desteklemektedir (Zhang ve Tao, 2020).

Edge AI yaklaşımı ise yapay zekâ işlemlerinin veri kaynağına yakın şekilde gerçekleştirilmesini hedeflemektedir. Böylece büyük miktardaki ham verinin merkezi sisteme gönderilmesine gerek kalmadan gerçek zamanlı analiz yapılabilmektedir. Edge AI çalışmalarında kaynak kısıtları nedeniyle model boyutu, işlem gücü, enerji tüketimi ve çıkarım süresi önemli tasarım kriterleri haline gelmektedir (Edge AI araştırmaları; Ahmad vd., 2023).

UYGULAMA KATMANI

Uygulama katmanı, IoT altyapısının kullanıcıya veya belirli bir işletme sürecine doğrudan hizmet sunduğu katmandır. Bu katmanda sensörlerden elde edilen veriler kullanıcı arayüzlerine, karar destek sistemlerine, otomasyon mekanizmalarına veya farklı bilgi sistemlerine aktarılmaktadır.

IoT uygulamalarının çeşitlenmesi nedeniyle uygulama katmanı oldukça geniş bir kullanım alanına sahiptir. Akıllı evler, akıllı şehirler, sağlık sistemleri, akıllı tarım, endüstriyel otomasyon, enerji yönetimi, ulaşım ve lojistik bunların başlıca alanlardır.

IoT Uygulama Alanları

Akıllı şehirlerde IoT sistemleri trafik yönetimi, park yönetimi, çevresel izleme, enerji tüketimi, atık yönetimi ve kamu güvenliği gibi alanlarda kullanılmaktadır. Bu uygulamalarda çok sayıda sensörün şehir genelinde dağıtılması nedeniyle ölçeklenebilirlik ve veri yönetimi önemli hale gelmektedir. Akıllı şehirlerde edge computing kullanımı, büyük veri hacmi ve gecikme gereksinimi nedeniyle özellikle önem kazanmıştır (Khan vd., 2020).

Sağlık alanında IoT cihazları hasta takibi, giyilebilir cihazlar, uzaktan izleme ve klinik karar destek uygulamalarında kullanılmaktadır. Sağlık uygulamalarında veri sürekliliği ve düşük gecikme kadar sistemin güvenilirliği de önemlidir. Fog tabanlı sağlık IoT mimarileri, cihazlardan gelen verilerin sağlık hizmetlerine daha yakın işlem noktalarında değerlendirilmesine olanak sağlamaktadır (Quy vd., 2022).

Akıllı tarım uygulamalarında sensörler aracılığıyla toprak, hava ve bitki koşulları izlenebilmekte; elde edilen veriler sulama, gübreleme ve üretim planlamasında kullanılabilmektedir. IoT tabanlı tarım sistemlerinde

sensörler, iletişim altyapısı, veri işleme ve karar destek bileşenlerinin birlikte çalışması gerekmektedir (Quy vd., 2022; Shaikh vd., 2022).

Endüstriyel IoT (IIoT) sistemlerinde ise sensörlerden elde edilen veriler üretim makinelerinin izlenmesi, kalite kontrol ve üretim optimizasyonu amacıyla kullanılmaktadır. IIoT mimarilerinin diğer IoT uygulamalarından farklı olarak yüksek güvenilirlik, düşük gecikme, süreklilik ve endüstriyel protokollerle bütünleşme gibi gereksinimleri bulunmaktadır (Mirani vd., 2022; Arnold vd., 2022).

Uygulama Gereksinimleri ve Hizmet Yönetimi

Farklı IoT uygulamalarının mimari gereksinimleri aynı değildir. Çevresel izleme sisteminde birkaç dakikalık gecikme kabul edilebilirken, endüstriyel kontrol veya otonom araç uygulamalarında milisaniye seviyesinde tepki gerekebilir.

Bu nedenle uygulama katmanı ile alt katmanlar arasında karşılıklı bir ilişki bulunmaktadır. Uygulamanın gerektirdiği gecikme, güvenilirlik, veri işleme kapasitesi ve hareketlilik özellikleri mimarinin alt katmanlarını da belirlemektedir.

Uygulama gereksinimleri ayrıca işlem yerinin seçimini de etkiler. Büyük verinin uzun dönemli analizinde cloud daha uygun olabilirken, anlık karar gerektiren uygulamalarda edge işlem daha avantajlı olabilir. Bu yaklaşım, IoT sistemlerinin tek bir işlem paradigması yerine farklı işlem noktalarının birlikte kullanıldığı hibrit mimarilere doğru gelişmesine neden olmuştur (Gkonis vd., 2023).

EDGE–FOG–CLOUD MİMARİLERİ

IoT sistemlerinin gelişiminde cloud computing önemli bir rol üstlenmiştir. Bulut altyapıları yüksek işlem gücü, büyük depolama kapasitesi ve merkezi yönetim imkânı sağlamaktadır. Ancak IoT cihazlarının sayısının artması ve gerçek zamanlı uygulamaların yaygınlaşması, bütün verilerin uzak bulut merkezlerine gönderilmesinin bazı durumlarda yetersiz kalmasına neden olmuştur.

Bu sorunlara çözüm olarak işlem ve depolama kaynaklarının veri kaynağına yaklaştırılması yaklaşımı geliştirilmiştir. Edge computing veri kaynağına yakın işlem yapılmasını sağlarken fog computing IoT cihazları ile cloud arasında dağıtık bir işlem katmanı oluşturmaktadır.

Cloud Tabanlı IoT Mimarileri

Cloud tabanlı IoT mimarisinde sensör ve cihazlardan elde edilen veriler ağ üzerinden merkezi bulut altyapısına gönderilmektedir. Bulutta yüksek işlem kapasitesi ve depolama imkânı sayesinde büyük miktardaki verinin analiz edilmesi mümkün olmaktadır.

Bu mimarinin önemli avantajları merkezi yönetim, yüksek ölçeklenebilirlik, güçlü işlem kapasitesi ve uzun dönemli veri depolamadır. Bununla birlikte uzak veri merkezine sürekli veri aktarılması bant genişliği kullanımını artırabilmekte ve gecikmeye neden olabilmektedir.

IoT sistemlerinde cloud merkezli mimariler, yüksek işlem gücü ve depolama kapasitesi gerektiren uygulamalar için önemli bir yaklaşım sunmaktadır. Bununla birlikte, gerçek zamanlı uygulamalarda düşük gecikme ve hızlı karar verme gereksinimleri, veri işleme süreçlerinin edge ve fog gibi daha yakın işlem noktalarına dağıtılmasını gerektirmektedir. Bu nedenle güncel IoT mimarilerinde cloud, edge ve fog kaynaklarının birlikte kullanıldığı daha dağıtık ve esnek işlem yapıları öne çıkmaktadır (Hemmati vd., 2026).

Edge Computing

Edge computing, işlem ve veri depolama kaynaklarının IoT cihazlarına veya veri üretim noktalarına yakın konumlandırılması yaklaşımıdır. Bu sayede ham verinin tamamının merkezi cloud sistemine gönderilmesine gerek kalmadan ilk işlem ve analizler edge düğümlerinde gerçekleştirilebilir.

Edge computing özellikle düşük gecikme gerektiren uygulamalar için önemlidir. Veri kaynağı ile işlem noktası arasındaki fiziksel mesafenin azalması, iletişim gecikmesini düşürebilmektedir. Bunun yanında edge üzerinde veri filtreleme ve ön işleme yapılması, cloud sistemlerine gönderilecek veri miktarını da azaltabilir.

Hamdan vd. (2020), IoT için edge computing mimarilerini veri yerleşimi, güvenlik ve büyük veri gibi farklı özellikler açısından sınıflandırmaktadır. Kong vd. (2022) ise edge computing'in IoT sistemlerinde cloud üzerindeki yükü azaltarak daha düşük gecikmeli ve ölçeklenebilir mimariler oluşturabileceğini ortaya koymaktadır.

Fog Computing

Fog computing, IoT cihazları ile cloud altyapısı arasında yer alan dağıtık işlem ve depolama katmanını ifade etmektedir. Fog düğümleri ağın farklı

noktalarında bulunabilir ve cihazlara cloud veri merkezlerinden daha yakın işlem hizmeti sunabilir.

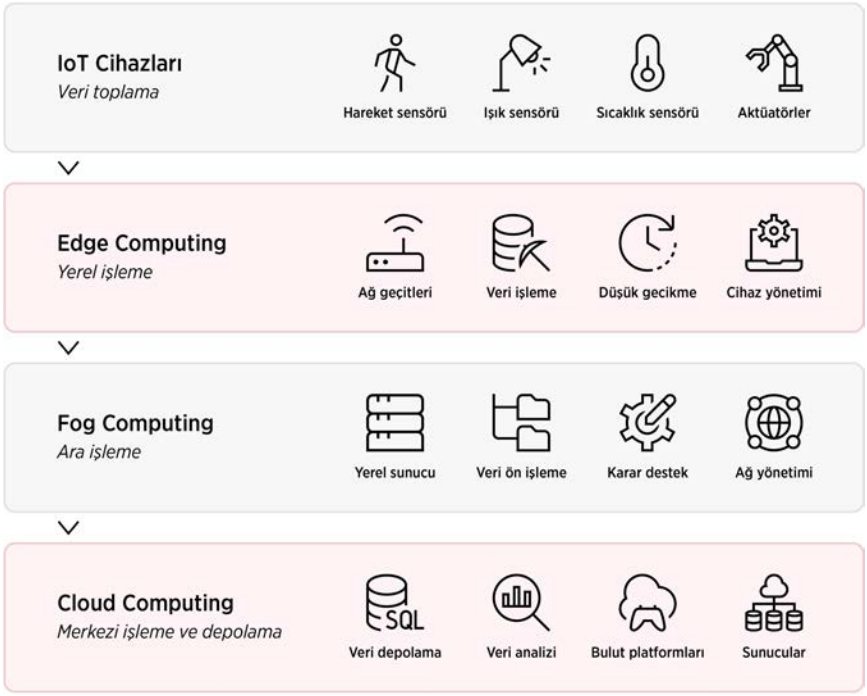
Fog yaklaşımı edge computing ile benzer amaçlara sahip olmakla birlikte daha geniş bir ağ perspektifine sahiptir. Fog düğümleri ağ geçitleri, yönlendiriciler, yerel sunucular veya diğer ağ bileşenleri üzerinde konumlandırılabilir. Böylece veri işleme, depolama ve ağ yönetimi IoT–cloud hattının farklı noktalarına dağıtılabilir.

Habibi vd. (2020), fog computing mimarisini çok katmanlı ve dağıtık bir yapı olarak ele almakta; hesaplama, ağ, depolama ve hızlandırma kaynaklarının uç noktaya yakın konumlandırılmasının IoT uygulamalarında performans avantajı sağlayabileceğini belirtmektedir. Benzer şekilde Laroui vd. (2021) cloud, fog ve edge yaklaşımlarını IoT ortamında gecikme, kaynak yönetimi ve uygulama gereksinimleri açısından değerlendirmektedir.

Fog mimarilerinin önemli özelliklerinden biri, cloud altyapısının tamamen ortadan kaldırılmasını gerektirmemesidir. Uzun dönemli depolama ve yoğun hesaplama işlemleri cloud üzerinde gerçekleştirilirken zaman kritik işlemler fog veya edge seviyesinde yapılabilir. Böylece farklı işlem seviyeleri arasında görev dağılımı sağlanabilir.

Edge–Fog–Cloud Bütünleşik Mimarisi

Güncel IoT sistemlerinde edge, fog ve cloud yaklaşımlarının birbirinin alternatifi olarak değil, birlikte çalışan katmanlar olarak ele alınması giderek önem kazanmaktadır.



Şekil 3. Edge–Fog–Cloud tabanlı IoT mimarisi

Şekil 3'de gösterilen yapıda sensörler ve cihazlar veriyi üretmekte, edge düğümleri veriye en yakın işlem seviyesini oluşturmaktadır. Fog katmanı farklı edge düğümlerinden gelen verilerin birleştirilmesi ve daha kapsamlı işlemlerin gerçekleştirilmesini sağlayabilir. Cloud ise merkezi depolama, büyük ölçekli analiz, model eğitimi ve uzun dönemli veri yönetimi gibi işlemleri üstlenebilir.

Kumar ve Agrawal (2023), özellikle endüstriyel IoT ortamlarında Edge–Fog–Cloud mimarilerinin yüksek hacimli ve çok boyutlu verilerin analizinde önemli bir yaklaşım olduğunu ortaya koymaktadır. Bu mimarilerde veri işleme seviyesinin seçimi; gecikme, enerji, veri hacmi ve uygulamanın işlem gereksinimleri dikkate alınarak yapılmaktadır.

Edge–Fog–Cloud mimarisinin temel avantajı işlem yükünün tek bir merkezi noktada toplanmamasıdır. Buna karşılık dağıtık mimari daha karmaşık kaynak yönetimi, görev planlama, veri senkronizasyonu ve sistem orkestrasyonu gerektirmektedir. Bu nedenle mimarinin avantajları ile yönetim karmaşıklığı birlikte değerlendirilmelidir (Martinez vd., 2020; Costa vd., 2022).

IOT SİSTEM MİMARİLERİNİN KARŞILAŞTIRILMASI

IoT sistem mimarisinin seçiminde tek bir modelin bütün uygulamalar için en iyi çözüm olduğu söylenemez. Mimari yaklaşım uygulamanın ihtiyaçlarına göre belirlenmelidir. Merkezi cloud mimarisi yüksek işlem ve depolama kapasitesi sağlarken edge yaklaşımı düşük gecikme ve yerel veri işleme açısından avantaj sağlamaktadır. Fog mimarisi ise bu iki yaklaşım arasında dağıtık bir işlem alanı oluşturabilmektedir.

Mimari Yaklaşımların Teknik Özellikleri

Tablo 1. IoT sistem mimarilerinin karşılaştırılması

	Cloud IoT	**Edge IoT**	**Fog IoT**	**Edge–Fog–Cloud**
Mimari	Merkezi	Dağıtık	Dağıtık	Çok seviyeli
İşlem Konumu	Merkezi veri merkezi	Veri kaynağına yakın	Ağ üzerinde dağıtık düğümler	Çok seviyeli
Gecikme	Yüksek/Orta	Düşük	Düşük/Orta	Uygulamaya bağlı
Ölçeklenebilirlik	Yüksek	Orta	Yüksek	Çok yüksek
Bant Genişliği Kullanımı	Yüksek	Düşük/Orta	Orta	Optimize edilebilir
Tipik Kullanım	Büyük veri, uzun dönemli analiz	Gerçek zamanlı uygulamalar	Dağıtık IoT sistemleri	Büyük ölçekli karmaşık IoT

Tablo 1'de mimariler arasında temel farklılık işlem kaynaklarının fiziksel konumundan kaynaklandığı görülmektedir. Cloud mimarisi merkezi işlem kapasitesi sağlarken, edge ve fog yaklaşımları ise işlemi veri kaynağına yaklaştırmaktadır. Bu farklılıkların özellikle gecikme, bant genişliği ve kaynak kullanımı açısından önemli olduğu belirtilmektedir (Yousefpour vd., 2019; Laroui vd., 2021; Andriulo vd., 2024).

Performans, Gecikme ve Kaynak Kullanımı

IoT mimarilerinin performans değerlendirmesinde gecikme en önemli kriterlerden biridir. Cloud merkezli sistemlerde verinin cihazdan veri merkezine taşınması ve sonuçların tekrar cihaza gönderilmesi ek iletişim

süresi oluşturabilir. Edge mimarilerinde ise işlem kaynağının veri üretim noktasına yakın olması bu gecikmeyi azaltabilir.

Bununla birlikte edge cihazlarının işlem kapasitesi cloud sistemlerine göre sınırlı olabilir. Dolayısıyla bütün işlemlerin edge üzerinde gerçekleştirilmesi de her zaman optimum çözüm değildir. Özellikle büyük makine öğrenmesi modellerinin eğitimi, uzun dönemli veri analizi ve büyük ölçekli depolama işlemleri cloud altyapılarında daha uygun şekilde gerçekleştirilebilir.

Görevlerin hangi katmana gönderileceğine ilişkin kararın otomatikleştirilmesi bu nedenle önemli bir araştırma alanıdır. IoT-cloud-edge iş birliğinde görev bölümlendirme ve offloading mekanizmaları, cihaz kaynaklarının yetersiz olduğu durumlarda işlem yükünün daha güçlü sistemlere aktarılmasını sağlamaktadır (Chen vd., 2022; Dayong vd., 2024).

Uygulama Alanlarına Göre Mimari Seçimi

Tablo 2. IoT uygulamalarında mimari seçimi

Uygulama	Temel Gereksinim	Mimari
Akıllı ev	Konfor, otomasyon, düşük maliyet	Edge–Cloud
Akıllı şehir	Ölçeklenebilirlik, çok sayıda cihaz	Fog–Edge–Cloud
Akıllı tarım	Geniş alan, düşük enerji	Edge–Cloud / Fog
Akıllı sağlık	Düşük gecikme, süreklilik	Edge–Fog–Cloud
Endüstriyel IoT	Gerçek zamanlı, güvenilirlik	Edge–Fog–Cloud
Büyük veri analizi	Yüksek işlem ve depolama	Cloud
Otonom sistemler	Düşük gecikme	Edge

Tablo 2'de uygulama alanı mimari seçiminin temel belirleyici özelliklerindendir. Örneğin akıllı tarımda geniş coğrafi alan ve düşük enerji tüketimi önemliyken endüstriyel IoT sistemlerinde gerçek zamanlılık ve yüksek güvenilirlik daha ön plandadır. Sağlık uygulamalarında ise düşük gecikmenin yanında veri işleme sürekliliği ve güvenilirliği önem taşımaktadır (Dwivedi vd., 2022; Al-Rawashdeh vd., 2022).

GÜNCEL YAKLAŞIMLAR VE GELECEK EĞİLİMLERİ

IoT sistem mimarileri günümüzde yalnızca sensör, ağ ve cloud bileşenlerinden oluşan klasik yapılardan daha karmaşık bir noktaya ulaşmıştır. Yapay zekâ, edge computing, 5G/6G, dijital ikiz, dağıtık öğrenme ve mikroservis gibi teknolojiler IoT mimarilerinin işlevlerini genişletmektedir.

AIoT ve Edge AI

AIoT, IoT cihazlarının ürettiği verilerin yapay zekâ algoritmaları kullanılarak analiz edilmesini ve sistemin belirli seviyelerde otomatik karar verebilmesini sağlamaktadır. Bu yaklaşımda sensörlerden gelen veriler yalnızca depolanmak yerine tahmin, sınıflandırma, anomali tespiti veya optimizasyon amacıyla kullanılmaktadır.

Edge AI ise yapay zekâ modellerinin edge seviyesinde çalıştırılmasını ifade etmektedir. Böylece görüntü, ses veya sensör verileri gibi yüksek hacimli verilerin tamamının cloud sistemine gönderilmesi yerine analiz işlemleri kaynağa yakın gerçekleştirilir. Bu yaklaşım özellikle gerçek zamanlı görüntü analizi, endüstriyel kalite kontrol, otonom sistemler ve akıllı sağlık uygulamalarında önem taşımaktadır.

Edge AI'ın önemli bir problemi, yapay zekâ modellerinin IoT cihazlarının sınırlı kaynaklarıyla uyumlu hale getirilmesidir. Model sıkıştırma, quantization, pruning ve dağıtık öğrenme gibi teknikler bu nedenle önem kazanmaktadır. Edge AI üzerine yapılan güncel araştırmalar, düşük kaynaklı cihazlarda model doğruluğu ile işlem maliyeti arasında denge kurulmasının temel sorunlardan biri olduğunu göstermektedir (Ahmad vd., 2023).

Federated learning de bu bağlamda önemli bir yaklaşım olarak öne çıkmaktadır. Modelin merkezi sunucuya ham veri göndermeden farklı edge cihazlarında eğitilmesi, veri aktarımını azaltabilmekte ve bazı uygulamalarda veri mahremiyetini destekleyebilmektedir. Federated learning'in edge-IoT ortamlarında veri güvenliği ve mahremiyet açısından sağladığı avantajlar tespit edilmiştir (Li vd., 2024).

5G/6G Destekli IoT Mimarileri

5G, IoT sistemlerinin yoğun cihaz bağlantısı ve düşük gecikme gereksinimlerine cevap verebilecek önemli bir teknoloji olarak

görülmektedir. Edge computing ile birlikte kullanıldığında 5G, haberleşme ve işlem altyapısının birbirine daha yakın bütünleşmesini sağlayabilir.

6G ile birlikte IoT sistemlerinin daha yüksek cihaz yoğunluğu, yapay zekâ destekli ağ yönetimi, gelişmiş algılama ve hesaplama-haberleşme bütünleşmesi gibi özelliklere sahip olması beklenmektedir. Bu durum gelecekte IoT mimarilerinin yalnızca veri taşıyan ağlar değil, algılama, iletişim, hesaplama ve karar verme işlevlerini birlikte gerçekleştiren dağıtık sistemlere dönüşmesini destekleyecektir.

Bu gelişmeler özellikle akıllı şehirler, otonom ulaşım, endüstriyel otomasyon ve geniş ölçekli sensör ağlarında önem kazanmaktadır. 5G ve LPWAN teknolojilerinin farklı kullanım senaryolarında birlikte değerlendirilmesi de geniş kapsamlı IoT bağlantısı açısından önemli bir yaklaşım oluşturmaktadır (Ogbodo vd., 2022).

Dijital İkiz ve Endüstriyel IoT

Dijital ikiz (Digital Twin), fiziksel bir varlığın veya sistemin dijital ortamda temsil edilmesini ve fiziksel sistemden gelen gerçek zamanlı verilerle bu temsilin güncellenmesini sağlayan bir yaklaşımdır. IoT sensörleri dijital ikizin fiziksel sistemden veri almasını sağlarken edge, fog ve cloud altyapıları bu verilerin işlenmesinde kullanılabilmektedir.

Endüstriyel IoT ortamında dijital ikizler üretim makinelerinin izlenmesi, kestirimci bakım, proses optimizasyonu ve sanal testler için kullanılabilmektedir. Bu nedenle dijital ikizler IoT mimarisinin yalnızca uygulama katmanında çalışan bir yazılım değil, fiziksel sistem ile veri işleme altyapısı arasında sürekli veri alışverişi sağlayan bütünleşik bir mimari bileşen olarak değerlendirilebilir.

Sharma vd. (2022), dijital ikiz teknolojisinin fiziksel ve dijital sistemler arasında sürekli etkileşim sağlayarak gerçek zamanlı izleme, analiz ve optimizasyon süreçlerini desteklediğini belirtmektedir. Dijital ikizlerin gelişiminde IoT, büyük veri ve makine öğrenmesi gibi teknolojilerin önemli rol oynadığı; özellikle endüstriyel uygulamalarda bu teknolojilerin dijital ikiz sistemlerinin yeteneklerini geliştirdiği görülmektedir (Lu vd., 2020; Xu vd., 2023).

SONUÇ VE ÖNERİLER

IoT sistem mimarileri, fiziksel dünyadan elde edilen verilerin anlamlı bilgiye ve hizmete dönüştürülmesini sağlayan bütünleşik yapıların temelini

oluşturmaktadır. IoT sistemlerinin gelişmesiyle birlikte mimari yapıların da yalnızca cihazların internete bağlanmasını sağlayan basit modellerden, dağıtık işlem, yapay zekâ, yüksek hızlı haberleşme ve akıllı karar mekanizmalarını içeren karmaşık sistemlere doğru geliştiği görülmektedir.

Bu çalışmada IoT sistem mimarileri algılama, ağ, veri işleme ve uygulama katmanları üzerinden ele alınmış; üç ve beş katmanlı mimarilerin temel özellikleri açıklanmıştır. Algılama katmanının fiziksel dünya ile sistem arasındaki bağlantıyı oluşturduğu, ağ katmanının farklı cihaz ve sistemler arasındaki veri iletişimini sağladığı, veri işleme katmanının ham verileri anlamlı bilgiye dönüştürdüğü ve uygulama katmanının elde edilen bilgileri kullanıcıya veya işletme süreçlerine sunduğu görülmektedir.

IoT mimarisinin seçiminde tek bir yapının bütün uygulamalar için uygun olmadığı sonucuna ulaşılmaktadır. Uygulamanın gecikme gereksinimi, veri hacmi, enerji tüketimi, işlem kapasitesi, ölçeklenebilirlik ve güvenilirlik gereksinimleri mimari seçiminde birlikte değerlendirilmelidir. Bu açıdan cloud, edge ve fog computing yaklaşımlarının birbirinin alternatifi olarak değil, farklı görevleri üstlenebilen tamamlayıcı mimari bileşenler olarak değerlendirilmesi daha uygun olacaktır.

Gelecekte IoT sistemlerinin daha fazla yapay zekâ destekli, dağıtık ve otonom hale gelmesi beklenmektedir. Özellikle Edge AI, 5G/6G, dijital ikiz ve AIoT gibi yaklaşımlar IoT sistem mimarilerinin işlem, haberleşme ve karar verme yeteneklerini önemli ölçüde geliştirecektir. Bununla birlikte mimari karmaşıklığın artması, sistemlerin yönetimi ve birlikte çalışabilirliği açısından yeni sorunları da beraberinde getirecektir. Bu nedenle IoT sistemlerinin tasarımında yalnızca performansın artırılması değil; farklı teknolojilerin birlikte çalışabilmesi, kaynakların etkin kullanılması, gecikmenin azaltılması, enerji tüketiminin kontrol edilmesi ve sistemlerin ölçeklenebilir olması gibi disiplinler birlikte ele alınmalıdır.

Sonuç olarak IoT sistem mimarilerinin geleceği, tek merkezli ve statik yapılardan ziyade fiziksel cihazlar, ağ altyapıları, dağıtık işlem kaynakları, yapay zekâ ve bulut sistemlerinin birlikte çalıştığı dinamik mimarilere doğru ilerlemektedir. Bu dönüşümün başarılı biçimde gerçekleştirilebilmesi için mimari tasarımın uygulamanın teknik gereksinimleri doğrultusunda yapılması ve farklı katmanların bir bütün olarak değerlendirilmesi gerekmektedir.

KAYNAKÇA

Abolhassani Khajeh, S., Saberikamarposhti, M., & Rahmani, A. M. (2022). Real-time scheduling in IoT applications: A systematic review. Sensors, 23(1), 232.

Ahmad, S., Shakeel, I., Mehfuz, S., & Ahmad, J. (2023). Deep learning models for cloud, edge, fog, and IoT computing paradigms: Survey, recent advances, and future directions. Computer Science Review, 49, 100568.

Al-Rawashdeh, M., Keikhosrokiani, P., Belaton, B., Alawida, M., & Zwiri, A. (2022). IoT adoption and application for smart healthcare: a systematic review. Sensors, 22(14), 5377.

Andriulo, F. C., Fiore, M., Mongiello, M., Traversa, E., & Zizzo, V. (2024, September). Edge computing and cloud computing for internet of things: A review. In Informatics (Vol. 11, No. 4, p. 71). MDPI.

Arnold, L., Jöhnk, J., Vogt, F., & Urbach, N. (2022). IIoT platforms' architectural features–a taxonomy and five prevalent archetypes: IIoT platforms' architectural features–a taxonomy and five prevalent archetypes. Electronic Markets, 32(2), 927-944.

Belli, D., Barsocchi, P., & Palumbo, F. (2024). Connectivity Standards Alliance Matter: State of the art and opportunities. Internet of Things, 25, 101005.

Beniwal, G., & Singhrova, A. (2022). A systematic literature review on IoT gateways. Journal of King Saud University-Computer and Information Sciences, 34(10), 9541-9563.

Choudhary, A. (2024). Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. Discover Internet of Things, 4(1), 31.

Chen, H., Qin, W., & Wang, L. (2022). Task partitioning and offloading in IoT cloud-edge collaborative computing framework: a survey. Journal of Cloud Computing, 11(1), 86.

Costa, B., Bachiega Jr, J., De Carvalho, L. R., & Araujo, A. P. (2022). Orchestration in fog computing: A comprehensive survey. ACM computing surveys (CSUR), 55(2), 1-34.

Dauda, A., Flauzac, O., & Nolot, F. (2024). A survey on IoT application architectures. Sensors, 24(16), 5320.

Dayong, W., Bakar, K. B. A., Isyaku, B., Eisa, T. A. E., & Abdelmaboud, A. (2024). A comprehensive review on internet of things task offloading in multi-access edge computing. Heliyon, 10(9).

Donta, P. K., Srirama, S. N., Amgoth, T., & Annavarapu, C. S. R. (2022). Survey on recent advances in IoT application layer protocols and machine learning scope for research directions. Digital Communications and Networks, 8(5), 727-744.

Dwivedi, R., Mehrotra, D., & Chandra, S. (2022). Potential of Internet of Medical Things (IoMT) applications in building a smart healthcare system: A systematic review. Journal of oral biology and craniofacial research, 12(2), 302-318.

Gkonis, P., Giannopoulos, A., Trakadas, P., Masip-Bruin, X., & D'Andria, F. (2023). A survey on IoT-edge-cloud continuum systems: Status, challenges, use cases, and open issues. Future Internet, 15(12), 383.

Habibi, P., Farhoudi, M., Kazemian, S., Khorsandi, S., & Leon-Garcia, A. (2020). Fog computing: a comprehensive architectural survey. IEEE access, 8, 69105-69133.

Hamdan, S., Ayyash, M., & Almajali, S. (2020). Edge-computing architectures for internet of things applications: A survey. Sensors, 20(22), 6441.

He, Y., Guo, X., Zheng, X., Yu, Z., Zhang, J., Jiang, H., ... & Zhang, J. (2022). Cross-technology communication for the Internet of Things: A survey. ACM Computing Surveys, 55(5), 1-29.

Hemmati, A., Khaledian, N., & Rahmani, A. M. (2026). Toward a unified computing paradigm: A survey and roadmap for data management in integrated IoT, fog, and cloud services. Peer-to-Peer Networking and Applications, 19(2), 54.

Khan, L. U., Yaqoob, I., Tran, N. H., Kazmi, S. A., Dang, T. N., & Hong, C. S. (2020). Edge-computing-enabled smart cities: A comprehensive survey. IEEE Internet of Things journal, 7(10), 10200-10232.

Kim, H. S., Kumar, S., & Culler, D. E. (2019). Thread/OpenThread: A compromise in low-power wireless multihop network architecture for the Internet of Things. IEEE Communications Magazine, 57(7), 55-61.

Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., ... & Das, S. K. (2022). Edge-computing-driven internet of things: A survey. ACM computing surveys, 55(8), 1-41.

Kumar, R., & Agrawal, N. (2023). Analysis of multi-dimensional Industrial IoT (IIoT) data in Edge–Fog–Cloud based architectural frameworks: A survey on current state and research challenges. Journal of Industrial Information Integration, 35, 100504.

Laroui, M., Nour, B., Moungla, H., Cherif, M. A., Afifi, H., & Guizani, M. (2021). Edge and fog computing for IoT: A survey on current research activities & future directions. Computer Communications, 180, 210-231.

Li, H., Ge, L., & Tian, L. (2024). Survey: federated learning data security and privacy-preserving in edge-Internet of Things: H. Li et al. Artificial Intelligence Review, 57(5), 130.

Lombardi, M., Pascale, F., & Santaniello, D. (2021). Internet of things: A general overview between architectures, protocols and applications. Information, 12(2), 87.

Lu, Y., Liu, C., Kevin, I., Wang, K., Huang, H., & Xu, X. (2020). Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robotics and computer-integrated manufacturing, 61, 101837.

Martinez, I., Hafid, A. S., & Jarray, A. (2020). Design, resource management, and evaluation of fog computing systems: a survey. IEEE Internet of Things Journal, 8(4), 2494-2516.

Mirani, A. A., Velasco-Hernandez, G., Awasthi, A., & Walsh, J. (2022). Key challenges and emerging technologies in industrial IoT architectures: A review. Sensors, 22(15), 5836.

Mota, A., Serôdio, C., & Valente, A. (2024). Matter protocol integration using Espressif's solutions to achieve smart home interoperability. Electronics, 13(11), 2217.

Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., Niyato, D., ... & Poor, H. V. (2021). 6G Internet of Things: A comprehensive survey. IEEE Internet of Things Journal, 9(1), 359-383.

Ogbodo, E. U., Abu-Mahfouz, A. M., & Kurien, A. M. (2022). A survey on 5G and LPWAN-IoT for improved smart cities and remote area applications: From the aspect of architecture and security. Sensors, 22(16), 6313.

Quy, V. K., Hau, N. V., Anh, D. V., Quy, N. M., Ban, N. T., Lanza, S., ... & Muzirafuti, A. (2022). IoT-enabled smart agriculture: architecture, applications, and challenges. Applied Sciences, 12(7), 3396.

Quy, V. K., Hau, N. V., Anh, D. V., & Ngoc, L. A. (2022). Smart healthcare IoT applications based on fog computing: architecture, applications and challenges. Complex & Intelligent Systems, 8(5), 3805-3815.

Sadri, A. A., Rahmani, A. M., Saberikamarposhti, M., & Hosseinzadeh, M. (2022). Data reduction in fog computing and internet of things: A systematic literature survey. Internet of Things, 20, 100629.

Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., & Calinescu, A. (2022). Digital twins: State of the art theory and practice, challenges, and open research questions. Journal of Industrial Information Integration, 30, 100383.

Shaikh, F. K., Karim, S., Zeadally, S., & Nebhen, J. (2022). Recent trends in internet-of-things-enabled sensor technologies for smart agriculture. IEEE Internet of Things Journal, 9(23), 23583-23598.

Silva, J. A. L., Costa, W., Islam, K. M. M., Deters, J. K., Bergsma, E., Rocha, H. R. O., ... & Wörtche, H. (2024). Improving the Power Consumption of Thread Mesh Networks Through Genetic Algorithm Optimization. IEEE Sensors Letters, 8(12), 1-4.

Sobin, C. C. (2020). A Survey on Architecture, Protocols and Challenges in IoT: CC Sobin. Wireless Personal Communications, 112(3), 1383-1429.

Vikash, Mishra, L., & Varma, S. (2021). Middleware Technologies for Smart Wireless Sensor Networks towards Internet of Things: A Comparative Review: Vikash et al. Wireless personal communications, 116(3), 1539-1574.

Xu, H., Wu, J., Pan, Q., Guan, X., & Guizani, M. (2023). A survey on digital twin for industrial internet of things: Applications, technologies and tools. IEEE Communications Surveys & Tutorials, 25(4), 2569-2598.

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., ... & Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of systems architecture, 98, 289-330.

Zhang, J., & Tao, D. (2020). Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. IEEE Internet of Things Journal, 8(10), 7789-7817.

SENSÖRLER, AKTÜATÖRLER VE GÖMÜLÜ SİSTEMLER

Sensör Türleri, Aktüatörler, Mikrodenetleyiciler ve Veri Toplama Süreçlerine Bütüncül Bir Bakış

Ayhan Arısoy*

GİRİŞ

Bilgi işlem sistemlerinin fiziksel dünya ile kurduğu bağ, büyük ölçüde sensörler, aktüatörler ve bu ikisi arasında karar mekanizması işlevi gören gömülü sistemler aracılığıyla gerçekleşir. Bir sensör, sıcaklık, basınç, ivme veya ışık gibi fiziksel bir büyüklüğü işlenebilir bir elektriksel ya da dijital veriye dönüştürürken; bir aktüatör, işlemcinin ürettiği karar veya komutu fiziksel bir harekete ya da etkiye çevirir. Bu iki bileşen arasında yer alan mikrodenetleyici veya mikroişlemci tabanlı gömülü sistem ise, algılanan veriyi işleyerek anlamlı bir çıktıya dönüştüren hesaplama katmanını oluşturur. Bu üçlü yapı, günümüzde çok sayıda teknolojik alanın temelini oluşturmaktadır.

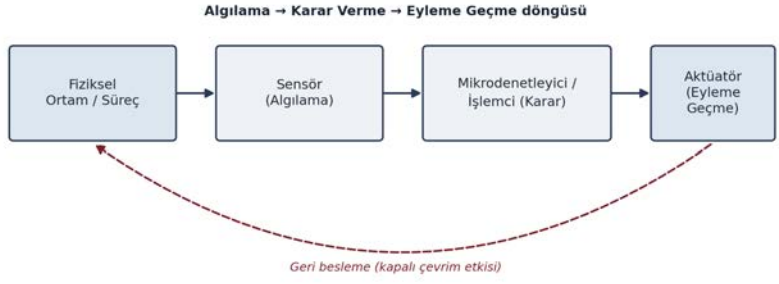
Nesnelerin İnterneti (Internet of Things – IoT) kavramı, birbirine bağlı sensör ve aktüatör düğümlerinin ağ altyapısı üzerinden veri paylaşarak ortak bir işlevi yerine getirdiği sistemleri tanımlamak için kullanılır (Chui vd., 2023). Bu sistemlerin fiziksel dünya ile hesaplama dünyası arasında sıkı bir etkileşim kurması, onları siber-fiziksel sistemler (cyber-physical systems) kategorisinde de değerlendirilebilir kılar; burada fiziksel süreçler sensörler aracılığıyla dijital ortama taşınır, hesaplama katmanında işlenir ve aktüatörler aracılığıyla yeniden fiziksel dünyaya geri döner. Bu döngüsel yapı, Endüstri 4.0 ve onun insan merkezli, sürdürülebilirlik odaklı devamı olarak tartışılan Endüstri 5.0 yaklaşımlarının teknik omurgasını oluşturur; sensör tabanlı veri akışının ürün tasarımı ve üretim süreçlerine doğrudan entegre edilmesi, bu dönüşümün somut bir görünümüdür (Adel, 2022; Rosário & Dias, 2023; Varshney vd., 2021). Üretim hatlarındaki durum izleme sensörlerinden akıllı şehirlerdeki trafik ve çevre izleme ağlarına, akıllı tarımda toprak nemi ve iklim verisi toplayan düğümlerden (Sinha & Dhanalakshmi, 2021) giyilebilir sağlık teknolojilerindeki biyomedikal

* Dr. Öğr. Üyesi, Manisa Celal Bayar Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesi, Veri Bilimi ve Analitiği Anabilim Dalı, Manisa, ayhan.arisoy@cbu.edu.tr, ORCID: orcid.org/0000-0001-6754-932X

sensörlere kadar uzanan geniş bir uygulama yelpazesi, aynı temel mimariyi -algılama, işleme ve eyleme geçme- farklı ölçeklerde tekrar eder.

Robotik ve otomasyon sistemleri de bu üçlü yapının doğrudan bir uzantısıdır: bir robot kolun çevresini algılaması, konumunu ve görevini hesaplaması ve ardından eklemlerini hareket ettirmesi, sensör-işlemci-aktüatör zincirinin klasik bir örneğidir. Yapay zekâ tabanlı sistemlerin yaygınlaşmasıyla birlikte bu zincire bir katman daha eklenmiştir: sensörlerden toplanan ham veri, yalnızca basit eşik değerleriyle değil, öğrenilmiş modeller aracılığıyla da yorumlanabilmektedir. Bu modellerin bir kısmının artık bulut sunucularında değil, doğrudan sensöre yakın uçtaki gömülü donanımda çalıştırılması -edge computing olarak adlandırılan yaklaşım-, gerçek zamanlı veri toplama sistemlerinde gecikmeyi azaltmak, ağ trafiğini sınırlamak ve mahremiyeti korumak açısından giderek daha fazla tercih edilmektedir.

Sensör, işlemci ve aktüatör arasındaki ilişki, Şekil 1'de gösterilen kapalı çevrim mimarisi üzerinden genelleştirilebilir. Bu mimaride sensör, fiziksel ortamdaki bir değişimi algılayarak işlemciye iletir; işlemci bu veriyi önceden tanımlanmış bir mantık, kural kümesi veya öğrenilmiş bir model aracılığıyla değerlendirir; aktüatör ise üretilen kararı yeniden fiziksel ortama uygular. Ortamda meydana gelen değişim, sensör tarafından yeniden algılanarak döngü sürdürülür. Bu geri beslemeli yapı, yalnızca basit bir otomasyon devresi için değil, karmaşık siber-fiziksel sistemlerin ve otonom sistemlerin de kavramsal temelini oluşturur.



Şekil 1. Sensör–İşlemci–Aktüatör Kapalı Çevrim Mimarisi

Bu bölümün amacı, sensörler, aktüatörler ve gömülü sistemlerin temel çalışma ilkelerini akademik bir çerçevede ele almak; Arduino, ESP32 ve Raspberry Pi gibi yaygın donanım platformlarını nesnel ölçütler üzerinden karşılaştırmak ve sensörlerden elde edilen verinin fiziksel ortamdan alınmasından işlenmesine, iletilmesine ve depolanmasına kadar uzanan veri

toplama sürecini bütüncül biçimde ortaya koymaktır. Bölüm, Bilgisayar Mühendisliği, Yazılım Mühendisliği, Yapay Zekâ, Veri Bilimi, Elektrik-Elektronik Mühendisliği ve Mekatronik gibi alanlardaki lisans öğrencileri ile konuya giriş yapmak isteyen araştırmacıların ihtiyaç duyacağı kavramsal ve teknik alt yapıyı sağlayacak biçimde düzenlenmiştir. İzleyen kısımlarda sırasıyla sensörlerin sınıflandırılması ve performans özellikleri, aktüatör türleri ve sürme yöntemleri, gömülü sistem mimarisi, mikroişlemci/mikrodenetleyici kavramları, Arduino, ESP32 ve Raspberry Pi platformlarının karşılaştırmalı incelemesi, sensör-mikrodenetleyici haberleşme protokolleri, uçtan buluta veri toplama süreci, IoT tabanlı mimariler, edge computing ve TinyML yaklaşımları, veri güvenliği konuları, örnek uygulama senaryoları ve alandaki güncel eğilimler ele alınacaktır.

SENSÖRLER

Sensör Kavramı ve Temel Tanımlar

Bir **sensör**, sıcaklık, ışık şiddeti, basınç, ivme veya nem gibi fiziksel ya da kimyasal bir büyüklüğü, işlenebilir bir elektriksel sinyale ya da dijital veriye dönüştüren bileşen olarak tanımlanır. Sensör kavramı sıklıkla **transdüser** (transducer) kavramıyla birlikte anılır; transdüser, bir enerji türünü başka bir enerji türüne dönüştüren daha genel bir bileşendir ve sensörler, girdisi fiziksel bir büyüklük, çıktısı ise ölçülebilir bir elektriksel işaret olan transdüser alt kümesini oluşturur (Fraden, 2016). Bu dönüşüm süreci, ölçülen büyüklüğün fiziksel doğasına bağlı olarak direnç, kapasitans, endüktans, gerilim veya frekans değişimi biçiminde ortaya çıkabilir.

Sensörler, ürettikleri çıktının yapısına göre **analog sensör** ve **dijital sensör** olarak ikiye ayrılır. Analog bir sensör, ölçülen büyüklükle sürekli ve genellikle doğrusal bir ilişki içinde değişen bir gerilim ya da akım üretir; bu sinyalin mikrodenetleyici tarafından işlenebilmesi için analog-dijital dönüştürücüden (ADC) geçirilmesi gerekir. Dijital bir sensör ise, çoğunlukla iç yapısında bulunan entegre bir ADC ve sinyal işleme devresi sayesinde ölçüm sonucunu doğrudan ikili (binary) veri veya seri haberleşme çerçevesi biçiminde sunar; bu durum mikrodenetleyici tarafındaki devre karmaşıklığını azaltır ancak sensörün kendi iç örnekleme hızı ve çözünürlüğüyle sınırlı kalınmasına yol açar.

İkinci önemli ayrım, sensörün ölçüm için dışarıdan enerjiye ihtiyaç duyup duymamasına dayanır. **Pasif sensör**, ölçüm sırasında dış ortamdan gelen enerjiyi (örneğin ışık veya ısı) doğrudan elektriksel sinyale dönüştürür ve

kendi başına bir uyarıcı sinyal üretmez; fotodiyot ve termokupl bu kategoriye örnektir. **Aktif sensör** ise ölçüm yapabilmek için kendisi bir uyarıcı sinyal (örneğin ultrasonik dalga veya kızılötesi ışın) yayar ve bu sinyalin ortamla etkileşiminden dönen tepkiyi değerlendirir; ultrasonik mesafe sensörleri ve radar tabanlı sensörler bu grupta yer alır. Bu ayrım, sensörün güç tüketimi, tepki süresi ve ortamdan bağımsızlık düzeyi açısından doğrudan sonuç doğurur.

Sensörlerin Temel Performans Özellikleri

Bir sensörün belirli bir uygulama için uygunluğu, yalnızca ölçtüğü büyüklükle değil, aynı zamanda bir dizi performans ölçütüyle de belirlenir. **Hassasiyet** (sensitivity), sensörün çıktısındaki değişimin, ölçülen büyüklükteki değişime oranı olarak tanımlanır ve sensörün transfer fonksiyonunun eğimine karşılık gelir:

$$S = \Delta Y / \Delta X \quad (1)$$

Burada S sensörün duyarlılığını, ΔY sensör çıktısındaki (örneğin gerilim) değişimi, ΔX ise ölçülen fiziksel büyüklükteki değişimi ifade eder. Yüksek duyarlılık, küçük fiziksel değişimlerin dahi algılanabilmesini sağlar; ancak aşırı yüksek duyarlılık, sensörü gürültüye karşı da daha savunmasız hale getirebilir.

Doğruluk (accuracy), sensörün ürettiği ölçüm sonucunun gerçek değere ne kadar yakın olduğunu ifade ederken; **kesinlik** ya da **tekrarlanabilirlik** (precision/repeatability), aynı koşullar altında tekrarlanan ölçümlerin birbirine ne ölçüde yakın sonuçlar verdiğini gösterir. Bir sensör yüksek kesinliğe sahip olduğu hâlde düşük doğruluğa sahip olabilir; bu durum, ölçümlerin birbirine yakın ancak gerçek değerden sistematik olarak sapmış olduğu anlamına gelir ve genellikle kalibrasyon eksikliğinden kaynaklanır.

Çözünürlük (resolution), sensörün ayırt edebildiği en küçük değişim miktarıdır ve dijital sensörlerde doğrudan çıktı bit sayısıyla ilişkilidir.

Ölçüm aralığı, sensörün güvenilir biçimde çalışabildiği alt ve üst sınırları tanımlar; bu aralığın dışında sensör doyuma ulaşabilir veya hatalı sonuçlar üretebilir.

Doğrusallık, sensörün gerçek transfer fonksiyonunun ideal doğrusal bir çizgiden ne ölçüde saptığını; **tepki süresi**, sensörün girişteki bir değişikliğe karşı kararlı çıktı değerine ulaşması için geçen süreyi ifade eder.

Histerezis, artan ve azalan yönde uygulanan aynı büyüklük için sensörün farklı çıktı değerleri üretmesi durumudur ve genellikle mekanik veya manyetik gecikmelerden kaynaklanır.

Gürültü, ölçülen büyüklükle ilişkili olmayan rastgele dalgalanmaları; **kalibrasyon**, sensör çıktısının bilinen referans değerlerle karşılaştırılarak düzeltilmesi sürecini ifade eder.

Drift ise sensör çıktısının zamanla, sıcaklık değişimi, yaşlanma veya bileşen bozunumu gibi nedenlerle yavaşça kaymasıdır ve periyodik yeniden kalibrasyon gerektirir. Bu özelliklerin birlikte değerlendirilmesi, bir sensörün belirli bir gömülü sistem uygulaması için ne ölçüde uygun olduğunu belirler.

Sensör Türlerinin Sınıflandırılması

Sıcaklık sensörleri

Sıcaklık sensörleri, bir ortamın veya nesnenin termal enerjisini elektriksel bir büyüklüğe dönüştürür ve başlıca üç teknoloji üzerine kuruludur: termokupllar, direnç sıcaklık dedektörleri (RTD) ve termistörler. Termokupllar, iki farklı metalin birleşim noktasında sıcaklık farkına bağlı olarak ortaya çıkan Seebeck gerilimini ölçer; geniş sıcaklık aralıklarında çalışabilmeleri nedeniyle endüstriyel fırın ve motor uygulamalarında tercih edilirler, ancak çıktıları düşük genlikli olduğundan hassas yükseltme devrelerine ihtiyaç duyarlar. RTD'ler, metal direncin sıcaklıkla değişimini temel alır ve yüksek doğruluk sunar, fakat maliyetleri ve tepki süreleri termistörlere kıyasla daha yüksektir. Termistörler ise yarı iletken malzemelerin direncinin sıcaklıkla güçlü biçimde (genellikle negatif yönde) değişmesi ilkesine dayanır; düşük maliyetleri ve yüksek duyarlılıkları nedeniyle tüketici elektroniği ve gömülü sistem projelerinde yaygın olarak kullanılırlar. Dijital sıcaklık sensörleri (örneğin I2C veya tek-hat arabirimli entegre devreler), iç yapılarında ADC ve doğrusallaştırma devresi barındırdıklarından mikrodenetleyiciyle doğrudan sayısal haberleşme kurabilir; bu da kalibrasyon yükünü büyük ölçüde azaltır. Sınırlılıkları arasında sınırlı sıcaklık aralığı ve nispeten düşük örnekleme hızı sayılabilir. Gömülü sistemlerde bu sensörler, çevresel izleme, soğutma kontrolü ve endüstriyel süreç otomasyonu gibi geniş bir yelpazede kullanılır.

Nem sensörleri

Nem sensörleri, havadaki su buharı oranını ölçmek için genellikle kapasitif veya rezistif prensiplere dayanır. Kapasitif nem sensörlerinde, iki

elektrot arasına yerleştirilen nem tutucu polimer tabakanın dielektrik sabiti bağıl nemle birlikte değişir ve bu değişim kapasitans üzerinden ölçülür; doğrusallıkları ve kararlılıkları nedeniyle günümüzde en yaygın kullanılan yaklaşımdır. Rezistif nem sensörleri ise nem tutucu bir tuz veya polimer tabakanın elektriksel direncinin nemle değişiminden yararlanır; daha düşük maliyetli olmakla birlikte doğrusallık ve uzun vadeli kararlılık açısından kapasitif muadillerinin gerisinde kalırlar. Gömülü sistem projelerinde yaygın olarak kullanılan entegre sıcaklık-nem sensörleri, kapasitif algılama elemanını bir ADC ve dijital arabirimle birleştirerek tek-hat veya I2C üzerinden doğrudan sayısal veri sağlar; bu da sensörün mikrodenetleyiciye entegrasyonunu büyük ölçüde basitleştirir. Nem sensörlerinin başlıca sınırlılığı, kirlenmeye ve zamanla oluşan drift'e duyarlı olmaları ve bu nedenle periyodik kalibrasyon gerektirmeleridir. Tarımsal izleme, iklimlendirme sistemleri ve depolama ortamı kontrolü bu sensörlerin tipik kullanım alanlarındandır.

Basınç sensörleri

Basınç sensörleri, günümüzde büyük ölçüde mikro-elektromekanik sistem (MEMS) teknolojisiyle üretilmekte olup piezorezistif ve kapasitif olmak üzere iki temel çalışma ilkesine dayanır. Piezorezistif basınç sensörlerinde, ince bir silikon diyaframın üzerine yerleştirilen direnç elemanları, basınç altında diyaframın esnemesiyle birlikte mekanik gerinime maruz kalır ve bu gerinim direnç değişimine, dolayısıyla ölçülebilir bir gerilim farkına dönüşür. Kapasitif basınç sensörlerinde ise diyaframın hareketi, sabit bir elektrotla arasındaki kapasitansı değiştirir; bu yaklaşım genellikle daha düşük güç tüketimi ve daha yüksek sıcaklık kararlılığı sunar (Han vd., 2023). MEMS tabanlı üretim, sensörlerin küçültülmesini, düşük maliyetli seri üretimini ve mikrodenetleyici tabanlı sistemlere doğrudan entegrasyonunu mümkün kılmıştır. Basınç sensörleri; hava basıncı ve rakım ölçümünden endüstriyel hidrolik/pnömatik hat izlemeye, giyilebilir cihazlardaki barometrik yükseklik tahminine kadar geniş bir uygulama alanına sahiptir. Dezavantajları arasında sıcaklığa bağlı ofset kayması ve yüksek doğruluk gerektiren uygulamalarda düzenli kalibrasyon ihtiyacı sayılabilir.

Işık sensörleri

Işık sensörleri, ortamdaki elektromanyetik ışımayı elektriksel sinyale dönüştürür. Fotodirenç (LDR – light dependent resistor), üzerine düşen ışık miktarına bağlı olarak direnci değişen pasif bir yarı iletken elemandır; düşük

maliyeti nedeniyle basit aydınlatma kontrolü uygulamalarında yaygındır, ancak tepki süresi görece yavaştır ve doğrusallığı sınırlıdır. Fotodiyotlar, ters kutuplanmış bir p-n eklemin ışığa maruz kaldığında ürettiği akımdan yararlanır ve hızlı tepki süresiyle öne çıkar; fototransistörler ise benzer bir ilkeyle çalışmakla birlikte iç kazanç sağladıklarından daha yüksek çıkış akımı üretir. Dijital ortam ışığı sensörleri, fotodiyot dizisini bir ADC ve I2C arabirimiyle birleştirerek lüks cinsinden doğrudan sayısal ölçüm sunar. Işık sensörleri; otomatik parlaklık ayarı, gün ışığına duyarlı aydınlatma sistemleri, güvenlik amaçlı hareket-ışık kombinasyonları ve tarımsal ışık yönetimi gibi uygulamalarda kullanılır.

Mesafe, yakınlık, ultrasonik ve kızılötesi sensörler

Mesafe ve yakınlık ölçümü, gömülü sistemlerde farklı fiziksel ilkelere dayanan birkaç sensör ailesiyle gerçekleştirilir. Ultrasonik sensörler, insan işitme aralığının üzerinde bir frekansta ses dalgası yayarak bu dalganın bir engelden yansıyıp geri dönmesi için geçen süreyi ölçer; ses hızının bilinmesiyle bu süre doğrudan mesafeye çevrilir. Bu yaklaşım, aktif bir sensör örneği olup optik olarak şeffaf veya koyu renkli yüzeylerde dahi güvenilir ölçüm sağlar, ancak yumuşak veya ses emici yüzeylerde ve dar açılı hedeflerde performansı düşebilir. Kızılötesi (infrared) mesafe ve yakınlık sensörleri ise ya yayılan kızılötesi ışının yansıma yoğunluğunu ya da üçgenleme (triangulation) ilkesiyle yansıma açısını değerlendirerek mesafe tahmini yapar; tepki süreleri kısa ve maliyetleri düşüktür, fakat parlak veya çok koyu yüzeylerden ve doğrudan güneş ışığından etkilenebilirler. Kızılötesi yakınlık sensörlerinin bir alt kümesi olan pasif kızılötesi (PIR) sensörler, hareket eden bir cismin yaydığı kızılötesi ısı değişimini algılayarak hareket tespiti yapar ve bu nedenle mesafe ölçümünden ziyade varlık/hareket algılama amacıyla kullanılır. Bu sensör ailesi; robotik engel algılama, otomatik kapı sistemleri, park yardımcıları, seviye ölçümü ve güvenlik sistemlerinde yoğun biçimde kullanılır ve mikrodenetleyicilere genellikle dijital tetikleme-yankı (trigger-echo) hatları veya doğrudan analog çıkış üzerinden bağlanır.

Gaz ve hava kalitesi sensörleri

Gaz sensörleri, havadaki belirli kimyasal bileşenlerin (karbon monoksit, karbon dioksit, uçucu organik bileşikler vb.) konsantrasyonunu elektriksel sinyale dönüştürür. En yaygın kullanılan aile, metal oksit yarı iletken (MOx) sensörlerdir; bu sensörlerde ısıtılmış bir metal oksit yüzeyin hedef gaz molekülleriyle etkileşimi, yüzey direncinde ölçülebilir bir değişime yol açar.

MOx sensörleri düşük maliyetli ve çok çeşitli gazlara duyarlı olmakla birlikte, seçicilikleri sınırlıdır ve sıcaklık-nem gibi çevresel etkenlere karşı hassastırlar; bu nedenle güvenilir ölçüm için ısınma süresi ve düzenli kalibrasyon gerektirirler (Al-Okby vd., 2022). Elektrokimyasal gaz sensörleri, hedef gazın bir elektrot yüzeyinde gerçekleştirdiği redoks reaksiyonundan doğan akımı ölçerek daha yüksek seçicilik sunar ve özellikle toksik gaz tespitinde tercih edilir. Dağınık kızılötesi (NDIR) sensörler ise gazın belirli bir kızılötesi dalga boyundaki soğurumunu ölçerek özellikle karbon dioksit tayininde yüksek doğruluk sağlar. Gömülü sistemlerle entegrasyonları genellikle analog çıkış üzerinden ADC'ye veya dijital arabirim üzerinden doğrudan mikrodenetleyiciye bağlanma biçiminde gerçekleşir. Bu sensörler; iç mekân hava kalitesi izleme, endüstriyel güvenlik, akıllı havalandırma ve çevresel kirlilik takibi gibi uygulamalarda kullanılır.

Eylemsizlik sensörleri: ivmeölçerler, jiroskoplar, manyetometreler ve ımu sistemleri

İvmeölçerler, bir cismin doğrusal ivmesini, genellikle MEMS yapılı küçük bir test kütlesinin eylemsizlik kuvveti altındaki yer değiştirmesinin kapasitif veya piezorezistif olarak ölçülmesiyle belirler; yer çekimi ivmesinin bileşenlerinden yararlanarak eğim tahmini de yapılabilir. Jiroskoplar, açısal hızı ölçer; MEMS jiroskoplarının büyük kısmı, titreşen bir kütlenin Coriolis kuvveti altında maruz kaldığı yer değiştirmeyi algılama ilkesine dayanır. Manyetometreler ise yerel manyetik alan bileşenlerini ölçerek pusula işlevi görür ve mutlak yön bilgisi sağlar. Bu üç sensör türü tek başına kullanıldığında farklı sınırlılıklar taşır: ivmeölçerler titreşime duyarlıdır, jiroskoplar zamanla drift gösterir ve manyetometreler yakın çevredeki manyetik parazitlerden etkilenir. Sensör füzyonu, farklı sensörlerden veya aynı büyüklüğü farklı ilkelerle ölçen sensörlerden gelen verilerin, tek başına hiçbir sensörün sağlayamayacağı kadar güvenilir ve tutarlı bir tahmin üretmek amacıyla birleştirilmesi ilkesine dayanır. Bu nedenle uygulamalarda genellikle üçü bir arada, **eylemsizlik ölçüm birimi (IMU – inertial measurement unit)** olarak paketlenir ve sensör füzyonu algoritmaları (örneğin tamamlayıcı filtre veya genişletilmiş Kalman filtresi) aracılığıyla birleştirilerek daha kararlı bir yönelim tahmini elde edilir (Nazarahari & Rouhani, 2021). IMU sistemleri; drone ve robot yönelim kontrolü, hareket tanıma, adım sayma, sanal/artırılmış gerçeklik başlıkları ve

araç dengeleme sistemlerinde yaygın olarak kullanılır ve mikrodenetleyicilere çoğunlukla I2C veya SPI arabirimi üzerinden bağlanır.

Kuvvet ve basınç algılama elemanları

Kuvvet ölçümüne yönelik sensörlerin başında, gerinim ölçer (strain gauge) tabanlı yük hücreleri (load cell) gelir; bu elemanlarda uygulanan kuvvet, metal bir yapının mikroskobik düzeyde deforme olmasına ve buna bağlı direnç değişimine yol açar. Bu değişim genellikle bir Wheatstone köprüsü ile yükseltilerek ölçülür ve yüksek doğruluk gerektiren tartım ve kuvvet ölçüm uygulamalarında tercih edilir. Daha düşük maliyetli bir alternatif olan kuvvete duyarlı dirençler (FSR – force sensitive resistor), üzerine uygulanan basınçla birlikte iletkenliği artan polimer tabanlı bir yapıya sahiptir; doğrusallıkları ve tekrarlanabilirlikleri yük hücrelerine kıyasla sınırlı olmakla birlikte, dokunma algılama ve basit basınç eşiği uygulamaları için yeterlidir. Bu sensör ailesi; endüstriyel tartım sistemleri, robotik tutma mekanizmalarında dokunsal geri bildirim, giyilebilir basınç haritalama ve insan-makine arayüzlerinde kullanılır.

Ses sensörleri

Ses sensörleri, hava basıncındaki akustik dalgalanmaları elektriksel sinyale dönüştürür. Elektret kondansatör mikrofonlar, sabit bir yüklü membran ile arka plaka arasındaki kapasitans değişimini ölçerken; MEMS mikrofonlar benzer bir kapasitif ilkeyi silikon tabanlı, minyatürize bir yapı üzerinde uygular ve yüzey montaj teknolojisiyle doğrudan devre kartına entegre edilebilir. Ses sensörleri, gömülü sistemlere genellikle analog çıkış üzerinden bir yükseltici devre ve ADC ile ya da dijital PDM/I2S arabirimi üzerinden bağlanır. Basit gürültü seviyesi algılamadan ses komutu tanıma sistemlerine, endüstriyel arıza teşhisinde akustik izlemeden güvenlik amaçlı cam kırılma algılamaya kadar geniş bir kullanım alanına sahiptirler; sınırlılıkları arasında ortam gürültüsüne duyarlılık ve yüksek örnekleme hızı gereksinimi nedeniyle artan işlem yükü sayılabilir.

Kamera ve görüntü sensörleri

Görüntü sensörleri, bir sahneden gelen ışığı piksel dizisi üzerinde elektrik yüküne, ardından dijital bir görüntüye dönüştürür. Günümüzde CMOS (complementary metal-oxide-semiconductor) tabanlı görüntü sensörleri, düşük güç tüketimi, düşük maliyet ve piksel düzeyinde entegre okuma devresi sunmaları nedeniyle CCD (charge-coupled device) sensörlere kıyasla gömülü sistemlerde baskın hâle gelmiştir. CMOS sensörler genellikle bir seri

kamera arabirimi (örneğin MIPI CSI) veya I2C tabanlı kontrol hattı üzerinden ana işlemciye bağlanır; buna karşılık ürettikleri veri hacminin büyüklüğü, işlemci ve bellek kaynakları açısından mikrodenetleyicilerden çok, daha güçlü işlem kapasitesine sahip tek kart bilgisayarları (örneğin Raspberry Pi) veya özel görüntü işleme çekirdekleri gerektirir. Kamera sensörleri; nesne tanıma, yüz tanıma, kalite kontrol amaçlı makine görüşü, güvenlik izleme ve otonom araç algılama sistemlerinde kullanılır ve genellikle görüntü işleme algoritmaları ya da derin öğrenme tabanlı modellerle birlikte çalıştırılır.

Biyomedikal sensörler

Biyomedikal sensörler, insan vücuduna ait fizyolojik sinyalleri ölçerek sağlık izleme ve teşhis amaçlı gömülü sistemlerde kullanılır. Elektrokardiyografi (ECG) sensörleri, kalbin elektriksel aktivitesini cilt yüzeyine yerleştirilen elektrotlar aracılığıyla ölçer ve kalp ritmi ile ilgili değerli bilgi sağlar; giyilebilir ECG sistemleri, klasik klinik cihazlara kıyasla daha az elektrotla ve sürekli izleme imkânıyla öne çıkar, ancak hareket kaynaklı sinyal bozulmasına (artifact) karşı daha hassastırlar (Dahiya vd., 2024). Fotopletismografi (PPG) sensörleri, deri altı kan damarlarındaki hacim değişimini bir ışık kaynağı ve fotodetektör çifti aracılığıyla optik olarak ölçer; bu yöntem kalp atış hızı ve oksijen satürasyonu tahmini için yaygın olarak giyilebilir cihazlarda kullanılır ve ECG'ye kıyasla daha az elektrot/temas gerektirmesi nedeniyle tüketici cihazlarında tercih edilir, ancak hareket artefaktlarına ve deri tonu/perfüzyon farklılıklarına karşı duyarlıdır (Castaneda vd., 2018). Elektromiyografi (EMG) sensörleri kas aktivitesini elektriksel sinyal olarak ölçerken, biyosensörler (örneğin glikoz sensörleri) belirli bir biyokimyasal reaksiyonu elektrokimyasal sinyale dönüştürür. Bu sensörler; giyilebilir sağlık takip cihazları, uzaktan hasta izleme sistemleri, spor performans analizi ve rehabilitasyon teknolojilerinde yaygın olarak kullanılmakta olup mikrodenetleyicilere genellikle analog ön yükseltme devresi ve ADC üzerinden ya da entegre dijital sağlık izleme yongaları aracılığıyla bağlanırlar.

Tablo 1. Başlıca Sensör Türleri ve Özellikleri

Sensör Türü	Ölçülen Büyüklük	Çalışma Prensibi	Çıkış Türü	Tipik Kullanım Alanı
Termistör / RTD / Termokupl	Sıcaklık	Direnç veya Seebeck geriliminin sıcaklıkla değişimi	Analog / Dijital	Çevresel izleme, endüstriyel süreç kontrolü
Kapasitif nem sensörü	Bağıl nem	Polimer dielektrik tabakanın kapasitans değişimi	Analog / Dijital	Tarım, iklimlendirme
MEMS piezorezistif/kapasitif	Basınç	Diyafram gerinimi veya kapasitans değişimi	Analog / Dijital	Rakım, hidrolik/pnömatik izleme
LDR / Fotodiyot	Işık şiddeti	Işığa bağlı direnç veya akım değişimi	Analog	Aydınlatma kontrolü, güvenlik
Ultrasonik sensör	Mesafe	Ses dalgası uçuş süresi (time-of-flight)	Dijital (darbe genişliği)	Engel algılama, seviye ölçümü
Kızılötesi (IR) sensör	Mesafe / Hareket	Yansıma yoğunluğu veya üçgenleme	Analog / Dijital	Yakınlık algılama, hareket tespiti
MOx / elektrokimyasal gaz sensörü	Gaz konsantrasyonu	Yüzey direnci veya redoks akımı değişimi	Analog / Dijital	Hava kalitesi, endüstriyel güvenlik
MEMS ivmeölçer/jiroskop (IMU)	İvme / Açısal hız / Yönelim	Kapasitif yer değiştirme, Coriolis kuvveti	Dijital (I2C/SPI)	Hareket tanıma, dengeleme, robotik
Yük hücresi / FSR	Kuvvet / Basınç	Gerinime bağlı direnç değişimi	Analog	Tartım, dokunsal algılama
MEMS mikrofon	Ses basıncı	Kapasitif membran hareketi	Analog / Dijital (PDM)	Ses tanıma, akustik izleme
CMOS görüntü sensörü	Görüntü / Işık deseni	Fotodiyot dizisinde yük birikimi	Dijital (paralel/seri)	Makine görüşü, güvenlik
ECG / PPG sensörü	Kalp elektriksel/optik sinyali	Biyoelektrik potansiyel veya optik absorpsiyon	Analog / Dijital	Giyilebilir sağlık izleme

AKTÜATÖRLER

Aktüatör, bir kontrol sinyalini fiziksel bir harekete, kuvvete veya başka bir gözlenebilir etkiye dönüştüren bileşendir ve bu yönüyle sensörün tam tersi bir işlev üstlenir: sensör fiziksel dünyadan bilgiyi dijital alana taşırken, aktüatör dijital alanda üretilen kararı fiziksel dünyaya geri uygular. Bu ilişki, "algılama–karar verme–eyleme geçme" döngüsü üzerinden somutlaştırılabilir: sensör ortamdaki bir durumu algılar, mikrodenetleyici bu veriyi önceden tanımlanmış bir kontrol mantığı veya algoritma aracılığıyla değerlendirerek bir karar üretir, aktüatör ise bu kararı motor hareketi, valf açılması veya röle anahtarlaması gibi somut bir eyleme dönüştürür. Bu üç bileşenin senkronize çalışması, otomatik sulama sisteminden endüstriyel robot koluna kadar uzanan geniş bir uygulama yelpazesinin temelini oluşturur.

Aktüatör Türleri

DC motorlar, uygulanan doğru akım gerilimiyle orantılı bir hızda sürekli dönme hareketi üretir; basit sürme devresi ve düşük maliyetleri nedeniyle yaygın olarak tercih edilirler, ancak hassas konum kontrolü için ek geri besleme (örneğin enkoder) gerektirirler. **Servo motorlar**, bir DC veya fırçasız motoru, dişli kutusu ve konum geri beslemeli bir kontrol devresiyle birleştirerek belirli bir açısal konuma hassas biçimde yerleşebilme yeteneği kazandırır; robotik eklem kontrolü ve konum hassasiyeti gerektiren uygulamalarda tercih edilirler, sınırlılıkları arasında sınırlı dönüş açısı (çoğu standart modelde yaklaşık 180°) ve nispeten yüksek maliyet sayılabilir. **Step (adım) motorlar**, her bir elektriksel darbeye karşılık sabit bir açısal adım kadar dönerek açık çevrim konum kontrolüne imkân tanır; enkodere ihtiyaç duymadan tekrarlanabilir konumlandırma sağlamaları nedeniyle 3B yazıcılar ve CNC eksenlerinde yaygındır, ancak yüksek hızlarda adım kaybı riski taşırlar ve DC motorlara kıyasla daha fazla güç tüketirler.

Solenoidler, bir bobinden geçen akımın oluşturduğu manyetik alan aracılığıyla bir demir çekirdeği doğrusal olarak hareket ettiren basit elektromekanik aktüatörlerdir; valf açma/kapama ve kilit mekanizmaları gibi iki konumlu (açık/kapalı) uygulamalarda kullanılırlar. **Röleler**, düşük güçlü bir kontrol sinyaliyle yüksek akım/gerilim taşıyan bir devreyi elektromekanik olarak anahtarlayan aktüatörlerdir ve mikrodenetleyicinin doğrudan süremeyeceği yükleri (örneğin AC motorlar, ısıtıcılar) devreye alıp çıkarmak için kullanılır. Bu iki eleman birlikte, geniş anlamda **elektromekanik aktüatörler** kategorisini oluşturur ve ortak özellikleri,

elektrik enerjisini doğrudan mekanik harekete veya anahtarlamaya dönüştürmeleridir.

Pnömatik aktüatörler, basınçlı hava kullanarak piston veya diyafram hareketi üretir; hızlı tepki süreleri ve nispeten hafif yapılarıyla endüstriyel otomasyonda yaygındır, ancak sıkıştırılmış hava kaynağı gerektirmeleri ve pozisyon hassasiyetlerinin sınırlı olması dezavantaj oluşturur. **Hidrolik aktüatörler** ise sıkıştırılamayan bir sıvının basıncından yararlanarak çok daha yüksek kuvvet yoğunluğu sağlar; ağır iş makineleri ve büyük ölçekli endüstriyel sistemlerde tercih edilirler, fakat sızıntı riski, bakım maliyeti ve sistem karmaşıklığı açısından pnömatik ve elektromekanik alternatiflere kıyasla daha dezavantajlıdırlar. **Piezoelektrik aktüatörler**, belirli kristal veya seramik malzemelerin elektrik alanı altında boyut değiştirmesi (ters piezoelektrik etki) ilkesine dayanır; nanometre mertebesinde konumlandırma hassasiyeti sundukları için hassas optik sistemler, mikro-pompalar ve titreşim kontrolü gibi mikro ölçekli uygulamalarda kullanılırlar, ancak hareket menzilleri son derece sınırlıdır ve yüksek sürme gerilimlerine ihtiyaç duyarlar.

Aktüatör Sürme Yöntemleri

Aktüatörlerin mikrodenetleyici çıkışlarıyla doğrudan sürülmesi, çoğu zaman akım ve gerilim sınırlamaları nedeniyle mümkün olmadığından, aradaki gücü uygun biçimde ölçeklendiren ara devrelere ihtiyaç duyulur. **Darbe genişlik modülasyonu (PWM – pulse width modulation)**, sabit genlikli bir sinyalin açık kalma süresinin (duty cycle) değiştirilmesiyle ortalama gerilimin ya da gücün ayarlanmasını sağlayan bir yöntemdir; DC motor hız kontrolünde ve servo motorların konum bilgisinin iletilmesinde temel sürme mekanizmasını oluşturur. **Motor sürücü entegre devreleri**, mikrodenetleyicinin ürettiği düşük akımlı kontrol sinyalini, motorun ihtiyaç duyduğu yüksek akıma dönüştüren ara katman görevi görür ve genellikle aşırı akım koruması gibi ek güvenlik işlevleri de sunar. **H-köprü devreleri**, dört anahtarlama elemanının uygun sırayla tetiklenmesiyle bir DC motorun her iki yönde de döndürülebilmesini sağlayan temel bir motor sürme topolojisidir; PWM sinyaliyle birlikte kullanıldığında hem yön hem de hız kontrolüne imkân tanır. PWM sinyalinin çalışma çevrimi (duty cycle) D = t_on / T biçiminde tanımlanır (t_on: anahtarın iletimde kaldığı süre, T: anahtarlama periyodu) ve buna bağlı ortalama çıkış gerilimi yaklaşık olarak Vort = D x Vkaynak ilişkisiyle hesaplanabilir; arızaya karşı toleranslı H-köprüsü tasarımlarında bu ilişkinin doğru yönetilmesi, kalıcı mıknatıslı DC

motorların güvenilir hız kontrolü açısından önemlidir (Ahmed, Amin, Wajid ve Ahmad, 2020). **Röle modülleri**, mikrodenetleyicinin ürettiği düşük gerilimli sinyali bir röle bobinini süren ara transistör devresiyle birleştirerek, yüksek güçlü AC/DC yüklerin güvenli biçimde anahtarlanmasını sağlar. Bu sürme yöntemlerinin ortak amacı, sınırlı akım/gerilim kapasitesine sahip mikrodenetleyici çıkışları ile daha yüksek güç talep eden aktüatörler arasında güvenli ve kontrollü bir arabirim oluşturmaktır.

Tablo 2. Başlıca Aktüatör Türleri ve Kullanım Alanları

Aktüatör Türü	Çalışma Prensibi	Tipik Sürme Yöntemi	Tipik Kullanım Alanı
DC motor	Manyetik alan altında dönme hareketi	PWM + H-köprü / motor sürücü	Fan, tekerlek tahriki, pompa
Servo motor	Geri beslemeli açısal konumlandırma	PWM (konum kodlama)	Robotik eklem, yönlendirme mekanizması
Step motor	Darbe başına sabit açısal adım	Adım sürücü entegre devre	3B yazıcı, CNC eksen kontrolü
Solenoid	Elektromıknatıs ile doğrusal hareket	Transistör anahtarlama / röle	Valf kontrolü, kilit mekanizması
Röle	Elektromekanik anahtarlama	Transistör + röle sürücü	Yüksek güçlü yük anahtarlama
Pnömatik aktüatör	Basınçlı hava ile piston hareketi	Solenoid valf kontrolü	Endüstriyel otomasyon, tutucular
Hidrolik aktüatör	Basınçlı sıvı ile yüksek kuvvet üretimi	Solenoid valf / servo valf	Ağır iş makineleri
Piezoelektrik aktüatör	Ters piezoelektrik etki (elektrik → boyut değişimi)	Yüksek gerilim sürücü devre	Hassas konumlandırma, mikro-pompa

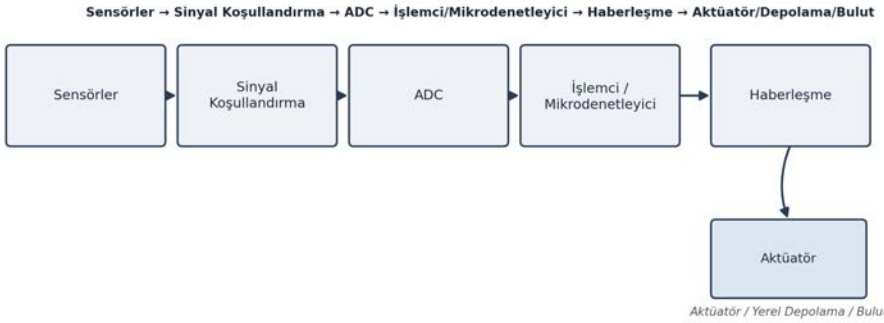
GÖMÜLÜ SİSTEMLER

Bir **gömülü sistem** (embedded system), belirli bir görevi yerine getirmek üzere tasarlanmış, donanım ve yazılımın sıkı biçimde bütünleştiği, çoğunlukla daha büyük bir mekanik ya da elektronik sistemin parçası olarak çalışan özel amaçlı bir hesaplama birimidir. Genel amaçlı bir bilgisayar, kullanıcının farklı zamanlarda farklı uygulamalar çalıştırabileceği esnek bir platform sunarken; gömülü bir sistem, tasarım aşamasında belirlenmiş sınırlı

bir işlevi -örneğin bir çamaşır makinesinin yıkama döngüsünü kontrol etmek ya da bir sensör düğümünden veri toplamak- yerine getirmek üzere optimize edilir. Bu temel fark, iki sistem sınıfı arasında bir dizi ikincil farkı da beraberinde getirir.

Gömülü sistemler, genellikle pil ile beslenen veya sürekli çalışması beklenen cihazlarda kullanıldıklarından **düşük güç tüketimi** ön planda tutularak tasarlanır; buna karşın genel amaçlı bilgisayarlar performans önceliğiyle tasarlanır ve güç tüketimi ikincil bir kısıttır. Gömülü sistemler tipik olarak **sınırlı işlem ve bellek kaynaklarına** sahiptir, çünkü hedeflenen görev genellikle sabit ve önceden bilinen bir karmaşıklığa sahiptir; bu sınırlama, maliyet ve güç tüketimi avantajı sağlar ancak yazılımın kaynak kullanımı açısından dikkatli tasarlanmasını gerektirir. Birçok gömülü sistem uygulaması, bir olayın belirli bir zaman penceresi içinde işlenmesini gerektiren **gerçek zamanlı çalışma** kısıtına tabidir; burada önemli olan yalnızca doğru sonucu üretmek değil, bu sonucu öngörülebilir bir süre içinde üretmektir. Gömülü sistemler çoğunlukla insan müdahalesinin sınırlı olduğu ortamlarda uzun süre kesintisiz çalışması beklenen cihazlarda yer aldığından **güvenilirlik** kritik bir tasarım hedefidir. **Maliyet** kısıtı, özellikle yüksek adetli tüketici ürünlerinde donanım seçimini doğrudan etkiler ve bu da genellikle daha düşük saat hızına ve daha az çevre birimine sahip bileşenlerin tercih edilmesine yol açar. Son olarak, gömülü sistemler sensörler ve aktüatörler aracılığıyla **fiziksel sistemlerle doğrudan etkileşim** kurar; bu, genel amaçlı bilgisayarların çoğunlukla klavye, fare ve ekran gibi insan arabirimleri üzerinden dolaylı etkileşim kurmasından temelde farklıdır.

Tipik bir gömülü sistem mimarisi, Şekil 2'de özetlendiği gibi, fiziksel dünyadan başlayıp yeniden fiziksel dünyaya veya bulut ortamına uzanan bir veri ve kontrol akışı izler: sensörlerden gelen ham sinyal önce sinyal koşullandırma devresinden geçirilir, ardından analog-dijital dönüştürücü (ADC) aracılığıyla sayısallaştırılır, işlemci veya mikrodenetleyici tarafından işlenir, gerekli durumlarda bir haberleşme arabirimi üzerinden iletilir ve son olarak bir aktüatörü tetikler, yerel belleğe kaydedilir ya da bulut sunucusuna gönderilir.



Şekil 2. Tipik Bir Gömülü Sistem Mimarisi

MİKROİŞLEMCİ VE MİKRODENETLEYİCİLER

Mikroişlemci ve Mikrodenetleyici Kavramları

Mikroişlemci (microprocessor), yalnızca merkezi işlem birimini (CPU) tek bir yonga üzerinde barındıran, bellek, giriş/çıkış birimleri ve diğer çevre birimlerinin ise ayrı yongalar olarak dışarıda bulunduğu bir hesaplama bileşenidir; bu yapı, genel amaçlı bilgisayarlarda olduğu gibi yüksek işlem gücü ve esnek bellek yapılandırması gerektiren uygulamalarda tercih edilir.

Mikrodenetleyici (microcontroller) ise CPU'nun yanı sıra RAM, flash bellek, giriş/çıkış pinleri ve çoğunlukla ADC, zamanlayıcı gibi çevre birimlerini tek bir yonga üzerinde bütünleştiren bir sistem-tek-yongadır (system-on-chip). Bu bütünleşik yapı, mikrodenetleyicilerin daha düşük maliyetli, daha az güç tüketen ve daha basit devre tasarımı gerektiren çözümler olmasını sağlar; buna karşılık işlem gücü ve bellek kapasitesi, ayrık bileşenlerle genişletilebilen mikroişlemci tabanlı sistemlere kıyasla daha sınırlıdır. Bu nedenle mikroişlemciler genellikle işletim sistemi çalıştıran, yüksek hesaplama gücü gerektiren sistemlerde (örneğin tek kart bilgisayarlar); mikrodenetleyiciler ise doğrudan sensör ve aktüatör kontrolü gerektiren, gerçek zamanlı ve düşük güçlü gömülü uygulamalarda tercih edilir.

Mikrodenetleyicilerin Temel Bileşenleri

Bir mikrodenetleyicinin işlevselliği, yonga üzerinde bütünleşik biçimde yer alan bir dizi bileşenin koordineli çalışmasına dayanır. **CPU**, komutları getirip yorumlayan ve yürüten merkezi hesaplama birimidir. **RAM (SRAM)**, program çalışırken kullanılan değişkenleri ve yığın verisini geçici olarak tutan uçucu bellektir. **Flash bellek**, programın kalıcı olarak saklandığı, güç kesildiğinde içeriğini kaybetmeyen yeniden yazılabilir bellek türüdür. **GPIO**

(genel amaçlı giriş/çıkış) pinleri, dijital sinyallerin okunmasını veya üretilmesini sağlayan yazılım tarafından yapılandırılabilir uçlardır. **ADC (analog-dijital dönüştürücü)**, analog sensör sinyallerini sayısal değerlere çevirirken; **DAC (dijital-analog dönüştürücü)** ters yönde bir dönüşüm yaparak sayısal bir değeri analog gerilime çevirir. **Timer/Counter (zamanlayıcı/sayıcı)** birimleri, hassas zaman aralıkları oluşturmak, olayları saymak veya PWM sinyali üretmek için kullanılır; **PWM** çıkışları da bu zamanlayıcı altyapısından yararlanarak motor hızı veya LED parlaklığı gibi analog benzeri etkiler üretir. **Kesme (interrupt)** mekanizması, işlemcinin ana program akışını, belirli bir olay (örneğin bir pinin durum değiştirmesi veya bir zamanlayıcının dolması) gerçekleştiğinde geçici olarak durdurup ilgili olay işleyicisine yönlendirmesini sağlar ve bu sayede sürekli yoklama (polling) yapılmasına gerek kalmadan zamana duyarlı olaylara hızlı tepki verilebilir. **UART, SPI ve I2C**, sensör ve diğer çevre birimleriyle seri haberleşme kurmak için kullanılan temel arabirimlerdir; **CAN (Controller Area Network)**, özellikle otomotiv ve endüstriyel ortamlarda gürültüye dayanıklı çoklu düğüm haberleşmesi için kullanılan bir protokoldür. **Watchdog timer (bekçi köpeği zamanlayıcısı)** ise yazılımın beklenmedik biçimde kilitlenmesi durumunda sistemin belirli bir süre sonunda otomatik olarak yeniden başlatılmasını sağlayarak güvenilirliği artıran bir güvenlik mekanizmasıdır.

Bu bileşenlerin tek bir yonga üzerinde bütünleşik olarak bulunması, mikrodenetleyicileri sensör ve IoT uygulamaları için özellikle uygun kılar: harici bileşen sayısının azalması devre kartı maliyetini ve karmaşıklığını düşürür, düşük güç modları pil ömrünü uzatır, entegre haberleşme arabirimleri sensörlerle doğrudan bağlantı kurulmasını sağlar ve gerçek zamanlı kesme/zamanlayıcı altyapısı, zamana duyarlı algılama ve kontrol görevlerinin öngörülebilir biçimde yerine getirilmesine imkân tanır.

Tablo 3. Mikroişlemci ve Mikrodenetleyici Karşılaştırması

Özellik	Mikroişlemci	Mikrodenetleyici
Bütünleşme düzeyi	Yalnızca CPU tek yongada; bellek ve çevre birimleri ayrık	CPU, RAM, flash, GPIO ve çevre birimleri tek yongada
İşlem gücü	Genellikle yüksek, çok çekirdekli olabilir	Genellikle sınırlı, çoğunlukla tek çekirdekli
Güç tüketimi	Görece yüksek	Düşük, çoğu zaman pille çalışacak biçimde optimize edilmiş
Tipik işletim biçimi	Çok görevli işletim sistemi (Linux vb.) üzerinde çalışır	Genellikle işletim sistemsiz veya RTOS ile çalışır
Maliyet	Sistem genelinde görece yüksek (ek yongalar dahil)	Düşük, tek yonga çözümü
Tipik kullanım alanı	Tek kart bilgisayarlar, sunucular, kişisel bilgisayarlar	Sensör/aktüatör kontrolü, gömülü IoT düğümleri

Gerçek Zamanlı İşletim Sistemleri

Gömülü sistemlerde, birden fazla görevin (task) belirli zaman kısıtları içinde öngörülebilir biçimde yürütülmesi gerektiğinde, ham döngü (bare-metal) yaklaşımı yetersiz kalabilir; bu noktada gerçek zamanlı işletim sistemleri (RTOS – real-time operating system) devreye girer. Bir RTOS, her biri bağımsız bir yürütme akışı olan görevleri, bir zamanlayıcı (scheduler) aracılığıyla önceliklerine göre işlemciye paylaştırır; öncelik tabanlı önleyici (preemptive) zamanlama, yüksek öncelikli bir görevin gerektiğinde düşük öncelikli bir görevin yürütülmesini keserek işlemciyi devralmasına imkân tanır ve böylece kritik görevlerin zaman kısıtları içinde tamamlanması güvence altına alınmaya çalışılır. FreeRTOS, kaynakları sınırlı mikrodenetleyiciler için yaygın olarak tercih edilen, açık kaynaklı ve küçük bellek ayak izine sahip bir RTOS örneğidir; görev oluşturma, sıralar (queue), semaforlar ve zamanlayıcılar gibi temel gerçek zamanlı işletim sistemi ilkellerini sunar. Farklı mikrodenetleyici platformlarında çalışan RTOS'lerin zamanlama gecikmesi ve bağlam değiştirme (context switch) süreleri gibi performans ölçütleri, kritik gerçek zamanlı uygulamalarda RTOS seçimini doğrudan etkileyen faktörler arasında yer alır (Ungurean, 2020).

ARDUINO PLATFORMU

Arduino, açık kaynaklı donanım ve yazılım felsefesiyle 2000'li yılların ortasında ortaya çıkmış ve gömülü sistemlerin eğitim ortamına ve hobi/prototipleme camiasına yaygın biçimde açılmasında belirleyici bir rol oynamıştır. Platformun önemi, karmaşık mikrodenetleyici programlama bilgisine sahip olmayan öğrencilerin ve mühendislerin dahi kısa sürede sensör ve aktüatör tabanlı sistemler geliştirebilmesini sağlayan basitleştirilmiş bir geliştirme ortamı ve standartlaştırılmış bir donanım arabirimi sunmasından kaynaklanır. Bu erişilebilirlik, Arduino'yu gömülü sistemler eğitiminde giriş seviyesi için sıklıkla tercih edilen bir platform hâline getirmiştir.

Arduino ailesinin en çok bilinen ve yaygın kullanılan üyesi olan **Arduino Uno**, ATmega328P mikrodenetleyicisi üzerine kuruludur ve 5V çalışma gerilimiyle, 16 MHz saat hızında çalışır (Arduino, t.y.). Kart üzerinde 14 dijital giriş/çıkış pini (bunlardan 6 tanesi PWM çıkışı olarak yapılandırılabilir) ve 6 analog giriş pini bulunur; bu analog girişler dahili 10-bit çözünürlüklü bir ADC üzerinden okunur. Bellek tarafında 32 KB flash bellek (0,5 KB'ı önyükleyici için ayrılmıştır), 2 KB SRAM ve 1 KB EEPROM yer alır. Haberleşme açısından kart, donanımsal UART desteğinin yanı sıra kütüphane düzeyinde I2C ve SPI iletişimini de destekler.

Arduino'nun asıl gücü, donanımından çok geliştirme ekosisteminden gelir. **Arduino IDE**, C/C++ tabanlı basitleştirilmiş bir programlama arabirimi sunarak "setup()" ve "loop()" fonksiyonları etrafında kurulan öngörülebilir bir program akışı sağlar. Geniş bir **kütüphane** ekosistemi, çok sayıda sensör ve aktüatör için hazır sürücü kodları sunarak donanım entegrasyonunu büyük ölçüde basitleştirir. Buna ek olarak **shield** adı verilen, Arduino kartının üzerine doğrudan takılabilen genişletme kartları (motor sürücü shield'leri, haberleşme shield'leri vb.), donanım tasarımı bilgisi gerektirmeden işlevsellik eklenmesine imkân tanır.

Arduino'nun akademik açıdan değerlendirilmesi gereken avantajları arasında düşük öğrenme eğrisi, geniş belgelendirme ve topluluk desteği, düşük maliyet ve kararlı/öngörülebilir gerçek zamanlı davranış sayılabilir; ATmega328P'nin işletim sistemi çalıştırmaması, kesme tabanlı işlemlerin gecikme süresinin görece deterministik olmasını sağlar. Sınırlılıkları ise sınırlı işlem gücü ve bellek kapasitesi, dahili kablosuz haberleşme desteğinin bulunmayışı (bu işlev için ek modüllere ihtiyaç duyulması) ve görüntü işleme ya da makine öğrenmesi gibi hesaplama yoğun görevler için yetersiz

kalmasıdır. Bu nedenle Arduino Uno, doğrudan sensör okuma ve aktüatör kontrolü gerektiren, ağ bağlantısının zorunlu olmadığı basit ve orta karmaşıklıktaki gömülü uygulamalar için uygun bir platform olarak konumlanır.

ESP32 PLATFORMU

ESP32 ailesi, Espressif Systems tarafından geliştirilen ve düşük maliyetli kablosuz bağlantılı gömülü sistemlerin yaygınlaşmasında önemli bir rol oynayan mikrodenetleyici serisidir. Ailenin önemi, klasik bir mikrodenetleyicinin sunduğu GPIO, ADC ve seri haberleşme yeteneklerini, Wi-Fi ve Bluetooth gibi kablosuz iletişim modülleriyle tek bir yonga üzerinde birleştirmesinden kaynaklanır; bu bütünleşik yapı, IoT düğümlerinin tasarımını önemli ölçüde basitleştirmiştir.

Orijinal ESP32 serisi, tek veya çift çekirdekli, 240 MHz'e kadar saat hızında çalışabilen 32-bit Xtensa LX6 mimarisine dayanır ve 520 KB SRAM barındırır (Espressif Systems, t.y.). Giriş/çıkış tarafında, yapılandırılabilen çok sayıda GPIO pini, 18 kanala kadar erişilebilen iki adet 12-bit SAR ADC ve iki adet 8-bit DAC bulunur. Haberleşme arabirimleri açısından üç adet UART, dört SPI denetleyicisi ve iki I2C arabirimi desteklenir; ayrıca ISO 11898-1 uyumlu bir CAN (TWAI) denetleyicisi de yer alır. Kablosuz iletişim tarafında 802.11 b/g/n Wi-Fi standardı ile Bluetooth v4.2 (klasik ve düşük enerjili BLE modları dahil) desteklenir. ESP32, aktif modun yanı sıra modem-uyku, hafif-uyku, derin-uyku ve hazırda bekletme (hibernation) gibi birden fazla düşük güç modu sunarak pil ile çalışan IoT uygulamalarında enerji tüketimini önemli ölçüde azaltabilir.

Bu özelliklerin bir arada bulunması, ESP32'yi doğrudan buluta veya yerel bir ağa bağlanması gereken sensör düğümleri için doğal bir seçim hâline getirir: bir sıcaklık/nem sensöründen okunan veri, ek bir kablosuz modül gerektirmeden doğrudan Wi-Fi üzerinden bir sunucuya veya MQTT aracısına iletilebilir. Bu noktada akademik açıdan önemli bir uyarı gereklidir: ESP32 ailesi tek bir ürün değil, farklı çekirdek mimarilerine (örneğin RISC-V tabanlı ESP32-C serisi) ve farklı çevre birimi konfigürasyonlarına sahip birden fazla model içeren geniş bir aile olduğundan, GPIO sayısı, ADC kanal sayısı veya belirli bir haberleşme arabiriminin varlığı gibi özellikler modelden modele değişebilir; bu nedenle burada sunulan değerler orijinal ESP32 serisi için geçerlidir ve tüm aile için genellenmemelidir.

Arduino Uno ile karşılaştırıldığında ESP32, çok daha yüksek işlem gücü, daha fazla bellek ve yerleşik kablosuz bağlantı sunar; buna karşılık daha yüksek güç tüketimi (özellikle aktif Wi-Fi kullanımı sırasında), 3,3V mantık seviyesi nedeniyle bazı 5V sensörlerle doğrudan uyumsuzluk ve göreli olarak daha karmaşık bir geliştirme öğrenme eğrisi taşır. ESP32'nin avantajları arasında entegre kablosuz bağlantı, çift çekirdekli işlem gücü sayesinde haberleşme ve sensör okuma görevlerinin paralel yürütülebilmesi ve düşük güç modlarının sunduğu esneklik sayılabilir; sınırlılıkları arasında ise Arduino Uno'ya kıyasla daha yüksek maliyet ve bazı düşük seviye çevre birimi yapılandırmalarının daha fazla teknik bilgi gerektirmesi yer alır.

RASPBERRY PI

Raspberry Pi, mikrodenetleyici kartlarından temel olarak mimari düzeyde ayrılır: bir mikrodenetleyici kartı genellikle işletim sistemi çalıştırmadan, doğrudan yazılan makine koduyla veya basit bir gerçek zamanlı çekirdekle çalışırken, Raspberry Pi bir **tek kart bilgisayar (Single Board Computer – SBC)** olarak tasarlanmıştır ve tam teşekküllü bir işletim sistemi (genellikle Linux tabanlı Raspberry Pi OS) üzerinde çalışır. Bu fark, Raspberry Pi'yi dosya sistemi yönetimi, çoklu görev (multitasking), ağ yığını ve yüksek seviyeli programlama dilleri gibi bir masaüstü bilgisayarda bulunan olanakları sunan, buna karşılık gerçek zamanlı garantiler açısından klasik mikrodenetleyicilerin gerisinde kalan bir platform hâline getirir.

Temsili bir örnek olarak Raspberry Pi 4 Model B ele alındığında, kart Broadcom BCM2711 sistem-üstü-yonga (SoC) üzerine kuruludur; bu yonga, standart sürümlerde 1,5 GHz saat hızında (bazı sonraki revizyonlarda ve Raspberry Pi 400 modelinde 1,8 GHz'e kadar) çalışan dört çekirdekli 64-bit ARM Cortex-A72 işlemci içerir (Raspberry Pi Ltd., t.y.). Bellek tarafında karta bağlı olarak 1 GB ile 8 GB arasında değişen LPDDR4 RAM seçenekleri bulunurken, kalıcı depolama işletim sistemi ve verinin yazıldığı bir microSD kart yuvası üzerinden sağlanır. Kart, önceki nesillerle uyumlu 40 pinli bir GPIO başlığı barındırır; bu başlık, dijital giriş/çıkış hatlarının yanı sıra donanımsal I2C, SPI ve UART hatlarını da içerir, fakat Arduino ve ESP32'den farklı olarak dahili bir analog-dijital dönüştürücü sunmaz ve analog sensör bağlantısı için harici bir ADC modülüne ihtiyaç duyulur. Bağlantı tarafında iki adet USB 3.0 ve iki adet USB 2.0 portu, Gigabit Ethernet, 802.11ac Wi-Fi (2,4 GHz ve 5 GHz) ile Bluetooth 5.0/BLE desteği; görüntü çıkışı için ise 4K çözünürlüğe kadar destekleyen iki adet mikro-HDMI portu bulunur. Raspberry Pi ailesinin farklı nesilleri arasında

işlem gücü bakımından belirgin bir ilerleme gözlenir: Raspberry Pi 5, önceki nesil Raspberry Pi 4'e kıyasla daha güçlü bir işlemci ve grafik birimine sahipken, Raspberry Pi 4 de kendisinden önceki Raspberry Pi 3'e göre gözle görülür bir performans artışı sunar; bu ilerleme, kartın daha talepkâr görüntü işleme ve kenar yapay zekâ (edge AI) iş yüklerini daha kısa sürede ve daha yüksek çözünürlük/kare hızıyla işleyebilmesine imkân tanır (Raspberry Pi Ltd., t.y.).

Bu donanımsal olanaklar, Raspberry Pi'yi klasik mikrodenetleyicilerin yetersiz kaldığı görevler için uygun kılar. **Edge computing** bağlamında, Raspberry Pi birden fazla sensör düğümünden gelen veriyi toplayan, ön işleyen ve gerektiğinde buluta ileten bir ağ geçidi (gateway) rolü üstlenebilir. **Görüntü işleme** ve **yapay zekâ** uygulamalarında, çok çekirdekli işlemcisi ve yeterli belleği sayesinde hafif derin öğrenme modellerinin doğrudan cihaz üzerinde (on-device) çalıştırılmasına imkân tanır; bu da onu bir **IoT ağ geçidi** olarak konumlandırır ve alt seviye mikrodenetleyici tabanlı sensör düğümlerinden toplanan verinin birleştirilmesi, ön işlenmesi ve ağ üzerinden iletilmesi görevlerini üstlenebilir. **Robotik** uygulamalarında yüksek seviyeli karar algoritmalarını (örneğin yol planlama) çalıştırırken, düşük seviye motor kontrolünü genellikle bir mikrodenetleyiciye devreder. **Veri toplama** ve **yerel veri analizi** görevlerinde ise, üzerinde çalışan tam bir veritabanı yönetim sistemi veya analiz kütüphaneleri sayesinde, sensör verisinin yerel olarak saklanması ve ön analizinin yapılması mümkündür. Belirtilmesi gereken önemli bir husus, farklı Raspberry Pi modellerinin (örneğin Raspberry Pi Zero, Raspberry Pi 5) işlemci mimarisi, bellek seçenekleri ve çevre birimi desteği açısından belirgin farklılıklar taşıdığı; bu nedenle burada aktarılan teknik özelliklerin Raspberry Pi 4 Model B'ye özgü olduğu ve resmi Raspberry Pi dokümantasyonundan doğrulanması gerektiğidir.

ARDUINO, ESP32 VE RASPBERRY PI KARŞILAŞTIRMASI

Önceki kısımlarda ayrı ayrı ele alınan üç platform, aşağıda Tablo 4'te temsili birer model üzerinden (Arduino Uno, orijinal ESP32 serisi ve Raspberry Pi 4 Model B) karşılaştırılmıştır. Bu karşılaştırmanın akademik açıdan doğru yorumlanabilmesi için, tablodaki bazı satırların (örneğin bellek kapasitesi, GPIO sayısı) belirli modele özgü değerleri, bazı satırların ise (örneğin sistem türü, gerçek zamanlı kontrol uygunluğu) platform ailesinin genel eğilimini yansıttığı ayrımının gözetilmesi gerekir.

Tablo 4. Arduino, ESP32 ve Raspberry Pi Karşılaştırması

Kriter	Arduino (Uno)	ESP32	Raspberry Pi (4 Model B)
Sistem türü	Mikrodenetleyici kartı	Mikrodenetleyici (kablosuz destekli SoC)	Tek kart bilgisayar (SBC)
İşlem gücü	8-bit, 16 MHz, tek çekirdek	32-bit, 240 MHz'e kadar, tek/çift çekirdek	64-bit, 1,5 GHz (bazı revizyonlarda/Pi 400'de 1,8 GHz), dört çekirdek
İşletim sistemi	İşletim sistemi yok (doğrudan makine kodu)	İşletim sistemi yok / hafif RTOS (FreeRTOS)	Linux tabanlı tam işletim sistemi
Gerçek zamanlı kontrol uygunluğu	Yüksek (deterministik kesme davranışı)	Orta-yüksek (RTOS ile yönetilebilir)	Düşük-orta (işletim sistemi zamanlama gecikmesi)
Bellek kapasitesi	32 KB flash / 2 KB SRAM	Tipik olarak birkaç MB flash / 520 KB SRAM	1-8 GB RAM + microSD depolama
Analog giriş	6 kanal, dahili 10-bit ADC	Dahili 12-bit SAR ADC (çoklu kanal)	Yok (harici ADC modülü gerekir)
GPIO	14 dijital G/Ç	34 programlanabilir GPIO	40 pinli GPIO başlığı
Wi-Fi	Yok (harici modül gerekir)	Dahili (802.11 b/g/n)	Dahili (802.11ac)
Bluetooth	Yok (harici modül gerekir)	Dahili (BR/EDR + BLE)	Dahili (Bluetooth 5.0/BLE)
Güç tüketimi	Düşük	Düşük-orta (Wi-Fi aktifken artar)	Orta-yüksek (tipik 5V/3A besleme)
Programlama kolaylığı	Çok yüksek (Arduino IDE, basit C/C++)	Yüksek (Arduino IDE veya ESP-IDF)	Yüksek (Python, C/C++, yüksek seviye diller)
Sensör uygulamalarına uygunluk	Çok uygun (doğrudan analog/dijital okuma)	Çok uygun (analog/dijital + kablosuz iletim)	Uygun (harici ADC ile birlikte)
IoT uygulamalarına uygunluk	Sınırlı (ek modül gerektirir)	Çok uygun (yerleşik kablosuz bağlantı)	Uygun (ağ geçidi/gateway rolü için ideal)
Görüntü işleme uygunluğu	Uygun değil	Sınırlı (düşük çözünürlüklü basit görevler)	Uygun (kamera arabirimi ve yeterli işlem gücü)

Kriter	Arduino (Uno)	ESP32	Raspberry Pi (4 Model B)
Yapay zekâ uygulamalarına uygunluk	Çok sınırlı (yalnızca TinyML düzeyinde)	Sınırlı-orta (TinyML çıkarım modelleri)	Orta-yüksek (hafif derin öğrenme modelleri)
Maliyet açısından genel değerlendirme	Düşük	Düşük-orta	Orta
Tipik kullanım alanı	Basit sensör/aktüatör kontrolü, eğitim	Kablosuz sensör düğümleri, IoT uçları	Edge/IoT ağ geçidi, görüntü işleme, veri analizi

Bu karşılaştırmadan çıkarılabilecek temel sonuç, üç platformun birbirinin yerini alacak biçimde değil, farklı problem sınıflarına yönelik tamamlayıcı çözümler olarak konumlandığıdır. Yalnızca doğrudan sensör okuma ve aktüatör tetikleme gerektiren, ağ bağlantısının zorunlu olmadığı basit otomasyon görevlerinde Arduino, düşük maliyeti ve deterministik gerçek zamanlı davranışı nedeniyle akademik açıdan makul bir tercihtir. Sensör verisinin kablosuz biçimde bir sunucuya veya IoT platformuna iletilmesi gereken, pille çalışan dağıtık düğümlerden oluşan uygulamalarda ESP32, entegre Wi-Fi/Bluetooth desteği ve düşük güç modları sayesinde öne çıkar. Buna karşılık, birden fazla sensör düğümünden gelen verinin toplanması, yerel olarak işlenmesi, görüntü tabanlı analiz yapılması veya bir IoT ağ geçidi işlevi görülmesi gereken senaryolarda Raspberry Pi, işletim sistemi desteği ve yüksek işlem gücü sayesinde daha uygun bir platform olarak öne çıkar. Uygulamada bu üç platform sıklıkla hiyerarşik bir mimaride birlikte kullanılır: Arduino veya ESP32 tabanlı uçtaki düğümler sensör verisini toplayıp ön işler, Raspberry Pi ise bu düğümlerden gelen veriyi birleştiren ve buluta ileten bir ara katman (edge gateway) olarak görev yapar.

SENSÖR VE MİKRODENETLEYİCİ HABERLEŞMESİ

Bir sensörün mikrodenetleyiciye bağlanma biçimi, sensörün çıktı türüne ve uygulamanın gereksinimlerine göre değişir. **Analog giriş**, sensörün ürettiği sürekli gerilim değerinin doğrudan ADC pinine bağlanmasıyla gerçekleştirilir; en basit bağlantı yöntemi olmakla birlikte tek bir sinyal hattı taşıdığından ek bir protokol karmaşıklığı gerektirmez, ancak uzun kablo mesafelerinde gürültüye duyarlıdır ve mikrodenetteki ADC kanal sayısıyla sınırlıdır. **Dijital GPIO** bağlantısı, sensörün yalnızca iki durumlu (yüksek/düşük) bir çıktı ürettiği durumlarda (örneğin basit bir hareket

sensörünün tetikleme çıkışı) kullanılır; düşük karmaşıklığına karşın yalnızca ikili bilgi taşıyabilmesi temel sınırlılığıdır.

UART (Universal Asynchronous Receiver-Transmitter), iki cihaz arasında saat sinyali paylaşılmadan, önceden anlaşılan bir baud hızında asenkron seri veri iletimi sağlayan bir protokoldür; verici ve alıcı hatları (TX/RX) üzerinden noktadan noktaya bağlantı kurar. Basitliği ve donanım gereksiniminin azlığı avantajıdır, ancak yerleşik bir adresleme mekanizması bulunmadığından ikiden fazla cihazın aynı hat üzerinde doğrudan haberleşmesi için ek düzenlemeler gerekir; veri hızı tipik olarak birkaç yüz kbps ile birkaç Mbps arasında değişir. **I2C (Inter-Integrated Circuit)**, iki hat üzerinden (SDA – veri, SCL – saat) çalışan, senkron ve adresli bir seri haberleşme protokolüdür; aynı hat üzerine onlarca cihazın bağlanabilmesine imkân tanıması ve pin sayısını minimumda tutması avantajıdır, ancak veri hızı SPI'ye kıyasla daha düşüktür ve hat uzunluğu arttıkça sinyal bütünlüğü sorunları ortaya çıkabilir (NXP Semiconductors, 2021). **SPI (Serial Peripheral Interface)**, ayrı veri gönderme (MOSI), veri alma (MISO), saat (SCK) ve cihaz seçme (CS/SS) hatları kullanan tam çift yönlü (full-duplex) bir senkron protokoldür; I2C'ye kıyasla çok daha yüksek veri hızına ulaşabilmesi temel avantajıdır, ancak her ek cihaz için ayrı bir cihaz seçme hattı gerektirmesi pin kullanımını artırır.

CAN (Controller Area Network), çift telli diferansiyel sinyal hattı üzerinden çalışan, yüksek elektromanyetik gürültü ortamlarında dahi güvenilir haberleşme sağlayan, çarpışma çözümlü çoklu erişim mekanizmasına sahip bir protokoldür; başlangıçta otomotiv uygulamaları için geliştirilmiş olup günümüzde endüstriyel otomasyon sistemlerinde de yaygın olarak kullanılır. CAN protokolünün UART, I2C ve SPI ile karşılaştırmalı özellikleri Tablo 5'te dördüncü bir sütun olarak sunulmuştur. **One-Wire**, tek bir veri hattı üzerinden (ortak toprak hattıyla birlikte) hem güç hem de veri iletimini mümkün kılan, düşük hızlı fakat pin kullanımı açısından son derece ekonomik bir protokoldür; her cihazın benzersiz bir seri numarası bulunması sayesinde aynı hat üzerinde birden fazla cihaz adreslenebilir, ancak veri hızı düşüktür ve zamanlama toleransları sıkıdır.

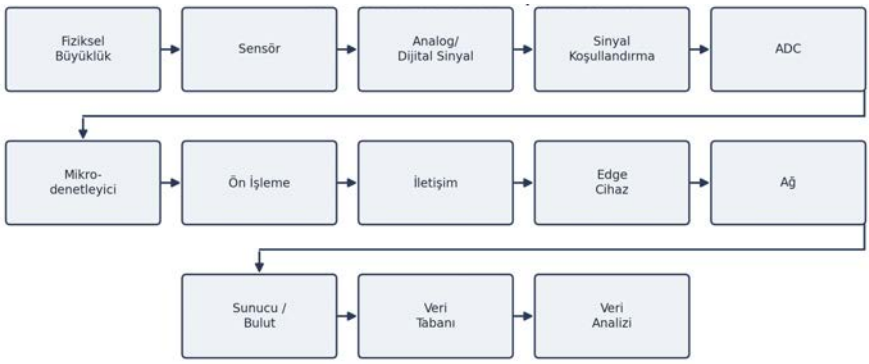
Tablo 5. UART, I2C, SPI ve CAN Karşılaştırması

Özellik	UART	I2C	SPI	CAN
Temel çalışma prensibi	Asenkron, saat sinyali paylaşılmaz	Senkron, adresli, iki hatlı	Senkron, tam çift yönlü, çok hatlı	Senkron, çift telli diferansiyel sinyal, çarpışma çözümlü çoklu erişim (CSMA/CR)
Avantaj	Basit, az donanım gerektirir	Az pin ile çok cihaz bağlanabilir	Yüksek veri hızı	Yüksek gürültü bağışıklığı, güvenilir çoklu düğüm haberleşmesi
Dezavantaj	Çoklu cihaz desteği sınırlı	Veri hızı görece düşük	Cihaz başına ayrı CS hattı gerekir	Daha karmaşık denetleyici donanımı gerektirir, veri çerçevesi UART/SPI'ye göre daha sınırlı (8 bayt klasik CAN)
Tipik veri hızı	~9,6 kbps - birkaç Mbps	100 kbps - 3,4 Mbps (moda göre)	Birkaç Mbps - >10 Mbps	Genellikle 125 kbps - 1 Mbps (klasik CAN)
Bağlanabilecek cihaz sayısı	Genellikle 2 (noktadan noktaya)	Onlarca (adres alanıyla sınırlı)	Cihaz seçme hattı sayısıyla sınırlı	Onlarca (bus üzerinde çoklu düğüm, arbitrasyon ile)
Hat yapısı	2 hat (TX, RX)	2 hat (SDA, SCL)	4 hat (MOSI, MISO, SCK, CS)	2 hat (CAN_H, CAN_L - diferansiyel çift)
Tipik kullanım alanı	GPS modülleri, bilgisayarla seri haberleşme	Sensör ağları, gerçek zaman saatleri, IMU	Ekranlar, SD kart, hızlı ADC/DAC	Otomotiv ağları, endüstriyel otomasyon

VERİ TOPLAMA SÜRECİ

Sensör verisinin fiziksel ortamdan alınıp anlamlı bir analiz çıktısına dönüşmesi, birbirine bağlı çok sayıda aşamadan oluşan bütüncül bir süreçtir. Şekil 3'te özetlenen bu akış, fiziksel büyüklüğün sensör tarafından

algılanmasıyla başlar, sinyal koşullandırma ve analog-dijital dönüşümle sayısallaşır, mikrodenetleyici düzeyinde ön işlemeden geçer, bir iletişim kanalı üzerinden edge cihaza, oradan ağ üzerinden sunucuya veya buluta taşınır ve son olarak bir veri tabanında depolanarak analiz edilir.



Şekil 3. Sensörden Buluta Veri Toplama Süreci

Örnekleme ve Analog-Dijital Dönüşüm

Sürekli zamanlı bir analog sinyalin sayısal ortamda işlenebilmesi için önce zaman ekseninde belirli aralıklarla **örneklenmesi** gerekir. Örnekleme sıklığını belirleyen **örnekleme frekansı** (sampling frequency), birim zamanda alınan örnek sayısını ifade eder ve sinyalin bilgi kaybına uğramadan yeniden oluşturulabilmesi için belirli bir alt sınırı aşması gerekir. Bu alt sınır, **Nyquist yaklaşımı** olarak bilinen ve Shannon'ın örnekleme teoremine dayanan aşağıdaki ilişkiyle ifade edilir (Shannon, 1949):

$$f_s \geq 2 \cdot f_max \quad (2)$$

Burada f_s örnekleme frekansını, f_max ise ölçülen sinyalde bulunan en yüksek frekans bileşenini ifade eder. Bu koşulun sağlanmaması durumunda **örtüşme (aliasing)** adı verilen bir hata ortaya çıkar; sinyalin yüksek frekanslı bileşenleri, örnekleme sonrası daha düşük bir frekansa sahipmiş gibi yanlış biçimde temsil edilir ve bu durum, yalnızca örnekleme frekansının artırılmasıyla değil, ADC öncesinde bir alçak geçiren filtre (anti-aliasing filter) kullanılmasıyla da önlenebilir.

Analog sinyalin sayısal değere dönüştürülmesi sürecinde, ADC'nin kullandığı bit sayısı (n), ayırt edebileceği en küçük gerilim adımını, yani **ADC çözünürlüğünü** belirler:

$$Q = V_FSR / 2^n \quad (3)$$

Burada Q bir bitlik en küçük anlamlı adımı (1 LSB), V_FSR ADC'nin tam skala referans gerilim aralığını, n ise ADC'nin bit çözünürlüğünü temsil eder. Örneğin 10-bit çözünürlüğe sahip ve 5V referans gerilimiyle çalışan bir ADC, yaklaşık 4,9 mV'luk bir adım genişliğine sahiptir; bu değerin altındaki gerilim farkları ADC tarafından ayırt edilemez. Bu ayrıklaştırma sürecinde kaçınılmaz olarak ortaya çıkan **kuantizasyon** hatası, gerçek analog değer ile en yakın sayısal seviyeye yuvarlanan değer arasındaki farktır ve en kötü durumda adım genişliğinin yarısı kadar olabilir:

$$e_q(maks) = \pm Q / 2 \quad (4)$$

Bu hata, temel olarak ADC'nin sonlu çözünürlüğünden kaynaklanan ve tamamen ortadan kaldırılamayan, yalnızca daha yüksek bit çözünürlüğüyle azaltılabilen sistematik bir belirsizlik kaynağıdır.

Gürültü, Filtreleme ve Kalibrasyon

Sensör ölçümlerine, ölçülen fiziksel büyüklükle ilişkili olmayan rastgele veya sistematik **gürültü** bileşenleri karışabilir; bu gürültünün kaynağı elektromanyetik girişim, güç kaynağı dalgalanmaları veya sensörün kendi iç devresindeki termal gürültü olabilir. Gürültünün etkisini azaltmak için sıklıkla dijital **filtreleme** teknikleri kullanılır. Bunlardan en basit ve hesaplama maliyeti en düşük olanı **hareketli ortalama filtresi**dir; bu filtre, son N örneğin aritmetik ortalamasını alarak yüksek frekanslı rastgele dalgalanmaları yumuşatır:

$$y[n] = (1/N) \cdot \Sigma\, x[n-k],\ k = 0..N-1 \quad (5)$$

Burada $y[n]$ filtrelenmiş çıktı örneğini, $x[n-k]$ ham örnekleri ve N pencere büyüklüğünü ifade eder. Pencere büyüklüğü arttıkça gürültü bastırma etkisi güçlenir, fakat sinyaldeki gerçek ve hızlı değişimlere tepki verme yeteneği (tepki süresi) de o ölçüde yavaşlar; bu nedenle N değerinin seçimi, uygulamanın gürültü toleransı ile tepki hızı gereksinimi arasında bir denge gözetilerek yapılmalıdır.

Filtrelemeden bağımsız olarak, sensör çıktısının gerçek fiziksel değerle örtüşmesini sağlamak amacıyla **kalibrasyon** işlemi uygulanır; bu süreçte sensör, bilinen referans değerlere sahip standart koşullar altında ölçüm yapar ve ölçülen değer ile referans değer arasındaki fark kullanılarak bir düzeltme fonksiyonu (genellikle doğrusal veya çok terimli bir denklem) belirlenir. **Sensör doğrulama** (validation), kalibrasyonu tamamlanmış bir sensörün bağımsız bir referans ölçüm sistemiyle karşılaştırılarak beklenen doğruluk

aralığında çalıştığının teyit edilmesi sürecidir ve özellikle güvenlik açısından kritik uygulamalarda düzenli aralıklarla tekrarlanması gerekir.

Zamanlama, Senkronizasyon ve Veri Kalitesi

Birden fazla sensörden veya birden fazla düğümden gelen verinin anlamlı biçimde bir araya getirilebilmesi için her ölçüme bir **zaman damgası** (timestamp) eklenmesi gerekir; bu, verinin ne zaman üretildiğinin kayıt altına alınmasını ve sonraki analizlerde zamana dayalı ilişkilerin doğru kurulabilmesini sağlar. Farklı sensör düğümlerinin saatlerinin birbirinden az da olsa sapması, **veri senkronizasyon** sorunlarına yol açabilir; bu nedenle dağıtık sensör ağlarında düğümler arasında bir zaman senkronizasyon protokolü (örneğin ağ zaman protokolü) kullanılması yaygın bir uygulamadır.

Gerçek dünya veri toplama sistemlerinde **eksik veri**, ağ kesintisi, sensör arızası veya güç kaybı gibi nedenlerle kaçınılmaz olarak ortaya çıkabilir ve analiz aşamasında ya uygun bir enterpolasyon yöntemiyle doldurulmalı ya da eksikliğin kendisi ayrı bir bilgi olarak işaretlenmelidir. **Aykırı değerler** (outliers), sensör arızası, geçici bir çevresel etkileşim veya iletim hatası sonucunda ortaya çıkan, genel eğilimden anlamlı biçimde sapan ölçümlerdir; bu değerlerin istatistiksel yöntemlerle (örneğin standart sapma tabanlı eşikleme) tespit edilip ayıklanması veya işaretlenmesi, sonraki analizlerin güvenilirliği açısından önemlidir.

Sensörden üretilen veri, ağ bağlantısının her zaman kesintisiz olmayabileceği gerçeği göz önünde bulundurularak genellikle edge cihaz üzerinde geçici olarak **veri tamponlanır (buffering)**; bu, ağ kesintisi sırasında verinin kaybolmasını önler ve bağlantı yeniden kurulduğunda birikmiş verinin toplu biçimde iletilmesine imkân tanır. **Veri iletimi**, tamponlanmış verinin bir haberleşme protokolü üzerinden (bir sonraki bölümde ayrıntılı ele alınacak MQTT, HTTP/HTTPS gibi protokoller aracılığıyla) sunucuya veya buluta taşınmasıdır. Son olarak **veri depolama**, iletilen verinin bir veri tabanında -yapılandırılmış ilişkisel veri tabanları ya da zaman serisi verisi için özelleşmiş veri tabanları biçiminde- kalıcı olarak saklanmasını kapsar ve bu depolanmış veri, sonraki veri analizi aşamasının girdisini oluşturur.

Tablo 6. Sensör Veri Toplama Sürecinde Karşılaşılabilecek Hata Kaynakları ve Çözüm Yöntemleri

Hata Kaynağı	Açıklama	Tipik Çözüm Yöntemi
Gürültü	Elektriksel veya çevresel kaynaklı rastgele dalgalanmalar	Hareketli ortalama / alçak geçiren filtre, ekranlama
Kalibrasyon hatası / drift	Sensör çıktısının referans değerden zamanla sapması	Periyodik yeniden kalibrasyon
Örtüşme (aliasing)	Yetersiz örnekleme frekansı nedeniyle yanlış frekans temsili	Nyquist kriterine uygun örnekleme + anti-aliasing filtre
Kuantizasyon hatası	ADC'nin sonlu çözünürlüğünden kaynaklanan yuvarlama farkı	Daha yüksek bit çözünürlüklü ADC kullanımı
Zaman senkronizasyon hatası	Düğümler arası saat sapması	Ağ zaman senkronizasyon protokolü, ortak zaman damgası
Eksik veri	Ağ kesintisi, güç kaybı veya sensör arızası	Yerel tamponlama, enterpolasyon, eksik veri işaretleme
Aykırı değerler (outlier)	Genel eğilimden anlamlı sapan ölçümler	İstatistiksel eşikleme, aykırı değer tespiti algoritmaları
Elektromanyetik girişim	Yakın çevredeki elektriksel kaynaklardan kaynaklanan bozulma	Ekranlı kablolama, diferansiyel sinyal hatları, topraklama

IoT TABANLI VERİ TOPLAMA

Dağıtık sensör ağlarında veri toplama mimarisi, sensör düğümünün topladığı veriyi bir kablosuz haberleşme teknolojisi üzerinden bir ağ geçidine veya doğrudan bir IoT platformuna taşımasını gerektirir. Bu noktada kullanılan kablosuz teknolojiler, kapsama alanı, veri hızı ve güç tüketimi açısından birbirinden belirgin biçimde ayrışır ve seçim, uygulamanın gereksinimlerine göre yapılır. **Wi-Fi**, görece yüksek veri hızı ve doğrudan internet erişimi sunar, fakat güç tüketimi diğer kısa menzilli teknolojilere kıyasla daha yüksektir ve bu nedenle sürekli güç kaynağına sahip veya sık şarj edilebilen cihazlarda tercih edilir. **Bluetooth Low Energy (BLE)**, kısa menzilli ve düşük güç tüketimli bir iletişim sağlar; giyilebilir cihazlar ve telefon ile eşleştirilen sensörler için uygundur, fakat kapsama alanı sınırlıdır. **Zigbee**, düşük güç tüketimi ile birlikte iyi bir çok

atlamalı (mesh) ağ desteği sunarak akıllı ev ve bina otomasyonu gibi çok sayıda düğümün bir arada çalıştığı senaryolarda tercih edilir. **LoRa/LoRaWAN**, çok düşük veri hızı pahasına kilometrelerce menzile ulaşabilen, düşük güçlü geniş alan ağı (LPWAN) sınıfında bir teknolojidir ve akıllı tarım veya şehir çapında dağıtılmış çevresel izleme gibi geniş coğrafi alan gerektiren, düşük veri hacimli uygulamalar için uygundur.

Fiziksel katmandaki bu kablosuz teknolojilerin üzerinde, verinin uygulama düzeyinde nasıl paketlenip iletileceğini belirleyen protokoller yer alır. **MQTT (Message Queuing Telemetry Transport)**, yayınla-abone ol (publish-subscribe) mimarisine dayanan, hafif ve düşük bant genişliği gerektiren bir mesajlaşma protokolüdür; sensör düğümleri ölçüm verisini bir konuya (topic) yayınlar, ilgilenen uygulamalar ise bu konuya abone olarak veriyi alır. Bu mimari, sensör düğümü ile veri tüketen uygulama arasında doğrudan bir bağlantı gerektirmediğinden, kaynak kısıtlı gömülü cihazlar için özellikle uygundur. MQTT protokolü, mesaj teslim güvencesini uygulamanın ihtiyacına göre ayarlayabilmek için üç farklı hizmet kalitesi (QoS) düzeyi tanımlar: QoS 0 (en fazla bir kez iletim, teslim garantisi verilmez), QoS 1 (en az bir kez iletim, mesaj yinelenebilir) ve QoS 2 (tam olarak bir kez iletim, en yüksek güvence fakat en yüksek protokol yükü); sensör ölçüm akışları gibi kayıpla başa çıkabilen veriler için genellikle QoS 0 veya QoS 1 tercih edilirken, kritik komut mesajlarında QoS 2 kullanılabilir (OASIS, 2019). **HTTP/HTTPS**, web tabanlı sistemlerde yaygın olarak kullanılan istek-yanıt tabanlı bir protokoldür; MQTT'ye kıyasla daha fazla protokol yükü taşımasına karşın, mevcut web altyapısıyla doğrudan uyumlu olması ve daha az sıklıkla veri gönderen uygulamalarda pratik olması nedeniyle tercih edilebilir.

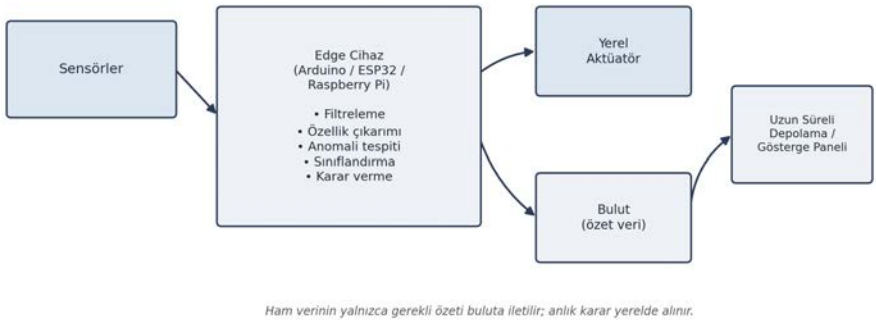
Bu bileşenlerin bir araya gelmesiyle oluşan örnek bir IoT tabanlı veri toplama mimarisi Şekil 4'te gösterilmiştir: bir sensörden okunan veri, ESP32 tarafından okunup ön işlenir, Wi-Fi üzerinden MQTT protokolüyle bir IoT platformuna veya mesaj aracısına (broker) yayınlanır, buradan bir veri tabanına kaydedilir ve son olarak bir gösterge paneli (dashboard) üzerinden görselleştirilir veya ileri analiz süreçlerine girdi olarak kullanılır. Bu akış, günümüzde uzaktan çevresel izleme, akıllı tarım ve endüstriyel durum izleme gibi çok sayıda uygulamada temel referans mimari olarak kullanılmaktadır (Sinha & Dhanalakshmi, 2021).



Şekil 4. IoT Tabanlı Sensör Veri Toplama Mimarisi

EDGE COMPUTING VE AKILLI SENSÖR SİSTEMLERİ

Klasik IoT mimarilerinde, sensörlerden toplanan ham verinin tamamı buluta gönderilir ve işleme/karar verme adımları merkezi sunucularda gerçekleştirilir. Sensör sayısının ve örnekleme sıklığının arttığı ölçekte bu yaklaşım, artan ağ bant genişliği maliyeti, yüksek gecikme (latency) ve sürekli internet bağlantısına bağımlılık gibi sınırlılıklar doğurur. **Edge computing**, bu sınırlılıkları azaltmak amacıyla filtreleme, özellik çıkarımı, anomali tespiti, sınıflandırma ve hatta karar verme gibi işlemlerin bir kısmının, veri kaynağına yakın konumdaki gömülü cihaz üzerinde gerçekleştirilmesini öngören bir yaklaşımdır. Bu sayede yalnızca anlamlı, özetlenmiş veya eşik değeri aşan veri buluta iletilir; bu durum ağ trafiğini azaltır, tepki süresini kısaltır, sürekli bağlantı gerekliliğini gevşetir ve ham verinin cihaz dışına çıkmaması sayesinde mahremiyet açısından da avantaj sağlar. Şekil 5, bu yaklaşımı somutlaştıran kavramsal bir mimariyi göstermektedir; burada sensörlerden toplanan veri doğrudan edge cihaz üzerinde işlenir, yerel bir aktüatör tetiklenebilir ve yalnızca özet veya anomali bilgisi buluta iletilir.



Şekil 5. Edge Computing Tabanlı Akıllı Sensör Sistemi

Bu yaklaşımın son yıllarda hızla gelişen bir alt dalı olan **TinyML**, makine öğrenmesi modellerinin, birkaç yüz kilobayt ile birkaç megabayt mertebesindeki bellek kısıtına sahip mikrodenetleyiciler üzerinde çalıştırılmasını konu alır (Ray, 2021). Bu yaklaşım, modelin eğitim aşamasının genellikle daha güçlü donanımlarda gerçekleştirilmesi, ardından niceleme (quantization) ve budama (pruning) gibi tekniklerle modelin boyutunun küçültülerek gömülü cihaza aktarılması ilkesine dayanır; bu

sayede basit görüntü sınıflandırma, anahtar kelime tespiti veya anomali tespiti gibi çıkarım görevleri, buluta hiç veri göndermeden doğrudan sensöre bitişik donanım üzerinde gerçekleştirilebilir. Niceleme ve budama gibi model küçültme teknikleri uygulandığında model boyutu genellikle büyük ölçüde azalır ve çıkarım hızı artar; ancak bu kazanım çoğunlukla model doğruluğunda sınırlı bir miktar kayıpla birlikte gelir, bu nedenle niceleme sonrası doğrulama adımı TinyML geliştirme sürecinin kritik bir parçası olarak değerlendirilir (Ray, 2021).

Arduino, ESP32 ve Raspberry Pi'nin edge intelligence açısından üstlendiği roller, işlem gücü ve bellek kapasiteleriyle doğru orantılı biçimde farklılaşır. Arduino Uno gibi 8-bit mikrodenetleyiciler, son derece sınırlı bellek kapasitesi nedeniyle karmaşık modellerden ziyade basit eşikleme ve kural tabanlı karar mantığıyla sınırlı kalır. ESP32, daha yüksek işlem gücü, daha fazla belleği ve entegre kablosuz bağlantısı sayesinde hafif TinyML modellerinin (örneğin basit ses/anahtar kelime tanıma) doğrudan cihaz üzerinde çalıştırılmasına imkân tanırken, aynı zamanda gerektiğinde sonucu buluta iletebilir. Raspberry Pi ise, tam bir işletim sistemi ve görece yüksek işlem gücü sayesinde daha karmaşık derin öğrenme tabanlı çıkarım modellerini çalıştırabilir ve genellikle bir edge ağ geçidi olarak, kendisine bağlı birden fazla mikrodenetleyici tabanlı sensör düğümünden gelen veriyi birleştirip analiz eden bir konumda yer alır.

VERİ GÜVENLİĞİ VE GİZLİLİĞİ

Sensör ve IoT tabanlı veri toplama sistemleri, çoğunlukla kişisel, endüstriyel veya çevresel açıdan hassas veriler taşıdığından güvenlik ve gizlilik konuları tasarımın ayrılmaz bir parçası olarak ele alınmalıdır. **Kimlik doğrulama** (authentication), yalnızca yetkili cihazların ağa katılabilmesini sağlarken, **yetkilendirme** (authorization) bu cihazların hangi kaynaklara erişebileceğini ve hangi işlemleri gerçekleştirebileceğini sınırlar. **Şifreleme**, hem depolanan verinin (data at rest) hem de iletim sırasındaki verinin (data in transit) yetkisiz taraflarca okunmasını engeller; bu bağlamda **güvenli haberleşme** protokolleri (örneğin TLS ile korunan MQTT veya HTTPS bağlantıları), sensör verisinin ağ üzerinde açık metin olarak taşınmasını önler. **Firmware güvenliği**, cihaz üzerinde çalışan yazılımın kötü amaçlı biçimde değiştirilmesini önlemeye yönelik imzalı güncelleme mekanizmalarını kapsarken, **fiziksel güvenlik**, cihaza doğrudan erişimi olan kişilerin donanımı kurcalayarak veri veya kimlik bilgilerini ele geçirmesini engellemeyi hedefler. Kişisel verilerin işlendiği sistemlerde **kişisel verilerin**

korunması ilkesi çerçevesinde, yalnızca amaç için gerekli olan verinin toplanması anlamına gelen **veri minimizasyonu** yaklaşımının benimsenmesi, hem mahremiyet risklerini azaltır hem de depolama ve iletim maliyetlerini düşürür. Bu unsurların tasarım aşamasından itibaren (privacy/security by design) sisteme dahil edilmesi, sonradan eklenen güvenlik önlemlerine kıyasla çok daha etkili bir koruma sağlar.

Şifreleme Yaklaşımları

Gömülü sistemlerde şifreleme işlevi, donanımsal veya yazılımsal biçimde gerçekleştirilebilir ve bu iki yaklaşım arasındaki seçim, kaynak kısıtları ile güvenlik gereksinimleri arasında bir ödünleşim (trade-off) oluşturur. Yazılımsal şifreleme, mikrodenetleyicinin genel amaçlı işlemcisi üzerinde çalışan kriptografik kütüphaneler aracılığıyla gerçekleştirilir; ek donanım gerektirmediğinden esnek ve düşük maliyetlidir, ancak işlemci döngülerini tüketir ve özellikle kaynakları sınırlı 8/16-bit mikrodenetleyicilerde gecikmeye ve artan enerji tüketimine yol açabilir. Donanımsal şifreleme ise, kriptografik işlemleri gerçekleştirmek üzere özelleştirilmiş bir yardımcı işlemci veya güvenli element (secure element) kullanır; bu sayede işlem hızı artar, ana işlemci üzerindeki yük azalır ve anahtarların güvenli biçimde saklanması sağlanır, ancak ek donanım maliyeti ve tasarım karmaşıklığı getirir. Kaynakları son derece kısıtlı IoT uç düğümlerinde, hafif kriptografi (lightweight cryptography) algoritmalarının kullanılması, güvenlik düzeyini büyük ölçüde düşürmeden hesaplama ve bellek yükünü azaltmayı hedefleyen bir ara çözüm olarak öne çıkar (Rana, Mamun ve Islam, 2021).

ÖRNEK UYGULAMA SENARYOLARI

Bu kısımda, önceki bölümlerde ele alınan kavramların somutlaştırılması amacıyla üç temsili uygulama senaryosu, sistem amacı, kullanılan bileşenler, veri akışı, kontrol süreci, avantajları ve sınırlılıkları bakımından incelenmiştir.

Uygulama 1: ESP32 Tabanlı Sıcaklık ve Nem İzleme Sistemi

Bu sistemin amacı, bir ortamın (örneğin bir sera, depo veya sunucu odası) sıcaklık ve nem koşullarının sürekli olarak uzaktan izlenmesidir. Sistem, dijital bir sıcaklık-nem sensörü, bir ESP32 geliştirme kartı, bir Wi-Fi erişim noktası, bir MQTT aracısı, bir sunucu/veri tabanı ve bir gösterge paneli bileşenlerinden oluşur; veri akışı Sensör → ESP32 → Wi-Fi → MQTT → Sunucu → Veri Tabanı → Dashboard biçiminde ilerler. Kontrol süreci basittir: ESP32 belirli aralıklarla sensörü okur, ölçüm sonucunu bir MQTT

konusuna yayınlar, sunucu bu veriyi alarak veri tabanına kaydeder ve gösterge paneli bu kayıtları görselleştirir; sistem isteğe bağlı olarak belirli eşik değerleri aşıldığında uyarı da üretebilir. Bu yaklaşımın avantajları arasında düşük donanım maliyeti, kablosuz kurulum kolaylığı ve düşük güç modları sayesinde uzun süreli pil ömrü sayılabilir. Sınırlılıkları ise Wi-Fi kapsama alanının sınırlı olması, ağ kesintisi durumunda veri kaybı riski (bu risk yerel tamponlama ile azaltılabilir) ve tek bir sensör düğümünün kapsadığı alanın sınırlı kalmasıdır.

Uygulama 2: Arduino Tabanlı Otomatik Sulama Sistemi

Bu sistemin amacı, toprak nem seviyesine bağlı olarak bir sulama pompasının otomatik biçimde devreye alınmasıdır. Kullanılan bileşenler arasında bir toprak nem sensörü, bir Arduino kartı, bir röle modülü ve bir su pompası yer alır; veri akışı Toprak Nem Sensörü → Arduino → Karar Algoritması → Röle → Su Pompası biçimindedir. Kontrol süreci, Arduino'nun toprak nem sensöründen okuduğu analog değeri önceden belirlenmiş bir eşik değeriyle karşılaştırmasına dayanır: ölçülen nem seviyesi eşik değerin altına düştüğünde Arduino, röle modülünü tetikleyerek pompayı belirli bir süre çalıştırır ve nem seviyesi yeterli düzeye ulaştığında pompayı durdurur. Bu sistemin avantajları arasında düşük maliyet, basit ve deterministik kontrol mantığı ile su tüketiminin optimize edilerek tasarruf sağlanması sayılabilir. Sınırlılıkları ise sistemin tek bir nokta ölçümüne dayanması nedeniyle toprak neminin alansal değişkenliğini yansıtamaması, kablosuz bağlantı bulunmadığından uzaktan izlemenin doğrudan mümkün olmaması (bu, isteğe bağlı bir ESP32 modülü eklenerek giderilebilir) ve sensör kalibrasyonunun toprak tipine göre ayarlanması gerekliliğidir.

Uygulama 3: Raspberry Pi Tabanlı Akıllı Görüntü/Sensör Sistemi

Bu sistemin amacı, bir kamera ve tamamlayıcı sensörlerden (örneğin hareket veya mesafe sensörü) gelen verinin birlikte değerlendirilerek yerel bir karar üretilmesi ve gerektiğinde bu kararın buluta iletilmesidir; tipik bir örnek, bir güvenlik kamerasının yalnızca insan varlığı tespit ettiğinde kayıt başlatması ve ilgili görüntüyü buluta göndermesidir. Kullanılan bileşenler bir kamera modülü, tamamlayıcı sensörler, bir Raspberry Pi kartı ve isteğe bağlı bir bulut bağlantısıdır; veri akışı Kamera + Sensörler → Raspberry Pi → Edge AI Modeli → Yerel Karar → Bulut biçimindedir. Kontrol süreci, Raspberry Pi üzerinde çalışan bir görüntü işleme veya hafif derin öğrenme modelinin, kamera görüntüsünü ve tamamlayıcı sensör verisini birlikte değerlendirerek bir sınıflandırma veya tespit kararı üretmesine ve bu karara

bağlı olarak yerel bir eylem tetiklemesine (örneğin kayıt başlatma) ya da yalnızca ilgili olayı buluta bildirmesine dayanır. Bu yaklaşımın avantajları arasında yüksek işlem gücü sayesinde karmaşık analiz görevlerinin yerelde gerçekleştirilebilmesi, ham görüntünün sürekli buluta gönderilmemesi sayesinde bant genişliği tasarrufu ve mahremiyetin görece daha iyi korunması sayılabilir. Sınırlılıkları ise klasik mikrodenetleyicilere kıyasla daha yüksek maliyet ve güç tüketimi ile görüntü işleme modellerinin gerektirdiği hesaplama yükünün, düşük maliyetli Raspberry Pi modellerinde performans sınırlamalarına yol açabilmesidir.

GÜNCEL GELİŞMELER VE GELECEK YÖNELİMLER

Sensör, aktüatör ve gömülü sistemler alanındaki güncel literatür, birkaç birbiriyle ilişkili eğilim etrafında şekillenmektedir. **Akıllı sensörler**, geleneksel algılama işlevine ek olarak yerel sinyal işleme ve kendi kendini teşhis edebilme (self-diagnostic) yeteneği barındıran, böylece ham veri yerine doğrudan anlamlı bilgi üreten bileşenler olarak tanımlanabilir; bu eğilim, sensör tasarımının giderek işlemci tabanlı zekâ ile bütünleşmesi anlamına gelir. Bu bütünleşmenin işlem tarafındaki karşılığı **edge AI** ve onun kaynak kısıtlı donanıma özelleşmiş biçimi olan **TinyML**'dir; bu yaklaşımlar, çıkarım işleminin veri kaynağına yakın konumda gerçekleştirilmesini sağlayarak gecikmeyi azaltmakta ve buluta bağımlılığı sınırlamaktadır (Ray, 2021).

Düşük güç tüketimli IoT yaklaşımları, hem donanım düzeyinde (düşük güç modları, verimli radyo protokolleri) hem de yazılım düzeyinde (uyku döngüleri, olay tabanlı örnekleme) enerji tüketimini azaltmayı hedefler; bu çabaların önemli bir tamamlayıcısı, sensör düğümlerinin çevresel enerjiden (güneş, titreşim, termal gradyan) yararlanarak kendi enerjisini üretmesini sağlayan **enerji hasadı (energy harvesting)** teknikleridir. Bu yaklaşım, özellikle erişimi güç veya pil değişimi açısından zor olan sensör düğümlerinde bakım gereksinimini azaltarak sürdürülebilir sensör ağlarının kurulmasına katkı sağlar (Adu-Manu vd., 2018). Enerji hasadı yöntemleri, kullanılan fiziksel ilkeye göre çeşitlenir: piezoelektrik hasat, mekanik titreşim veya deformasyonu elektrik enerjisine dönüştürürken; termoelektrik hasat, bir sıcaklık gradyanından (örneğin bir motor gövdesi ile ortam arasındaki farktan) yararlanır; güneş enerjisi hasadı ise fotovoltaik hücreler aracılığıyla ışık enerjisini elektriğe çevirir. Bu yöntemlerin sağladığı değişken ve genellikle düşük güç, mikrodenetleyicinin uyku (sleep) ve derin uyku (deep-sleep) gibi düşük güç modlarıyla birlikte kullanılmasını

gerektirir; böylece düğüm, çoğu zaman düşük güç modunda beklerken yalnızca kısa aralıklarla uyanıp ölçüm alarak enerji bütçesini dengeleyebilir (Naifar, Kanoun ve Trigona, 2024).

Sensör füzyonu, birden fazla sensörden gelen verinin, tek bir sensörün sunabileceğinden daha güvenilir ve kapsamlı bir tahmin üretmek amacıyla birleştirilmesi sürecidir; ivmeölçer, jiroskop ve manyetometre verilerinin birleştirilerek daha kararlı bir yönelim tahmini elde edilmesi bu yaklaşımın tipik bir örneğidir (Nazarahari & Rouhani, 2021). Sensör füzyonu ve edge AI'nin birlikte kullanımı, **otonom sistemlerin** ve **yapay zekâ destekli sensör sistemlerinin** gelişiminde belirleyici bir rol oynamaktadır; bu sistemlerde algılama, karar verme ve eyleme geçme döngüsü, insan müdahalesi olmadan veya sınırlı müdahaleyle sürdürülebilir hâle gelmektedir.

Dijital ikiz (digital twin) kavramı, fiziksel bir sistemin sensörlerden toplanan gerçek zamanlı veriyle sürekli güncellenen sanal bir modelini ifade eder; bu model, fiziksel sistemin davranışının simüle edilmesine, arıza öngörüsüne ve bakım planlamasına imkân tanır (Wu vd., 2023). Tipik bir dijital ikiz akışında, fiziksel varlık üzerindeki sensörlerden toplanan veri önce bir iletişim katmanı aracılığıyla bulut veya kenar sunucudaki sanal modele aktarılır; sanal model bu veriyle sürekli güncellenerek fiziksel sistemle eşzamanlı hâle getirilir, üzerinde simülasyonlar ve tahminler koşturulur, elde edilen öngörüler (örneğin bir arızanın erken belirtileri) ise aktüatörlere veya operatöre geri beslenerek fiziksel sistem üzerinde eyleme dönüştürülür; bu çift yönlü veri akışı, dijital ikizi statik bir simülasyon modelinden ayıran temel özelliktir (Tao, Qi, Wang ve Nee, 2019). Dijital ikiz yaklaşımı, üretim sistemlerinin insan ihtiyaçları, sürdürülebilirlik ve dayanıklılık ekseninde yeniden konumlandırılmasını öngören **Endüstri 5.0** vizyonuyla da yakından ilişkilidir; bu vizyonda sensör ve aktüatör tabanlı otomasyon, insan operatörün yerini almaktan çok onun karar verme sürecini destekleyen bir araç olarak konumlandırılır (Adel, 2022). Bu eğilimlerin ortak paydası, sensör ve aktüatör donanımının artık yalnızca veri toplayan veya komut uygulayan pasif bileşenler değil, işleme ve karar verme kapasitesine sahip, ağa bağlı ve giderek daha özerk hâle gelen sistem bileşenleri olarak konumlanmasıdır.

SONUÇ

Bu bölümde ele alınan sensörler, aktüatörler, mikrodenetleyiciler ve gömülü sistem platformları, birbirinden bağımsız teknolojiler olarak değil, fiziksel dünya ile hesaplama dünyası arasında sürekli bir bilgi ve eylem alışverişi kuran bütüncül bir mimarinin bileşenleri olarak değerlendirilmelidir. Bir sensörün ürettiği ham sinyal, ancak uygun bir sinyal koşullandırma, örnekleme, filtreleme ve kalibrasyon sürecinden geçtiğinde güvenilir bir bilgiye dönüşür; bu bilgi, ancak uygun bir mikrodenetleyici veya işlemci mimarisi tarafından zamanında işlenip yorumlandığında anlamlı bir karara dönüşür; bu karar da ancak uygun bir aktüatör ve sürme devresi aracılığıyla fiziksel dünyaya doğru biçimde uygulandığında bir etki yaratır. Veri toplama sürecinin her aşamasında (örnekleme frekansının seçiminden zaman senkronizasyonuna, aykırı değer tespitinden güvenli iletime kadar) ortaya çıkabilecek hatalar, sistemin nihai güvenilirliğini doğrudan etkilediğinden, bu sürecin bütüncül biçimde tasarlanması ayrı bir önem taşır.

Arduino, ESP32 ve Raspberry Pi gibi platformların karşılaştırmalı incelemesi, tek bir "en iyi" platformun bulunmadığını, bunun yerine her platformun farklı bir problem sınıfına yanıt verecek biçimde konumlandığını göstermektedir. Arduino'nun düşük maliyetli, deterministik ve doğrudan donanım kontrolü gerektiren görevlerdeki uygunluğu; ESP32'nin kablosuz bağlantılı, düşük güçlü ve dağıtık sensör düğümü senaryolarındaki üstünlüğü; Raspberry Pi'nin ise işlem yoğun, çok bileşenli ve ağ geçidi işlevi gerektiren uygulamalardaki avantajı, platform seçiminin genel bir tercihten çok, uygulamanın işlem gücü, güç bütçesi, haberleşme ve maliyet gereksinimlerine bağlı mühendislik kararı olması gerektiğini ortaya koymaktadır. Bu üç platformun sıklıkla hiyerarşik biçimde bir arada kullanılması, gerçek dünya sistemlerinin nadiren tek bir donanım sınıfıyla sınırlı kaldığını, bunun yerine katmanlı bir mimari içinde birbirini tamamlayan bileşenlerden oluştuğunu göstermektedir.

Edge computing, TinyML ve sensör füzyonu gibi yaklaşımların yaygınlaşması, sensör-işlemci-aktüatör döngüsünün giderek daha fazla zekâyı veri kaynağına yakın konuma taşıdığını göstermektedir; bu eğilim, yalnızca teknik bir tercih değil, aynı zamanda ağ, gecikme, mahremiyet ve enerji kısıtlarına verilen bir yanıttır. Dijital ikiz ve Endüstri 5.0 gibi kavramların yükselişi ise, gömülü sensör ve aktüatör sistemlerinin artık yalnızca otomasyonun değil, insan merkezli, sürdürülebilir ve öngörücü

sistemlerin de temel yapı taşı olmaya devam edeceğine işaret etmektedir. Bu bakımdan sensör, aktüatör ve gömülü sistem teknolojilerinin bütüncül biçimde anlaşılması, yalnızca belirli bir donanımın nasıl kullanılacağını bilmekten öte, fiziksel ve dijital dünyaları birbirine bağlayan sistemlerin nasıl tasarlanacağını kavramanın bir ön koşulu olarak değerlendirilmelidir.

KAYNAKÇA

Adel, A. (2022). Future of industry 5.0 in society: human-centric solutions, challenges and prospective research areas. *Journal of Cloud Computing*, 11(1), 40. https://doi.org/10.1186/s13677-022-00314-5

Adu-Manu, K. S., Adam, N., Tapparello, C., Ayatollahi, H., & Heinzelman, W. (2018). Energy-harvesting wireless sensor networks (EH-WSNs): A review. *ACM Transactions on Sensor Networks*, 14(2), 1–50. https://doi.org/10.1145/3183338

Ahmed, S., Amin, A. A., Wajid, Z., & Ahmad, F. (2020). Reliable speed control of a permanent magnet DC motor using fault-tolerant H-bridge. *Advances in Mechanical Engineering*, 12(11). https://doi.org/10.1177/1687814020970311

Al-Okby, M. F. R., Roddelkopf, T., Fleischer, H., & Thurow, K. (2022). Evaluating a novel gas sensor for ambient monitoring in automated life science laboratories. *Sensors*, 22(21), 8161. https://doi.org/10.3390/s22218161

Arduino S.r.l. (t.y.). Arduino Uno Rev3 – Teknik özellikler. Erişim adresi: https://docs.arduino.cc/hardware/uno-rev3/ ve https://store.arduino.cc/products/arduino-uno-rev3

Castaneda, D., Esparza, A., Ghamari, M., Soltanpur, C., & Nazeran, H. (2018). A review on wearable photoplethysmography sensors and their potential future applications in health care. *International Journal of Biosensors & Bioelectronics*, 4(4), 195–202. https://doi.org/10.15406/ijbsbe.2018.04.00125

Chui, K. T., Gupta, B. B., Liu, J., Arya, V., Nedjah, N., Almomani, A., & Chaurasia, P. (2023). A survey of Internet of Things and cyber-physical systems: Standards, algorithms, applications, security, challenges, and future directions. *Information*, 14(7), 388. https://doi.org/10.3390/info14070388

Dahiya, E. S., Kalra, A., Lowe, A., & Anand, G. (2024). Wearable technology for monitoring electrocardiograms (ECGs) in adults: A scoping review. *Sensors*, 24(4), 1318. https://doi.org/10.3390/s24041318

Espressif Systems. (t.y.). ESP32 Series Datasheet, versiyon 5.3. Erişim adresi: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Fraden, J. (2016). *Handbook of modern sensors: Physics, designs, and applications* (5. bs.). Springer.

Han, X., Huang, M., Wu, Z., Gao, Y., Xia, Y., Yang, P., Shu, F., Lu, X., Yang, X., Lin, L., Su, W., Wang, L., Cui, Z., Zhao, Y., Li, Z., Zhao, L., & Jiang, Z. (2023). Advances in high-performance MEMS pressure sensors: Design, fabrication, and packaging. *Microsystems & Nanoengineering*, 9(1), 156. https://doi.org/10.1038/s41378-023-00620-1

International Organization for Standardization. (2024). ISO 11898-1:2024 — Road vehicles: Controller area network (CAN) — Part 1: Data link layer and physical signalling. Erişim adresi: https://www.iso.org/standard/86260.html

Naifar, S., Kanoun, O., & Trigona, C. (2024). Energy harvesting technologies and applications for the Internet of Things and wireless sensor networks. *Sensors*, 24(14), 4688. https://doi.org/10.3390/s24144688

Nazarahari, M., & Rouhani, H. (2021). Sensor fusion algorithms for orientation tracking via magnetic and inertial measurement units: An experimental comparison survey. *Information Fusion*, 76, 8–23. https://doi.org/10.1016/j.inffus.2021.04.009

NXP Semiconductors. (2021). *UM10204: I2C-bus specification and user manual* (Rev. 7.0). Erişim adresi: https://www.nxp.com/docs/en/user-guide/UM10204.pdf

OASIS. (2019). MQTT Version 5.0 — OASIS Standard. Erişim adresi: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

Rana, M., Mamun, Q., & Islam, R. (2021). Lightweight cryptography in IoT networks: A survey. *Future Generation Computer Systems*, 129, 77–89. https://doi.org/10.1016/j.future.2021.11.011

Ray, P. P. (2021). A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University – Computer and Information Sciences*, 34(4), 1595–1623. https://doi.org/10.1016/j.jksuci.2021.11.019

Raspberry Pi Ltd. (t.y.). Raspberry Pi 4 Model B – Teknik özellikler. Erişim adresi: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

Rosário, A. T., & Dias, J. C. (2023). How Industry 4.0 and sensors can leverage product design: Opportunities and challenges. *Sensors*, 23(3), 1165. https://doi.org/10.3390/s23031165

Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1), 10–21. https://doi.org/10.1109/JRPROC.1949.232969

Sinha, B. B., & Dhanalakshmi, R. (2021). Recent advancements and challenges of Internet of Things in smart agriculture: A survey. *Future Generation Computer Systems*, 126, 169–184. https://doi.org/10.1016/j.future.2021.08.006

Tao, F., Qi, Q., Wang, L., & Nee, A. Y. C. (2019). Digital twins and cyber-physical systems toward smart manufacturing and Industry 4.0: Correlation and comparison. *Engineering*, 5(4), 653–661. https://doi.org/10.1016/j.eng.2019.01.014

Ungurean, I. (2020). Timing comparison of the real-time operating systems for small microcontrollers. *Symmetry*, 12(4), 592. https://doi.org/10.3390/sym12040592

Varshney, A., Garg, N., Nagla, K. S., Nair, T. S., Jaiswal, S. K., Yadav, S., & Aswal, D. K. (2021). Challenges in sensors technology for Industry 4.0 for futuristic metrological applications. *MAPAN*, 36(2), 215–226. https://doi.org/10.1007/s12647-021-00453-1

Wu, H., Ji, P., Ma, H., & Xing, L. (2023). A comprehensive review of digital twin from the perspective of total process: Data, models, networks and applications. *Sensors*, 23(19), 8306. https://doi.org/10.3390/s23198306

OTONOM ARAÇLARDA DDOS SALDIRILARIN ENGELLENMESİ İÇİN YAPAY ZEKA DESTEKLİ SALDIRI TESPİT SİSTEMİN GELİŞTİRİLMESİ

İsa Tuncer*, Ahmet Ali Süzen**

GİRİŞ

Otomotiv endüstrisi son yirmi yılda mekanik odaklı bir yapıdan yazılım ve veri odaklı akıllı sistemlere doğru köklü bir dönüşüm geçirmiştir. Bu dönüşümün en uç noktası olan otonom araçlar (AV), gelişmiş sürüş destek sistemleri(ADAS), sensör füzyonu ve karar destek algoritmaları aracılığıyla insan müdahalesine duyulan ihtiyacı minimize etmeyi hedeflemektedir. Otonom araçların güvenli bir şekilde seyredebilmesi, araç içi ağlar (IVN) üzerinden akan verilerin doğruluğuna ve sürekliliğine doğrudan bağlıdır. Ancak, araç içi haberleşme altyapıları tasarlanırken siber güvenlik tehditlerinden ziyade operasyonel verimlilik ve düşük maliyet önceliklendirilmiştir (Avatefipour vd., 2019).

Otonom araçlarda kontrol üniteleri (ECU) arasındaki veri akışını sağlayan temel haberleşme omurgası, Denetleyici Alan Ağı (CAN bus) protokolüdür. 1980'li yıllarda geliştirilen CAN bus, yüksek güvenilirlik ve düşük hata oranı sunmasına rağmen, modern siber güvenlik mimarilerinden yoksundur (Avatefipour vd., 2019). Protokolün doğası gereği mesajların kimlik doğrulaması ve şifreleme mekanizmalarına sahip olmaması, ağın herhangi bir noktasına erişim sağlayan bir saldırganın tüm ağa kötü niyetli veri enjekte etmesine olanak tanımaktadır.

Bu zafiyetlerin en kritik yansımalarından biri Dağıtık Hizmet Engelleme (DDoS) saldırılarıdır. CAN bus üzerindeki öncelik tabanlı mesajlaşma yapısı kullanılarak ağın yoğun trafikle manipüle edilmesi, kritik güvenlik verilerinin ECU'lara ulaşmasını engelleyebilmekte ve otonom sistemin karar

* Yüksek Lisans, Isparta Uygulamalı Bilimler Üniversitesi, Lisansüstü Eğitim Enstitüsü, Türkiye, isatuncer@gmail.com, ORCID: https://orcid.org/0009-0000-4220-6224
** Doç. Dr., Isparta Uygulamalı Bilimler Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, Isparta, ahmetsuzen@isparta.edu.tr, ORCID: https://orcid.org/0000-0002-5871-1652

mekanizmasını felç ederek can ve mal kaybına yol açabilecek senaryolara zemin hazırlamaktadır (Marchetti ve Stabili, 2017).

Geleneksel imza tabanlı güvenlik duvarları, hızla evrilen ve karmaşıklaşan siber saldırı türlerini tespit etmekte yetersiz kalmaktadır. Bu noktada, otonom araç ağlarındaki normal trafik örüntülerini öğrenen ve bu örüntülerden sapmaları anomali olarak nitelendirebilen Yapay Zekâ (YZ) destekli Saldırı Tespit Sistemleri (IDS), proaktif bir savunma katmanı olarak öne çıkmaktadır (Hossain vd., 2020). Makine öğrenmesi ve derin öğrenme algoritmaları, CAN bus trafiği gibi yüksek boyutlu ve zaman serisi niteliğindeki verilerin analizinde yüksek başarı sergilemektedir (Ali Alheeti vd., 2015).

Bu çalışmada otonom araçların CAN bus ağları üzerinde gerçekleştirilen DDoS saldırılarını yüksek doğruluk ve düşük gecikme süresi ile tespit edebilen hibrit bir savunma mimarisi önermektedir. Çalışma kapsamında, klasik makine öğrenmesi yöntemlerinden Extra Trees ve Random Forest ile modern derin öğrenme yaklaşımlarından 1D-KSA ve DSA gibi farklı mimariler tasarlanmıştır. Çalışmasında sadece yüksek tespit oranlarına ulaşmak değil aynı zamanda otonom sürüşün en kritik gereksinimi olan gerçek zamanlılık kriterini karşılamaktır. Bu doğrultuda modellerin performansları standart sınıflandırma metriklerinin ötesinde milisaniye mertebesindeki tahmin süreleri ile de değerlendirilmiştir. Elde edilen bulgular özellikle HistGradientBoosting ve 1D-KSA modellerinin, araç içi ağ güvenliğinde otonom sistemin tepki süresini riske atmadan uygulanabilir bir çözüm sunduğunu kanıtlamaktadır.

LİTERATÜR ÖZETİ

Otonom ve bağlantılı araç teknolojilerindeki hızlı gelişim, bu sistemlerin operasyonel sürekliliğini ve yolcu güvenliğini doğrudan tehdit eden siber güvenlik risklerini de beraberinde getirmiştir. Literatürde yer alan çalışmalar özellikle CAN ve Internet of Vehicles (IoV) gibi kritik haberleşme altyapılarının yapısal zafiyetlerine odaklanmakta ve bu zafiyetlerin giderilmesi için çok katmanlı savunma stratejileri önermektedir. İlk dönemlerde kural tabanlı ve istatistiksel anomali tespiti yöntemleriyle başlayan güvenlik yaklaşımları, günümüzde büyük veri analitiği, makine öğrenmesi ve hibrit derin öğrenme modellerine doğru evrilerek daha proaktif ve yüksek doğruluklu bir yapıya kavuşmuştur.

Bu kapsamda gerçekleştirilen araştırmalar yeni veri setlerinin oluşturulması, simülasyon tabanlı saldırı testleri, karşılaştırmalı performans analizleri ve gerçek zamanlı uygulanabilirliği artırmayı hedefleyen hafifletilmiş mimariler üzerinde yoğunlaşmaktadır. Literatürdeki bu gelişim süreci sadece saldırı tespit oranlarını artırmayı değil, aynı zamanda otonom sürüşün gerektirdiği düşük gecikme süresi ve sınırlı kaynak kullanımı gibi operasyonel kısıtları da optimize etmeyi amaçlamaktadır. Aşağıda sunulan literatür özeti otonom araç güvenliğini sağlamaya yönelik geliştirilen bu teknolojik yaklaşımların metodolojik çeşitliliğini ve elde edilen deneysel başarıları kapsamlı bir şekilde ortaya koymaktadır.

Otonom ve bağlantılı araç ekosisteminde siber güvenlik, artık sadece ek bir özellik değil, fonksiyonel güvenliğin ayrılmaz bir parçası olarak kabul edilmektedir. Turgut (2025) ve Khokha (2024), otonom sistemlerdeki yazılım karmaşıklığı ve veri bağımlılığının yarattığı zafiyetlerin, ISO 26262 ve ISO/SAE 21434 gibi standartlar ışığında henüz tasarım aşamasındayken ele alınması gerektiğini vurgulamaktadır. Literatürdeki kapsamlı derleme çalışmaları, saldırı yüzeylerinin genişliğini ortaya koymaktadır; Morić vd. (2025) ve Yadav vd. (2025), CAN bus'tan V2X haberleşmesine kadar uzanan kritik hedefleri analiz ederken, Rajapaksha ve Kalutarage (2023) ile Banafshehvaragh ve Rahmani (2023), yapay zekâ tabanlı çözümlerin bu tehditlere karşı en güçlü aday olduğunu, ancak gerçek zamanlılık ve açıklanabilirlik gibi aşılması gereken engellerin bulunduğunu belirtmektedir. Mijwil vd. (2022) ise YZ tabanlı savunmaların adaptif saldırgan davranışlarına karşı sunduğu potansiyeli ve hesaplama maliyeti gibi kısıtları teorik bir çerçevede tartışmaktadır.

Saldırı Tespit Sistemlerinin (IDS) başarısı kullanılan verinin kalitesine ve test ortamının gerçekçiliğine doğrudan bağlıdır. Literatürde standart veri seti eksikliğini gidermek amacıyla Lampe ve Meng (2024) "can-train-and-test" adlı küratörlü bir CAN veri seti sunarken, Alhaidari ve Alrehan (2021) simülasyon tabanlı VANET trafik verileriyle yeni bir DDoS veri seti oluşturmuştur. Deneysel metodoloji tarafında ise Weaver vd. (2025), vehicle-in-the-loop (VIL) test yataklarının IDS doğrulaması için kritik olduğunu savunmaktadır. Benzer şekilde Khadka ve Efstathopoulos (2021), farklı güvenlik çözümlerinin aynı koşullar altında kıyaslanmasına olanak tanıyan sistematik bir benchmarking çerçevesi önermiş, ancak bu çalışmaların odağı bir algoritma eğitmekten ziyade metodolojik altyapı sağlamak olmuştur.

Otonom araçların iç iletişim omurgası olan CAN bus, DoS ve mesaj enjeksiyonu saldırılarına karşı oldukça savunmasızdır. Bu alandaki çalışmalar, derin öğrenme mimarilerinin yüksek başarısını göstermektedir. Gao vd. (2023) ve Hossain ve Kadobayashi (2020), CAN mesajlarını görüntü tabanlı veya zamansal temsillere dönüştürerek Konvolüsyonel Sinir Ağları ile sırasıyla %99.2 ve %97.8 gibi yüksek doğruluk oranlarına ulaşmışlardır. Amato ve Santone (2021) ile Rai vd. (2025) de benzer şekilde derin sinir ağlarının geleneksel yöntemlerden daha etkin olduğunu kanıtlamıştır. Performansı artırmak adına Altalbe (2024) özellik füzyonu ve stacking tabanlı topluluk öğrenmesini kullanırken; Khatri vd. (2023) transfer öğrenme ile veri kısıtı olan senaryolarda hızlı yakınsama sağlamıştır. Daha geleneksel makine öğrenmesi tarafında ise Avatefipour ve Mosavi (2019) ile Alfardus ve Rawat (2021), Rastgele Orman ve SVM gibi modellerin %95-97 bandında kararlı sonuçlar verdiğini raporlamıştır.

Araçların dış dünya ile iletişimi (V2X/IoV), saldırı hacminin büyüdüğü ve DDoS riskinin arttığı bir alandır. Ali Alheeti ve McDonald-Maier (2015) tarafından temelleri atılan bu alanda, günümüzde çok daha karmaşık yapılar önerilmektedir. Ahmed ve Anisi (2025), AIDAS sistemi ile saldırı tespiti ve kimlik doğrulamayı birleştirerek %99.1 doğruluğa ulaşırken; Sadaf ve Gadekallu (2024) belirsiz ağ koşullarında bulanık mantık (fuzzy logic) kullanarak esneklik sağlamıştır. Alladi vd. (2021) ve Anand vd. (2023), IoV ekosistemleri için ölçeklenebilir YZ mimarileri üzerinde durmuştur. Ağ yönetimi düzeyinde ise Amari vd. (2021) ve Butt vd. (2024), Yazılım Tanımlı Ağlar (SDN) üzerinden trafik izleme ve derin öğrenme ile DDoS saldırılarını %99'un üzerinde başarıyla engellemenin mümkün olduğunu göstermiştir. Al-Khulaidi vd. (2024) ve Badukale vd. (2021) ise VANET'lerdeki spesifik saldırı türlerini (Sybil, Blackhole vb.) derleyerek çözüm yöntemlerini sınıflandırmıştır.

Son dönem çalışmaları, sadece yüksek doğruluk oranına değil, aynı zamanda güvenilirlik ve dayanıklılığa odaklanmaktadır. Seo vd. (2018), Üretken Çekişmeli Ağlar (GAN) kullanarak bilinmeyen saldırı türlerini tespit etmede klasik denetimli öğrenmeyi geride bırakmıştır. Alsaade ve Al-Adhaileh (2023) ise derin otoenkoderler ile anomali tespiti yaparak %98 başarı sağlamıştır. Veri gizliliği ve dağıtık yapı ihtiyacı için Taslimasa vd. (2023) federatif öğrenme tabanlı ImageFed yaklaşımını önermiştir. Sistemin kararlılığı ve yanıt kapasitesi açısından Niyomdi ve Oluoch (2025) SOAR yaklaşımını, Prabha vd. (2024) ise bilişsel IDS yapılarını geliştirmiştir. Son

olarak, Ali vd. (2024) açıklanabilir yapay zekâ (XAI) tekniklerini (SHAP vb.) kullanarak IDS kararlarının yorumlanabilirliğini artırmış; Khemissa vd. (2024) ise performans değerlendirmesinde dayanıklılık ve saldırı sonrası toparlanma gibi çok boyutlu metriklerin önemini vurgulamıştır.

Literatürde yer alan otonom araç güvenliği çalışmalarının karşılaştırmalı analizi Tablo 1'de verilmiştir. IDS, ilk dönem kural tabanlı yaklaşımlardan günümüzün karmaşık derin öğrenme ve topluluk öğrenme modellerine doğru metodolojik bir evrim geçirdiğini göstermektedir. Hossain vd. (2020), Gao vd. (2023) ve Ahmed vd. (2025) gibi güncel çalışmaların %97 ile %99 arasında değişen yüksek doğruluk oranlarına ulaştığı görülse de bu araştırmaların büyük bir kısmının otonom sürüşün en kritik kısıtı olan gerçek zamanlı tepki sürelerini raporlamadığı saptanmıştır.

Tablo 1. *IDS tabanlı literatürün karşılaştırılması*

Çalışma	**Algoritma / Yöntem**	**Doğruluk**	**Gerçek Zamanlılık Analizi**
Ali Alheeti ve McDonald-Maier (2015)	Kural tabanlı ve istatistiksel anomali tespiti	Raporlanmadı	Belirtilmemiş
Seo vd. (2018)	GAN (Üretici-Ayırt Edici Ağ)	~%99	Belirtilmemiş
Avatefipour ve Mosavi (2019)	Random Forest, SVM, KNN	>%95	Belirtilmemiş
Hossain ve Kadobayashi (2020)	Konvolüsyonel Sinir Ağları (KSA)	%97.8	Belirtilmemiş
Reddy ve Velliangiri (2021)	Hibrit KSA ve LSTM	%98.6	Belirtilmemiş
Gao vd. (2023)	Hafif (Lightweight) KSA	%99.2	Düşük Gecikme
Sadaf ve Gadekallu (2024)	Bulanık Mantık (Fuzzy Logic)	%96.2	Esnek Yapı
Altalbe (2024)	Stacking Ensemble (SVM, RF, KNN)	~%99	Belirtilmemiş
Taslimasa vd. (2023)	Federatif Öğrenme + KSA	~%99	Belirtilmemiş
Khatri vd. (2023)	Transfer Öğrenme + DSA	%97–99	Hızlı Yakınsama
Aljabri vd. (2025)	Karar Ağaçları ve Lojistik Regresyon	%96–98	Düşük Gecikme
Ahmed ve Anisi (2025)	Derin Öğrenme + Kimlik Doğrulama	%99.1	Çok Katmanlı

DDOS TESPİT MODELİNİN GELİŞTİRİLMESİ

Bu bölümde otonom araçlarda DDoS saldırılarının engellenmesine yönelik geliştirilen yapay zekâ destekli saldırı tespit sisteminin metodolojik yapısı ayrıntılı olarak açıklanmaktadır. Yöntemsel süreç veri setinin hazırlanması, yapay zekâ tabanlı model tasarımı, modelin eğitimi ve test edilmesi, performans değerlendirmesi ve gerçek zamanlı uygulanabilirlik analizini içermektedir. Belirlenen bu aşamalar sistematik bir yaklaşımla ele alınarak saldırı tespit sürecinin bütüncül biçimde değerlendirilmesini amaçlamaktadır.

Otonom araçlarda DDoS saldırılarının tespitine yönelik geliştirilen yapay zekâ tabanlı saldırı tespit sistemlerinde, farklı algoritmaların birlikte değerlendirilmesi saldırı davranışlarının daha kapsamlı biçimde analiz edilmesine olanak sağlamaktadır. Araç içi ağlar ve otonom araç iletişim sistemlerinden elde edilen veriler, doğrusal olmayan ilişkiler, zamansal bağımlılıklar ve yüksek boyutlu özellikler içermektedir. Bu nedenle, tek bir algoritma yerine farklı öğrenme yaklaşımlarına sahip modellerin kullanılması, saldırı tespit performansının daha sağlıklı biçimde değerlendirilmesini mümkün kılmaktadır. Literatürde, otonom araç ve CAN bus tabanlı saldırı tespit çalışmalarında klasik makine öğrenmesi algoritmaları ile derin öğrenme tabanlı modellerin birlikte kullanıldığı görülmektedir. Bu yaklaşımlar hem yorumlanabilirliği yüksek temel modellerin hem de karmaşık veri örüntülerini öğrenebilen derin öğrenme algoritmalarının avantajlarından yararlanılmasını sağlamaktadır (Amato ve Santone, 2021). Bu kapsamda ele alınan Extra Trees, Rastgele Orman, Derin Sinir Ağları, Konvolüsyonel Sinir Ağları ve HistGradientBoosting farklı saldırı senaryolarındaki performansları ve güçlü yönleri açısından ayrı ayrı değerlendirilmiştir.

Veri Kümesinin Eğitim ve Test Olarak Ayrılması

Bu tez çalışmada otonom ve bağlantılı araçların ağ güvenliği araştırmalarında literatürde referans bir kıyaslama noktası olarak kabul edilen ve Hacking and Countermeasure Research Lab (HCRL) tarafından sunulan Car Hacking Dataset kullanılmıştır. Söz konusu veri kümesi gerçek bir aracın On-Board Diagnostics portu üzerinden operasyonel süreçler esnasında kaydedilen gerçek zamanlı CAN trafiğini içermektedir. Veri seti normal sürüş kayıtlarının yanı sıra sisteme dışarıdan enjekte edilen Hizmet Engelleme, Bulanık ve Spoofing gibi kritik saldırı türlerini zaman damgası,

CAN ID ve DLC/ Payload gibi özniteliklerle birlikte sunmaktadır (Seo vd., 2025).

Yapay zekâ tabanlı saldırı tespit modellerinin güvenilir ve nesnel biçimde değerlendirilebilmesi için kullanılan veri setinin eğitim ve test alt kümelerine ayrılması gerekmektedir. Bu ayrım modelin öğrenme süreci ile genelleme yeteneğinin birbirinden bağımsız olarak analiz edilmesine olanak sağlamaktadır.

Bu çalışma kapsamında kullanılan veri kümesi ve içerdiği CAN Bus parametreleri Tablo 3'de gösterilmiştir. Bu veri kümesine normal trafik (Label = 0) ve DDoS saldırı trafiği (Label = 1) içerecek şekilde etiketlendikten sonra literatürde yaygın olarak kullanılan %80 eğitim – %20 test oranına göre bölünmüştür. Eğitim veri seti, modelin saldırı ve normal trafik örüntülerini öğrenmesi amacıyla kullanılırken; test veri seti, eğitilen modelin daha önce görmediği veriler üzerindeki performansını değerlendirmek için ayrılmıştır. Bu yaklaşım, aşırı öğrenme riskinin azaltılmasına ve model performansının genelleme kabiliyetinin daha sağlıklı biçimde ölçülmesine katkı sağlamaktadır. Veri setinin sınıflara göre dağılımı ve eğitim/test ayrımı sonrasında elde edilen örnek sayıları Tablo 2'de sunulmaktadır.

Tablo 2. *Veri setinin eğitim ve test kümelerine göre dağılımı*

Veri Kümesi	**Normal**	**DDoS**	
Eğitim Kümesi (%70)	649 695	2 566 039	3 215 734
Doğrulama (%10)	92 814	366 577	459 391
Test Kümesi (%20)	185 627	733 154	918 781
Toplam	**928 136**	**3 665 770**	**4 593 906**

Tablo 3. *CAN bus veri setinde kullanılan parametreler ve açıklamaları*

Parametre	**Açıklama**
Timestamp	CAN mesajının ağ üzerinde iletildiği zaman bilgisini ifade eder. Mesajlar arasındaki zaman farklarının hesaplanmasında kullanılır.
ID	CAN mesajının kimliğini belirtir. Mesajın önceliğini ve hangi ECU tarafından gönderildiğini temsil eder. Düşük ID değeri daha yüksek önceliğe sahiptir.
DLC (Data Length Code)	CAN mesajı içerisindeki veri alanının uzunluğunu (0–8 byte) gösterir.

DATA0	CAN çerçevesindeki ilk veri baytıdır. Araç içi sensör veya kontrol bilgisi içerebilir.
DATA1	CAN çerçevesindeki ikinci veri baytıdır.
DATA2	CAN çerçevesindeki üçüncü veri baytıdır.
DATA3	CAN çerçevesindeki dördüncü veri baytıdır.
DATA4	CAN çerçevesindeki beşinci veri baytıdır.
DATA5	CAN çerçevesindeki altıncı veri baytıdır.
DATA6	CAN çerçevesindeki yedinci veri baytıdır.
DATA7	CAN çerçevesindeki sekizinci veri baytıdır.
time_diff	Ardışık CAN mesajları arasındaki zaman farkını temsil eder. DDoS saldırılarında mesaj frekansındaki ani artışları yakalamak için kullanılır.
id_freq	Belirli bir CAN ID'sinin kısa zaman aralığında kaç kez tekrarlandığını gösterir. DDoS ve flooding saldırılarında ayırt edici bir özelliktir.
rolling_10	Son 10 mesajlık kayan pencere (rolling window) içinde mesaj yoğunluğunu ifade eder. Trafik patlamalarının tespitinde kullanılır.

Çalışmada kullanılan veri setinin 4.593.906 örnekten oluşan yüksek hacimli yapısı dikkate alınarak model değerlendirme sürecinde K-katlı çapraz doğrulama yerine Hold-out yöntemi tercih edilmiştir. Literatürde veri setinin milyonlarca örnek içerdiği durumlarda, test kümesinin ana kütleyi temsil etme kabiliyetinin arttığı ve istatistiksel varyansın minimize olduğu belirtilmektedir (Goodfellow vd., 2016). Yaklaşık 918.782 örnekten oluşan test seti modellerin genelleme performansını ölçmek için istatistiksel olarak yeterli güven düzeyini sağlamaktadır. Ayrıca derin öğrenme modellerinin (1D-KSA ve DSA) eğitim maliyetleri ve zaman gereksinimleri göz önüne alındığında Hold-out yönteminin kullanılmasının deneysel verimlilik açısından daha uygun olduğu değerlendirilmiştir.

Tahmin Modelinin Geliştirilmesi

Bu çalışma kapsamında önerilen makine öğrenmesi ve derin öğrenme modellerinin eğitim, optimizasyon ve test süreçleri, bulut tabanlı bir geliştirme ortamı olan Google Colab platformu üzerinde gerçekleştirilmiştir. Yaklaşık 4.6 milyon örnekten oluşan yüksek hacimli veri setinin işlenmesi ve özellikle 1D-KSA ile DSA gibi derin öğrenme mimarilerinin karmaşık hesaplama gereksinimleri nedeniyle platformun sunduğu yüksek performanslı Grafik İşlemci Birimi (GPU) ve genişletilmiş RAM imkanlarından yararlanılmıştır. Yazılım geliştirme sürecinde Python

programlama dili tercih edilmiş, klasik makine öğrenmesi modelleri (Extra Trees, Rastgele Orman, HistGradientBoosting) için Scikit-learn, derin öğrenme modelleri (DSA, 1D-KSA) için ise TensorFlow ve Keras kütüphaneleri kullanılmıştır.

Bu çalışma kapsamında kullanılan makine öğrenmesi ve derin öğrenme tabanlı modellerin sınıflandırma performanslarını en üst düzeye çıkarmak amacıyla hiper parametre optimizasyonu gerçekleştirilmiştir. Hiperparametreler modelin öğrenme sürecini doğrudan etkileyen ve eğitim sırasında otomatik olarak öğrenilmeyen ayarları ifade etmektedir. Bu parametrelerin uygun şekilde belirlenmesi, özellikle karmaşık ve gerçek zamanlılık gerektiren DDoS saldırı tespit problemlerinde model başarımı açısından kritik bir öneme sahiptir. Hiperparametre optimizasyonu sürecinde GridSearchCV yöntemi kullanılmıştır. GridSearchCV önceden tanımlanan hiper parametre aralıkları üzerinde tüm olası kombinasyonları sistematik olarak deneyen ve her bir kombinasyonu çapraz doğrulama yaklaşımı ile değerlendiren bir arama yöntemidir. Bu sayede, modellerin yalnızca eğitim verisi üzerinde değil, daha önce görülmemiş veriler üzerinde de tutarlı ve genellenebilir sonuçlar üretmesi hedeflenmiştir.

Bu çalışmada GridSearchCV yöntemi ile gerçekleştirilen hiper parametre arama süreci kullanılan model türüne göre farklı parametreler üzerinden yürütülmüştür. Ağaç tabanlı algoritmalarda, modelin karmaşıklığını ve aşırı öğrenme eğilimini kontrol eden ağaç sayısı, maksimum derinlik ve bölünme kriterleri gibi parametreler dikkate alınmıştır. Özellikle Extra Trees ve Random Forest modellerinde, rastgelelik derecesi ve ağaç yapısının kontrol edilmesi yoluyla hem sınıflandırma doğruluğunun artırılması hem de modelin genelleme yeteneğinin iyileştirilmesi amaçlanmıştır.

HistGradientBoosting modelinde ise öğrenme sürecinin kademeli yapısı nedeniyle öğrenme oranı ve ağaç derinliği gibi parametreler ön plana çıkmıştır. Bu parametreler, modelin saldırı trafiğindeki ani değişimleri ne ölçüde hassasiyetle öğrenebileceğini doğrudan etkilemektedir. Derin öğrenme tabanlı modellerde (DSA ve KSA) ise ağ mimarisinin yapısını belirleyen katman sayısı, nöron sayıları, aktivasyon fonksiyonları, optimizasyon algoritmaları ve öğrenme oranı gibi hiperparametreler dikkate alınarak en uygun yapılandırma belirlenmiştir.

GridSearchCV yöntemi ile gerçekleştirilen bu optimizasyon süreci sonucunda, her bir model için en yüksek sınıflandırma başarımını sağlayan hiperparametre kombinasyonları belirlenmiş ve modeller bu parametreler

kullanılarak yeniden eğitilmiştir. Kullanılan hiperparametre aralıkları ve her bir model için elde edilen en uygun değerler Tablo 4'de sunulmuştur.

Tablo 4. GridSearchCV yöntemi ile kullanılan modeller için hiperparametre aralıkları ve optimum değerler

Model Adı	**Hiperparametre**	**Aralıklar**	**Değer**
Extra Trees	n_estimators	[50, 100, 200]	100
	max_depth	[None, 10, 20]	20
	max_features	['auto', 'sqrt']	sqrt
Random Forest	n_estimators	[100, 150]	150
	max_depth	[10, 20]	10
	min_samples_split	[3, 5]	5
	min_samples_leaf	[2, 5]	2
	max_features	['auto', 'sqrt']	sqrt
HistGradientBoosting	learning_rate	[0.01, 0.05, 0.1]	0.05
	max_depth	[8, 10, 12]	12
	max_iter	[200, 300]	300
DSA	epochs	[5, 10]	10
	batch_size	[32, 64]	64
	learning_rate	[0.001, 0.0005]	0.001
	activation	['relu']	relu
1D-KSA	filters	[32, 64]	64
	kernel_size	[3, 5]	3
	epochs	[5, 10]	10
	batch_size	[32, 64]	64
	learning_rate	[0.001, 0.0005]	0.001

Modellerin Sınıflandırma Performans Bulguları

Bu bölümde DoS/DDoS saldırılarının tespiti amacıyla kullanılan makine öğrenmesi ve derin öğrenme tabanlı modellerin sınıflandırma performansları sunulmaktadır. Geliştirilen modellerin başarımı test veri seti üzerinde hesaplanan doğruluk, hassasiyet, duyarlılık, F1-skoru ve ROC-AUC performans metrikleri kullanılarak değerlendirilmiştir. Değerlendirme sürecinde modellerin yalnızca doğru sınıflandırma oranları değil, aynı zamanda saldırı sınıfını ayırt edebilme yetenekleri de dikkate alınmıştır.

Otonom araç içi ağ güvenliği gibi hata toleransının hayati riskler taşıdığı alanlarda modellerin başarısı değerlendirilirken Duyarlılık metriği temel performans göstergesi olarak kabul edilmiştir. Literatürde siber-fiziksel sistemlerde bir saldırının tespit edilememesinin can ve mal güvenliği üzerinde yıkıcı etkiler yaratabileceği buna karşın hatalı alarmların yaratacağı operasyonel maliyetin tolere edilebilir olduğu vurgulanmaktadır.

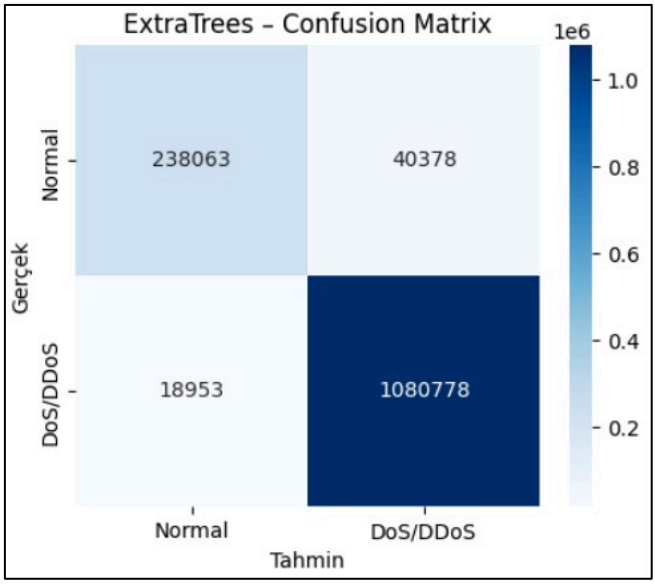
Tablo 5'de verilen ve karışıklık matrislerinden elde edilen verilerle doğrulanan performans metrikleri geliştirilen modellerin otonom araç içi CAN bus ağındaki siber saldırıları yüksek bir başarıyla ayırt edebildiğini kanıtlamaktadır. Genel sınıflandırma başarısı temel alındığında Rastgele Orman modeli 0.9579 doğruluk ve 0.9739 F1-skoru ile en dengeli genel performansı sergilerken, Extra Trees modelinin 0.9640 değeriyle en yüksek kesinlik oranına ulaştığı ve böylece normal trafiğin hatalı etiketlenme olasılığını minimize ettiği görülmektedir. Bununla birlikte otonom sürüş güvenliği gibi hata toleransının hayati risk taşıdığı bir uygulama alanında en kritik başarı kriteri olan duyarlılık değerleri incelendiğinde derin öğrenme temelli modellerin konvansiyonel makine öğrenmesi algoritmalarına karşı belirgin bir üstünlük sağladığı tespit edilmiştir.

Özellikle 1D-KSA modeli matris verilerinden hesaplanan 0.9996 duyarlılık oranıyla gerçek saldırıların neredeyse tamamını yakalayarak güvenlik açısından en güvenilir savunma katmanını sunmaktadır. Benzer şekilde HistGradientBoosting modeli, 0.9971 duyarlılık değeriyle yanlış negatif oranını düşük seviyede tutarken DSA mimarisi de 0.9985 duyarlılık oranı ile benzer bir güvenlik marjı sağlamaktadır. Sonuç olarak yüksek hacimli veri seti üzerinde elde edilen bu bulgular otonom araçların siber dayanıklılığını artırmada 1D-KSA ve HistGradientBoosting mimarilerinin kilit bir rol oynayabileceğini ve özellikle atlanan saldırı riskini minimize etmede derin öğrenme yaklaşımlarının tercih edilmesi gerektiğini akademik düzeyde doğrulamaktadır.

***Tablo 5**. Modellerin sınıflandırma performans metrikleri*

Model	Doğruluk	Kesinlik	**Duyarlılık**	F1-Skoru
Extra Trees	0.9569	0.9640	**0.9828**	0.9733
Rastgele Orman	0.9579	0.9633	**0.9847**	0.9739
HistGradientBoosting	0.9567	0.9510	**0.9971**	0.9735
DSA	0.9481	0.9401	**0.9985**	0.9684
1D-KSA	0.9448	0.9356	**0.9996**	0.9665

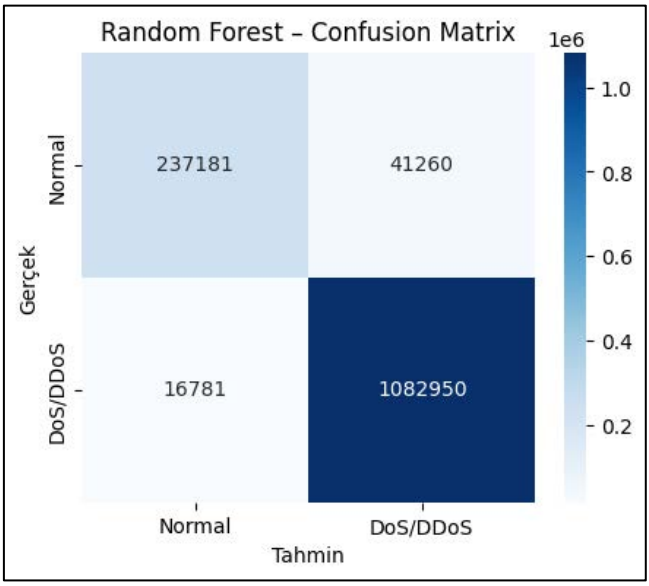
Şekil 1'de verilen Extra Trees modeline ait karışıklık matrisi bu topluluk öğrenme yönteminin otonom araç içi CAN bus ağındaki saldırı trafiğini tespit etme performansını sergilemektedir. Sonuçlar incelendiğinde modelin 1.099.731 saldırı mesajından 1.080.778'ini doğru şekilde sınıflandırarak yaklaşık %98.28 oranında bir duyarlılık değerine ulaştığı görülmektedir.



***Şekil 1.** ExtraTrees modeline ait karışıklık matrisi*

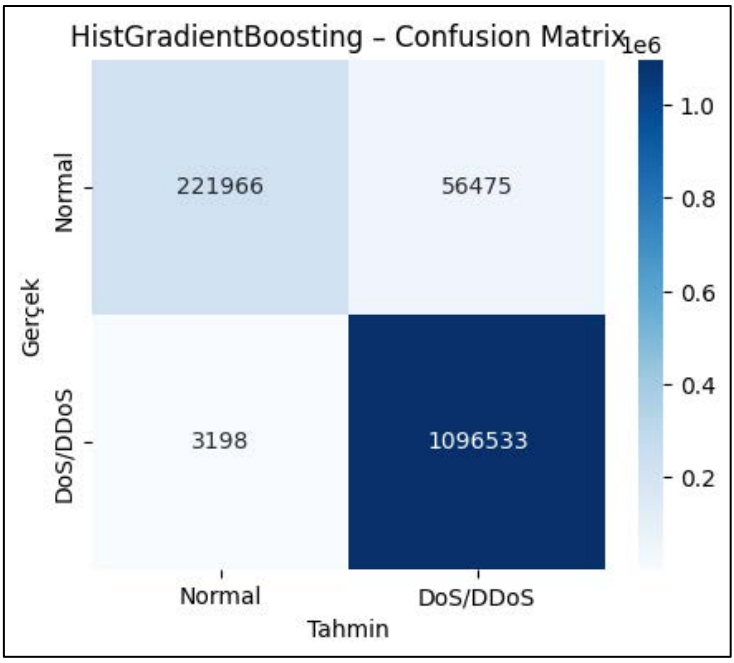
Şekil 2'de sunulan Rastgele Orman modeline ait karışıklık matrisi geliştirilen sistemin otonom araç içi CAN bus ağındaki normal trafik ile DDoS saldırılarını ayırt etme kabiliyetini detaylı olarak yansıtmaktadır. Toplam 1.378.172 örnek içeren test seti üzerindeki performans

incelendiğinde modelin özellikle saldırı trafiğini tespit etmedeki başarısı dikkat çekicidir. 1.099.731 saldırı örneğinin 1.082.950'sinin doğru tahmin edilmesiyle elde edilen yaklaşık %98.47'lik duyarlılık oranı güvenlik kritik bir sistem olan otonom araçlarda saldırıların çok düşük bir hata payı ile yakalandığını kanıtlamaktadır. Diğer yandan normal trafik sınıflarında gözlemlenen 41.260 yanlış pozitif değeri, sistemin normal mesajları saldırı olarak etiketleme eğilimini gösterse de, saldırıların kaçırılma oranını temsil eden yanlış negatif değerinin bu denli düşük kalması, can ve mal güvenliği açısından öncelikli olan saldırıyı atlamama kriterinin başarıyla karşılandığını ortaya koymaktadır.



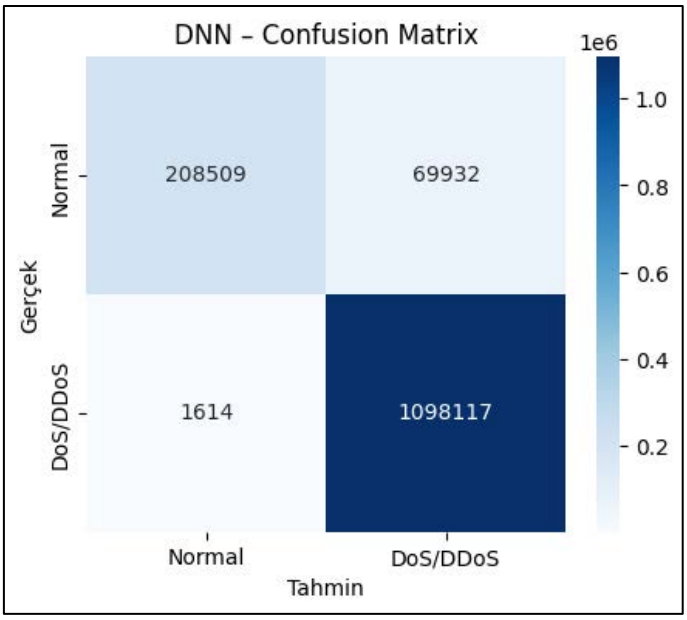
Şekil 1. *Rastgele orman modeline ait karışıklık matrisi*

Şekil 3'de sunulan HistGradientBoosting modeline ait karışıklık matrisi gradyan artırma temelli bu yaklaşımın otonom araç içi CAN bus ağındaki saldırı trafiğini ayırt etmedeki karakteristik performansını ortaya koymaktadır. Toplam 1.378.172 test örneği üzerindeki sınıflandırma sonuçları incelendiğinde modelin gerçek saldırı trafiğini yakalama başarısını temsil eden duyarlılık oranının yaklaşık %99.71 gibi oldukça yüksek bir seviyeye ulaştığı görülmektedir. 1.099.731 saldırı mesajından yalnızca 3.198 adedinin Normal olarak hatalı etiketlenmiş olması, sistemin güvenlik açısından en kritik hata türü olan saldırıyı atlama riskini minimize ettiğini kanıtlamaktadır.



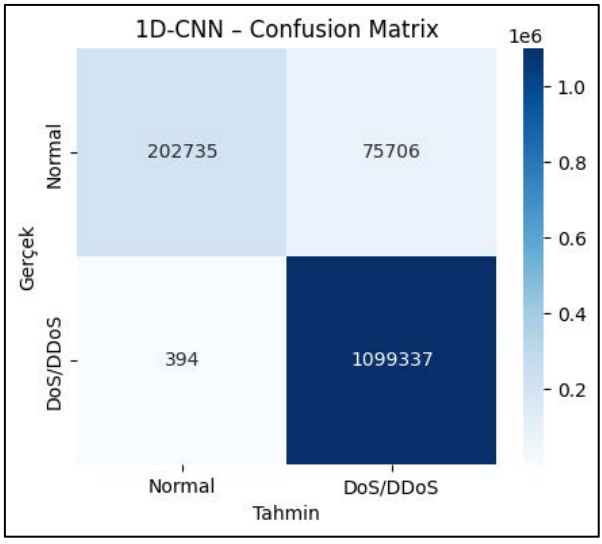
***Şekil 2**. HistGradientBoosting modeline ait karışıklık matrisi*

Şekil 4'te sunulan DSA modeline ait karışıklık matrisi derin öğrenme temelli bu yaklaşımın otonom araç içi iletişim ağındaki siber saldırıları karakterize etme ve ayırt etme noktasındaki üstün başarısını ortaya koymaktadır. Modelin 1 099 731 saldırı mesajından 1 098 117'sini doğru şekilde sınıflandırarak yaklaşık %99.85 oranında bir duyarlılık değerine ulaştığı görülmektedir. Bu performans incelenen önceki modellerle (RF ve HGB) kıyaslandığında yanlış negatif sayısının 1614'e kadar düşürülmesini sağlamış ve sistemin saldırıları kaçırma olasılığını en düşük seviyeye indirmiştir. Otonom araç güvenliği gibi hata payının hayati riskler taşıdığı bir alanda yanlış negatiflerin bu denli minimize edilmesi sistemin güvenilirliğini akademik açıdan pekiştirmektedir. Her ne kadar 69 932 örnekte gözlemlenen yanlış pozitif değeri modelin normal trafiği saldırı olarak niteleme konusunda hassas bir eğilim sergilediğini gösterse de saldırıların %99.85 kesinlikle yakalanması önerilen DSA mimarisinin CAN bus güvenliği için son derece etkili bir savunma katmanı sunduğunu kanıtlamaktadır.



Şekil 3. *DSA modeline ait karışıklık matrisi*

Şekil 5'te sunulan 1D-KSA modeline ait karışıklık matrisi bu derin öğrenme mimarisinin otonom araç içi ağlardaki DDoS saldırılarını karakterize etmedeki olağanüstü başarısını ortaya koymaktadır. Toplam 1.378.172 test örneği üzerinden yapılan analizde, 1.099.731 gerçek saldırı örneğinin 1.099.337'sinin doğru tahmin edilmesiyle yaklaşık %99.96 oranında bir duyarlılık değerine ulaşılmıştır. Bu sonuç çalışmada yer alan tüm modeller arasındaki en yüksek saldırı yakalama oranını temsil etmektedir. Sadece 394 saldırı mesajının Normal olarak hatalı etiketlenmesi 1D-KSA modelinin CAN bus üzerindeki siber tehditleri neredeyse tam bir kesinlikle izole edebildiğini kanıtlamaktadır. Her ne kadar 75.706 örnekte görülen yanlış pozitif değeri, modelin normal trafik örüntülerini saldırı olarak niteleme konusunda diğer modellere göre biraz daha agresif bir tutum sergilediğini gösterse de güvenlik kritik otonom sistemlerde sıfıra yakın saldırı kaçırma hedefi hayati önem taşımaktadır.



Şekil 4. *1D-KSA modeline ait karışıklık matrisi*

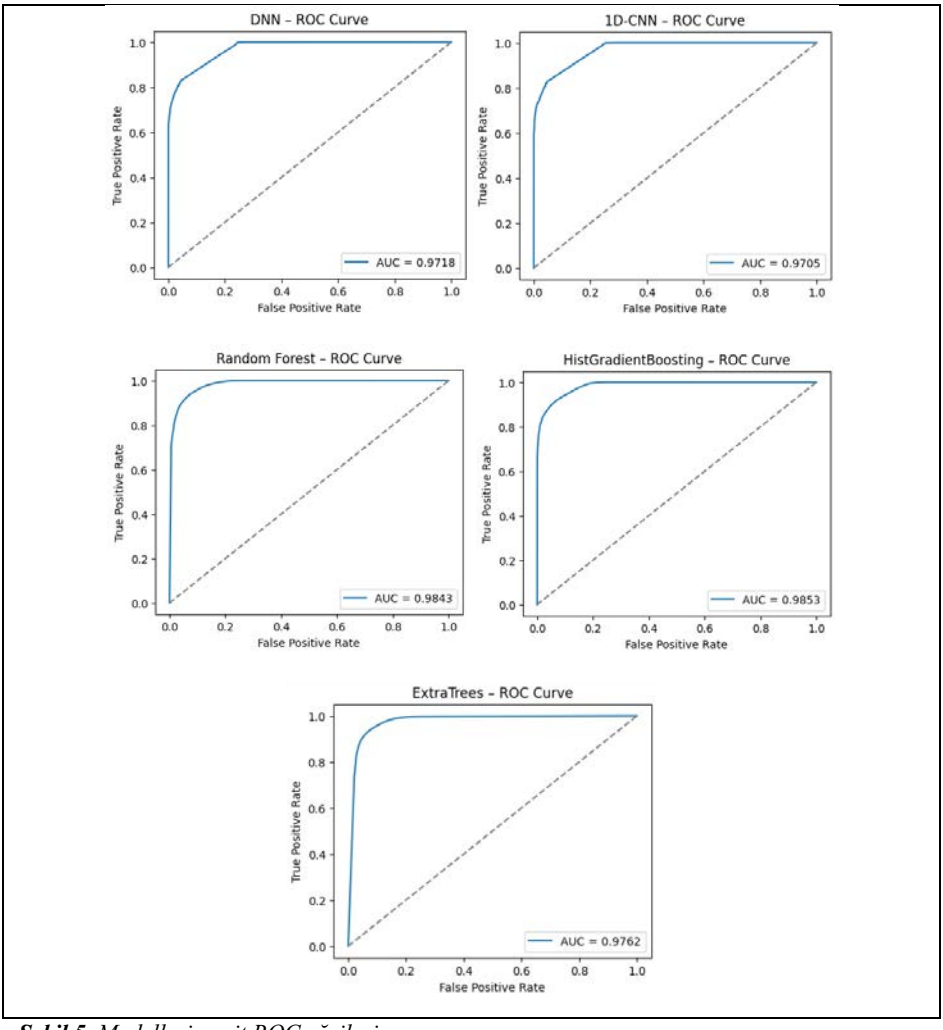
ROC Eğrisi ve Ayırt Edicilik Bulguları

Bu bölümde çalışmada kullanılan modellerin saldırı ve normal trafik sınıflarını ayırt edebilme yetenekleri, Receiver Operating Characteristic (ROC) eğrileri ve AUC (Area Under Curve) değerleri kullanılarak değerlendirilmiştir. ROC eğrileri farklı karar eşik değerleri altında doğru pozitif oranı ile yanlış pozitif oranı arasındaki ilişkiyi göstermesi açısından önemli bir performans göstergesidir.

Şekil 4.6'da geliştirilen beş farklı modelin (Extra Trees, Rastgele Orman, HistGradientBoosting, DSA ve 1D-KSA CAN bus üzerindeki saldırı trafiğini normal trafikten ayırt etme kapasitesini gösteren ROC eğrileri ve bu eğrilerin altında kalan alan AUC değerleri sunulmuştur. Şekil 4.6'da verilen ROC eğrileri geliştirilen modellerin otonom araç içi CAN bus ağındaki saldırı trafiğini normal trafikten ayırt etme noktasındaki yüksek hassasiyetini ve kararlılığını göstermektedir. Tüm modellere ait AUC değerlerinin 0.9705 ile 0.9853 gibi dar ve yüksek bir bantta yer alması modellerin rastgele tahminin çok ötesinde bir ayırt edicilik kapasitesine sahip olduğunu göstermektedir. Toplam alan bazında yapılan değerlendirmede HistGradientBoosting ve Rastgele Orman modelleri en yüksek skorları elde ederek sınıflandırma dengesi açısından öne çıkarken, Extra Trees modelinin de 0.9762 AUC değeriyle bu başarıyı desteklediği görülmektedir. Derin öğrenme mimarileri olan DNN ve 1D-CNN modellerinin eğrileri, toplam alan bakımından ağaç tabanlı modellerin bir miktar gerisinde kalsa da, çok

düşük Yanlış Pozitif Oranlarında dahi %100 Doğru Pozitif Oranına hızla ulaşmaları çalışmanın temel önceliği olan yüksek duyarlılık hedefini istatistiksel olarak doğrulamaktadır.

ROC eğrisinin Y ekseni aslında modelin duyarlılık değeridir. Modellerinizin Çizelge 4.5'te 0.98 ile 0.99 arasında değişen çok yüksek duyarlılık değerlerine sahip olması ROC eğrilerinin grafik başlangıcında neden bu kadar dik bir ivmeyle yukarı tırmandığını açıklamaktadır. Bu durum modellerin çok düşük yanlış alarm oranlarında dahi saldırıların neredeyse tamamını yakalayabildiğini kanıtlamaktadır.



Şekil 5. *Modellerine ait ROC eğrileri*

Gerçek Zamanlı Performans Bulguları

Bu bölümde çalışmada kullanılan makine öğrenmesi ve derin öğrenme tabanlı modellerin gerçek zamanlı ortamlarda uygulanabilirliği tahmin süresi ölçütü temel alınarak değerlendirilmiştir. Gerçek zamanlı saldırı tespit sistemlerinde modelin sınıflandırma kararını ne kadar sürede üretebildiği kritik bir performans kriteri olarak kabul edilmektedir. Özellikle otonom araç sistemlerinde düşük gecikme süresi güvenli ve kesintisiz çalışmanın sağlanabilmesi açısından büyük önem taşımaktadır.

Tablo 7'de geliştirilen saldırı tespit sisteminin otonom araç içi ağlarda gerçek zamanlı çalışabilirliğini, milisaniye cinsinden ölçülen dört temel metrik üzerinden analiz edilmektedir. Çizelge 4.2'de yer alan Ortalama Tahmin Süresi, modelin tek bir veri girişi için harcadığı tipik işlem süresini ifade ederken test setindeki tüm tahmin sürelerinin toplamının örnek sayısına bölünmesiyle elde edilir.

p5 Tahmin Süresi, tahminlerin %95'inin altında kaldığı üst sınırı belirleyerek emniyet kritik sistemlerdeki gecikme kararlılığını temsil etmektedir. Tüm tahmin süreleri küçükten büyüğe sıralandığında en yavaş %5'lik dilimin başlangıç sınırını temsil eder. Bu değer sistemin en kötü senaryolarda dahi ne kadar gecikme yaşatacağını gösterdiği için otonom araç güvenliğinde güvenilirlik ölçütü olarak kullanılır.

Modellerin tahmin yürütebilmesi için ağdan veri toplama süresini simgeleyen Window Gecikmesi, tüm mimariler için 10.0 ms olarak sabitlenmiş yani sistem tahmin yapmadan önce her zaman 10 milisaniyelik bir ağ trafiğini gözlemler. Toplam Tespit Süresi ise p95 gecikmesi ile Windows gecikmesi süresinin toplamından oluşarak bir siber tehdidin fark edilmesine kadar geçen nihai süreyi belirtmektedir.

Ölçümler genellikle milisaniye hassasiyetinde sonuç veren yüksek çözünürlüklü zamanlayıcılardan olan Python'da time.perf_counter() kullanılarak gerçekleştirilmiştir.

Tablo 7'deki veriler genel olarak değerlendirildiğinde derin öğrenme tabanlı DSA ve 1D-KSA modellerinin sırasıyla 10.64 ms ve 10.90 ms toplam tespit süreleriyle en yüksek performansı sergilediği görülmektedir. Bu modellerin p95 tahmin sürelerinin 1 ms'nin dahi altında kalması (0.64 ms ve 0.90 ms), CAN bus üzerindeki yüksek trafik yoğunluğunda dahi işlemci yükünü minimize ederek anlık savunma yapabildiklerini kanıtlamaktadır. HistGradientBoosting modeli, 14.58 ms toplam süresiyle özet metinde

hedeflenen 10–15 ms bandını başarıyla karşılarken; Extra Trees (86.89 ms) ve Rastgele Orman (70.58 ms) modellerinin yüksek hesaplama maliyetleri nedeniyle gerçek zamanlı otonom sürüş güvenliği için kısıtlı bir uygulanabilirlik sunduğu müşahede edilmektedir.

Tablo 7. *Modellerin gerçek zamanlı DoS/DDoS tespit performansları*

Model	Ortalama Tahmin Süresi (ms)	p95 Tahmin Süresi (ms)	Window Gecikmesi (ms)	Toplam Tespit Süresi (ms)
Extra Trees	68.43	76.89	10.0	**86.89**
Rastgele Orman	53.70	60.58	10.0	**70.58**
HistGradientBoosting	4.37	4.58	10.0	**14.58**
DSA	0.60	0.64	10.0	**10.64**
1D-KSA	0.81	0.90	10.0	**10.90**

Modellerin Literatürle Karşılaştırmalı Performansı

Bu bölümde literatürde otonom araçlar ve CAN bus için önerilen saldırı tespit yaklaşımları ile bu çalışmada geliştirilen yöntemlerin performansları karşılaştırmalı olarak ele alınmıştır. Karşılaştırma özellikle gerçek zamanlı tepki süresi açısından gerçekleştirilmiştir. Gerçek zamanlı saldırı tespit sistemlerinde bir saldırının ne kadar sürede tespit edilebildiği, sistemin pratikte uygulanabilirliği açısından kritik bir değerlendirme ölçütü olarak kabul edilmektedir.

Tablo 8'de sunulan veriler doğrultusunda, otonom araçların CAN bus ağ güvenliği için önerilen hibrit saldırı tespit sisteminin (HGB, DSA ve 1D-KSA) literatürdeki öncü çalışmalarla olan kıyaslamalı analizi gerçekleştirilmiştir. Yapılan analizler neticesinde literatürdeki mevcut araştırmaların büyük bir çoğunluğunun modelin sınıflandırma doğruluğunu maksimize etmeye odaklandığı ancak otonom sürüş sistemlerinin operasyonel güvenliği için kritik bir parametre olan gerçek zamanlı tepki süresi kriterini raporlamadığı gözlemlenmiştir.

Örneğin Gao vd. (2023), %99.2 ile literatürdeki en yüksek doğruluk oranlarından birine ulaşmış, Hossain vd. (2020) ve Reddy vd. (2021) ise,

sırasıyla %97.8 ve %98.6 gibi rekabetçi başarılar sergilemişlerdir. Ancak bu çalışmaların hiçbirinde sistemin saldırıyı tespit etme anındaki gecikme süresine dair sayısal bir veri sunulmamıştır. Buna karşın, bu tez çalışmasında elde edilen %97.62'lik doğruluk başarısı yaklaşık 10–15 ms aralığında değişen tespit süresiyle birlikte değerlendirildiğinde önerilen yöntemin otonom sistemlerdeki karar verme süreçlerini aksatmadan çalışabileceğini kanıtlamaktadır. Bu bulgu çalışmanın sadece teorik bir yüksek başarı sunmakla kalmayıp, aynı zamanda gerçek zamanlı kısıtlara sahip araç içi ağ ekosistemlerinde uygulanabilir ve proaktif bir güvenlik katmanı oluşturduğunu ortaya koymaktadır.

Tablo 8. *Literatürdeki çalışmalar ile önerilen yöntemin tepki süresi karşılaştırması*

Çalışma	**Yöntem**	**Uygulama Alanı**	**Raporlanan Tepki Süresi**	**F1 skoru**
Hossain vd. (2020)	KSA tabanlı IDS	CAN Bus	Belirtilmemiş	%97.8
Avatefipour vd. (2019)	Makine öğrenmesi tabanlı IDS	CAN Bus	Belirtilmemiş	%98.8
Amato vd. (2021)	Derin öğrenme tabanlı IDS	CAN Bus	Belirtilmemiş	%95.0
Reddy vd. (2021)	Hibrit derin öğrenme	Ağ tabanlı DDoS	Belirtilmemiş	%98.6
Sadaf vd. (2024)	Bulanık mantık tabanlı IDS	Otonom araç	Belirtilmemiş	%96.2
Gao vd. (2023)	Hafif KSA mimarisi	CAN Bus	Belirtilmemiş	%99.2
Ahmed ve Anisi (2025)	Yapay zekâ tabanlı IDS	Otonom araç	Belirtilmemiş	%97.0
Bu çalışma	**HGB / DSA / 1D-KSA**	**CAN Bus**	**≈ 10–15 ms**	%97.62

SONUÇ VE ÖNERİLER

Bu çalışma kapsamında, otonom araçların temel haberleşme omurgasını oluşturan CAN bus protokolünün DoS/DDoS saldırılarına karşı korunması amacıyla yapay zekâ tabanlı kapsamlı bir saldırı tespit mimarisi geliştirilmiştir. Çalışma süresince Extra Trees, Rastgele Orman,

HistGradientBoosting gibi geleneksel makine öğrenmesi yöntemleri ile DSA ve 1D-Konvolüsyonel Sinir Ağları (1D-KSA) gibi derin öğrenme mimarileri karşılaştırmalı performans analizine tabi tutulmuştur.

Elde edilen deneysel bulgular CAN bus verilerinin yapısal karakteristikleri ve dengesiz sınıf dağılımları göz önüne alındığında topluluk tabanlı öğrenme yöntemlerinin üstün bir genelleme başarımı sergilediğini kanıtlamıştır. Özellikle HistGradientBoosting modeli doğruluk F1-skoru ve hassasiyet metriklerinde en yüksek başarıyı elde ederek öne çıkmıştır. Modelin saldırı sınıfına yönelik yüksek duyarlılık performansı, yanlış negatif oranlarını minimize ederek kritik otonom sürüş fonksiyonlarının güvenliğini tehdit edebilecek tespit hatalarının önüne geçmektedir. Extra Trees ve RF modelleri ise düşük hesaplama maliyetleri ve veri gürültüsüne karşı gösterdikleri direnç ile gerçek zamanlı sistemler için güçlü birer alternatif olduklarını kanıtlamışlardır.

Derin öğrenme tarafında ise DSA ve 1D-KSA modelleri CAN trafiğindeki lineer olmayan karmaşık ilişkileri ve zamansal örüntüleri başarıyla modellemiştir. Ancak bu modellerin eğitim aşamasındaki donanım gereksinimleri ve yüksek hesaplama maliyetleri, kısıtlı kaynaklara sahip gömülü araç içi sistemlerde ek optimizasyon süreçlerini zorunlu kılmaktadır. Gerçek zamanlı uygulanabilirlik analizleri topluluk tabanlı modellerin milisaniye altı tahmin sürelerine sahip olduğunu; toplam saldırı tespit gecikmesinin ise 10–15 ms aralığında kaldığını göstermiştir. Bu sonuçlar, önerilen sistemin araç içi anlık karar mekanizmalarında herhangi bir darboğaz yaratmadan operasyonel sürüş güvenliğine katkı sağlayabileceğini doğrulamaktadır.

Gelecek çalışmalar kapsamında, mevcut savunma mekanizmasının farklı saldırı vektörlerini kapsayacak şekilde çok sınıflı senaryolarla genişletilmesi hedeflenmektedir. Ayrıca, geliştirilen modellerin gerçek dünya fiziksel katman koşullarında ve donanım kısıtlı gömülü sistemler üzerinde test edilmesi sistemin pratik uygulanabilirliğini pekiştirecektir. Son olarak transfer öğrenme yaklaşımlarının entegrasyonu ve saldırı tespit sisteminin araç içi diğer pasif güvenlik katmanlarıyla bütünleşik çalıştırılması, otonom araç siber güvenliğinde daha dayanıklı ve otonom bir koruma kalkanı oluşturulması açısından kritik bir araştırma alanı olarak değerlendirilmektedir.

KAYNAKÇA

Ahmed, S., & Hossein Anisi, M. (2025). AIDAS: AI-enhanced ıntrusion detection and authentication for autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, 26(8), 12548-12557. https://doi.org/10.1109/TITS.2025.3562919

Alfardus, A., & Rawat, D. B. (2021). Intrusion detection system for CAN Bus In-vehicle network based on machine learning algorithms. In 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). (pp. 0944-0949)

Alhaidari, F. A., & Alrehan, A. M. (2021). A simulation work for generating a novel dataset to detect distributed denial of service attacks on Vehicular Ad hoc NETwork systems. International Journal of Distributed Sensor Networks, 17(3), 155014772110002. https://doi.org/10.1177/15501477211000287

Ali Alheeti, K. M., Gruebler, A., & McDonald-Maier, K. D. (2015). An intrusion detection system against malicious attacks on the communication network of driverless cars. In 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). (pp. 916-921)

Ali, M. N., Imran, M., Bahoo, G., & Kim, B. S. (2024). DoS Attack detection enhancement in autonomous vehicle systems with explainable AI. In 2024 IEEE International Conference on Future Machine Learning and Data Science (FMLDS). (pp. 228-233)

Aljabri, W., Hamid, Md. A., & Mosli, R. (2025). Lightweight and adaptive data-driven ıntrusion detection system for autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, 26(2), 2282-2292. https://doi.org/10.1109/TITS.2024.3509459

Al-Khulaidi, N. A., Zahary, A. T., Al-Shargabi, A. A., & Hazaa, M. A. S. (2024). Machine learning for ıntrusion detection in vehicular ad-hoc networks (VANETs): a survey. In 2024 4th International Conference on Emerging Smart Technologies and Applications (eSmarTA). (pp. 1-10)

Alladi, T., Kohli, V., Chamola, V., Yu, F. R., & Guizani, M. (2021). Artificial intelligence (AI)-empowered intrusion detection architecture for the internet of vehicles. IEEE Wireless Communications, 28(3), 144-149. https://doi.org/10.1109/MWC.001.2000428

Alsaade, F. W., & Al-Adhaileh, M. H. (2023). Cyber attack detection for self-driving vehicle networks using deep autoencoder algorithms. Sensors, 23(8), 4086. https://doi.org/10.3390/s23084086

Altalbe, A. (2024). Enhanced Intrusion detection in ın-vehicle networks using advanced feature fusion and stacking-enriched learning. IEEE Access, 12, 2045-2056. https://doi.org/10.1109/ACCESS.2023.3347619

Amari, H., Louati, W., Khoukhi, L., & Belguith, L. H. (2021). Securing software-defined vehicular network architecture against DDoS attack. In 2021 IEEE 46th Conference on Local Computer Networks (LCN). (pp. 653-656)

Amato, F., Coppolino, L., Mercaldo, F., Moscato, F., Nardone, R., & Santone, A. (2021). CAN-Bus attack detection with deep learning. IEEE Transactions on Intelligent Transportation Systems, 22(8), 5081-5090. https://doi.org/10.1109/TITS.2020.3046974

Anand, M., Kumar, S. P., Selvi, M., Kumar Svn, S., Ram, G. D., & Kannan, A. (2023). Deep learning model based IDS for detecting cyber attacks in IoT based smart vehicle network. In 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS). (pp. 281-286)

Avatefipour, O., Al-Sumaiti, A. S., El-Sherbeeny, A. M., Awwad, E. M., Elmeligy, M. A., Mohamed, M. A., & Malik, H. (2019). An intelligent secured framework for cyberattack detection in electric vehicles' CAN Bus using machine learning. IEEE Access, 7, 127580-127592. https://doi.org/10.1109/ACCESS.2019.2937576

Badukale, P., Thorat, S., & Rojatkar, D. (2021). Sum up work on intrusion detection system in vehicular ad-hoc networks. In 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). (pp. 641-645)

Banafshehvaragh, S. T., & Rahmani, A. M. (2023). Intrusion, anomaly, and attack detection in smart vehicles. Microprocessors and Microsystems, 96, 104726. https://doi.org/10.1016/j.micpro.2022.104726

Butt, H. A., Fang, K., Shah, M. A., Ur Rehman, M., Jhaveri, R., Feng, H., & Ali, J. (2024). Deep learning empowered DDoS attack detection in software defined network. In 2024 IEEE Globecom Workshops (GC Wkshps). (pp. 1-6)

Delwar Hossain, M., Inoue, H., Ochiai, H., Fall, D., & Kadobayashi, Y. (2020). An effective in-vehicle CAN bus intrusion detection system using CNN deep learning approach. In GLOBECOM 2020 - 2020 IEEE Global Communications Conference. (pp. 1-6)

Gao, S., Zhang, L., He, L., Deng, X., Yin, H., & Zhang, H. (2023). Attack detection for intelligent vehicles via CAN- Bus: a lightweight ımage network approach. IEEE Transactions on Vehicular Technology, 72(12), 16624-16636. https://doi.org/10.1109/TVT.2023.3296705

Khadka, A., Karypidis, P., Lytos, A., & Efstathopoulos, G. (2021). A benchmarking framework for cyber-attacks on autonomous vehicles. Transportation Research Procedia, 52, 323-330. https://doi.org/10.1016/j.trpro.2021.01.038

Khatri, N., Lee, S., & Nam, S. Y. (2023). Transfer learning-based ıntrusion detection system for a controller area network. IEEE Access, 11, 120963-120982. https://doi.org/10.1109/ACCESS.2023.3328182

Khemissa, H., Bouchouia, M. L., & Gherbi, E. (2024). Poster: cyber resiliency metrics for AI-based in-vehicle ıntrusion detection systems. In 2024 IEEE Vehicular Networking Conference (VNC). (pp. 269-270)

Khokha, S. (2024). From standards to ımplementation: functional safety and cybersecurity in modern autonomous and electric vehicles. In 2024 International Conference on Cybernation and Computation (CYBERCOM). (pp. 52-56)

Lampe, B., & Meng, W. (2024). can-train-and-test: A curated CAN dataset for automotive intrusion detection. Computers & Security, 140, 103777. https://doi.org/10.1016/j.cose.2024.103777

Marchetti, M., & Stabili, D. (2017). Anomaly detection of CAN bus messages through analysis of ID sequences. In 2017 IEEE Intelligent Vehicles Symposium (IV). (pp. 1577-1583)

Mıjwıl, M. M., Sadıkoğlu, E., Cengiz, E., & Candan, H. (2022). Siber güvenlikte yapay zekanın rolü ve önemi: Bir derleme. Veri Bilimi, 5(2), 97-105.

Morić, Z., Matahlija, D., & Kapulica, A. (2025). Automotive hacking: security risks in connected and autonomous vehicles. In 2025 15th International Conference on Advanced Computer Information Technologies (ACIT). (pp. 584-588)

Niyomdi, M., & Oluoch, J. (2025). A Comprehensive security orchestration, automation, and response system (SOAR) for Connected and autonomous vehicles (CAVs). In 2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC). (pp. 1-14)

Prabha, R. S., Kaushik, S., Rahul, R., & Varun, R. (2024). Cognitive intrusion detection system in autonomous vehicles using machine learning. In 2024 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS). (pp. 1-7)

Rai, R., Grover, J., Sharma, P., & Pareek, A. (2025). Securing the CAN bus using deep learning for intrusion detection in vehicles. Scientific Reports, 15(1), 13820. https://doi.org/10.1038/s41598-025-98433-x

Rajapaksha, S., Kalutarage, H., Al-Kadri, M. O., Petrovski, A., Madzudzo, G., & Cheah, M. (2023). AI-based ıntrusion detection systems for in-vehicle networks: a survey. ACM Computing Surveys, 55(11), 1-40. https://doi.org/10.1145/3570954

Reddy, K. P., Kodati, S., Swetha, M., Parimala, M., & Velliangiri, S. (2021). A hybrid neural network architecture for early detection of DDOS attacks using deep learning models. In 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC). (pp. 323-327)

Sadaf, M., Iqbal, Z., Anwar, Z., Noor, U., Imran, M., & Gadekallu, T. R. (2024). A novel framework for detection and prevention of denial of service attacks on autonomous vehicles using fuzzy logic. Vehicular Communications, 46, 100741. https://doi.org/10.1016/j.vehcom.2024.100741

Seo, E., Song, H. M., & Kim, H. K. (2025). GIDS: GAN based ıntrusion detection system for in-vehicle network. In 2025 16th Annual Conference on Privacy, Security and Trust (PST). (pp. 1-6)

Taslimasa, H., Dadkhah, S., Carlos Pinto Neto, E., Xiong, P., Iqbal, S., Ray, S., & Ghorbani, A. A. (2023). ImageFed: practical privacy preserving ıntrusion detection system for in-vehicle CAN bus protocol. In 2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS). (pp. 122-129)

Turgut, K. (2025). Yapay zeka destekli otonom sistemler ve güvenlik sorunları. Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi, 9(1), 1-9. https://doi.org/10.33461/uybisbbd.1600576

Weaver, A., Von Jouanne, A., Sicker, D., & Yokochi, A. (2025). Cybersecurity dynamometer testbed: a review to advance vehicle-in-the-loop testing of traditional, connected and autonomous vehicles. IEEE Open Journal of Vehicular Technology, 6, 2925-2943. https://doi.org/10.1109/OJVT.2025.3623913

Yadav, A. K., Seema, Kumar, A., Anand, A., Rana, P. S., & Singh, M. (2025). Cyber attacks: the dark side of electric vehicles. In 2025 IEEE International Conference on Computer, Electronics, Electrical Engineering & Their Applications (IC2E3). (pp. 1-7)

IOT VERİLERİNİN TOPLANMASI, SAKLANMASI VE YÖNETİMİ

Oğuzhan Kilim[**], Cem Deniz Kumral[***]

GİRİŞ

Internet of Things (IoT) ekosistemlerinde fiziksel ortam ile sayısal sistem arasındaki temel bağ, sensörler ve bağlı cihazlar tarafından sürekli üretilen veridir. Sıcaklık, nem, basınç, titreşim, enerji tüketimi, konum, hız, kalp atım hızı veya makine durum bilgisi gibi ölçümler çoğunlukla zaman damgası ile üretilir ve bu nedenle doğal olarak zaman serisi niteliği taşır. IoT mimarilerinde sensörlerin kesintisiz veya yüksek frekanslı veri üretmesi, geleneksel statik veri kümelerinden farklı olarak verinin sıralı, bağımlı, akışkan ve zaman duyarlı biçimde ele alınmasını gerektirir. IoT'nin gelişimine ilişkin temel çalışmalarda da algılama, iletişim ve bulut/dağıtık hesaplama bileşenleri arasındaki veri akışının sistemin merkezinde bulunduğu vurgulanmaktadır (Gubbi vd., 2013; Zanella vd., 2014).

Bu bağlamda zaman serisi verilerinin doğru biçimde toplanması kadar, toplanan verinin kalitesinin korunması da kritik önemdedir. Sensör arızası, kalibrasyon kayması, ağ paket kaybı, gecikme, zaman senkronizasyonu bozukluğu, enerji kesintisi, yinelenen iletim veya çevresel etkenler; ölçümlerin eksik, gecikmiş, tutarsız ya da fiziksel olarak geçersiz hale gelmesine yol açabilmektedir. IoT veri kalitesi üzerine yapılan sistematik ve kapsamlı çalışmalar, veri kalitesinin yalnızca "doğruluk" ile sınırlandırılamayacağını; tamlık, zamanlılık, tutarlılık, geçerlilik ve bağlama uygunluk gibi birden fazla boyutla değerlendirilmesi gerektiğini göstermektedir (Karkouch vd., 2016; Liu vd., 2019; Wang & Strong, 1996; Zhang vd., 2021). Bu nedenle zaman serisi yönetimi ile veri kalitesi yönetimi birbirinden bağımsız iki işlem değil, aynı veri yaşam döngüsünün tamamlayıcı parçalarıdır.

Bu bölümde önce IoT bağlamında zaman serisi verilerinin yapısal özellikleri, örnekleme düzenleri, zaman damgası yönetimi ve temel ön

[**] Öğr. Gör. Dr., Isparta Uygulamalı Bilimleri Üniversitesi, Isparta Bilişim Teknolojileri Meslek Yüksekokulu, Veri Tabanı, Ağ Tasarımı ve Yönetimi Bölümü, Isparta, oguzhankilim@isparta.edu.tr, ORCID: https://orcid.org/0000-0003-3365-7327

[***] Öğr. Gör. Dr., Isparta Uygulamalı Bilimleri Üniversitesi, Uzaktan Eğitim Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Isparta, cemkumral@isparta.edu.tr, ORCID: https://orcid.org/0000-0002-1326-4537

işleme gereksinimleri açıklanmaktadır. Ardından veri kalitesinin IoT zaman serilerine özgü boyutları, bozulma kaynakları, kalite ölçütleri, eksik veri ve aykırı değerlerin yönetimi ele alınmaktadır. Devamında IoT verilerinin ilişkisel ve zaman serisi veritabanlarında saklanması, NoSQL veri modellerinin ölçeklenebilir IoT iş yüklerindeki rolü ve bulut tabanlı veri depolama yaklaşımları incelenmektedir. Son olarak, veri kalitesi güvence döngüsü ile depolama, yönetişim ve analitik süreçlerinin birlikte nasıl ele alınabileceği bütüncül bir çerçevede sunulmaktadır. IoT zaman serisi verisinin ölçümden analitiğe uzanan yaşam döngüsü Şekil 1'de verilmiştir.



Şekil 1. IoT zaman serisi verisinin ölçümden analitiğe uzanan yaşam döngüsü.

Şekil 1'de gösterildiği gibi veri kalitesi problemi yalnızca nihai depolama katmanında ortaya çıkmaz. Sensör düzeyinde yanlış ölçüm üretilebilir; ağ geçidinde tamponlama nedeniyle gecikme oluşabilir, mesajlaşma katmanında yinelenen paketler veya sıra bozulması görülebilir; depolama katmanında hatalı zaman damgaları ya da tip uyuşmazlıkları oluşabilir. Bu nedenle kalite denetimlerinin veri yaşam döngüsünün farklı katmanlarına dağıtılması gerekir.

IOT ORTAMLARINDA ZAMAN SERİSİ VERİLERİ

Bir zaman serisi, gözlemlerin zaman eksenine göre sıralandığı veri dizisi olarak tanımlanabilir. Tek değişkenli bir zaman serisi x_t biçiminde gösterilirken, birden fazla sensör veya özellik aynı anda ölçülüyorsa her zaman anındaki gözlem bir vektör olarak x_t = [x_(t,1), x_(t,2), …, x_(t,m)]

biçiminde ifade edilebilir. Zaman serilerini klasik tablosal veriden ayıran temel özellik, gözlemlerin bağımsız kabul edilememesidir. Bir t anındaki değer, çoğu uygulamada önceki zaman adımlarındaki değerlerle ilişkilidir. Zaman serisi veri madenciliği literatürü bu zamansal bağımlılığı; temsil, benzerlik ölçümü, örüntü bulma, sınıflandırma, tahmin ve anomali belirleme gibi işlemlerin temel belirleyicilerinden biri olarak ele almaktadır (Esling & Agon, 2012).

IoT sistemlerinde zaman serileri çok farklı ölçeklerde oluşabilir. Bir akıllı tarım sensörü toprağın nemini her 15 dakikada bir ölçerken, endüstriyel titreşim sensörü saniyede binlerce örnek üretebilir. Akıllı sayaçlar tüketim verisini dakikalık veya saatlik aralıklarla kaydedebilir; giyilebilir sağlık cihazları ise kalp ritmi, ivme veya oksijen doygunluğu gibi çok değişkenli ve yüksek frekanslı akışlar oluşturabilir. Dolayısıyla zaman serisinin karakteri yalnızca değerlerin türüne değil, örnekleme hızına, cihaz sayısına, veri üretim sürekliliğine ve uygulamanın gecikme toleransına da bağlıdır. Görüntü tabanlı izleme sistemlerinde de düzenli aralıklarla elde edilen gözlemler zamansal bir veri dizisi oluşturabilir. Örneğin Genç ve arkadaşları, soğuk depolama koşullarındaki karideslerin kalite değişimini değerlendirmek amacıyla görüntüleri 15 dakikalık aralıklarla toplamış ve ardışık görüntülerdeki zamansal değişimi CNN ve LSTM bileşenlerini birleştiren hibrit bir mimariyle modellemiştir. Bu tür uygulamalar, IoT ve akıllı izleme sistemlerinde zaman serisi kavramının yalnızca sayısal sensör ölçümleriyle sınırlı olmadığını; görüntü gibi yüksek boyutlu verilerin de zaman damgalı ve ardışık biçimde elde edilerek zamansal analize dahil edilebileceğini göstermektedir (Genç vd., 2025).Tablo 1'de IoT zaman serilerinin temel özellikleri ve yönetim etkileri sunulmuştur.

Tablo 1. IoT zaman serilerinin temel özellikleri ve yönetim etkileri.

Özellik	Açıklama	IoT örneği	Yönetim açısından etkisi
Zaman bağımlılığı	Ardışık gözlemler arasında ilişki bulunur.	Makine titreşimindeki ardışık örnekler	Sıra korunmalı; zaman tabanlı sorgu/analiz gerekir.
Örnekleme sıklığı	Ölçümlerin alınma periyodudur.	1 Hz sıcaklık, 10 kHz titreşim	Veri hacmini, tamponlamayı ve çözünürlüğü belirler.
Çok değişkenlilik	Birden fazla sensör aynı süreç için birlikte izlenir.	Sıcaklık-nem-CO_2	Senkronizasyon ve çapraz sensör tutarlılığı önem kazanır.

Düzensiz örnekleme	Ölçümler eşit aralıklarla gelmeyebilir.	Kablosuz sensörde paket kaybı	Yeniden örnekleme veya eksik veri işlemi gerekir.
Trend/mevsimsellik	Uzun dönem yönelim ve periyodik yapı görülebilir.	Günlük enerji tüketimi	Baz çizgi ve anomali eşikleri zamana göre değişebilir.
Akış niteliği	Veri çoğu zaman sürekli gelir.	Gerçek zamanlı üretim hattı	Çevrim içi kalite kontrol ve düşük gecikme gerekir.

Düzenli ve düzensiz örnekleme

İdeal bir sensör akışında ardışık iki gözlem arasındaki zaman farkı sabittir. Beklenen örnekleme aralığı Δt olmak üzere zaman damgaları t_i için $\Delta t_i = t_i - t_{(i-1)} \approx \Delta t$ koşulunun sağlanması beklenir. Ancak gerçek IoT sistemlerinde iletişim gecikmesi, ağ yoğunluğu, enerji tasarrufu modları, cihaz yeniden başlatmaları ve paket kaybı gibi nedenlerle Δt_i değişken hale gelebilir. Bu durum, düzensiz örnekleme olarak adlandırılır ve özellikle tahmin, spektral analiz, korelasyon hesabı veya makine öğrenmesi modelleri öncesinde dikkate alınmalıdır.

$$\Delta t_i = t_i - t_{i-1}$$

Düzensiz örnekleme, doğrudan "veri hatası" anlamına gelmez; bazı IoT uygulamalarında olay tetiklemeli ölçüm bilinçli bir tasarım tercihidir. Örneğin bir titreşim sensörü yalnızca belirli bir eşik aşıldığında daha yüksek frekanslı veri üretmeye başlayabilir. Bununla birlikte analiz algoritması sabit aralıklı veri bekliyorsa, zaman ekseni yeniden örneklenmeli ve oluşan boşluklar uygun yöntemlerle ele alınmalıdır. Bu nedenle yeniden örnekleme sırasında ortalama, medyan, minimum, maksimum veya son değer gibi toplulaştırma operatörlerinin seçimi uygulamaya göre yapılmalıdır.

Olay zamanı, varış zamanı ve zaman damgası bütünlüğü

IoT verilerinde tek bir "zaman" kavramı yerine en az iki zamanın ayrıştırılması yararlıdır: olay zamanı (event time), fiziksel ölçümün gerçekten gerçekleştiği anı; varış veya işleme zamanı (arrival/processing time) ise kaydın ağ geçidine, sunucuya ya da analiz motoruna ulaştığı anı gösterir. Bu iki zaman arasındaki fark gecikmeyi ifade eder. Özellikle gerçek zamanlı alarm sistemlerinde verinin doğru ölçülmüş olması tek başına yeterli değildir; karar verilebilecek sürede sisteme ulaşması gerekir.

$$\text{Gecikme}_i = t_{i,\text{varış}} - t_{i,\text{olay}}$$

Cihaz saatlerinin birbirinden sapması, saat dilimi dönüşümleri veya yaz/kış saati uygulamaları da zaman damgası kalitesini bozabilir. Bu nedenle IoT sistemlerinde zamanın mümkün olduğunca ortak bir referansa göre tutulması, zaman dilimi bilgisinin açık biçimde saklanması ve cihaz saatlerinin NTP/PTP benzeri senkronizasyon mekanizmalarıyla denetlenmesi gerekir. Aynı fiziksel olayı ölçen çoklu sensörlerin birleştirilmesi sırasında milisaniye düzeyindeki sapmalar bile yüksek frekanslı uygulamalarda yanlış korelasyon veya yanlış olay sıralaması üretebilir.

Trend, mevsimsellik, durağanlık ve kavram kayması

IoT zaman serileri çoğunlukla trend, mevsimsellik, çevrimsel davranış ve gürültünün bileşiminden oluşur. Bina enerji tüketimi günün saatine ve haftanın gününe göre tekrar eden örüntüler sergilerken, tarımsal sensörlerde mevsimsel sıcaklık değişimleri baskın olabilir. Bu yapıların fark edilmemesi, normal bir mevsimsel tepenin aykırı değer olarak etiketlenmesine veya gerçek bir arızanın olağan değişim olarak değerlendirilmesine neden olabilir. Bu nedenle kalite kontrolünde tek bir sabit eşik yerine bağlama duyarlı ve mümkünse zamanla güncellenen eşikler tercih edilmelidir.

Durağanlık, zaman serisinin ortalama, varyans ve bağımlılık yapısının zaman içinde belirli ölçüde kararlı olmasıyla ilişkilidir. IoT sistemlerinde cihaz eskimesi, sensör değişimi, üretim reçetesi değişikliği veya çevresel koşullar veri dağılımını değiştirebilir. Bu olgu kavram/veri kayması ile ilişkilidir ve yalnızca tahmin modellerini değil, veri kalitesi kurallarını da etkiler. Örneğin yeni kalibre edilen bir sensörün doğal çalışma aralığı değişmişse eski kalite eşikleri yanlış alarm üretebilir. Bu nedenle IoT zaman serisi yönetiminde kalite profillerinin periyodik olarak yeniden değerlendirilmesi gerekir.

Zaman serisi verisinin ön işlenmesi

Zaman serisi ön işleme; ham verinin analize uygun, zamansal olarak tutarlı ve karşılaştırılabilir hale getirilmesini kapsar. Başlıca işlemler zaman damgası ayrıştırma ve standartlaştırma, yinelenen kayıt kontrolü, sıralama, yeniden örnekleme, ölçekleme, eksik değer işlemleri, gürültü azaltma ve aykırı değer incelemesidir. Ancak ön işleme sırasında ham verinin izlenebilirliği korunmalıdır. Bir değerin silinmesi, düzeltilmesi veya imputasyonla tahmin edilmesi durumunda bu işlemin veri kökeni

(provenance) bilgisiyle birlikte kaydedilmesi bilimsel tekrarlanabilirlik ve denetlenebilirlik açısından önemlidir.

Zaman serilerine özgü veri yönetim sistemleri; zaman damgasını birincil erişim ekseni olarak ele almaları, sıkıştırma, sütun tabanlı saklama, zaman aralığı sorguları ve yüksek hızlı ekleme operasyonları gibi nedenlerle IoT iş yüklerinde avantaj sağlayabilmektedir. Apache IoTDB gibi sistemler milyonlarca zaman serisinin ve çok büyük ölçekli sensör verisinin yönetimi için geliştirilmiştir (WangChen vd., 2023). Edge ortamında yapılan karşılaştırmalar da zaman serisi iş yüklerinde veritabanı seçiminin ekleme ve sorgu performansını anlamlı biçimde etkileyebildiğini göstermektedir (Grzesik & Mrozek, 2020). Bununla birlikte hangi depolama teknolojisi kullanılırsa kullanılsın, hatalı zaman damgası veya düşük kaliteli ölçüm depolandığında sonraki analitik adımların güvenilirliği azalacaktır.

IOT ZAMAN SERİLERİNDE VERİ KALİTESİ

Veri kalitesi, verinin belirli bir kullanım amacı için uygunluğunu ifade eden çok boyutlu bir kavramdır. Klasik veri kalitesi çalışmalarında kullanıcı açısından kalite; doğruluk yanında tamlık, temsil tutarlılığı, zamanlılık ve erişilebilirlik gibi boyutlarla ele alınmıştır (Wang & Strong, 1996). IoT'de bu kavram daha da önemlidir; çünkü veri otomatik olarak, farklı üreticilerin cihazlarından, heterojen ağ koşullarında ve çoğu zaman insan doğrulaması olmadan üretilmektedir. Karkouch ve arkadaşları IoT'ye özgü veri kalitesi sorunlarının sensör, iletişim, çevre ve veri işleme zincirinin farklı noktalarında ortaya çıkabildiğini göstermiştir (Karkouch vd., 2016). Liu ve arkadaşlarının sistematik incelemesi de IoT veri kalitesi çalışmalarında kalite boyutları, problem belirtileri ve ölçüm yöntemleri arasında güçlü bir ilişki bulunduğunu ortaya koymaktadır (Liu vd., 2019).

Bu nedenle kaliteli IoT verisi yalnızca fiziksel gerçeğe yakın bir sensör değeri değildir. Ölçümün beklenen zamanda üretilmesi, eksiksiz biçimde iletilmesi, tekil olması, birim ve veri tipi açısından geçerli olması, ilişkili sensörlerle çelişmemesi ve kullanım amacı için yeterince güncel olması da gerekir. Zhang ve arkadaşları veri kalitesi yönetiminin IoT veri yaşam döngüsünün bütününe yayılması gerektiğini vurgulamakta ve kalite yönetimini edinim, iletim, işleme ve kullanım aşamalarını kapsayan sürekli bir süreç olarak ele almaktadır (Zhang vd., 2021). Tablo 2'de IoT zaman serilerinde başlıca veri kalitesi boyutları gösterilmektedir.

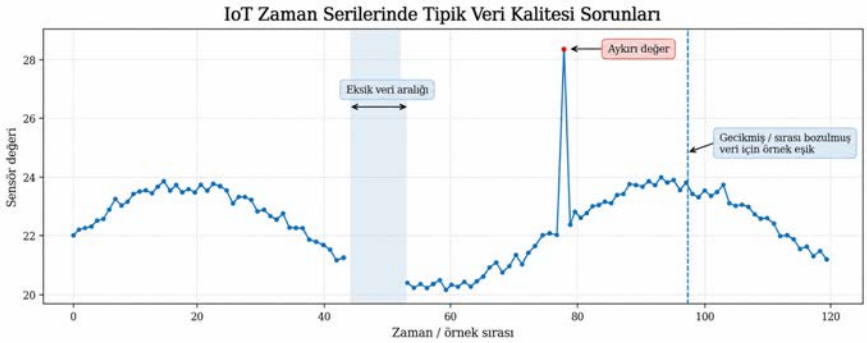
Tablo 2. IoT zaman serilerinde başlıca veri kalitesi boyutları.

Kalite boyutu	IoT zaman serisi açısından tanım	Örnek problem	Basit ölçüm yaklaşımı
Doğruluk (accuracy)	Ölçümün gerçek/fiziksel değere yakınlığı	Kalibrasyonu bozulmuş sıcaklık sensörü	Referans sensöre göre hata, MAE/RMSE
Tamlık (completeness)	Beklenen gözlemlerin mevcut olma oranı	Paket kaybı nedeniyle 5 dakikalık boşluk	N_obs / N_exp
Zamanlılık (timeliness)	Verinin ihtiyaç duyulan süre içinde ulaşması	Alarm verisinin 30 s gecikmesi	Zamanında gelen kayıt oranı
Tutarlılık (consistency)	Kayıtların tekrar, sıra ve çapraz kaynak açısından çelişmemesi	Aynı zaman damgalı yinelenen paket	Tekrarsız/uyumlu kayıt oranı
Geçerlilik (validity)	Değerin tip, aralık ve fizik kurallarına uyması	Nem=%135 veya negatif basınç	Kuralları geçen kayıt oranı
Güncellik (currency)	Verinin karar anında hâlâ temsil gücüne sahip olması	Eski konum verisiyle rota kararı	Veri yaşı / maksimum kabul edilebilir yaş

Kalite sorunlarının kaynakları

IoT zaman serilerindeki bozulmalar dört ana kaynaktan düşünülebilir. Birincisi sensör kaynaklı sorunlardır: kalibrasyon hatası, doygunluk, sürüklenme (drift), gürültü, batarya zayıflaması veya fiziksel arıza. İkincisi iletişim kaynaklı sorunlardır: paket kaybı, gecikme, tekrarlı iletim, bağlantı kopması ve sıra bozulması. Üçüncüsü zaman yönetimi sorunlarıdır: cihaz saatinin geri/ileri kayması, yanlış saat dilimi, senkronizasyon eksikliği veya zaman damgasının ölçüm yerine iletim anında üretilmesi. Dördüncüsü ise veri işleme ve entegrasyon hatalarıdır: farklı birimler, yanlış veri tipi, dönüşüm hataları, şema değişikliği veya sensör kimliğinin yanlış eşlenmesi.

Şekil 2, tek bir sensör serisi üzerinde sık karşılaşılan üç kalite problemini kavramsal olarak göstermektedir. Eksik bir zaman aralığı tamlık sorununa, aşırı sapmış bir değer doğruluk/geçerlilik sorununa, zaman eksenine geç eklenen kayıtlar ise zamanlılık ve kimi durumda tutarlılık sorununa dönüşebilir. TsQuality çalışması da zaman serisi kalite değerlendirmesini tamlık, tutarlılık, zamanlılık ve geçerlilik boyutları altında sistemleştirerek bu problemlerin milyonlarca seri üzerinde ölçülebileceğini göstermektedir (Qiu vd., 2023). IoT zaman serilerinde eksik veri, aykırı değer ve zamanlama sorunlarının temsili görünümü Şekil 2'de sunulmuştur. Tablo 3'te ise bu sorunların üstesinden gelmek için kullanılan yaklaşımlar verilmiştir.



Şekil 2. IoT zaman serilerinde eksik veri, aykırı değer ve zamanlama sorunlarının temsili görünümü.

Tablo 3. IoT zaman serilerindeki yaygın kalite sorunları ve müdahale yaklaşımları.

Sorun	Olası neden	Kalite boyutu	Önerilen kontrol
Eksik gözlem	Paket kaybı, enerji kesintisi, sensör arızası	Tamlık	Beklenen frekansa göre boşluk analizi; gerekiyorsa imputasyon
Yinelenen kayıt	Yeniden iletim, mesaj kuyruğu tekrarı	Tutarlılık	Cihaz kimliği + olay zamanı + sıra numarası ile tekilleştirme
Gecikmiş kayıt	Ağ yoğunluğu, tamponlama, edge işleme gecikmesi	Zamanlılık	Olay-varış zamanı farkı için SLA/eşik
Aykırı değer	Sensör arızası, ani fiziksel olay, dönüşüm hatası	Doğruluk/geçerlilik	Fiziksel sınırlar + istatistiksel/ML tabanlı anomali analizi
Sensör sürüklenmesi	Yaşlanma, kalibrasyon kayması	Doğruluk	Referans sensör, periyodik kalibrasyon, trend tabanlı drift kontrolü
Zaman kayması	Saat senkronizasyonu bozukluğu	Tutarlılık/zamanlılık	NTP/PTP kontrolü; olay zamanı standardizasyonu
Birim/şema uyuşmazlığı	Heterojen cihaz/entegrasyon	Geçerlilik/tutarlılık	Şema doğrulama, birim sözlüğü ve metadata kontrolü

Veri kalitesinin nicel olarak ölçülmesi

Kaliteyi yönetebilmek için önce ölçülebilir hale getirmek gerekir. IoT zaman serilerinde basit ve yorumlanabilir oranlar, gerçek zamanlı kalite izleme için yararlıdır. Belirli bir zaman penceresinde beklenen gözlem sayısı

N_exp ve alınan geçerli gözlem sayısı N_obs ise tamlık oranı aşağıdaki gibi tanımlanabilir:

$$Q_tamlık = N_obs / N_exp$$

Benzer biçimde zamanlılık, belirlenen gecikme eşiği τ içinde ulaşan kayıtların oranı; tutarlılık, yinelenmeyen ve sıra/kimlik kurallarına uyan kayıtların oranı; geçerlilik ise tanımlı fiziksel ve sözdizimsel kısıtları sağlayan değerlerin oranı olarak hesaplanabilir. TsQuality yaklaşımı bu dört boyutu zaman serisi veri tabanı içinde hesaplanabilen kalite fonksiyonları olarak uygulamaktadır (Qiu vd., 2023). Kalite ölçümü tek bir global skorla da özetlenebilir; ancak farklı uygulamaların kalite öncelikleri farklı olduğundan ağırlıkların sabit kabul edilmesi her zaman doğru değildir.

$$Q_toplam = \Sigma(k=1..K)\ w_k\ q_k\ ,\quad \Sigma w_k = 1$$

Örneğin gerçek zamanlı sağlık alarmında zamanlılığın ağırlığı yüksekken, geçmişe dönük enerji tüketimi analizinde tamlık ve doğruluk daha yüksek önceliğe sahip olabilir. Byabazaire ve arkadaşları farklı IoT uygulamalarının veri kalitesi gereksinimlerini tek bir veri hattında karşılamak amacıyla kalite boyutlarının adaptif ağırlıklandırılması ve füzyonunu incelemiş; gerçek zamanlı ve tarihsel analitik ihtiyaçlarının aynı kalite boyutlarına farklı önem verebildiğini göstermiştir (Byabazaire vd., 2023). Bu yaklaşım, kalite skorunun uygulama bağlamına göre yorumlanması gerektiğini desteklemektedir.

Eksik verilerin belirlenmesi ve yönetimi

Zaman serilerinde eksik veri, yalnızca hücrede NULL/NaN değerinin bulunması değildir. Beklenen örnekleme planına göre hiç oluşmamış bir zaman damgası da eksik gözlem anlamına gelir. Bu nedenle IoT'de eksiklik kontrolü çoğunlukla zaman damgaları arasındaki farkın beklenen örnekleme aralığı ile karşılaştırılmasına dayanır. Bir cihazın 60 saniyede bir veri üretmesi beklenirken iki kayıt arasındaki fark 180 saniye ise arada iki gözlemin kaybolmuş olabileceği sonucuna varılır. Bu tür boşlukların uzunluğu ve ardışıklığı, basit paket kaybı ile cihaz arızasını ayırt etmek için de kullanılabilir.

Eksik değerlerin nasıl doldurulacağı, veri dinamiğine ve eksikliğin süresine bağlıdır. Kısa ve düzgün değişen serilerde doğrusal veya spline interpolasyonu yeterli olabilir. Son değerin taşınması (LOCF), yavaş değişen proseslerde kısa kesintiler için kullanılabilir ancak hızlı değişen sinyallerde

hatalı plato etkisi oluşturabilir. Mevsimsel veya güçlü otokorelasyonlu serilerde Kalman filtreleme, durum-uzayı modelleri veya zaman serisine özgü istatistiksel yöntemler daha uygundur. Moritz ve Bartz-Beielstein, zaman serisi eksik değerlerinin interpolasyon, hareketli ortalama, Kalman tabanlı ve mevsimsel yöntemler gibi farklı yaklaşımlarla ele alınabileceğini göstermiştir (Moritz vd., 2017). Junninen ve arkadaşlarının çevresel zaman serilerindeki karşılaştırması da tek bir imputasyon yönteminin her eksiklik deseninde üstün olmadığını; performansın eksik veri yapısı ve serinin özelliklerine bağlı olduğunu ortaya koymuştur (Junninen vd., 2004).

İmputasyon yapılırken en önemli ilke, tahmin edilmiş değerin gerçek gözlem gibi değerlendirilmemesidir. İmpute edilen noktalara kalite bayrağı eklenmeli, kullanılan yöntem ve parametreler kaydedilmeli ve mümkünse ham veri korunmalıdır. Özellikle kritik karar sistemlerinde uzun kesintilerin agresif biçimde doldurulması sahte bir süreklilik yaratabilir. Böyle durumlarda eksiklik bilgisini modele açıkça taşımak veya ilgili pencereyi analiz dışında bırakmak daha güvenli olabilir. Tablo 4'te IoT zaman serilerinde eksik veri için başlıca imputasyon seçenekleri gösterilmektedir.

Tablo 4. IoT zaman serilerinde eksik veri için başlıca imputasyon seçenekleri.

Yöntem	Uygun olduğu durum	Avantaj	Başlıca risk
Doğrusal interpolasyon	Kısa boşluk, yumuşak değişim	Basit ve düşük maliyetli	Ani değişimleri düzleştirir
Spline interpolasyon	Düzgün fakat doğrusal olmayan seri	Daha yumuşak geçiş	Aşırı salınım üretebilir
LOCF/önceki değer	Yavaş değişen proses, çok kısa boşluk	Çevrim içi uygulaması kolay	Yapay plato ve gecikme oluşturur
Hareketli ortalama/medyan	Gürültülü lokal yapı	Gürültüye dayanıklı	Tepe ve çukurları bastırabilir
Kalman/durum uzayı	Dinamik modeli olan zaman serisi	Belirsizliği ve zamansal yapıyı kullanır	Model varsayımlarına duyarlıdır
ML/DL tabanlı	Çok değişkenli, karmaşık ve yeterli veri	Doğrusal olmayan ilişkileri öğrenebilir	Hesaplama maliyeti ve aşırı öğrenme riski

Aykırı değerler ve anomali yönetimi

Aykırı değer, serinin beklenen davranışından önemli ölçüde sapan gözlemdir; ancak her aykırı değer hatalı veri değildir. Endüstriyel sensörde ani titreşim artışı sensör hatasından kaynaklanabileceği gibi gerçek bir rulman arızasının da habercisi olabilir. Bu nedenle veri temizleme ile olay tespiti arasındaki sınır dikkatli kurulmalıdır. Zaman serisi anomali literatürü noktasal anomaliler, bağlamsal anomaliler ve ardışık/kollektif anomaliler arasında ayrım yapmaktadır (Blázquez-García vd., 2022). IoT özelindeki incelemeler ise sensör izleme, akıllı şehirler, güvenlik ve altyapı uygulamalarında anomali tespitinin önemli bir kalite ve güvenilirlik bileşeni olduğunu göstermektedir (Chatterjee & Ahmed, 2022).

Basit yöntemler fiziksel min–maks sınırları, z-skoru, çeyrekler açıklığı veya hareketli medyan temelli Hampel filtresi gibi kurallara dayanabilir. Bununla birlikte mevsimsellik veya güçlü trend içeren IoT verilerinde sabit eşikler çok sayıda yanlış pozitif üretebilir. Daha gelişmiş yaklaşımlar otoregresif modeller, yoğunluk tabanlı yöntemler, Isolation Forest, tek-sınıf modelleri, otomatik kodlayıcılar veya derin zaman serisi modellerini kullanabilir. Blázquez-García ve arkadaşlarının kapsamlı incelemesi, zaman serisi anomali tespitinde yöntemin veri boyutu, çevrim içi/çevrim dışı çalışma biçimi, pencereleme, bağlamsallık ve etiket bulunabilirliğine göre seçilmesi gerektiğini göstermektedir (Blázquez-García vd., 2022).

Gerçek zamanlı IoT akışlarında ayrıca anomali tespitinin ne kadar erken yapıldığı önemlidir. Ahmad ve arkadaşları akış verisinde çevrim içi ve etiketsiz anomali tespitinin, tüm veri kümesinin önceden mevcut olmadığı koşullarda değerlendirilmesi gerektiğini vurgulamış ve gerçek zamanlı anomali tespiti için bir karşılaştırma yaklaşımı sunmuştur (Ahmad vd., 2017). Bu yaklaşım IoT kalite yönetimi açısından önemlidir; çünkü düşük kaliteli veya şüpheli veri, nihai depolama ve karar katmanına ulaşmadan önce işaretlenebilir.

Yinelenen kayıtlar, sıra bozukluğu ve zamansal tutarlılık

Mesaj tabanlı IoT mimarilerinde “en az bir kez teslim” gibi güvenilirlik politikaları bazı mesajların birden fazla kez gelmesine neden olabilir. Aynı cihaz kimliği, olay zamanı ve ölçüm değeriyle gelen kayıtların yinelenmesi; sayım, ortalama, toplam enerji veya olay frekansı gibi hesapları bozabilir. Bu nedenle tekilleştirme için yalnızca değer karşılaştırması değil, cihaz kimliği,

zaman damgası, mesaj/sıra numarası ve mümkünse benzersiz olay kimliği birlikte kullanılmalıdır.

Sıra bozukluğu ise özellikle kablosuz ağlarda farklı gecikmelere sahip paketların varış sırasının olay sırasından farklı olmasıyla oluşur. Akış analitiği platformlarında belirli bir gecikme toleransı tanımlanarak geç gelen verinin ilgili zaman penceresine eklenmesine izin verilebilir. Fakat kabul penceresi sınırsız tutulamaz; aksi halde gerçek zamanlı sonuçlar sürekli geriye dönük değişir. Bu nedenle veri kalitesi politikası "ne kadar gecikmiş veri hâlâ kullanılabilir?" sorusuna uygulama bazında cevap vermelidir.

Doğruluk, sensör kalibrasyonu ve çapraz sensör kontrolü

Doğruluk doğrudan gerçek değerin bilinmesini gerektirdiğinden IoT ortamlarında ölçülmesi en zor kalite boyutlarından biridir. Referans cihazın bulunduğu laboratuvar veya kalibrasyon ortamında ölçüm hatası doğrudan hesaplanabilir. Saha koşullarında ise komşu sensörler, fiziksel model, geçmiş davranış veya tahmin modeli referans olarak kullanılabilir. Aynı çevresel olguyu ölçen yakın sensörler arasında yüksek korelasyon bekleniyorsa tek bir sensördeki kalıcı sapma kalibrasyon kaymasına işaret edebilir.

Burada temel risk, gerçek çevresel değişimin yanlışlıkla sensör hatası olarak düzeltilmesidir. Bu nedenle otomatik düzeltme öncesinde sensörün konumu, birimi, cihaz modeli, kalibrasyon tarihi, ölçüm aralığı ve doğruluk sınıfı gibi metadata dikkate alınmalıdır. Veri kalitesi yalnızca sayıların değil, sayıların anlamını belirleyen metadata'nın da kalitesine bağlıdır.

IOT VERİLERİNİN VERİTABANLARINDA SAKLANMASI

IoT sistemlerinde veri saklama katmanı, yalnızca ölçümlerin kalıcı olarak tutulduğu pasif bir bileşen değildir. Sensörlerden yüksek hızla gelen kayıtların yazılması, belirli zaman aralıklarının sorgulanması, cihaz metadata'sının yönetilmesi, geçmiş verinin arşivlenmesi ve analitik iş yüklerinin beslenmesi aynı veri altyapısı üzerinde farklı gereksinimler oluşturur. IoT veri yönetimi literatürü; veri hacmi, veri üretim hızı, heterojenlik, dağıtıklık, erişim gecikmesi ve yaşam döngüsü politikalarının depolama tasarımını birlikte belirlediğini göstermektedir (Abu-Elkheir vd., 2013). Bu nedenle uygun veritabanı seçimi, yalnızca veri modeline değil iş yükünün yazma-okuma oranına, tutarlılık gereksinimine ve beklenen ölçeklenme biçimine göre yapılmalıdır.

IoT iş yüklerinin depolama gereksinimleri

Klasik kurumsal uygulamalarda veritabanı iş yükü çoğunlukla kullanıcı kaynaklı işlem kayıtlarından oluşurken, IoT'de çok sayıda cihaz küçük fakat sürekli kayıtlar üretir. Bu durum yüksek ekleme hızı, ardışık zaman damgalı yazmalar, kısa zaman pencerelerinde yoğun sorgular ve uzun dönemli arşiv gereksinimini aynı anda ortaya çıkarır. Ayrıca cihaz modeli, konum, birim, kalibrasyon tarihi veya kullanıcı bilgisi gibi görece durağan metadata ile yüksek hacimli telemetri verisinin erişim örüntüleri birbirinden farklıdır. Bu ayrım, tek bir veri tabanı motorunun her yük için en uygun çözüm olmayabileceğini gösterir (Abu-Elkheir vd., 2013; Hemmati vd., 2026).

Depolama tasarımında ilk adım, veriyi işlevsel sınıflara ayırmaktır. Cihaz kayıtları ve yetkilendirme bilgileri işlemsel bütünlük gerektirebilir; sürekli sensör ölçümleri zaman damgasına göre hızlı ekleme ve zaman aralığı sorgularına ihtiyaç duyar; ham dosyalar, görüntüler veya büyük ikili nesneler ise nesne depolama katmanında daha ekonomik tutulabilir. Bu yaklaşım, aynı IoT uygulamasında ilişkisel veritabanı, zaman serisi veritabanı, NoSQL veritabanı ve bulut nesne depolamanın birlikte kullanılabildiği poliglot veri kalıcılığı (polyglot persistence) yaklaşımını doğurur.

İlişkisel veritabanlarının IoT'deki rolü

İlişkisel veritabanları veriyi tablolar, satırlar, sütunlar, anahtarlar ve ilişkiler aracılığıyla düzenler. SQL tabanlı sorgulama, olgun indeksleme mekanizmaları ve ACID (atomicity, consistency, isolation, durability) işlem özellikleri; cihaz envanteri, kullanıcı hesapları, abonelikler, konfigürasyon kayıtları ve iş kuralları gibi yapılandırılmış IoT verilerinde önemli avantaj sağlar. Bir sensörün hangi lokasyona, kullanıcıya veya bakım planına bağlı olduğunun izlenmesi gibi ilişkisel bütünlük gerektiren işlemler için bu model güçlüdür.

Bununla birlikte çok yüksek frekansta üretilen telemetri verisinin tek bir merkezi ilişkisel şemada saklanması; indeks büyümesi, tablo parçalama, yüksek yazma maliyeti ve yatay ölçekleme gereksinimleri nedeniyle zorlaşabilir. Bu nedenle ilişkisel veritabanı kullanıldığında zaman temelli bölümleme, toplu yazma, uygun bileşik indeksler ve eski verinin daha ekonomik katmanlara taşınması önem kazanır. Bu durum ilişkisel sistemlerin IoT için uygun olmadığı anlamına gelmez; doğru yaklaşım, ilişkisel modeli işlemsel ve güçlü tutarlılık gerektiren veri için, diğer veri modellerini ise yüksek hacimli akışlar için konumlandırmaktır (Cattell, 2010).

Zaman serisi veritabanları ve saklama politikaları

Zaman serisi veritabanları, zaman damgasını temel erişim ekseni kabul ederek yüksek hızlı ekleme, zaman aralığı sorguları, pencere tabanlı toplulaştırma, sıkıştırma ve saklama süresi (retention) yönetimi gibi işlevleri öne çıkarır. Bu bölümün önceki kısmında ele alınan Apache IoTDB gibi sistemler, çok sayıda sensör serisini ölçekli biçimde yönetmek üzere tasarlanmıştır (WangChen vd., 2023). Edge ortamındaki deneysel çalışmalar da zaman serisi iş yüklerinde veritabanı mimarisinin ekleme ve sorgu performansını doğrudan etkileyebildiğini göstermektedir (Grzesik & Mrozek, 2020).

IoT zaman serilerinde tüm verinin sonsuza kadar aynı ayrıntı düzeyinde tutulması çoğu zaman gereksizdir. Örneğin son yedi günün saniyelik verisi çevrim içi analiz için saklanırken, daha eski dönemler dakikalık veya saatlik özetlere dönüştürülebilir. Bu yaklaşım downsampling olarak adlandırılır. Benzer biçimde retention policy ile ham verinin belirli bir süre sonunda otomatik silinmesi veya arşive taşınması sağlanabilir. Bu politikalar belirlenirken bilimsel tekrar üretilebilirlik, hukuki saklama gereksinimleri ve anomali araştırmalarında geçmiş ham veriye duyulan ihtiyaç birlikte değerlendirilmelidir. IoT iş yükleri için başlıca veri saklama yaklaşımlarının karşılaştırılması Tablo 5'te gösterilmektedir.

Tablo 5. IoT iş yükleri için başlıca veri saklama yaklaşımlarının karşılaştırılması.

Ölçüt	İlişkisel DB	Zaman Serisi DB	NoSQL	Bulut nesne depolama
Başlıca veri	Yapılandırılmış işlem ve metadata	Zaman damgalı telemetri	Esnek/dağıtık veri	Ham dosya, arşiv, büyük nesne
Güçlü yön	ACID, SQL, ilişkiler	Yüksek yazma, zaman sorgusu, sıkıştırma	Yatay ölçek, esnek şema	Yüksek dayanıklılık, düşük arşiv maliyeti
Tipik erişim	JOIN, işlem, noktasal sorgu	Zaman aralığı, agregasyon	Anahtar/belge/partition tabanlı	Nesne bazlı toplu erişim
IoT örneği	Cihaz envanteri, kullanıcı/konfigürasyon	Sensör ölçümleri	Telemetri, cihaz durumu, olay kayıtları	Ham veri gölü, görüntü, uzun dönem arşiv
Dikkat edilmesi gereken	Yazma ölçeği ve indeks maliyeti	Kardinalite, retention, partition	Tutarlılık ve veri modelleme	Gecikme, egress ve yaşam döngüsü
Ölçüt	İlişkisel DB	Zaman Serisi DB	NoSQL	Bulut nesne depolama

Tablo 5'te görüldüğü üzere teknoloji seçimi bir 'en iyi veritabanı' arayışından çok, veri türü ve erişim örüntüsü eşleştirmesidir. Operasyonel olarak kritik metadata güçlü tutarlılık gerektirirken, yüksek hacimli telemetri daha farklı ölçekleme ve saklama stratejilerine ihtiyaç duyabilir.

NOSQL VERİTABANLARI VE IOT

NoSQL terimi, ilişkisel modelin yerine tek bir veri modeli önermekten çok, dağıtık ve yüksek ölçekli iş yükleri için farklı veri modellerini kapsayan bir şemsiye kavramdır. Cattell'in karşılaştırmalı çalışmasında ölçeklenebilir SQL ve NoSQL sistemlerinin veri modeli, tutarlılık, depolama, dayanıklılık, kullanılabilirlik ve sorgu desteği açısından farklı ödünleşimler sunduğu gösterilmiştir (Cattell, 2010). IoT açısından NoSQL'in temel çekiciliği; yatay ölçeklenebilme, şema esnekliği ve yüksek yazma hacminin çok sayıda düğüme dağıtılabilmesidir.

Bu özellikler özellikle cihaz firmware'i veya sensör tipleri zaman içinde değiştiğinde önemlidir. Aynı veri koleksiyonundaki tüm kayıtların aynı sütunları taşıması gerekmeyen belge tabanlı modeller, farklı cihaz sürümlerinden gelen JSON benzeri kayıtların saklanmasını kolaylaştırabilir. Benzer biçimde geniş sütunlu sistemler, cihaz ve zaman temelli bölümleme ile çok büyük telemetri tablolarını dağıtık biçimde yönetebilir. Buna karşılık karmaşık çoklu JOIN sorguları veya çok tablolulu güçlü işlemler gerekli olduğunda ilişkisel model daha doğal olabilir.

Başlıca NoSQL veri modelleri

NoSQL veritabanları pratikte anahtar-değer, belge, geniş sütun ve grafik olmak üzere dört temel veri modeli altında incelenebilir. Büyük ölçekli dağıtık veri altyapılarının gelişiminde Bigtable'ın sütun-aileli modeli (Chang vd., 2008), Dynamo'nun yüksek kullanılabilir anahtar-değer yaklaşımı (DeCandiaGiuseppe vd., 2007) ve Cassandra'nın merkezi olmayan dağıtık mimarisi (Lakshman & Malik, 2010) önemli referans noktalarıdır. Bu sistemler IoT için doğrudan tek bir reçete sunmaz; ancak yüksek ölçekli veri saklamada partition, replikasyon ve erişim örüntüsüne göre veri modelleme ilkelerini açık biçimde ortaya koyar. Tablo 6'da NoSQL veri modelleri ve IoT kullanım senaryoları birlikte gösterilmiştir.

Tablo 6. NoSQL veri modelleri ve IoT kullanım senaryoları.

NoSQL modeli	Temel yapı	IoT'de uygun kullanım	Avantaj	Sınırlılık
Anahtar-değer	key → value	Son cihaz durumu, oturum, önbellek	Çok hızlı noktasal erişim	Zengin sorgular sınırlı
Belge	JSON/BSON benzeri belge	Heterojen cihaz telemetrisi, olay kayıtları	Esnek şema, doğal nesne yapısı	Çapraz belge ilişkileri karmaşıklaşabilir
Geniş sütun	Partition key + sütun aileleri	Büyük hacimli sensör serileri	Dağıtık yazma ve yatay ölçek	Sorgu örüntüsü önceden tasarlanmalı
Grafik	Düğüm + kenar + özellik	Cihaz-topoloji-bağımlılık ilişkileri	İlişki yoğun sorgularda güçlü	Yüksek frekanslı ham telemetri için ilk tercih değildir

Tablo 6, veri modelinin uygulama semantiğine göre seçilmesi gerektiğini göstermektedir. Örneğin bir akıllı fabrika uygulamasında makine ölçümleri geniş sütunlu veya zaman serisi veritabanında tutulurken, makineler arasındaki üretim bağımlılıkları grafik veritabanında; cihaz konfigürasyonları ise ilişkisel ya da belge tabanlı sistemde saklanabilir.

Bölümleme, replikasyon ve tutarlılık

Dağıtık NoSQL sistemlerinde ölçeklenebilirliğin temel mekanizmalarından biri verinin partition veya shard adı verilen parçalara ayrılmasıdır. IoT zaman serileri için yaygın yaklaşım, cihaz kimliği ile zaman penceresini birlikte kullanan bir bölümleme anahtarı tasarlamaktır. Yalnızca zaman damgasına göre bölümleme, güncel verinin tek düğümde yoğunlaşmasına ve 'hot partition' problemine yol açabilir; yalnızca cihaz kimliğine göre bölümleme ise çok uzun ömürlü cihazlarda bölüm boyutlarının aşırı büyümesine neden olabilir. Bu nedenle device_id + time_bucket benzeri bileşik tasarımlar dengeli dağılım sağlayabilir.

Replikasyon, aynı verinin birden fazla düğümde tutulmasıyla hata toleransını ve kullanılabilirliği artırır. Ancak dağıtık sistemlerde ağ bölünmeleri sırasında tutarlılık ve kullanılabilirlik arasında uygulamaya özgü bir denge kurulması gerekir. CAP yaklaşımının sonraki yorumlarında da tasarımın basitçe 'üç özellikten ikisini seçmek' olarak görülmemesi; partition koşullarının açık biçimde yönetilmesi gerektiği vurgulanmıştır (Brewer, 2012). IoT'de alarm veya komut durumu gibi kritik kayıtlar güçlü tutarlılık

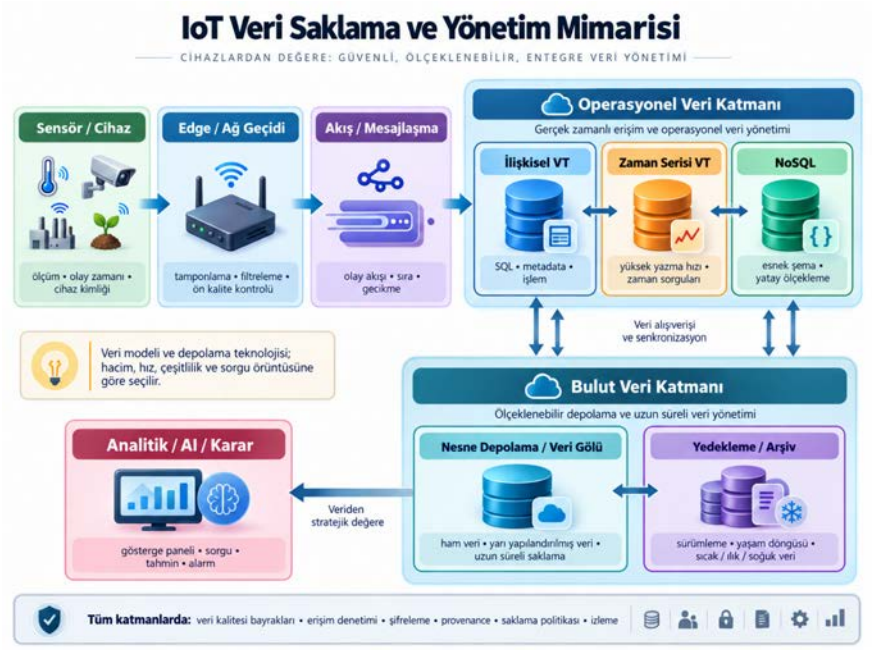
isterken, çevresel telemetri bazı senaryolarda kısa süreli eventual consistency ile çalışabilir.

Veritabanı seçiminin gerçek iş yüküyle test edilmesi de önemlidir. YCSB, bulut veri sunma sistemlerini farklı okuma, yazma ve güncelleme oranları altında karşılaştırmak için geliştirilmiş yaygın bir kıyaslama çerçevesidir (Cooper vd., 2010). IoT için benzer yaklaşım benimsenerek beklenen cihaz sayısı, saniyedeki yazma miktarı, sorgu penceresi, replikasyon faktörü ve ağ gecikmesi kullanılarak performans testleri yapılmalıdır.

BULUT VERİ DEPOLAMA VE IOT

Bulut bilişim, hesaplama ve depolama kaynaklarının ihtiyaç oldukça sağlanabildiği, kaynakların elastik biçimde ölçeklenebildiği bir hizmet modelidir. Bulut paradigmasının temel avantajları arasında talebe göre kaynak sağlama, ölçek ekonomisi ve altyapı yönetim yükünün azaltılması bulunmaktadır (Armbrust vd., 2010). IoT sistemleri açısından bu model, cihaz sayısı ve veri hacmi zaman içinde öngörülemeyen biçimde arttığında özellikle değerlidir.

IoT ile bulutun bütünleşmesi; çok sayıdaki cihazdan gelen verinin merkezi veya bölgesel veri merkezlerinde toplanmasını, uzun süre saklanmasını, büyük ölçekli analitik ve makine öğrenmesi süreçlerinin çalıştırılmasını kolaylaştırır. Botta ve arkadaşları bulut-IoT bütünleşmesinin ölçeklenebilir depolama ve hesaplama sağlarken ağ gecikmesi, güvenlik, heterojenlik ve hizmet kalitesi gibi yeni zorlukları da beraberinde getirdiğini göstermiştir (Botta vd., 2016). Daha güncel çalışmalar ise IoT-fog-edge-cloud sürekliliğinin tek bir merkezi mimariden ziyade katmanlı veri yönetimi yaklaşımı olarak ele alınması gerektiğini vurgulamaktadır (Hemmati vd., 2026; Kong vd., 2023). Şekil 3'te IoT veri saklama ve yönetimi mimarisi sunulmuştur.



Şekil 3. IoT verilerinin operasyonel veritabanları ile bulut depolama katmanlarında bütünleşik yönetimi.

Şekil 3'te gösterilen bütünleşik mimaride sensör verisi önce edge/ağ geçidi ve mesajlaşma katmanlarından geçerek operasyonel veri depolarına ulaşır. İlişkisel veritabanı işlemsel metadata için, zaman serisi veritabanı sık zaman tabanlı sorgular için, NoSQL ise esnek ve dağıtık veri modelleri için kullanılabilir. Ham ve uzun dönemli veriler bulut nesne depolama veya veri gölü katmanına taşınırken, kalite bayrakları, erişim politikaları ve provenance bilgileri tüm katmanlarda korunmalıdır.

Sıcak, ılık ve soğuk veri katmanları

Bulut veri depolamada temel maliyet-optimizasyon ilkelerinden biri, tüm veriyi aynı erişim performansına sahip katmanda tutmamaktır. Sık sorgulanan güncel veri 'sıcak' katmanda; daha seyrek kullanılan fakat makul erişim süresi gerektiren veri 'ılık' katmanda; denetim, yasal saklama veya uzun dönemli bilimsel analiz için korunan veri ise 'soğuk/arşiv' katmanında tutulabilir. Bu katmanlar arasında otomatik yaşam döngüsü politikaları uygulanması, büyük IoT kurulumlarında depolama maliyetini önemli ölçüde azaltabilir. Bulut ortamında IoT verisi için sıcak-ılık-soğuk saklama katmanları Tablo 7'de verilmiştir.

Tablo 7. Bulut ortamında IoT verisi için sıcak-ılık-soğuk saklama katmanları.

Katman	Erişim sıklığı	Tipik IoT içeriği	Öncelik	Önerilen işlem
Sıcak veri	Sürekli / çok sık	Son saatler-günler telemetrisi, alarm verisi	Düşük gecikme	Hızlı DB/TSDB, indeks, kısa retention
Ilık veri	Periyodik	Haftalık-aylık geçmiş, raporlama	Maliyet + sorgulanabilirlik	Özetleme, sütunlu dosya/analitik depo
Soğuk veri	Nadiren	Ham arşiv, yasal kayıt, model yeniden eğitimi	Düşük maliyet ve dayanıklılık	Nesne arşivi, sıkıştırma, uzun saklama

Tablo 7'deki katmanlama yalnızca maliyet konusu değildir. Sıcak katmanda veri daha hızlı sorgulanırken, uzun dönemli ham verinin arşivde korunması model yeniden eğitimi, hata kök-neden analizi veya geriye dönük bilimsel doğrulama için değer taşır. Bu nedenle özetleme yapıldığında ham verinin ne kadar süre korunacağı kullanım amacına göre açıkça tanımlanmalıdır.

Edge-bulut sürekliliği

Tüm IoT verisinin anında buluta gönderilmesi her zaman verimli değildir. Yüksek frekanslı sensörlerde bant genişliği maliyeti, bağlantı kesintileri ve gecikme gereksinimleri nedeniyle ön işleme edge katmanında yapılabilir. Edge computing odaklı IoT incelemeleri, düşük gecikme gerektiren işlevlerin veri kaynağına yakın çalıştırılmasının; bulutun ise daha büyük depolama kapasitesi ve küresel analitik için kullanılmasının uygun bir iş bölümü oluşturduğunu göstermektedir (Kong vd., 2023). Güncel IoT-fog-cloud literatürü de veri yönetiminin bu katmanlar arasında bütünleşik olarak tasarlanmasını önermektedir (Hemmati vd., 2026).

Pratik bir örnekte edge düğümü saniyede binlerce titreşim örneğini alabilir, temel filtreleme ve özellik çıkarımı yapabilir, kritik anomali durumunu hemen yerel olarak üretebilir; buluta ise özet özellikleri, kalite skorlarını ve gerektiğinde ham olay penceresini gönderebilir. Böylece hem ağ trafiği azaltılır hem de kritik kararlar bağlantı gecikmesinden bağımsızlaştırılır.

Güvenlik, yönetişim ve yaşam döngüsü

Bulutta saklanan IoT verisi için güvenlik ve yönetişim, depolama teknolojisinin ayrılmaz parçasıdır. İletim sırasında ve depoda şifreleme, en az ayrıcalık ilkesine dayalı kimlik ve erişim yönetimi, anahtar yönetimi, denetim günlükleri, yedekleme, sürümleme ve felaket kurtarma politikaları birlikte uygulanmalıdır. Ayrıca hangi cihaz verisinin hangi coğrafi bölgede tutulacağı, ne kadar süre saklanacağı ve kimler tarafından erişilebileceği veri egemenliği ve mevzuat gereksinimleri açısından açıkça tanımlanmalıdır.

Veri kalitesi açısından da bulut depolama sürecinde provenance kaybolmamalıdır. Bir ölçümün ham mı, filtrelenmiş mi, impute edilmiş mi veya gecikmeli mi olduğu depolama katmanında izlenebilir olmalıdır. Böylece daha sonra yapılan analizlerde yalnızca sensör değeri değil, değerin kalite ve işlem geçmişi de sorgulanabilir. Bu bağlantı, veri kalitesi yönetimi ile veritabanı ve bulut veri yönetimini tek bir yaşam döngüsünde birleştirir.

SÜREKLİ VERİ KALİTESİ GÜVENCE YAKLAŞIMI

IoT sistemlerinde veri kalitesi tek seferlik "temizleme" işlemi olarak tasarlanmamalıdır. Sensörlerin ve ağ koşullarının zaman içinde değişmesi nedeniyle kalite sürekli izlenmeli ve kalite metrikleri de birer zaman serisi olarak saklanmalıdır. Örneğin bir sensörün günlük tamlık oranının %99'dan %93'e düşmesi henüz tamamen arızalanmamış bir cihazdaki iletişim veya enerji probleminin erken göstergesi olabilir. Benzer şekilde ortalama gecikmenin kademeli artışı ağ kapasitesi sorununun habercisi olabilir.

Şekil 4'te önerilen döngüsel yaklaşım beş aşamadan oluşur: edinim, doğrulama, kalite ölçümü, iyileştirme ve kullanım/izleme. Edinim aşamasında cihaz kimliği, olay zamanı ve birim gibi temel metadata eksiksiz üretilir. Doğrulama aşamasında veri tipi, aralık ve sıra kuralları kontrol edilir. Kalite ölçümünde tamlık, zamanlılık, tutarlılık ve geçerlilik gibi göstergeler hesaplanır. İyileştirme aşaması imputasyon, kalibrasyon, tekilleştirme veya yeniden senkronizasyon gibi işlemleri kapsar. Son aşamada analitik sonuçlar ve kalite eğilimleri izlenerek sensör ve kalite politikalarına geri besleme sağlanır. Şekil 4'te veri kalitesi güvence döngüsü sunulmuştur.



Şekil 4. IoT zaman serileri için sürekli veri kalitesi güvence döngüsü.

Kalite güvencesinin uygulamada etkili olabilmesi için her kayıt veya zaman penceresi için kalite bayraklarının tutulması önerilir. Örneğin raw, validated, imputed, late, duplicate, outlier_candidate ve calibrated gibi durum etiketleri verinin hangi işlemlerden geçtiğini gösterir. Böylece kullanıcı yalnızca düzeltilmiş değeri değil, değerin güvenilirlik geçmişini de görebilir. Bu yaklaşım özellikle bilimsel araştırma, sağlık, endüstriyel güvenlik ve regülasyona tabi uygulamalarda izlenebilirliği artırır.

Ayrıca kalite eşikleri uygulamanın hizmet seviyesi hedefleriyle ilişkilendirilmelidir. Örneğin bir çevresel izleme sistemi için saatlik veride %95 tamlık kabul edilebilirken, yoğun bakım alarm sisteminde saniyelik akışta aynı oran yetersiz olabilir. Benzer biçimde 10 saniyelik gecikme akıllı tarım uygulamasında önemsizken endüstriyel güvenlik kapatmasında kabul edilemez olabilir. Dolayısıyla kalite metrikleri bağlamdan bağımsız mutlak sayılar olarak değil, kullanım amacına göre tanımlanmış sınırlar içinde değerlendirilmelidir.

IOT UYGULAMALARI İÇİN ÖNERİLEN UYGULAMA İLKELERİ

Zaman serisi verisi ve veri kalitesi birlikte ele alındığında IoT veri yönetimi için bazı temel ilkeler ortaya çıkmaktadır. İlk olarak olay zamanı ile işleme zamanı ayrıştırılmalı ve zaman damgasının kaynağı açık biçimde tanımlanmalıdır. İkinci olarak sensör kimliği, birim, örnekleme periyodu, kalibrasyon bilgisi ve fiziksel ölçüm aralığı veriyle birlikte yönetilmelidir. Üçüncü olarak kalite kontrolleri yalnızca veri gölüne veya veritabanına

yazma sonrasında değil, edge/ağ geçidi katmanından başlayarak mümkün olduğunca erken uygulanmalıdır.

Dördüncü olarak, eksik veya şüpheli değer üzerinde yapılan her otomatik işlem izlenebilir olmalı; ham ölçüm silinmemelidir. Beşinci olarak kalite metrikleri periyodik raporlar yerine gerçek zamanlı veya yakın gerçek zamanlı izlenmeli ve sensör bazında kalite profilleri oluşturulmalıdır. Altıncı olarak, tek bir kalite eşiğinin tüm IoT uygulamalarına uygulanması yerine kullanım amacı, risk ve gecikme toleransına göre kalite politikaları tanımlanmalıdır. Son olarak, anomali tespitinde "anormal = hatalı" varsayımından kaçınılmalı; fiziksel olay ile veri bozulmasını ayırt etmek için alan bilgisi ve çoklu sensör bağlamı kullanılmalıdır.

SONUÇ

IoT sistemlerinde veri çoğunlukla zaman içinde sürekli oluştuğu için zaman serisi yapısı, veri toplama ve yönetiminin merkezinde yer alır. Zaman bağımlılığı, farklı örnekleme hızları, düzensiz gözlemler, olay-varış zamanı ayrımı, trend, mevsimsellik ve kavram kayması; IoT verisinin klasik statik veri kümelerinden farklı işlenmesini gerektirir. Bu özellikler aynı zamanda kullanılacak veritabanının yazma hızı, bölümleme biçimi, zaman tabanlı sorgulama ve saklama politikalarını da doğrudan etkiler.

IoT veri yönetiminde tek bir depolama teknolojisinin tüm gereksinimleri karşılaması beklenmemelidir. İlişkisel veritabanları işlemsel bütünlük ve yapılandırılmış metadata için; zaman serisi veritabanları yoğun zaman damgalı telemetri için; NoSQL sistemleri esnek şema ve yatay ölçeklenebilirlik için; bulut nesne depolama ise ham veri, uzun dönemli arşiv ve veri gölü iş yükleri için güçlü seçenekler sunar. Edge-bulut sürekliliği, düşük gecikmeli işlemleri veri kaynağına yaklaştırırken büyük ölçekli saklama ve analitiği bulutta sürdürebilir.

Bu altyapının güvenilir sonuç üretmesi ise veri kalitesinin doğruluk, tamlık, zamanlılık, tutarlılık ve geçerlilik boyutlarıyla sürekli izlenmesine bağlıdır. Eksik gözlemler, yinelenen paketler, gecikmeler, aykırı değerler, sensör sürüklenmesi ve zaman senkronizasyonu hataları; yalnızca temizleme aşamasında değil ölçüm, iletim, depolama, arşiv ve analitik boyunca yönetilmelidir. Dolayısıyla etkili IoT veri yönetimi; zaman serisi işleme, uygun veri modeli, ölçeklenebilir depolama, bulut/edge mimarisi ve sürekli kalite güvencesinin birlikte tasarlandığı bütüncül bir yaşam döngüsü yaklaşımını gerektirir.

KAYNAKÇA

Abu-Elkheir, M., Hayajneh, M., & Ali, N. A. (2013). Data Management for the Internet of Things: Design Primitives and Solution. *Sensors, 13*(11), 15582-15612. https://doi.org/10.3390/S131115582

Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, *262*, 134-147. https://doi.org/10.1016/J.NEUCOM.2017.04.070

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58. https://doi.org/10.1145/1721654.1721672

Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2022). A Review on Outlier/Anomaly Detection in Time Series Data. *ACM Computing Surveys*, *54*(3). https://doi.org/10.1145/3444690

Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, *56*, 684-700. https://doi.org/10.1016/J.FUTURE.2015.09.021

Brewer, E. (2012). CAP twelve years later: How the "rules" have changed. *Computer*, *45*(2), 23-29. https://doi.org/10.1109/MC.2012.37

Byabazaire, J., O'Hare, G. M. P., Collier, R., & Delaney, D. (2023). IoT Data Quality Assessment Framework Using Adaptive Weighted Estimation Fusion. *Sensors, 23*(13), 5993. https://doi.org/10.3390/S23135993

Cattell, R. (2010). Scalable SQL and NoSQL data stores. *SIGMOD Record*, *39*(4), 12-27. https://doi.org/10.1145/1978915.1978919

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., & Gruber, R. E. (2008). Bigtable: A Distributed Storage System for Structured Data. *ACM Transactions on Computer Systems (TOCS)*, *26*(2). https://doi.org/10.1145/1365815.1365816

Chatterjee, A., & Ahmed, B. S. (2022). IoT anomaly detection methods and applications: A survey. *Internet of Things*, *19*, 100568. https://doi.org/10.1016/J.IOT.2022.100568

Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., & Sears, R. (2010). Benchmarking cloud serving systems with YCSB. *Proceedings of the 1st ACM Symposium on Cloud Computing, SoCC '10*, 143-154. https://doi.org/10.1145/1807128.1807152

DeCandiaGiuseppe, HastorunDeniz, JampaniMadan, KakulapatiGunavardhan, LakshmanAvinash, PilchinAlex, SivasubramanianSwaminathan, VosshallPeter, & VogelsWerner. (2007). Dynamo: Amazon's Highly Available Key-value Store. *ACM SIGOPS Operating Systems Review*, *41*(6), 205-220. https://doi.org/10.1145/1323293.1294281

Esling, P., & Agon, C. (2012). Time-series data mining. *ACM Computing Surveys*, *45*(1). https://doi.org/10.1145/2379776.2379788

Genç, İ. Y., Gürfidan, R., & Açikgözoğlu, E. (2025). Quality Determination of Frozen-Thawed Shrimp Using Machine Learning Algorithms Powered by Explainable Artificial Intelligence. *Food Analytical Methods, 18*(6), 935-945. https://doi.org/10.1007/S12161-025-02768-0

Grzesik, P., & Mrozek, D. (2020). Comparative analysis of time series databases in the context of edge computing for low power sensor networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12141 LNCS*, 371-383. https://doi.org/10.1007/978-3-030-50426-7_28

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645-1660. https://doi.org/10.1016/J.FUTURE.2013.01.010

Hemmati, A., Khaledian, N., & Rahmani, A. M. (2026). Toward a unified computing paradigm: a survey and roadmap for data management in integrated IoT, fog, and cloud services. *Peer-to-Peer Networking and Applications*, *19*(2), 54-. https://doi.org/10.1007/S12083-025-02183-6

Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J., & Kolehmainen, M. (2004). Methods for imputation of missing values in air quality data sets. *Atmospheric Environment*, *38*(18), 2895-2907. https://doi.org/10.1016/J.ATMOSENV.2004.02.026

Karkouch, A., Mousannif, H., Al Moatassime, H., & Noel, T. (2016). Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications*, *73*, 57-81. https://doi.org/10.1016/J.JNCA.2016.08.002

Kong, L., Tan, J., Huang, J., Chen, G., Wang, S., Jin, X., Zeng, P., Khan, M., & Das, S. K. (2023). Edge-computing-driven Internet of Things: A Survey. *ACM Computing Surveys*, *55*(8). https://doi.org/10.1145/3555308

Lakshman, A., & Malik, P. (2010). Cassandra - A decentralized structured storage system. *Operating Systems Review (ACM)*, *44*(2), 35-40. https://doi.org/10.1145/1773912.1773922

Liu, C., Nitschke, P., Williams, S. P., & Zowghi, D. (2019). Data quality and the Internet of Things. *Computing*, *102*(2), 573-599. https://doi.org/10.1007/S00607-019-00746-Z

Moritz, S., Bartz-Beielstein, T., Moritz, S., & Bartz-Beielstein, T. (2017). imputeTS: Time Series Missing Value Imputation in R. *The R Journal*, *453*. https://digitalcommons.unl.edu/r-journal/453

Qiu, Y., Fang, C., Song, S., Huang, X., Wang, C., & Wang, J. (2023). TsQuality: Measuring Time Series Data Quality in Apache IoTDB. *Proceedings of the VLDB Endowment*, *16*(12), 3982-3985. https://doi.org/10.14778/3611540.3611601

Wang, R. Y., & Strong, D. M. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems*, *12*(4), 5-34. https://doi.org/10.1080/07421222.1996.11518099

WangChen, QiaoJialin, HuangXiangdong, SongShaoxu, HouHaonan, JiangTian, RuiLei, WangJianmin, & SunJiaguang. (2023). Apache IoTDB: A Time Series Database for IoT Applications. *Proceedings of the ACM on Management of Data*, *1*(2), 1-27. https://doi.org/10.1145/3589775

Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, *1*(1), 22-32. https://doi.org/10.1109/JIOT.2014.2306328

Zhang, L., Jeong, D., Lee, S., Al-Masri, E., Chen, C.-H., Souri, A., & Kotevska, O. (2021). Data Quality Management in the Internet of Things. *Sensors, 21*(17), 5834. https://doi.org/10.3390/S21175834

BULUT DESTEKLİ NESNELERİN İNTERNETİ SİSTEMLERİ

Platformlar, Mikroservis Mimarileri, API Entegrasyonu ve Ölçeklenebilirlik

Enes Açıkgözoğlu[*], Serkan Keskin[**]

GİRİŞ

Nesnelerin İnterneti, sensörler, aktüatörler, gömülü işlemciler ve ağ arabirimleri ile donatılmış fiziksel nesnelerin veri üretebildiği, diğer sistemlerle haberleşebildiği ve elde edilen verilere bağlı olarak belirli işlevleri yerine getirebildiği geniş kapsamlı bir bilişim paradigmasıdır. IoT kavramı ilk dönem çalışmalarında fiziksel nesnelerin internet'e bağlanması ve tanımlanması üzerinde yoğunlaşmış olsa da günümüzde kavram; veri toplama, gerçek zamanlı karar verme, dağıtık hesaplama, yapay zekâ ve kurumsal uygulama entegrasyonunu kapsayan çok katmanlı bir ekosisteme dönüşmüştür (Atzori vd., 2010; Gubbi vd., 2013; Al-Fuqaha vd., 2015).

IoT sistemlerini geleneksel bilgi sistemlerinden ayıran başlıca özelliklerden biri, sistem içerisinde çok sayıda heterojen ve çoğu zaman kaynak bakımından kısıtlı cihazın bulunmasıdır. Bu cihazların işlem gücü, bellek kapasitesi, enerji kaynağı, haberleşme teknolojisi ve çalışma ortamı birbirinden önemli ölçüde farklı olabilmektedir. Buna ek olarak sensör verileri sürekli, dönemsel veya olay tabanlı olarak üretilebilmekte; veri hızı uygulama türüne bağlı olarak birkaç ölçümden yüksek frekanslı endüstriyel sinyallere ve görüntü akışlarına kadar değişebilmektedir. Bu heterojenlik; birlikte çalışabilirlik, cihaz yönetimi, güvenlik, veri yönetimi ve ölçeklenebilirlik gibi konuları IoT mimarisinin temel tasarım problemleri hâline getirmektedir (Sethi & Sarangi, 2017; Lin vd., 2017).

Bulut bilişim, IoT sistemlerinin ihtiyaç duyduğu hesaplama ve depolama kapasitesinin talep doğrultusunda sağlanmasını mümkün kıldığı için IoT ekosisteminin gelişiminde belirleyici bir rol üstlenmiştir. NIST bulut bilişim

[*] Dr. Öğr. Üyesi, Isparta Uygulamalı Bilimler Üniversitesi, Isparta Bilişim Teknolojileri Meslek Yüksekokulu, Veri Tabanı, Ağ Tasarımı Ve Yönetimi Bölümü, Isparta, enesacikgozoglu@isparta.edu.tr, ORCID: orcid.org/0000-0001-7293-883X

[**] Öğretim Görevlisi, Burdur Mehmet Akif Ersoy Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Veri Tabanı, Ağ Tasarımı Ve Yönetimi Bölümü, Burdur, skeskin@mehmetakif.edu.tr, ORCID: orcid.org/0000-0001-9404-5039

modelinde isteğe bağlı kendi kendine hizmet, geniş ağ erişimi, kaynak havuzlama, hızlı esneklik ve ölçülebilir hizmet olmak üzere beş temel özellik tanımlanmaktadır (Mell & Grance, 2011). Bu özellikler, cihaz sayısı ve telemetri trafiği zaman içerisinde değişebilen IoT uygulamalarının altyapı gereksinimleriyle doğrudan örtüşmektedir. Bulut kaynakları sayesinde cihazların sahip olmadığı uzun dönemli depolama, büyük ölçekli analitik ve yapay zekâ iş yükleri merkezi veya dağıtık veri merkezlerinde yürütülebilmektedir (Buyya vd., 2009; Botta vd., 2016).

Bununla birlikte IoT verisinin tamamının uzak bir bulut veri merkezine taşınması her uygulama için uygun değildir. Gecikmeye duyarlı endüstriyel kontrol, otonom sistemler, video analizi ve sağlık izleme gibi uygulamalarda kararların veri kaynağına daha yakın noktalarda verilmesi gerekebilmektedir. Benzer şekilde bağlantı süreksizliği, bant genişliği maliyeti ve veri yerelliği gereksinimleri de hesaplamanın ağın uç noktalarına yaklaştırılmasını teşvik etmektedir. Edge computing ve fog computing bu gereksinimlere cevap veren tamamlayıcı yaklaşımlar olarak gelişmiştir (Bonomi vd., 2012; Shi vd., 2016; Satyanarayanan, 2017). Güncel literatürde bu katmanların birbirinden bağımsız sistemler olarak değil, IoT uç cihazlarından merkezi buluta uzanan bir hesaplama sürekliliğinin parçaları olarak ele alınması öne çıkmaktadır (Gkonis vd., 2023; Al-Dulaimy vd., 2024; Maia vd., 2024).

IoT ile bulut bilişimin yakınsaması yazılım mimarisi açısından da önemli bir dönüşüm yaratmaktadır. IoT platformları cihaz yaşam döngüsü, telemetri toplama, kural işleme ve uygulama entegrasyonu gibi ortak işlevleri merkezileştirirken; mikroservis mimarileri bu işlevlerin bağımsız hizmetler hâlinde geliştirilmesine ve ölçeklenmesine imkân sağlamaktadır. REST tabanlı API'ler ile MQTT, CoAP ve olay veri yolları gibi mekanizmalar ise cihazlar, servisler ve uygulamalar arasındaki veri alışverişinin temelini oluşturmaktadır. Bu mimari dönüşüm ölçeklenebilirlik ve esneklik sağlarken servisler arası iletişim, dağıtık veri tutarlılığı, güvenlik ve gözlemlenebilirlik gibi yeni sorunları beraberinde getirmektedir (Di Francesco vd., 2019; Li vd., 2021; Siddiqui vd., 2023).

Bu çalışmanın amacı IoT platformları, bulut mimarileri, mikroservisler, API kullanımı ve ölçeklenebilirliği birbirinden kopuk kavramlar olarak değil, uçtan uca bir IoT sistem mimarisinin ilişkili bileşenleri olarak incelemektir. Bu doğrultuda önce IoT sistemlerinin mimari yapısı ve platform katmanı açıklanmakta, ardından bulut ve edge katmanlarının görev dağılımı ele alınmaktadır. Sonraki bölümlerde mikroservis mimarisi, API ve

mesajlaşma mekanizmaları, ölçeklenebilirlik teknikleri ve güvenlik incelenmekte; son bölümde ise bu bileşenleri bir araya getiren bütünleşik bir referans mimari ve güncel araştırma yönelimleri tartışılmaktadır.

IoT Sistemlerinin Mimari Yapısı

IoT sistemlerini açıklamak amacıyla literatürde üç, dört, beş veya daha fazla katmandan oluşan farklı mimari modeller önerilmiştir. Kullanılan katman sayısı değişse de temel süreç; fiziksel ortamdan verinin algılanması, ağ üzerinden taşınması, ara katmanlarda işlenmesi, platform servislerine aktarılması ve uygulamalar tarafından kullanılmasına dayanmaktadır. ISO/IEC 30141:2024 standardı, IoT sistemlerinin tanımlanması ve tasarlanması için güncel bir referans mimari ve ortak terminoloji sağlamaktadır (ISO/IEC, 2024). Bu bölümde pratik sistem tasarımını kolaylaştırmak amacıyla cihaz/algılama, iletişim, edge/gateway, platform/bulut ve uygulama katmanlarından oluşan işlevsel bir model benimsenmiştir.

Cihaz ve Algılama Katmanı

Cihaz ve algılama katmanı fiziksel dünya ile dijital sistem arasındaki ilk bağlantıyı kurmaktadır. Sıcaklık, nem, basınç, titreşim, ışık, hareket, konum, akım veya biyometrik ölçümler gerçekleştiren sensörler bu katmanda yer alırken röle, motor, vana ve benzeri aktüatörler sistemin fiziksel ortam üzerinde eylem gerçekleştirmesini sağlar. Cihazlar çoğu zaman sınırlı işlem gücü, bellek ve enerji kapasitesine sahiptir. Bu nedenle protokol seçimi, kriptografik mekanizmalar ve yerel veri işleme yöntemleri cihaz kaynakları dikkate alınarak tasarlanmalıdır.

IoT cihazları yalnızca ham sensör verisi üreten birimler olarak düşünülmemelidir. Güncel mikrodenetleyiciler ve gömülü platformlar temel filtreleme, veri sıkıştırma, eşik tabanlı karar verme ve hafif makine öğrenmesi çıkarımı gerçekleştirebilmektedir. Nitekim gömülü kontrol platformları üzerinde kamera ve sensör verilerinin işlenerek makine ve derin öğrenme yöntemleri aracılığıyla gerçek zamanlı kontrol kararlarının üretilebildiği uygulamalar bulunmaktadır (Aksoy vd., 2022). Bu durum buluta gönderilmesi gereken veri miktarını azaltabilir ve bağlantı kesintisi sırasında belirli işlevlerin yerel olarak sürdürülmesine katkı sağlayabilir.

İletişim Katmanı

İletişim katmanı cihazlardan elde edilen verilerin gateway, edge düğümü veya doğrudan IoT platformuna taşınmasını sağlar. Ethernet ve Wi-Fi gibi IP

tabanlı teknolojilerin yanında Bluetooth Low Energy, IEEE 802.15.4/Zigbee, LoRaWAN, NB-IoT ve hücresel haberleşme teknolojileri farklı uygulamalarda kullanılmaktadır. Haberleşme teknolojisinin seçimi menzil, veri hızı, enerji tüketimi, maliyet, topoloji ve çalışma ortamı gibi gereksinimlere bağlıdır. Uygulama katmanında ise HTTP, MQTT ve CoAP gibi protokoller öne çıkmaktadır (Al-Fuqaha vd., 2015).

Edge ve Gateway Katmanı

Gateway ve edge katmanı heterojen cihaz ağlarıyla üst seviye platformlar arasında ara katman görevi görmektedir. Gateway birimi protokol dönüştürme, veri filtreleme, veri birleştirme, güvenli tünelleme, yerel önbellekleme ve cihaz keşfi gibi işlevleri gerçekleştirebilir. Örneğin Modbus kullanan endüstriyel sensörlerden alınan kayıtlar gateway üzerinde normalize edilerek MQTT mesajlarına dönüştürülebilir.

Edge computing yaklaşımında gateway yalnızca protokol çeviren bir cihaz olmaktan çıkarak veri işleme ve karar verme yeteneği kazanır. Veri kaynağına yakın konumlandırılan edge düğümleri gecikmeyi azaltabilir, ağ trafiğini sınırlayabilir ve bulut bağlantısının geçici olarak kesildiği durumlarda yerel hizmet sürekliliği sağlayabilir (Shi vd., 2016; Satyanarayanan, 2017).

Platform ve Bulut Servisleri Katmanı

Platform ve bulut servisleri katmanı cihaz kimliklerinin yönetimi, telemetri alımı, mesaj yönlendirme, veri depolama, kural işleme, olay üretimi, analitik ve API sunumu gibi görevleri yerine getirir. IoT platformları bu katmanda ortak cihaz ve veri yönetimi işlevlerini sağlarken, genel amaçlı bulut servisleri uygulama sunucuları, veri tabanları, kapsayıcı kümeleri, mesaj kuyrukları ve yapay zekâ hizmetleri için altyapı sunar. Bu iki katman pratik sistemlerde çoğu zaman iç içe geçmektedir.

Uygulama Katmanı

Uygulama katmanı IoT verisini kullanıcı veya kurum açısından anlamlı hizmetlere dönüştürür. Web panoları, mobil uygulamalar, karar destek sistemleri, üretim yönetimi çözümleri, dijital ikiz uygulamaları, enerji izleme sistemleri ve harici kurumsal bilgi sistemleri bu katmanda yer alabilir. Uygulamaların platformla doğrudan veri tabanı üzerinden değil iyi tanımlanmış API'ler veya olay arayüzleri üzerinden haberleşmesi gevşek bağlılık, güvenlik ve sürdürülebilirlik açısından tercih edilmektedir.

IoT Platformları

Küçük bir prototipte sensörlerin doğrudan uygulama sunucusuna veri göndermesi yeterli olabilir. Ancak cihaz sayısı, veri hacmi ve entegrasyon gereksinimleri arttığında cihazların tek tek yönetilmesi, farklı protokollerin desteklenmesi ve uygulama kodunun cihaz ayrıntılarıyla iç içe geçmesi ciddi bir yönetilebilirlik problemi doğurur. IoT platformları cihazlar ile uygulamalar arasında ortak bir yönetim ve servis katmanı oluşturarak bu karmaşıklığı azaltmayı amaçlamaktadır.

Mineraud vd. (2016), IoT platformlarını cihazlar, hizmetler ve uygulamalar arasında ara katman sağlayan çözümler olarak incelemiş ve platformlar arasında cihaz yönetimi, veri kontrolü, API desteği ve ekosistem özellikleri bakımından önemli farklılıklar bulunduğunu göstermiştir. Domínguez-Bolaño vd. (2022) ise IoT mimarileri, protokoller ve açık kaynak projeleri üzerine yaptıkları kapsamlı incelemede bağlantı, veri yönetimi ve uygulama geliştirme işlevlerinin platform düzeyinde birlikte değerlendirilmesinin önemini ortaya koymuştur.

Güncel platform seçimi, yalnızca desteklenen protokoller veya fiyat üzerinden değerlendirilemeyecek kadar çok boyutlu bir karardır. Panagou vd. (2025), on iki IoT bulut platformunu güvenlik, ölçeklenebilirlik ve performans, birlikte çalışabilirlik, veri analitiği ve yapay zekâ entegrasyonu, edge desteği, maliyet, geliştirici araçları, standartlara uyum ve kablosuz yazılım güncelleme yetenekleri dâhil dokuz ölçüt üzerinden karşılaştırmıştır. Bu yaklaşım, platform seçiminin sistemin uzun dönemli sürdürülebilirliğini etkileyen stratejik bir mimari karar olduğunu göstermektedir.

Tablo 1. IoT platformlarının temel işlevleri

İşlev	Açıklama
Cihaz kaydı ve kimlik yönetimi	Cihazların benzersiz kimliklerle platforma kaydedilmesi, yetkilendirilmesi ve yaşam döngüsünün izlenmesi.
Bağlantı ve protokol desteği	MQTT, HTTP, CoAP veya ağ geçitleri üzerinden farklı cihaz türlerinin sisteme bağlanması.
Telemetri toplama	Sensör ölçümleri, durum bilgileri ve olayların güvenilir biçimde alınması.
Mesaj yönlendirme	Gelen verinin veri tabanı, kural motoru, akış işleme veya harici uygulamalara aktarılması.

Cihaz gölgesi / dijital ikiz	Fiziksel cihazın son bilinen veya hedef durumunun yazılım ortamında temsil edilmesi.
Kural ve olay işleme	Belirli telemetri veya olay koşullarında otomatik işlemlerin tetiklenmesi.
API ve entegrasyon	Web, mobil ve kurumsal sistemlerin cihaz/veri işlevlerine kontrollü erişimi.
Veri yönetimi ve analitik	Zaman serisi verilerinin saklanması, sorgulanması ve gerçek zamanlı/geçmiş analizi.
Güvenlik	Kimlik doğrulama, yetkilendirme, sertifika ve anahtar yönetimi, şifreli iletişim.
OTA güncelleme	Cihaz yazılımı veya yapılandırmasının uzaktan ve kontrollü biçimde güncellenmesi.

IoT platformlarının önemli mimari katkılarından biri cihaz soyutlamasıdır. Farklı üreticilere ait sensörler farklı protokollerle haberleşse de platform katmanı bu farklılıkları ortak veri modeli ve API'ler üzerinden uygulamalardan gizleyebilir. Böylece uygulama geliştiricilerinin her cihaz modelinin iletişim ayrıntılarını bilmesi gerekmez. Cihaz gölgesi veya dijital ikiz yaklaşımı da fiziksel cihazın çevrimdışı olduğu durumlarda son bilinen durumun uygulamalar tarafından erişilebilir kalmasına yardımcı olur.

Platformlar ayrıca kural motorları ve olay yönlendirme yetenekleri aracılığıyla uygulama mantığının belirli bölümlerini cihaz iletişiminden ayırabilir. Örneğin bir sıcaklık değerinin eşik üzerinde olması durumunda alarm oluşturulması, bildirim mikroservisinin çağrılması ve verinin ayrı bir analitik akışına yönlendirilmesi tek bir telemetri olayı üzerinden gerçekleştirilebilir.

Ticari ve açık kaynak platformlar farklı işletim modelleri sunmaktadır. AWS IoT Core ve Microsoft Azure IoT Hub yönetilen bulut hizmetleri aracılığıyla cihaz bağlantısı ve mesajlaşma işlevleri sağlarken; ThingsBoard gibi açık kaynak platformlar kurum içi veya bulut üzerinde kurulabilmektedir. Bu örnekler ürün karşılaştırmasından ziyade farklı sahiplik ve işletim modellerini göstermektedir (Amazon Web Services, 2026; Microsoft, 2026; ThingsBoard, 2026). Yönetilen hizmetler operasyon yükünü azaltabilirken açık kaynak çözümler veri egemenliği ve özelleştirme bakımından avantaj sağlayabilir.

Platform seçimi sırasında vendor lock-in riski de değerlendirilmelidir. Platforma özgü SDK, veri modeli ve API'lere yoğun bağımlılık, sistemin daha sonra başka bir sağlayıcıya taşınmasını güçleştirebilir. MQTT, HTTP, OAuth, standart sertifika altyapıları ve taşınabilir veri formatlarının kullanılması bu riski azaltır. Sonuç olarak platform seçimi; işlevsellik, güvenlik, ölçeklenebilirlik, birlikte çalışabilirlik, edge desteği, API olgunluğu, maliyet, veri taşınabilirliği ve standart uyumu birlikte değerlendirilerek yapılmalıdır.

IoT Bağlamında Bulut Bilişim Mimarileri

Bulut bilişim, bilgi işlem kaynaklarının paylaşımlı bir havuzdan gereksinim doğrultusunda sağlanmasına dayanan bir hizmet modelidir. NIST tanımı bulut bilişimi yalnızca uzaktaki sunucular olarak değil, hızlı biçimde tahsis edilebilen ve serbest bırakılabilen ağ, sunucu, depolama ve uygulama kaynaklarını kapsayan bir model olarak ele almaktadır (Mell & Grance, 2011). IoT açısından bulutun önemi, cihazların sınırlı kaynaklarını tamamlayan elastik hesaplama ve depolama kapasitesi sunmasından kaynaklanmaktadır.

Hizmet Modelleri

Altyapı Hizmeti (Infrastructure as a Service, IaaS), sanal makineler, ağlar ve depolama gibi temel kaynakların hizmet olarak sağlanmasını ifade eder. Bir kurum kendi MQTT broker'ını, veri tabanını ve mikroservislerini IaaS üzerinde çalıştırabilir; işletim sistemi ve uygulama katmanının yönetim sorumluluğu büyük ölçüde kuruma aittir.

Platform Hizmeti (Platform as a Service, PaaS), altyapı ayrıntılarını daha fazla soyutlayarak yönetilen veri tabanları, mesajlaşma servisleri, akış işleme ve uygulama çalışma ortamları sağlar. IoT uygulamalarında yönetilen IoT hub'ları, kuyruk servisleri ve sunucusuz fonksiyonlar PaaS yaklaşımıyla birlikte kullanılabilir.

Yazılım Hizmeti (Software as a Service, SaaS) ise son kullanıcıya hazır bir uygulama işlevinin sağlanmasıdır. Filo izleme, enerji yönetimi veya hazır IoT panoları gibi hizmetler SaaS örneği olarak değerlendirilebilir. IoT çözümünün farklı katmanlarında bu üç model aynı anda kullanılabilir.

Dağıtım Modelleri

NIST; public, private, community ve hybrid cloud dağıtım modellerini tanımlamaktadır. IoT uygulamalarında özellikle hibrit bulut modeli

önemlidir. Kritik veya mevzuata tabi veriler kurum içindeki özel altyapıda tutulurken büyük ölçekli analitik ve yedekleme iş yükleri genel bulut kaynaklarına aktarılabilir. Kurumsal IoT mimarileri üzerine güncel bir inceleme, işletmelerin legacy sistemlerle IoT servislerini ve edge/fog bileşenlerini birleştiren farklı mimari desenlere yöneldiğini göstermektedir (Li vd., 2025).

Elastiklik ve IoT İş Yükleri

Ölçeklenebilirlik ve elastiklik yakın kavramlar olmakla birlikte aynı değildir. Ölçeklenebilirlik, artan yükü daha fazla kaynak kullanarak karşılayabilme yeteneğini; elastiklik ise kaynakların yük değişimine göre dinamik biçimde artırılıp azaltılabilmesini ifade eder. IoT iş yüklerinde olay anında çok sayıda cihazın eş zamanlı veri göndermesi, mevsimsel kullanım veya üretim vardiyası değişimleri gibi nedenlerle trafik kısa sürede önemli ölçüde değişebilir.

Otomatik ölçekleme stratejileri CPU ve bellek kullanımının yanında yanıt süresi, kuyruk uzunluğu, istek hızı ve uygulama özelinde tanımlanan ölçümleri kullanabilir (Qu vd., 2018). IoT sistemlerinde aktif bağlantı sayısı, mesaj/saniye, tüketici gecikmesi ve telemetri kuyruğu uzunluğu çoğu zaman CPU kullanımından daha anlamlı ölçekleme sinyalleri sunar.

Bulut-Edge-IoT Bütünleşmesi

Merkezi bulut, yüksek hesaplama kapasitesi ve küresel görünürlük sağlasa da ağ gecikmesi, bant genişliği tüketimi, bağlantı bağımlılığı ve veri yerelliği bakımından sınırlamalar taşıyabilir. Edge computing hesaplama ve depolama kaynaklarını veri kaynağına yaklaştırarak bu sınırlamaların bir bölümünü azaltmayı amaçlar. Shi vd. (2016), edge computing yaklaşımını veri üretim noktalarına yakın kaynakların kullanılması üzerinden tanımlarken Satyanarayanan (2017), düşük gecikme ve bağlantı kesintilerine dayanıklılık gibi faydaları vurgulamaktadır.

Fog computing, bulut hizmetlerinin ağın uç noktalarına doğru genişletilmesini ifade eden daha geniş bir ara katman yaklaşımıdır. Bonomi vd. (2012), fog computing'in düşük gecikme, coğrafi dağılım ve çok sayıda düğüm gereksinimi bulunan IoT uygulamalarına uygunluğunu ortaya koymuştur. Günümüzde edge ve fog kavramları uygulama bağlamına göre farklı biçimlerde kullanılmakla birlikte temel fikir, tüm verinin ve hesaplamanın merkezi bulutta toplanmamasıdır.

Güncel çalışmalar cihaz, edge ve bulut kaynaklarının bir "hesaplama sürekliliği" içerisinde değerlendirilmesini önermektedir. Gkonis vd. (2023) IoT-edge-cloud continuum sistemlerinde görev yerleştirme, kaynak tahsisi, güvenilirlik ve ölçeklenebilirliği temel araştırma konuları arasında göstermektedir. Al-Dulaimy vd. (2024) ise farklı uygulama gereksinimlerinin tek bir katmanla karşılanamayacağını, hesaplama kaynaklarının uç cihazlardan merkezi buluta kadar bütünleşik biçimde yönetilmesi gerektiğini belirtmektedir.

Örneğin endüstriyel titreşim izleme uygulamasında sensör ham veriyi toplar, edge düğümü sinyal özelliklerini çıkarıp anomali ön tespiti yapar, bölgesel sunucu farklı makinelerden gelen verileri birleştirir ve merkezi bulut uzun dönemli model eğitimi ile kurumsal raporlamayı gerçekleştirir. Böyle bir dağılım hem ağ yükünü azaltabilir hem de kritik kararların düşük gecikmeyle alınmasını sağlayabilir.



Şekil 1. IoT-edge-bulut hesaplama sürekliliği ve iş yükü dağılımı.

IoT Sistemlerinde Mikroservis Mimarileri

Geleneksel monolitik yazılım mimarisinde sistemin büyük bölümü tek bir dağıtım birimi olarak geliştirilir ve çalıştırılır. Bu yapı küçük sistemlerde geliştirme ve işletim bakımından basit olabilir; ancak sistem büyüdükçe tek bir işlevdeki değişiklik bütün uygulamanın yeniden dağıtılmasını gerektirebilir ve yalnızca belirli bir bileşenin bağımsız olarak ölçeklenmesi güçleşebilir. Mikroservis mimarisi bu sorunu iş yeteneklerinin küçük, bağımsız dağıtılabilir hizmetlere ayrılmasıyla çözmeyi amaçlar.

Di Francesco vd. (2019) tarafından gerçekleştirilen sistematik haritalama çalışması, mikroservis mimarilerinin yazılım mimarisi alanında hızla yaygınlaştığını ve tasarım, dağıtım, test ve işletim süreçlerinin birlikte ele alınması gerektiğini göstermiştir. Li vd. (2021), mikroservis mimarilerinde ölçeklenebilirlik, performans, erişilebilirlik, güvenlik, test edilebilirlik ve gözlemlenebilirlik gibi kalite özelliklerinin kritik olduğunu ortaya koymaktadır.

IoT sistemleri mikroservis yaklaşımı için doğal bir uygulama alanıdır; çünkü cihaz yönetimi, telemetri toplama, kimlik doğrulama, kural yürütme, bildirim, analitik ve raporlama gibi işlevler farklı değişim ve yük profillerine sahiptir. Siddiqui vd. (2023), mikroservis tabanlı IoT mimarilerinin özellikle güvenilirlik ve erişilebilirlik açısından güçlü potansiyel taşıdığını, ancak servis bağımlılıkları ve dağıtık hata yönetimi konularının önemli araştırma sorunları olduğunu belirtmektedir.

Tablo 2. Monolitik ve mikroservis tabanlı IoT mimarilerinin karşılaştırılması

Özellik	Monolitik Mimari	Mikroservis Mimarisi
Dağıtım	Uygulama büyük ölçüde tek birim olarak dağıtılır.	Servisler bağımsız dağıtılabilir.
Ölçekleme	Çoğu durumda tüm uygulama ölçeklenir.	Yük altındaki servis bağımsız ölçeklenebilir.
Hata alanı	Bir hata daha geniş uygulama alanını etkileyebilir.	Doğru tasarımda servis bazında izolasyon mümkündür.
Teknoloji seçimi	Daha homojen teknoloji yığını yaygındır.	Servis bazında farklı teknoloji/veri deposu seçilebilir.
Geliştirme	Küçük ekip ve sistemlerde daha basittir.	Büyük sistemlerde ekip bağımsızlığı sağlayabilir.
İletişim	Süreç içi çağrılar daha fazladır.	Ağ üzerinden servisler arası çağrı ve olay trafiği artar.
Gözlemlenebilirlik	Görece daha basit günlükleme ve izleme.	Dağıtık izleme, merkezi günlük ve korelasyon gerekir.
Operasyon	Dağıtım topolojisi daha basit olabilir.	Kapsayıcı, orkestrasyon ve servis keşfi ihtiyacı artabilir.

Servis Sınırlarının Belirlenmesi

Mikroservis tasarımında amaç mümkün olan en fazla sayıda servis oluşturmak değildir. Hizmet sınırları değişim sıklığı, veri sahipliği, ölçekleme gereksinimi ve iş sorumluluğu dikkate alınarak belirlenmelidir. Bir IoT platformunda Cihaz Kayıt Servisi, Kimlik ve Yetkilendirme Servisi, Telemetri Alım Servisi, Kural Motoru, Dijital İkiz Servisi, Bildirim Servisi ve Analitik Servisi ayrı iş yetenekleri olarak ele alınabilir. Buna karşılık

yalnızca birkaç satır iş mantığı içeren işlevlerin gereksiz yere ayrı servis yapılması ağ trafiğini ve operasyonel karmaşıklığı artırabilir.

Senkron ve Asenkron İletişim

Mikroservisler REST veya gRPC gibi request-response mekanizmalarıyla senkron biçimde haberleşebileceği gibi MQTT, AMQP veya olay akış platformları üzerinden asenkron iletişim de kurabilir. Senkron iletişim anlık cevap gerektiren kullanıcı işlemleri için uygunken telemetri, alarm ve olay iş yüklerinde asenkron mesajlaşma servisler arasındaki zamansal bağımlılığı azaltır.

Olay güdümlü bir IoT mimarisinde cihaz tarafından yayımlanan aynı telemetri olayı depolama, anomali tespiti, alarm ve raporlama servisleri tarafından bağımsız biçimde tüketilebilir. Üretici cihazın hangi uygulamaların veriyi kullanacağını bilmesine gerek kalmaz. Bu gevşek bağlılık, yeni tüketicilerin sisteme mevcut üretici kodunu değiştirmeden eklenmesini kolaylaştırır.

Dağıtım ve Operasyonel Karmaşıklık

Mikroservislerin bağımsızlığı, servis keşfi, yapılandırma yönetimi, sır yönetimi, sağlık kontrolleri, merkezi günlükleme, dağıtık izleme, devre kesici ve yeniden deneme politikaları gibi yeni operasyonel ihtiyaçlar doğurur. El Akhdar vd. (2024), mikroservislerin IoT güvenliği açısından fırsatlar sunduğunu ancak servis ve iletişim noktalarının çoğalmasının saldırı yüzeyini genişletebildiğini göstermektedir.

El Fellah vd. (2026) edge üzerinde mikroservis tabanlı IoT dağıtımlarını sistematik inceleme, gri literatür analizi ve deneysel değerlendirmeyle ele almıştır. Çalışmada kapsayıcı mikroservisler, API gateway ve service-per-database yaklaşımı yaygın uygulamalar arasında belirlenmiş; uygulanan pratiklerin incelenen senaryoda throughput'u artırdığı, gecikmeyi ve bellek kullanımını azaltabildiği, buna karşılık mimari karmaşıklığın CPU maliyetini artırabildiği gösterilmiştir. Bu bulgu mikroservislerin faydalarının operasyonel ödünleşimlerle birlikte değerlendirilmesi gerektiğini ortaya koymaktadır.

API Kullanımı ve IoT Haberleşme Mekanizmaları

IoT ekosisteminde cihazlar, platformlar, mikroservisler ve son kullanıcı uygulamaları arasında iyi tanımlanmış arayüzlere ihtiyaç bulunmaktadır. Uygulama Programlama Arayüzü (Application Programming Interface,

API), bir bileşenin işlevlerinin diğer bileşenler tarafından hangi kurallar çerçevesinde kullanılacağını tanımlar. API tasarımı yalnızca teknik iletişimi değil sürümleme, güvenlik, hata yönetimi ve bağımlılıkların kontrolünü de etkileyen mimari bir karardır.

REST ve HTTP Tabanlı API'ler

REST, Fielding (2000) tarafından ağ tabanlı yazılım mimarileri bağlamında tanımlanan bir mimari stildir. REST yaklaşımında kaynakların adreslenebilir olması, tek biçimli arayüz, durumsuz etkileşim ve ara bileşenlerin kullanılabilmesi gibi ilkeler ölçeklenebilir dağıtık sistem tasarımına katkı sağlar. Güncel IoT platformlarında RESTful API'ler cihaz kayıtları, kullanıcı işlemleri, raporlama ve kurumsal entegrasyon için yaygın biçimde kullanılmaktadır.

Örneğin GET /api/devices/{id} çağrısı belirli bir cihazın metadata bilgisini döndürürken POST /api/devices/{id}/commands çağrısı cihaza komut oluşturmak amacıyla kullanılabilir. REST API'lerin web ve mobil uygulamalarla uyumu güçlüdür; ancak her sensör ölçümünün ayrı HTTP isteği olarak gönderilmesi kaynakları kısıtlı cihazlarda bağlantı ve başlık maliyetini artırabilir.

MQTT

MQTT, hafif publish/subscribe mesajlaşma modelini kullanan bir protokoldür. OASIS MQTT 5.0 standardı, protokolü özellikle bant genişliği ve kaynak kısıtlarının bulunduğu makineden makineye ve IoT iletişimi için uygun bir mekanizma olarak tanımlamaktadır (Banks vd., 2019). MQTT'de istemciler doğrudan birbirine bağlanmak yerine broker üzerinden topic'lere mesaj yayımlar veya abone olur.

Örneğin binaA/kat2/oda12/sicaklik başlığına gönderilen bir telemetri mesajını veri saklama, anomali tespiti ve pano servisleri ayrı ayrı tüketebilir. Bu yapı üreticiler ile tüketiciler arasında gevşek bağlılık sağlar. QoS seviyeleri, retained message, session state ve MQTT 5 özellikleri farklı teslimat ve oturum gereksinimlerinin yönetilmesine yardımcı olur.

CoAP

Constrained Application Protocol (CoAP), düşük güçlü ve kaynakları kısıtlı cihazlar için tasarlanmış bir web transfer protokolüdür. RFC 7252, CoAP'i UDP üzerinde düşük ek yükle request-response etkileşimi sağlayan, URI ve içerik türü kavramlarını destekleyen bir protokol olarak tanımlar

(Shelby vd., 2014). CoAP özellikle sınırlı cihaz ağlarında HTTP'ye daha hafif bir alternatif sunabilir; gözlem (observe) gibi uzantılar kaynak durumlarının değişimini takip etmeyi kolaylaştırır.

Tablo 3. REST/HTTP, MQTT ve CoAP'in genel karşılaştırması

Özellik	REST/HTTP	MQTT	CoAP
Temel model	İstek/yanıt	Yayınla/abone ol	İstek/yanıt
Tipik taşıma	TCP/TLS	TCP/TLS	UDP/DTLS veya güncel güvenli taşıma seçenekleri
Bağlantı/başlık maliyeti	Görece yüksek	Düşük	Çok düşük
Sürekli telemetri	Orta	Çok uygun	Uygun
Web/kurumsal entegrasyon	Çok güçlü	Broker/bridge üzerinden	Proxy veya uygulama katmanı üzerinden
Kaynak kısıtlı cihaz	Sınırlı/orta	Uygun	Çok uygun
Mikroservis kullanımı	API ve sorgu işlemleri	Olay ve telemetri	Daha çok cihaz/edge katmanı

API Gateway ve Olay Veri Yolu

Mikroservis sayısı arttığında istemcilerin her servisin adresini ve güvenlik politikasını bilmesi istenmez. API gateway dış istemciler için tek giriş noktası sağlayarak kimlik doğrulama, yetkilendirme, hız sınırlama, yönlendirme, API sürümleme ve gözlemleme işlevlerini merkezileştirebilir. Bununla birlikte gateway tek hata noktası hâline gelmemeli; yüksek erişilebilirlik ve yatay ölçekleme ile tasarlanmalıdır.

Olay veri yolu veya mesaj kuyruğu ise üretici ve tüketici servislerin zaman açısından birbirinden ayrılmasını sağlar. Ani telemetri artışlarında tüketici servisler geçici olarak yavaşlasa bile mesajların kontrollü biçimde tamponlanması veri kaybını azaltabilir. Ancak kuyrukların sınırsız büyümesine izin verilmemeli; retention, dead-letter queue, idempotency, backpressure ve yeniden deneme politikaları açıkça tasarlanmalıdır.

Ölçeklenebilir IoT Sistemlerinin Tasarımı

Ölçeklenebilirlik, bir IoT sisteminin cihaz sayısı, bağlantı sayısı, mesaj trafiği, veri hacmi veya uygulama isteği arttığında kabul edilebilir hizmet kalitesini koruyabilme yeteneğidir. IoT sistemlerinde ölçeklenebilirlik tek bir sunucunun daha güçlü hâle getirilmesiyle sınırlı değildir; cihaz bağlantı katmanı, broker, API, mikroservis, veri tabanı ve analitik bileşenlerinin birlikte ölçeklenebilmesi gerekir.

Panagou vd. (2025) tarafından yapılan güncel IoT platform incelemesinde ölçeklenebilirlik ve performans, platform değerlendirmesinin temel ölçütlerinden biri olarak ele alınmıştır. Büyük ölçekli sistemlerde doğru kapasite planlaması için CPU ve bellek kullanımının yanında aktif cihaz sayısı, bağlantı açma hızı, mesaj/saniye, API istek/saniye, p95/p99 gecikme, kuyruk derinliği, tüketici gecikmesi ve veri tabanı yazma hızı gibi uygulamaya özgü ölçümlerin izlenmesi gerekir.

Dikey ve Yatay Ölçekleme

Dikey ölçekleme mevcut sunucunun CPU, bellek veya depolama kapasitesinin artırılmasıdır. Uygulanması görece kolaydır ancak tek makinenin ulaşabileceği kaynak sınırları ve bakım sırasında kesinti riski nedeniyle uzun vadede sınırlıdır.

Yatay ölçekleme aynı hizmetin birden fazla örneğinin çalıştırılması esasına dayanır. Telemetri alım servisi sekiz örnek hâlinde çalıştırılırken daha düşük yük altındaki raporlama servisi iki örnekte kalabilir. Bu yaklaşım mikroservislerin temel avantajlarından biridir. Yatay ölçeklemenin etkin olabilmesi için servislerin mümkün olduğunca durumsuz (stateless) tasarlanması ve kalıcı durumun uygun paylaşımlı veya dağıtık depolarda tutulması gerekir.

Yük Dengeleme ve Otomatik Ölçekleme

Yük dengeleyici gelen istekleri veya bağlantıları aynı servisin farklı örneklerine dağıtarak tek örneğin aşırı yüklenmesini önlemeye yardımcı olur. IoT sistemlerinde MQTT bağlantı yükünün, REST API isteklerinin ve arka plan veri işleme işlerinin farklı yük dengeleme mekanizmaları gerektirebileceği unutulmamalıdır.

Otomatik ölçekleme, sistem ölçümlerine bağlı olarak servis örneği veya kaynak sayısının dinamik biçimde artırılıp azaltılmasıdır. Qu vd. (2018), bulut uygulamalarında otomatik ölçeklemenin performans hedefleri ile

kaynak maliyeti arasında denge kurmak için kullanıldığını göstermektedir. IoT iş yüklerinde CPU eşiklerinin yanında aktif bağlantı sayısı, MQTT mesaj hızı, kuyruk derinliği ve akış işleme gecikmesi gibi alan özelinde ölçütler daha erken ve anlamlı sinyaller sağlayabilir.

Kapsayıcılar ve Orkestrasyon

Kapsayıcı (container) teknolojileri mikroservislerin uygulama bağımlılıklarıyla birlikte taşınabilir bir çalışma birimi hâlinde paketlenmesini kolaylaştırır. Çok sayıda kapsayıcının üretim ortamında yönetilmesi ise orkestrasyon ihtiyacını doğurur. Kubernetes benzeri sistemler servis keşfi, sağlık kontrolü, yeniden başlatma, rolling update, yük dengeleme ve yatay pod ölçekleme gibi işlevleri otomatikleştirebilir.

Bununla birlikte kapsayıcı veya Kubernetes kullanmak tek başına ölçeklenebilirlik sağlamaz. Veri tabanı tek noktada darboğaz oluşturuyorsa veya bütün istekler senkron bir servis zincirinden geçiyorsa uygulama örneklerinin sayısını artırmak beklenen performans kazanımını sağlamayabilir. Ölçeklenebilirlik uygulama, veri ve mesajlaşma katmanlarının birlikte tasarlanmasını gerektirir.

Mesajlaşma ve Olay Güdümlü Mimari

Yüksek ölçekli IoT sistemlerinde cihazların ürettiği verinin doğrudan işleme servisine gönderilmesi ani trafik artışlarını uygulama sunucularına yansıtabilir. Araya broker veya event streaming katmanı eklenmesi üretici ve tüketicileri birbirinden ayırır. IoT Devices -> MQTT Broker -> Event Bus/Queue -> Consumer Microservices akışı kısa süreli hız farklarının tamponlanmasına ve tüketicilerin bağımsız ölçeklenmesine imkân sağlayabilir. Ancak teslimat garantisi, sıralama, tekrar tüketim ve idempotent işleme ilkeleri iş gereksinimine göre belirlenmelidir.

Veri Katmanının Ölçeklenmesi

IoT sistemlerinde zaman damgalı verilerin sürekli üretilmesi veri tabanını en hızlı büyüyen bileşenlerden biri hâline getirir. Zaman serisi veritabanları, zaman temelli bölümleme, yatay parçalama (sharding), replikasyon, önbellekleme ve sıcak-ılık-soğuk veri katmanları bu yükün yönetilmesinde kullanılabilir. Son günlerin verileri hızlı depolamada tutulurken eski ham veriler nesne depolama sistemlerine taşınabilir.

Mikroservis mimarisinde her servisin kendi veri deposunu yönetmesi servis bağımsızlığını artırabilir; ancak servisler arası transaction ve veri

tutarlılığı problemlerini karmaşıklaştırır. Bu nedenle database-per-service yaklaşımı kullanıldığında olay tabanlı veri eşitleme, saga benzeri dağıtık işlem kalıpları ve eventual consistency gibi mekanizmalar değerlendirilmelidir.

Edge-Cloud Sürekliliği ile Ölçeklenebilirlik

Ölçeklenebilirliğin yalnızca merkezi buluta daha fazla kaynak eklenerek çözülmesi ağ ve gecikme sorunlarını ortadan kaldırmaz. Edge katmanında veri filtreleme, özetleme ve yerel çıkarım yapılması merkezi sisteme ulaşan veri hacmini azaltabilir. Al-Dulaimy vd. (2024) ve Maia vd. (2024), hesaplama, iletişim ve depolama kaynaklarının uçtan buluta kadar birlikte yönetilmesinin yeni nesil uygulamalar için kritik olduğunu vurgulamaktadır.

Bu nedenle uçtan uca ölçeklenebilirlik; cihaz ölçeklenebilirliği, bağlantı ölçeklenebilirliği, mesajlaşma ölçeklenebilirliği, mikroservis ölçeklenebilirliği, veri ölçeklenebilirliği ve edge-bulut kaynak ölçeklenebilirliğinin birlikte ele alınmasını gerektirir. Bu bileşenlerden yalnızca birinin ölçeklenebilir olması sistemin tamamının ölçeklenebilir olduğu anlamına gelmez.

Gözlemlenebilirlik ve Dayanıklılık

Dağıtık IoT sistemlerinde performans sorunlarının kaynağını belirlemek için günlük, metrik ve dağıtık iz (trace) verilerinin ilişkilendirilebilmesi gerekir. API gateway gecikmesi, broker kuyrukları, mikroservis yanıt süreleri ve veri tabanı yazma gecikmeleri aynı olay kimliğiyle izlenebilirse uçtan uca darboğaz analizi kolaylaşır.

Dayanıklılık açısından zaman aşımı, sınırlı yeniden deneme, devre kesici, bulkhead, health check ve kontrollü degradation mekanizmaları önemlidir. Kontrolsüz retry politikaları bir servis arızası sırasında trafiği katlayarak zincirleme arızaya neden olabilir. Bu nedenle yüksek erişilebilirlik yalnızca replikasyon değil hata yayılımının sınırlandırılmasıyla birlikte düşünülmelidir.

Tablo 4. Ölçeklenebilir IoT sistemlerinde başlıca mekanizmalar

Mekanizma	Temel Amaç	IoT Bağlamında Örnek Ölçüt
Yatay ölçekleme	Servis kapasitesini örnek sayısını artırarak büyütmek	Mesaj/saniye, API istek/saniye
Otomatik ölçekleme	Kaynakları yüke göre dinamik ayarlamak	Aktif bağlantı, kuyruk derinliği, p95 gecikme
Yük dengeleme	Trafiği hizmet örneklerine dağıtmak	Bağlantı/istek dağılımı
Mesaj kuyruğu / event bus	Üretici ve tüketiciyi ayırmak, ani yükü tamponlamak	Consumer lag, queue depth
Veri bölümleme	Yüksek hacimli veriyi dağıtmak	Yazma hızı, bölüm boyutu
Edge işleme	Merkezi yükü ve ağ trafiğini azaltmak	Edge-to-cloud veri hacmi, yerel gecikme
Önbellekleme	Sık erişilen veriye hızlı erişim sağlamak	Cache hit ratio, API gecikmesi
Gözlemlenebilirlik	Darboğaz ve hata kök nedenini belirlemek	Trace süresi, hata oranı, servis bağımlılıkları

IoT-Bulut Sistemlerinde Güvenlik ve Gizlilik

IoT güvenliği fiziksel cihazdan bulut uygulamasına kadar bütün mimari boyunca ele alınmalıdır. IoT cihazlarının kimlik bilgilerinin ele geçirilmesi, zayıf güncelleme mekanizmaları, açık API'ler veya yanlış yapılandırılmış bulut servisleri saldırganın yalnızca veriye değil fiziksel süreçlere de erişmesine neden olabilir. Bu nedenle güvenlik tek bir ağ güvenlik duvarına bırakılabilecek bir işlev değildir.

NIST IR 8259A, IoT cihazlarında cihaz kimliği, yapılandırma, veri koruma, arayüz erişimi, yazılım güncelleme ve siber güvenlik durum farkındalığı gibi temel güvenlik yeteneklerinin ürün tasarımına dâhil edilmesini önermektedir (Fagan vd., 2020). NIST IR 8259 Rev. 1 ise 2026 güncellemesiyle IoT üreticilerinin ürün yaşam döngüsü boyunca güvenlik faaliyetlerini daha geniş bir çerçevede ele almaktadır (Fagan vd., 2026).

Cihaz Kimliği ve Kimlik Doğrulama

Her cihazın benzersiz bir kimlikle sisteme kaydedilmesi, paylaşımlı varsayılan kullanıcı adı ve parola kullanımından kaçınılması temel güvenlik gereksinimlerindendir. X.509 sertifikaları, cihaz başına anahtarlar ve güvenli provisioning süreçleri cihaz kimliğinin yönetilmesinde kullanılabilir. Kimlik bilgilerinin cihaz dosya sisteminde açık metin olarak saklanması yerine güvenli donanım veya korumalı anahtar depoları tercih edilmelidir.

Yalli vd. (2025), IoT kimlik doğrulama mekanizmalarını sistematik olarak incelemiş ve güvenlik düzeyinin yanında hesaplama yükü, iletişim maliyeti ve enerji tüketiminin de kaynakları kısıtlı cihazlar için birlikte değerlendirilmesi gerektiğini vurgulamıştır.

İletişim ve API Güvenliği

Cihaz-broker, gateway-bulut ve servis-servis iletişiminde verinin gizliliği ve bütünlüğü için uygun TLS/DTLS tabanlı güvenli taşıma mekanizmaları kullanılmalıdır. MQTT güvenli taşıma üzerinde çalıştırılmalı; CoAP kullanılan senaryolarda uygulamaya uygun güvenlik mekanizmaları uygulanmalıdır. Sertifika doğrulamasının devre dışı bırakılması veya zayıf şifre takımlarının kullanılması güvenli protokol kullanımının faydasını ortadan kaldırır.

API katmanında kimlik doğrulama, yetkilendirme, hız sınırlama, giriş doğrulama ve ayrıntılı kayıt mekanizmaları birlikte uygulanmalıdır. OAuth 2.0 / OpenID Connect kullanıcı ve uygulama erişiminde, servis kimlikleri ise mikroservisler arası iletişimde kullanılabilir. API anahtarlarının kaynak koduna gömülmesi yerine sır yönetimi sistemlerinde saklanması gerekir.

Sıfır Güven ve En Az Yetki

Sıfır güven (zero trust) yaklaşımı ağ içerisinde bulunmayı otomatik güven göstergesi olarak kabul etmez; kullanıcı, servis ve cihazların kimliği ile erişim bağlamının sürekli doğrulanmasını esas alır. Azad vd. (2024), IoT çağında sıfır güven yaklaşımının cihaz kimliği, sürekli doğrulama ve en az yetki ilkeleriyle birlikte değerlendirilmesi gerektiğini göstermektedir. Mikroservis mimarisinde her servisin yalnızca ihtiyaç duyduğu veri ve API'lere erişmesi saldırı alanını sınırlandırabilir.

API ve web tabanlı IoT uygulamalarında güvenlik yalnızca kullanıcı kimliğinin doğrulanmasıyla sınırlı olmayıp, otomatik erişim ve bot tabanlı saldırıların engellenmesini de kapsamalıdır. Bu bağlamda CAPTCHA

mekanizmaları, insan kullanıcılar ile otomatik sistemlerin ayrıştırılmasında tamamlayıcı bir güvenlik katmanı olarak kullanılabilir. Gürfidan vd. (2025), görüntü işleme teknikleri ile evrişimli sinir ağı (CNN) tabanlı değerlendirmeyi birleştiren etkileşimli ve akıllı bir CAPTCHA modeli önererek, otomatik çözücülere karşı daha dayanıklı doğrulama mekanizmalarının geliştirilebileceğini göstermiştir.

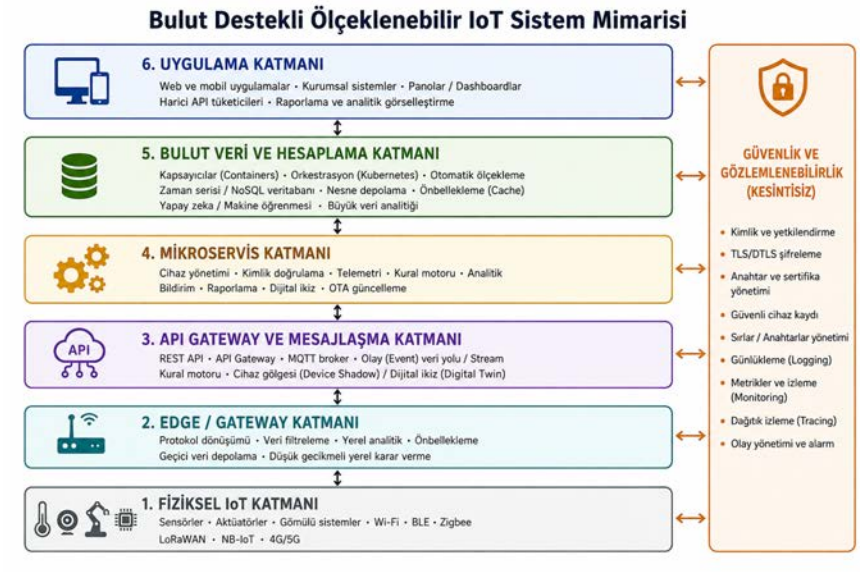
Güncelleme ve Yaşam Döngüsü Güvenliği

IoT cihazlarının uzun süre sahada kalması güvenli yazılım güncellemesini kritik hâle getirir. OTA güncellemelerinin imzalı olması, bütünlük ve kaynak doğrulamasının yapılması, başarısız güncellemede geri dönüş mekanizması bulunması ve destek süresinin açık biçimde tanımlanması gerekir. Güvenlik yaşam döngüsü cihaz üretiminden provisioning, işletim, güncelleme ve güvenli hizmetten çıkarma aşamasına kadar devam etmelidir.

Bulut Destekli IoT Sistemleri İçin Bütünleşik Referans Mimari

Önceki bölümlerde tartışılan teknolojiler birlikte değerlendirildiğinde yüksek ölçekli bir IoT çözümünün fiziksel cihaz, bağlantı, edge/gateway, mesajlaşma, API, mikroservis, veri/hesaplama ve uygulama katmanlarından oluşan modüler bir yapı ile tasarlanması uygun görünmektedir. Önerilen mimaride güvenlik ve gözlemlenebilirlik bütün katmanları dikey olarak kesen ortak yeteneklerdir.

Fiziksel katmanda sensör ve aktüatörler veri üretirken edge/gateway katmanı protokol dönüşümü, veri filtreleme ve düşük gecikmeli yerel karar görevlerini yerine getirir. IoT platformu ve mesajlaşma katmanında cihaz kayıtları, MQTT broker, kural motoru, olay veri yolu ve dijital ikiz işlevleri bulunur. API gateway dış istemciler ile mikroservisler arasındaki kontrollü giriş noktasıdır. Mikroservis katmanı cihaz yönetimi, telemetri, analitik, kimlik, bildirim ve raporlama gibi iş yeteneklerini ayrıştırır. Bulut veri ve hesaplama katmanı ise kapsayıcılar, otomatik ölçekleme, veri tabanları, nesne depolama ve AI/ML iş yüklerini barındırır.



Şekil 2. Bulut destekli ölçeklenebilir IoT sistemleri için bütünleşik referans mimari.

Bu mimaride tipik bir veri akışı şu şekilde gerçekleşebilir: sensör ölçümü MQTT aracılığıyla gateway veya broker'a iletilir; telemetri servisi ilgili topic'i tüketerek veriyi doğrular ve zaman serisi veri tabanına yazar; kural motoru kritik eşik tespit ettiğinde bildirim servisine olay üretir; analitik servisi aynı olayı farklı tüketici olarak işleyebilir; REST API ise web ve kurumsal uygulamalara sorgulanabilir sonuç sunar. Bu akışta üretici cihazın sonraki servislerin ayrıntılarını bilmesine gerek yoktur.

Mimarinin önemli özelliği her katmanın farklı ölçekleme ölçütüne sahip olabilmesidir. MQTT broker aktif bağlantı ve mesaj hızına, telemetri servisi kuyruk uzunluğuna, API servisi istek hızı ve p95 gecikmeye, analitik servisleri CPU/GPU kullanımına göre ölçeklenebilir. Edge katmanı ise merkezi trafiği azaltarak bulut ölçeklendirme ihtiyacını doğrudan etkiler. Bu nedenle sistemin kapasitesi tek bir bileşen üzerinden değil uçtan uca ölçülmelidir.

Güncel Eğilimler ve Gelecek Yönelimleri

Serverless Computing

Sunucusuz bilişim (serverless computing), uygulama işlevlerinin sürekli çalışan sunucu süreçleri yerine olaylarla tetiklenen fonksiyonlar veya yönetilen servisler olarak yürütülmesini sağlar. IoT iş yüklerinin önemli bir bölümü olay tabanlı olduğu için serverless yaklaşımı cihaz kaydı, dosya

işleme, alarm üretimi veya dönemsel dönüşüm görevlerinde kaynak verimliliği sağlayabilir.

Cassel vd. (2022), IoT bağlamında serverless computing çalışmalarını sistematik olarak incelemiş ve fonksiyonların cloud, fog ve edge katmanlarında yürütülmesine yönelik büyüyen bir araştırma alanı olduğunu göstermiştir. Shafiei vd. (2022) ise serverless sistemlerin otomatik ölçekleme ve kullanım kadar ödeme avantajlarına karşın cold start, durum yönetimi ve sağlayıcı bağımlılığı gibi zorluklar taşıdığını vurgulamaktadır.

AIoT ve Edge AI

Artificial Intelligence of Things (AIoT), IoT tarafından üretilen verilerin makine öğrenmesi ve yapay zekâ yöntemleriyle işlenmesini ifade etmektedir. Model eğitimi çoğunlukla yüksek hesaplama kapasitesi nedeniyle bulutta yürütülürken çıkarım işlemleri giderek edge veya uç cihazlara taşınmaktadır. Kamera görüntüsünün tamamını buluta göndermek yerine edge üzerinde nesne tespiti yaparak yalnızca sonuçların iletilmesi gecikme ve bant genişliği açısından önemli avantajlar sağlayabilir. Bu yaklaşım aynı zamanda kişisel veya hassas ham verinin cihaz dışına çıkmasını sınırlayabilir.

Dijital İkizler

Dijital ikiz, fiziksel bir nesne veya sürecin dijital ortamda güncel durumunu temsil eden ve fiziksel varlıkla veri alışverişi yapan modeldir. IoT platformlarından gelen telemetri dijital ikizin durumunu güncellerken bulut analitiği kestirimci bakım, senaryo analizi ve optimizasyon için kullanılabilir. Dijital ikiz yaklaşımı yalnızca bir cihaz gölgesinden daha geniştir; fiziksel sistemin davranışını, ilişkilerini ve kimi durumlarda simülasyon modelini de kapsayabilir.

IoT-Edge-Cloud Hesaplama Sürekliliği

Gelecek nesil sistemlerde "iş yükü edge'de mi yoksa bulutta mı çalışmalı?" sorusu ikili bir seçim olmaktan çıkmaktadır. Al-Dulaimy vd. (2024) tarafından tanımlanan hesaplama sürekliliği yaklaşımında görevler uç cihaz, edge, bölgesel kaynak ve merkezi bulut arasında gereksinimlere göre dinamik olarak dağıtılabilir. Maia vd. (2024) iletişim, hesaplama ve önbellekleme kaynaklarının bu süreklilikte birlikte yönetilmesinin özellikle düşük gecikme ve yüksek güvenilirlik isteyen yeni uygulamalar için önemli olduğunu göstermektedir.

Bu modelde basit ön işleme uç cihazda, gerçek zamanlı çıkarım edge'de, bölgesel toplulaştırma fog katmanında, büyük ölçekli model eğitimi ve uzun dönemli analitik merkezi bulutta yürütülebilir. Gelecekte görev yerleştirme kararlarının ağ koşulu, enerji, maliyet, veri gizliliği ve gecikme hedeflerini birlikte dikkate alan otonom orkestrasyon mekanizmalarıyla verilmesi beklenmektedir.

Tartışma

IoT platformları, bulut bilişim, mikroservisler, API'ler ve ölçekleme mekanizmaları uygulamada aynı dağıtık sistemin birbirini tamamlayan bileşenleridir. Platform katmanı heterojen cihazların yönetimini ve telemetri akışını standartlaştırırken bulut altyapısı elastik hesaplama ve depolama kaynakları sunar. Mikroservisler işlevlerin bağımsız geliştirilmesine ve ölçeklenmesine imkân verir; API ve olay mekanizmaları ise bu bileşenler arasındaki sözleşmeleri oluşturur.

Bununla birlikte modern mimariye ait her teknoloji her sistem için gerekli değildir. Az sayıda cihaz ve düşük trafik içeren bir uygulamada monolitik bir arka uç ve tek bir yönetilen veri tabanı daha ekonomik ve yönetilebilir olabilir. Buna karşılık yüz binlerce cihazın sürekli telemetri gönderdiği, bağımsız ekiplerin farklı işlevleri geliştirdiği ve kesintisiz hizmet hedefinin bulunduğu bir sistemde olay güdümlü mikroservis mimarisi, yatay ölçekleme ve dağıtık veri işleme gerekli hâle gelebilir. Mimari karar sistemin gerçek ölçeği ve kalite gereksinimlerinden türetilmelidir.

Mikroservisler ölçeklenebilirliği otomatik olarak garanti etmez. Aşırı servis parçalanması ağ gecikmesini, hata olasılığını ve operasyon yükünü artırabilir. El Fellah vd. (2026) tarafından gösterildiği gibi edge-IoT ortamlarında belirli mikroservis pratikleri performans kazanımı sağlayabilirken CPU maliyeti ve mimari karmaşıklık artabilir. Benzer biçimde edge computing merkezi bulut yükünü azaltabilir ancak çok sayıda edge düğümünün güvenli güncellenmesi ve gözlemlenmesi yeni bir işletim problemi oluşturur.

Bu nedenle IoT sistemi tasarımında temel yaklaşım "teknoloji odaklı" değil "gereksinim ve ölçüm odaklı" olmalıdır. Sistem geliştirme sürecinin başında cihaz sayısı, veri hızı, gecikme hedefi, veri saklama süresi, güvenlik sınıfı, bağlantı sürekliliği, maliyet ve büyüme beklentileri tanımlanmalı; mimari kararlar bu gereksinimlere göre verilmelidir. Üretim ortamında ise

kapasite ve mimari varsayımlar gerçek telemetri ölçümleriyle sürekli doğrulanmalıdır.

SONUÇ VE ÖNERİLER

IoT uygulamalarının artan cihaz sayısı, veri hacmi ve gerçek zamanlı işlem gereksinimleri, klasik merkezi yazılım mimarilerinin ötesinde esnek ve dağıtık çözümler gerektirmektedir. Bulut bilişim IoT sistemlerine yüksek hesaplama ve depolama kapasitesi ile elastik kaynak yönetimi sağlarken IoT platformları cihaz yönetimi, telemetri toplama, mesajlaşma ve uygulama entegrasyonu için ortak bir ara katman oluşturmaktadır.

Mikroservis mimarileri IoT işlevlerinin bağımsız hizmetler hâlinde tasarlanmasına imkân vererek özellikle yatay ölçekleme, bağımsız dağıtım ve hata izolasyonu bakımından avantaj sağlamaktadır. Bununla birlikte servisler arası ağ iletişimi, dağıtık veri yönetimi, güvenlik ve gözlemlenebilirlik gibi yeni karmaşıklıklar ortaya çıkarmaktadır. Bu nedenle mikroservis kullanımı sistem boyutu ve bağımsız ölçekleme ihtiyacıyla gerekçelendirilmelidir.

REST API'ler uygulamalar ve servisler arasında standart entegrasyon sağlarken MQTT publish/subscribe modeli sürekli telemetri ve olay akışları için, CoAP ise kaynakları kısıtlı cihazlar için uygun seçenekler sunmaktadır. Büyük ölçekli IoT sistemlerinin tek bir protokole dayandırılması yerine farklı görevler için uygun iletişim modellerinin birlikte kullanılması daha esnek bir mimari sağlar.

Ölçeklenebilirlik yalnızca daha güçlü sunucular kullanmakla sağlanamaz. Yatay ölçekleme, yük dengeleme, durumsuz servis tasarımı, mesaj kuyrukları, kapsayıcı orkestrasyonu, veri bölümlendirme, edge işleme ve gözlemlenebilirlik birlikte ele alınmalıdır. Güvenlik ise cihaz kimliğinden API erişimine ve yazılım güncellemesine kadar bütün katmanlarda tasarımın ayrılmaz bir parçası olmalıdır.

Sonuç olarak geleceğin IoT mimarilerinin merkezi bulut, edge computing, mikroservisler, serverless computing, yapay zekâ ve dijital ikiz teknolojilerinin bütünleştiği hesaplama sürekliliklerine doğru gelişeceği değerlendirilmektedir. Başarılı bir IoT mimarisinin temel ölçütü belirli bir teknolojiye sahip olması değil; gecikme, güvenlik, maliyet, performans, yönetilebilirlik ve ölçeklenebilirlik gereksinimleri arasında ölçülebilir ve sürdürülebilir bir denge kurabilmesidir.

KAYNAKÇA

Aksoy, B., Özsoy, K., Yücel, M., Ekrem, Ö., & Salman, O. K. M. (2022). Motion control of the robot arm manufactured with a three-dimensional printer and hardness detection of objects. Bilişim Teknolojileri Dergisi, 15(3), 289-300.

Al-Dulaimy, A., Jansen, M., Johansson, B., Trivedi, A., Iosup, A., Ashjaei, M., Galletta, A., Kimovski, D., Prodan, R., Tserpes, K., Kousiouris, G., Giannakos, C., Brandic, I., Ali, N., Bondi, A. B., & Papadopoulos, A. V. (2024). The computing continuum: From IoT to the cloud. Internet of Things, 27, 101272. https://doi.org/10.1016/j.iot.2024.101272

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376. https://doi.org/10.1109/COMST.2015.2444095

Amazon Web Services. (2026). How AWS IoT works - AWS IoT Core. AWS Documentation. Erişim tarihi: 2 Eylül 2026, https://docs.aws.amazon.com/iot/latest/developerguide/aws-iot-how-it-works.html

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54(15), 2787-2805. https://doi.org/10.1016/j.comnet.2010.05.010

Azad, M. A., Abdullah, S., Arshad, J., Lallie, H., & Ahmed, Y. (2024). Verify and trust: A multidimensional survey of zero-trust security in the age of IoT. Internet of Things, 27, 101227. https://doi.org/10.1016/j.iot.2024.101227

Banks, A., Briggs, E., Borgendale, K., & Gupta, R. (Eds.). (2019). MQTT Version 5.0. OASIS Standard. https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 13-16. https://doi.org/10.1145/2342509.2342513

Botta, A., de Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and Internet of Things: A survey. Future Generation Computer Systems, 56, 684-700. https://doi.org/10.1016/j.future.2015.09.021

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6), 599-616. https://doi.org/10.1016/j.future.2008.12.001

Cassel, G. A. S., Rodrigues, V. F., Righi, R. R., Bez, M. R., Nepomuceno, A. C., & da Costa, C. A. (2022). Serverless computing for Internet of Things: A systematic literature review. Future Generation Computer Systems, 128, 299-316. https://doi.org/10.1016/j.future.2021.10.020

Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. Journal of Systems and Software, 150, 77-97. https://doi.org/10.1016/j.jss.2019.01.001

Domínguez-Bolaño, T., Campos, O., Barral, V., Escudero, C. J., & García-Naya, J. A. (2022). An overview of IoT architectures, technologies, and existing open-source projects. Internet of Things, 20, 100626. https://doi.org/10.1016/j.iot.2022.100626

El Akhdar, A., Baidada, C., Kartit, A., Hanine, M., García, C. O., Lara, R. G., & Ashraf, I. (2024). Exploring the potential of microservices in Internet of Things: A systematic review of security and prospects. Sensors, 24(20), 6771. https://doi.org/10.3390/s24206771

El Fellah, Y., Minani, J. B., Moha, N., Gascon-Samson, J., & Guéhéneuc, Y.-G. (2026). Analysis of microservices-based IoT systems: Deployment challenges, industry practices, and performance insights. Internet of Things, 36, 101867. https://doi.org/10.1016/j.iot.2025.101867

Fagan, M., Megas, K. N., Scarfone, K., & Smith, M. (2020). IoT Device Cybersecurity Capability Core Baseline (NISTIR 8259A). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8259A

Fagan, M., Megas, K., Cuthill, B., Marron, J., & Hoehn, B. (2026). Foundational Cybersecurity Activities for IoT Product Manufacturers (NIST IR 8259 Rev. 1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8259r1

Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation). University of California, Irvine.

Gkonis, P., Giannopoulos, A., Trakadas, P., Masip-Bruin, X., & D'Andria, F. (2023). A survey on IoT-edge-cloud continuum systems: Status, challenges, use cases, and open issues. Future Internet, 15(12), 383. https://doi.org/10.3390/fi15120383

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660. https://doi.org/10.1016/j.future.2013.01.010

Gürfidan, R., Kilim, O., & Yiğit, T. (2025). Robust, interactive, and intelligent Captcha model based on image processing. *International Journal of Intelligent Systems and Applications, 17*(3), 1–14. https://doi.org/10.5815/ijisa.2025.03.01

ISO/IEC. (2024). ISO/IEC 30141:2024 Internet of Things (IoT) - Reference architecture (2nd ed.). International Organization for Standardization.

Li, S., Zhang, H., Jia, Z., Zhong, C., Zhang, C., Shan, Z., Shen, J., & Babar, M. A. (2021). Understanding and addressing quality attributes of microservices architecture: A systematic literature review. Information and Software Technology, 131, 106449. https://doi.org/10.1016/j.infsof.2020.106449

Li, X., Xu, L. D., Sigov, A., Ratkin, L., & Ivanov, L. A. (2025). Enterprise architecture of IoT-based applications: A review. Future Generation Computer Systems, 166, 107584. https://doi.org/10.1016/j.future.2024.107584

Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet of Things Journal, 4(5), 1125-1142. https://doi.org/10.1109/JIOT.2017.2683200

Maia, A., Boutouchent, A., Kardjadja, Y., Gherari, M., Gelal Soyak, E., Saqib, M., Boussekar, K., Cilbir, I., Habibi, S., Ouledsidi Ali, S., Ajib, W., Elbiaze, H., Erçetin, O., Ghamri-Doudane, Y., & Glitho, R. (2024). A survey on integrated computing, caching, and communication in the cloud-to-edge continuum. Computer Communications, 219, 128-152. https://doi.org/10.1016/j.comcom.2024.03.005

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing (NIST SP 800-145). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145

Microsoft. (2026). Azure IoT Hub documentation. Microsoft Learn. Erişim tarihi: 2 Eylül 2026, https://learn.microsoft.com/azure/iot-hub/

Mineraud, J., Mazhelis, O., Su, X., & Tarkoma, S. (2016). A gap analysis of Internet-of-Things platforms. Computer Communications, 89-90, 5-16. https://doi.org/10.1016/j.comcom.2016.03.015

Panagou, I. C., Katsoulis, S., Nannos, E., Zantalis, F., & Koulouras, G. (2025). A comprehensive evaluation of IoT cloud platforms: A feature-driven review with a decision-making tool. Sensors, 25(16), 5124. https://doi.org/10.3390/s25165124

Qu, C., Calheiros, R. N., & Buyya, R. (2018). Auto-scaling web applications in clouds: A taxonomy and survey. ACM Computing Surveys, 51(4), Article 73. https://doi.org/10.1145/3148149

Satyanarayanan, M. (2017). The emergence of edge computing. Computer, 50(1), 30-39. https://doi.org/10.1109/MC.2017.9

Sethi, P., & Sarangi, S. R. (2017). Internet of Things: Architectures, protocols, and applications. Journal of Electrical and Computer Engineering, 2017, 9324035. https://doi.org/10.1155/2017/9324035

Shafiei, H., Khonsari, A., & Mousavi, P. (2022). Serverless computing: A survey of opportunities, challenges, and applications. ACM Computing Surveys, 54(11s), Article 239. https://doi.org/10.1145/3510611

Shelby, Z., Hartke, K., & Bormann, C. (2014). The Constrained Application Protocol (CoAP) (RFC 7252). Internet Engineering Task Force. https://doi.org/10.17487/RFC7252

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637-646. https://doi.org/10.1109/JIOT.2016.2579198

Siddiqui, H., Khendek, F., & Toeroe, M. (2023). Microservices based architectures for IoT systems - State-of-the-art review. Internet of Things, 23, 100854. https://doi.org/10.1016/j.iot.2023.100854

ThingsBoard. (2026). ThingsBoard documentation: Architecture, protocols and APIs. Erişim tarihi: 2 Eylül 2026, https://thingsboard.io/docs/

Yalli, J. S., Hasan, M. H., Jung, L. T., & Al-Selwi, S. M. (2025). Authentication schemes for Internet of Things (IoT) networks: A systematic review and security assessment. Internet of Things, 30, 101469. https://doi.org/10.1016/j.iot.2024.101469

EDGE VE FOG COMPUTING İLE GERÇEK ZAMANLI İŞLEME

Uçta Veri İşleme, Düşük Gecikme ve Edge AI

Serkan Keskin*, Enes Açıkgözoğlu**

GİRİŞ

Bulut bilişim, ölçeklenebilir işlem gücü, merkezi yönetim ve ekonomik kaynak paylaşımı sayesinde günümüz sayısal hizmetlerinin temel altyapılarından biri hâline gelmiştir. Bununla birlikte IoT sensörleri, akıllı kameralar, endüstriyel kontrol birimleri, mobil cihazlar ve araçlar tarafından üretilen verinin hacmi ve zaman duyarlılığı arttıkça, bütün verinin önce uzak bir veri merkezine taşındığı ve bütün kararların bulutta üretildiği tek merkezli mimarinin sınırları daha görünür hâle gelmektedir. Özellikle milisaniye düzeyinde tepki gerektiren kontrol döngülerinde, yüksek çözünürlüklü video akışlarında ve bağlantının kesintili olduğu saha uygulamalarında ağın kendisi uygulama performansının belirleyici bir parçasına dönüşmektedir.

Edge computing yaklaşımı, işleme ve depolama kaynaklarını ağın merkezinden veri kaynağına doğru taşıyarak bu soruna yanıt verir. Shi vd. (2016), edge computing'i verinin ağın uç noktasına yakın işlenmesini öne çıkaran ve yanıt süresi, bant genişliği maliyeti, enerji ve mahremiyet gibi sorunları hedefleyen bir paradigma olarak ele almaktadır. Satyanarayanan (2017) ise özellikle etkileşimli ve mobil uygulamalarda, kullanıcıya yakın hesaplama kaynaklarının buluta erişim gecikmesini azaltabileceğini ve daha öngörülebilir hizmet sağlayabileceğini vurgulamaktadır.

Fog computing kavramı, edge yaklaşımını daha hiyerarşik bir bakışla genişletir. Bonomi vd. (2012), fog katmanını bulut hizmetlerinin ağın kenarına doğru uzantısı olarak ele almış; düşük gecikme, konum farkındalığı, coğrafi dağılım ve mobilite desteğini temel özellikler arasında göstermiştir. NIST'in kavramsal modeli de fog computing'i uç cihazlarla

* Öğretim Görevlisi, Burdur Mehmet Akif Ersoy Üniversitesi, Teknik Bilimler Meslek Yüksekokulu, Veri Tabanı, Ağ Tasarımı Ve Yönetimi Bölümü, Burdur, skeskin@mehmetakif.edu.tr, ORCID: orcid.org/0000-0001-9404-5039
** Dr. Öğr. Üyesi, Isparta Uygulamalı Bilimler Üniversitesi, Isparta Bilişim Teknolojileri Meslek Yüksekokulu, Veri Tabanı, Ağ Tasarımı Ve Yönetimi Bölümü, Isparta, enesacikgozoglu@isparta.edu.tr, ORCID: orcid.org/0000-0001-7293-883X

merkezi bulut arasında dağıtık, federatif ve çok katmanlı bir işlem modeli olarak tanımlamaktadır (Iorga vd., 2018). Bu bakış, edge ve fog kavramlarının mutlak biçimde birbirinden ayrılmasından çok, farklı yerellik düzeylerini temsil eden tamamlayıcı katmanlar olarak değerlendirilmesini sağlar.

Yapay zekânın uç sistemlere taşınması, edge computing'in önemini daha da artırmıştır. Derin öğrenme modellerinin eğitimi çoğunlukla güçlü veri merkezi kaynaklarında sürdürülse de çıkarımın cihaz üzerinde veya kullanıcıya yakın edge sunucuda yapılabilmesi; düşük gecikme, mahremiyet ve bağlantıdan bağımsız çalışma açısından önemli kazanımlar sağlayabilmektedir. Chen ve Ran (2019) ile Zhou vd. (2019), yapay zekâ yeteneklerinin ağ ucuna taşınmasını Edge AI/Edge Intelligence bağlamında ele alarak hesaplama, iletişim ve model tasarımının birlikte değerlendirilmesi gerektiğini ortaya koymuştur.

Bu çalışmanın amacı edge computing, fog computing ve gerçek zamanlı işleme arasındaki ilişkiyi; ağ, sistem ve yapay zekâ katmanlarını birlikte kapsayan bütüncül bir çerçevede açıklamaktır. Bölümde önce edge-fog-cloud sürekliliği açıklanmakta, ardından uçta veri işleme, gecikme ve bant genişliği, gerçek zamanlı işlem, iş yükü yerleştirme ve offloading konuları incelenmektedir. Daha sonra cihaz üzerinde çıkarım ve Edge AI teknikleri değerlendirilmekte; güvenlik, mahremiyet, enerji ve güvenilirlik sorunları tartışılmaktadır. Son kısımda MEC, TinyML, güvenilir Edge Intelligence ve uçta büyük dil modelleri gibi güncel yönelimler ele alınmaktadır.

Edge-Fog-Cloud Sürekliliği ve Kavramsal Çerçeve

Bulut Bilişim

Bulut bilişim; işlemci, bellek, depolama, veri tabanı, analitik ve yapay zekâ hizmetlerinin büyük ölçekli veri merkezlerinden talebe bağlı olarak sağlanmasına dayanır. Merkezi kaynak havuzu; elastik ölçekleme, yüksek kullanılabilirlik, büyük model eğitimi ve kurumsal yönetim açısından güçlüdür. Buna karşın uç cihaz ile veri merkezi arasındaki fiziksel mesafe, operatör ağı, İnternet yönlendirmesi ve sunucu kuyrukları, uygulamaya ek gecikme ve değişkenlik getirebilir. Bu nedenle bulutun güçlü olduğu 'yüksek kapasite ve küresel görünürlük' ile edge'in güçlü olduğu 'yerellik ve hızlı tepki' birbirinden farklı tasarım hedefleridir.

Edge Computing

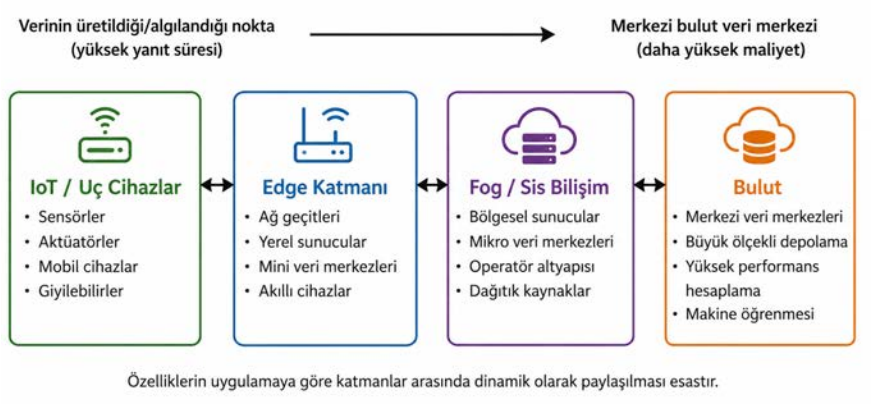
Edge computing, hesaplamayı veri kaynağının kendisine veya veri kaynağına ağ topolojisi açısından çok yakın bir düğüme yerleştirir. Akıllı kamera, endüstriyel PC, ağ geçidi, mobil cihaz, araç içi bilgisayar, baz istasyonu yakınındaki sunucu veya mikro veri merkezi bir edge düğümü olabilir. Edge yaklaşımının ayırt edici yönü, yalnızca küçük cihaz kullanması değildir; karar döngüsünü verinin üretildiği yere yaklaştırmasıdır. Bu nedenle güçlü bir yerel sunucu da uygulamanın veri kaynağına yakınsa edge rolü üstlenebilir (Shi vd., 2016; Satyanarayanan, 2017).

Fog Computing ve MEC

Fog computing, uç cihaz ile merkezi bulut arasında bir veya daha fazla ara hesaplama katmanı kurulmasına dayanır. Bu katmanlarda veri toplama, bölgesel analitik, önbellekleme, görev planlama, güvenlik politikası uygulama ve birden fazla edge düğümünün koordinasyonu gerçekleştirilebilir. NIST modeli, fog düğümlerinin hiyerarşik veya ağ biçiminde düzenlenebileceğini ve yerel hizmetlerin bulutla birlikte çalışabileceğini belirtmektedir (Iorga vd., 2018). Telekomünikasyon tarafında ETSI'nin Multi-access Edge Computing (MEC) çerçevesi, erişim ağına yakın hesaplama ortamlarının standart bir mimari içerisinde yönetilmesine yönelik işlevsel bileşenleri ve referans noktalarını tanımlamaktadır (ETSI, 2024).

Kavramların Birlikte Değerlendirilmesi

Edge, fog ve cloud kavramlarını birbirinin alternatifi olarak görmek çoğu gerçek sistem için yanıltıcıdır. Gerçek dünya uygulamalarında verinin bir bölümü cihazda, bir bölümü yakın edge/fog düğümünde, uzun dönemli analiz ve model eğitimi ise bulutta işlenebilir. Bu yaklaşım giderek 'edge-cloud continuum' veya hesaplama sürekliliği kavramıyla ifade edilmektedir. Süreklilik yaklaşımında temel problem, hangi iş yükünün hangi anda hangi katmanda yürütüleceğinin belirlenmesidir.



Şekil 1. Edge-fog-cloud sürekliliğinin kavramsal görünümü.

Tablo 1. Cloud, fog ve edge computing yaklaşımlarının karşılaştırılması

Özellik	Cloud Computing	Fog Computing	Edge Computing
İşlem konumu	Merkezi/büyük veri merkezleri	Uç ile bulut arasındaki dağıtık düğümler	Veri kaynağına çok yakın veya cihaz üzerinde
Tipik gecikme	Ağ koşullarına bağlı; görece yüksek olabilir	Düşük-orta	En düşük olabilecek yerleşim
Bant genişliği ihtiyacı	Ham veri aktarımında yüksek	Ön işleme ile azaltılabilir	Yerel işlem nedeniyle düşük olabilir
Kaynak kapasitesi	Çok yüksek	Orta-yüksek	Cihaza göre düşükten yükseğe
Yerel karar verme	Sınırlı	Yüksek	Çok yüksek
Büyük model eğitimi	Çok uygun	Belirli senaryolarda	Genellikle sınırlı
Bağlantı kesintisine dayanım	Düşük-orta	Orta	Yerel işlevlerde yüksek
Yönetim kapsamı	Küresel	Bölgesel/hiyerarşik	Yerel

Uçta Veri İşleme

IoT sistemlerinde sensörlerin ürettiği her veri örneğinin aynı bilgi değerine sahip olduğu varsayılamaz. Bir sıcaklık sensörü çoğu zaman yavaş değişen düşük hacimli veri üretirken, bir kamera veya LiDAR saniyede çok büyük veri akışları oluşturabilir. Ham verinin tamamını buluta göndermek yerine uçta filtreleme ve anlamlandırma yapılması, ağ yükünü azaltırken karar süresini de kısaltabilir. Ananthanarayanan vd. (2017), gerçek zamanlı video analitiğinde veri hacmi, işlem yükü ve gecikme gereksiniminin birlikte ele alınması gerektiğini ve kamera ile bulut arasında hiyerarşik bir işlem mimarisinin daha uygun olduğunu vurgulamaktadır.

Uçta veri işleme tipik olarak bir ardışık düzen biçiminde gerçekleşir. İlk aşamada veri toplanır ve zaman damgası, bütünlük veya örnekleme açısından doğrulanır. Ardından gürültü filtreleme, yeniden örnekleme, normalizasyon, veri sıkıştırma veya eksik veri işleme uygulanabilir. Bir sonraki aşamada özellik çıkarımı, olay algılama veya makine öğrenmesi çıkarımı çalıştırılır. Son aşamada yalnızca olay, özet, özellik vektörü veya sınıflandırma sonucu gibi daha düşük hacimli ve daha yüksek bilgi değerli çıktılar üst katmanlara gönderilebilir.

- Gürültü giderme, filtreleme ve sinyal ön işleme
- Örnekleme hızının uyarlanması ve veri azaltma
- Özellik çıkarımı ve veri birleştirme
- Yerel anomali veya olay tespiti
- Makine öğrenmesi / derin öğrenme çıkarımı
- Veri sıkıştırma, anonimleştirme veya maskeleme
- Aktüatörler için yerel kontrol kararlarının üretilmesi

Uçta işleme, veri kalitesini ve bağlamını da iyileştirebilir. Örneğin farklı sensörlerden gelen ölçümler aynı edge ağ geçidinde zaman eşleştirilerek birleştirilebilir. Bu, merkezi buluta gönderilen verinin yalnızca hacmini değil semantik karmaşıklığını da azaltabilir. Bununla birlikte ön işleme agresif biçimde yapılırsa daha sonra gerekli olabilecek ayrıntıların kaybedilmesi riski vardır. Bu nedenle hangi verinin ham olarak saklanacağı, hangi verinin özetleneceği ve hangi olayların buluta iletileceği uygulama gereksinimlerine göre belirlenmelidir.

Gecikme, Bant Genişliği ve Gerçek Zamanlı İşleme

Gecikmenin Bileşenleri

Bir edge uygulamasında yalnızca modelin veya algoritmanın çalışma süresini ölçmek, uçtan uca tepki süresini açıklamak için yeterli değildir. Toplam gecikme; verinin hazırlanması, kuyrukta beklemesi, ağ üzerinden iletilmesi, hesaplanması ve sonucun geri taşınması gibi birden fazla bileşenin toplamıdır. Basitleştirilmiş biçimde aşağıdaki ifade kullanılabilir:

T_toplam = T_hazırlama + T_kuyruk + T_iletim + T_işleme + T_dönüş

Edge yerleşiminin temel avantajı, özellikle T_iletim ve bazı durumlarda T_kuyruk bileşenlerini azaltmasıdır. Ancak yoğun yük altındaki küçük bir edge cihazında T_işleme ve T_kuyruk artabilir. Bu nedenle 'edge her zaman daha hızlıdır' şeklindeki genelleme doğru değildir; doğru olan, iş yükünün gereksinimleri ile mevcut hesaplama ve ağ kaynaklarının birlikte değerlendirilmesidir.



Şekil 2. Uçtan uca gecikmenin temel bileşenleri.

Jitter ve Kuyruk Gecikmeleri

Gerçek zamanlı uygulamalarda ortalama gecikme kadar gecikmenin değişkenliği de önemlidir. Jitter, ardışık isteklerin tepki süreleri arasındaki değişkenliği ifade eder. Ortalama gecikmesi düşük bir sistem, nadiren de olsa çok yüksek kuyruk gecikmeleri üretiyorsa kontrol, artırılmış gerçeklik veya etkileşimli robotik gibi uygulamalarda güvenilir olmayabilir. Bu nedenle yalnızca ortalama değer değil; medyan, standart sapma, P95 ve P99 gibi kuyruk yüzdelikleri de raporlanmalıdır.

Bant Genişliği

Bir sensör ağının ham veri üretimi yaklaşık olarak cihaz sayısı (N), bir cihazın birim zamandaki örnek sayısı (R) ve örnek boyutunun (S) çarpımıyla düşünülebilir: $B \approx N \times R \times S$. Kamera, mikrofon dizisi ve LiDAR gibi sensörlerde R ve S büyük olduğu için merkezi buluta sürekli ham veri aktarımı ekonomik ve teknik olarak zorlayıcı olabilir. Edge tarafında olay tespiti, özellik çıkarımı veya sıkıştırma uygulanması, üst katmanlara gönderilecek veri miktarını önemli ölçüde azaltabilir (Shi vd., 2016; Mao vd., 2017).

Gerçek Zamanlı Sistem Perspektifi

Gerçek zamanlı kavramı, yalnızca 'çok hızlı' hesaplama anlamına gelmez. Bir sistemin gerçek zamanlı olması, çıktının belirli bir zaman sınırı içinde üretilmesinin doğruluğun bir parçası olması anlamına gelir. Buttazzo (2024), öngörülebilir zamanlama ve son teslim sürelerinin gerçek zamanlı sistem tasarımının merkezinde bulunduğunu vurgular. Bu bakımdan edge computing, gecikmeyi azaltan bir altyapı yaklaşımıdır; gerçek zaman garantisi ise görev zamanlama, işletim sistemi, ağ, önceliklendirme ve en kötü durum yürütme süresi analizlerini de gerektirir.

Tablo 2. Gerçek zamanlı işlem sınıfları ve edge bağlamındaki örnekleri

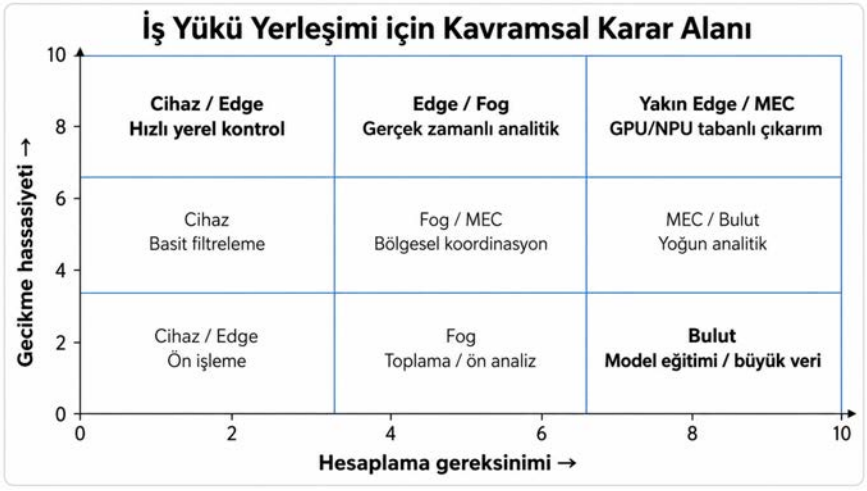
Sınıf	Temel özellik	Zaman sınırı aşıldığında	Edge örneği
Hard real-time	Deadline kesin gereksinimdir	Kabul edilemez / güvenlik riski oluşturabilir	Kritik kontrol döngüsü, belirli robotik/otomotiv fonksiyonları
Firm real-time	Geç sonuç büyük ölçüde değerini kaybeder	Tekil ihlaller tolere edilebilir	Endüstriyel olay sınıflandırma, kısa ömürlü kararlar
Soft real-time	Gecikme hizmet kalitesini düşürür	Sistem çalışır ancak deneyim kötüleşir	Video analitiği, AR/VR, etkileşimli medya

İş Yükü Yerleştirme ve Computation Offloading

Edge mimarilerinde önemli tasarım kararlarından biri, bir görevin hangi işlem katmanında çalıştırılacağıdır. Computation offloading, cihazın bir görevi tamamen veya kısmen daha güçlü bir edge/fog ya da bulut düğümüne devretmesidir. Mach ve Becvar (2017) ile Mao vd. (2017), mobil edge computing bağlamında hesaplama ve iletişim kaynaklarının birlikte optimize edilmesinin temel sorunlardan biri olduğunu göstermektedir.

Offloading kararı yalnızca CPU hızına bakılarak verilemez. Uzak düğüm daha güçlü olsa bile verinin aktarım süresi ve iletişim enerjisi toplam maliyeti yükseltebilir. Benzer biçimde cihaz üzerinde çalışmak düşük ağ gecikmesi sağlasa da enerji tüketimi, termal sınırlamalar veya bellek yetersizliği nedeniyle sürdürülebilir olmayabilir. Pratik bir yerleştirme kararı aşağıdaki değişkenleri birlikte dikkate almalıdır:

- Gecikme bütçesi ve deadline
- Anlık ağ bant genişliği, RTT, jitter ve paket kaybı
- Cihaz, edge ve bulut işlem kapasitesi
- Görevin giriş/çıkış veri boyutu ve ara aktivasyon hacmi
- Enerji ve batarya seviyesi
- Modelin bellek ve hızlandırıcı gereksinimi
- Veri mahremiyeti, mevzuat ve veri yerelliği gereksinimleri
  Bulut/edge kullanım maliyeti ve servis yoğunluğu



Şekil 3. Gecikme hassasiyeti ve hesaplama gereksinimine göre kavramsal iş yükü yerleşimi.

Statik kurallar küçük sistemlerde yeterli olabilir; ancak ağ ve cihaz koşullarının sürekli değiştiği sistemlerde dinamik orkestrasyon daha uygundur. Örneğin görüntü sınıflandırma uygulaması güçlü Wi-Fi bağlantısında yakın MEC sunucusuna yüksek doğruluklu modeli aktarabilir, bağlantı zayıfladığında cihazda daha küçük bir model çalıştırabilir. Böylece sistem, doğruluk ile gecikme ve enerji arasında çalışma anında seçim yapabilir.

Cihaz Üzerinde Çıkarım ve Edge AI

Eğitim ve Çıkarımın Ayrıştırılması

Makine öğrenmesi sistemlerinde model eğitimi ve çıkarım farklı kaynak profillerine sahiptir. Eğitim; büyük veri, çok sayıda matris işlemi, geriye yayılım ve uzun çalışma süreleri nedeniyle çoğunlukla GPU/TPU kümeleri veya güçlü sunucular üzerinde gerçekleştirilir. Çıkarım ise eğitilmiş modelin yeni girdiler üzerinde tahmin üretmesidir. Edge AI'ın yaygın uygulama modeli, eğitimin merkezi veya güçlü altyapıda, çıkarımın ise kullanıcıya yakın cihazlarda yapılmasıdır (Chen & Ran, 2019). Son yıllarda cihaz üzerinde ince ayar ve kısmi eğitim de araştırılmakla birlikte, çıkarım hâlen edge sistemlerin en yaygın yapay zekâ iş yüküdür.

Cihaz üzerinde çıkarım yaklaşımının robotik sistemlerde de uygulanabildiği görülmektedir. Aksoy vd. (2022), Raspberry Pi tabanlı bir robotik sistemde kamera ve basınç sensörlerinden elde edilen verileri makine ve derin öğrenme modelleriyle yerel olarak işleyerek hareket ve kavrama kararlarının gerçek zamanlı olarak üretilmesini sağlamıştır. Cihaz üzerinde çıkarımın temel kazanımları düşük iletişim gecikmesi, çevrimdışı çalışma, ham verinin cihaz dışına çıkmaması ve bulut maliyetinin azalmasıdır. Buna karşılık CPU/GPU/NPU kapasitesi, RAM, depolama, güç bütçesi ve termal sınırlar model seçiminde belirleyicidir. Bu nedenle bulutta yalnızca doğruluk için optimize edilen bir modelin hiçbir değişiklik yapılmadan edge cihazına taşınması çoğu zaman uygun değildir.

Hafif Sinir Ağı Mimarileri

Mobil ve gömülü sistemler için tasarlanan ağ mimarileri, yüksek doğruluğu daha düşük işlem ve bellek maliyetiyle sağlamayı hedefler. MobileNet ailesi bu yaklaşımın önemli örneklerindendir. Howard vd. (2017), depthwise separable convolution kullanarak standart evrişimin işlem maliyetini azaltan MobileNet mimarisini önermiştir. MobileNetV2 ise inverted residual ve linear bottleneck yapılarıyla mobil ortamlar için bellek ve işlem verimliliğini geliştirmiştir (Sandler vd., 2018). EfficientNet, ağ derinliği, genişliği ve giriş çözünürlüğünü bileşik ölçekleme ile dengeli biçimde artırmayı amaçlayan başka bir verimlilik yaklaşımıdır (Tan & Le, 2019).

Edge uygulamalarında modelin FLOP veya parametre sayısı tek başına yeterli performans göstergesi değildir. Aynı model iki farklı hızlandırıcıda farklı kernel optimizasyonları, bellek bant genişliği ve desteklenen veri

tipleri nedeniyle çok farklı gerçek gecikmelere sahip olabilir. Bu nedenle model seçimi hedef donanım üzerinde ölçülmelidir.

Model Sıkıştırma: Nicemleme, Budama ve Bilgi Damıtma

Nicemleme (quantization), ağırlık ve aktivasyonların daha düşük bit genişlikleriyle temsil edilmesini sağlar. Jacob vd. (2018), ağırlık ve aktivasyonların 8 bit tam sayı biçimine dönüştürülmesinin 32 bit kayan noktalı temsile kıyasla bellek ayak izini yaklaşık dört kat azaltabildiğini ve mobil CPU'larda verimli integer-only çıkarım sağlayabildiğini göstermiştir. Nicemleme, eğitim sonrasında uygulanabileceği gibi quantization-aware training ile eğitim sırasında da hesaba katılabilir.

Budama (pruning), modelin katkısı düşük bağlantı, ağırlık, kanal veya bloklarının kaldırılmasıdır. Han vd. (2016), budama ve nicemlemeyi birleştiren Deep Compression yaklaşımıyla büyük sinir ağlarının depolama gereksinimini önemli ölçüde azaltmıştır. Bununla birlikte düzensiz sparsity her donanımda gerçek hızlanma sağlamaz. Edge donanımı açısından structured pruning, yani filtre/kanal/blok gibi donanımın kolay işleyebildiği yapıların kaldırılması çoğu zaman daha kullanışlıdır.

Bilgi damıtma (knowledge distillation), büyük bir öğretmen modelin bilgi ve çıktı dağılımını daha küçük bir öğrenci modele aktarmayı amaçlar. Hinton vd. (2015), büyük modeller veya model toplulukları tarafından öğrenilen bilginin daha küçük bir modelde temsil edilebileceğini göstermiştir. Edge AI açısından damıtma, model boyutu ile doğruluk arasında pratik bir denge kurmak için önemlidir.

Donanıma Duyarlı Tasarım

Donanıma duyarlı tasarımda hedef, teorik olarak en küçük modeli değil hedef cihaz üzerinde gecikme, enerji ve bellek açısından en uygun modeli seçmektir. Cai vd. (2020) tarafından önerilen Once-for-All yaklaşımı, tek bir süper ağdan farklı donanım kısıtlarına göre özelleştirilebilen alt ağlar elde etmeyi amaçlamaktadır. Bu yaklaşım, edge cihaz heterojenliğinin model geliştirme sürecine doğrudan dahil edilmesinin bir örneğidir.

Tablo 3. Edge AI için başlıca model optimizasyon teknikleri

Teknik	Temel amaç	Olası avantaj	Temel risk / sınırlılık
Hafif mimari	Modeli baştan verimli tasarlamak	Düşük FLOP, düşük bellek ve gecikme	Göreve özgü doğruluk tavanı
Nicemleme	Daha düşük bit genişliği	Model ve bellek küçülmesi, integer hızlandırma	Doğruluk kaybı ve operatör uyumluluğu
Budama	Gereksiz yapıların kaldırılması	Parametre ve işlem azalması	Sparsity her donanımda hızlanmayabilir
Bilgi damıtma	Büyük model bilgisini küçük modele aktarmak	Daha küçük modelle yüksek doğruluk	Ek eğitim süreci gerekir
Donanıma duyarlı NAS/OFA	Cihaza göre alt ağ seçmek	Gerçek gecikmeye göre özelleştirme	Arama/eğitim karmaşıklığı

Bölünmüş Çıkarım ve Edge-Cloud İşbirliği

Cihaz üzerinde çıkarım ile tamamen bulut tabanlı çıkarım arasında ara bir yaklaşım, modelin katmanlar arasında bölünmesidir. Split inference yaklaşımında sinir ağının ilk katmanları cihazda çalıştırılır, belirli bir ara katmanda oluşan aktivasyonlar edge veya bulut sunucusuna gönderilir ve kalan katmanlar uzakta yürütülür. Kang vd. (2017) tarafından geliştirilen Neurosurgeon, derin sinir ağlarının mobil cihaz ile bulut arasında dinamik bölümlenmesini; gecikme, enerji ve veri merkezi yükü bakımından inceleyen öncü sistemlerden biridir.

Bölünme noktasının seçimi kritik önemdedir. Ham girdi büyük, ara aktivasyon küçükse cihazda ilk katmanları çalıştırmak iletişim maliyetini azaltabilir. Ancak bazı ağlarda ara aktivasyon, ham girdiden daha büyük olabilir. Ayrıca cihaz tarafında ilk katmanları çalıştırmanın enerji maliyeti de hesaba katılmalıdır. Bu nedenle bölünme noktası sabit olmaktan çok bant genişliği ve cihaz yüküne göre uyarlanabilir.

Edge-cloud işbirliği yalnızca model bölmek anlamına gelmez. Model önbellekleme, uzman modellerin edge düğümlerine yerleştirilmesi, batch oluşturma, erken çıkış (early exit), kademeli kalite ve sonuç önbelleği gibi

teknikler de birlikte kullanılabilir. Böylece edge sistemi, uygulamanın kalite hizmet düzeyine göre farklı işlem yolları seçebilir.

Federated Learning ve Dağıtık Öğrenme

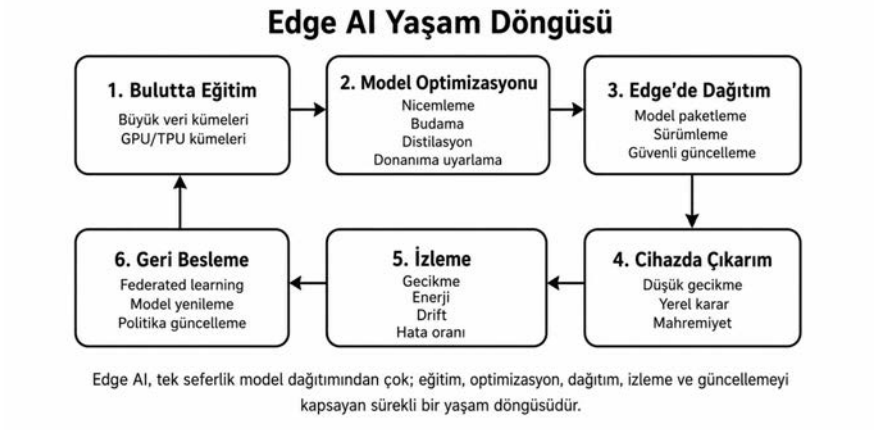
Edge AI'ın önemli bir uzantısı, verinin yalnızca çıkarım için değil öğrenme için de yerelde tutulmasıdır. Federated learning (FL), istemcilerin yerel verileri üzerinde model güncellemesi yapıp ham veriyi merkezi sunucuya göndermeden güncellemeleri birleştirdiği dağıtık öğrenme yaklaşımıdır. McMahan vd. (2017), FedAvg yaklaşımıyla merkezi olmayan ve dengesiz veri dağılımlarında iletişim açısından verimli ortak model öğrenimini göstermiştir. Bonawitz vd. (2019) ise üretim ölçeğinde federated learning sistemlerinin cihaz kullanılabilirliği, kesintili bağlantı, heterojen donanım ve orkestrasyon gibi pratik sorunlarını ayrıntılandırmıştır.

Federated learning'in temel avantajı, ham verinin cihazdan ayrılmasını zorunlu kılmamasıdır; ancak bu özellik tek başına tam gizlilik garantisi vermez. Güncelleme bilgilerinden veri çıkarımı, poisoning saldırıları, kötü niyetli istemciler ve non-IID veri dağılımı önemli araştırma problemleridir. Kairouz vd. (2021), federated learning'in mahremiyet, optimizasyon, iletişim ve sistem heterojenliği boyutlarında çok sayıda açık probleme sahip olduğunu göstermektedir.

Edge ortamında FL tasarımının bir başka sorunu enerji ve iletişim maliyetidir. Model güncellemelerinin sık gönderilmesi, özellikle bataryalı TinyML cihazlarında önemli maliyet yaratabilir. Bu nedenle istemci seçimi, güncelleme sıkıştırma, seyrek haberleşme ve yerel epoch sayısının ayarlanması gibi teknikler sistem seviyesinde ele alınmalıdır.

Edge AI Yaşam Döngüsü, İzleme ve Güncelleme

Üretim ortamında Edge AI, modeli bir kez cihaza kopyalayıp çalıştırmaktan daha geniş bir yaşam döngüsüdür. Modelin eğitilmesi, optimize edilmesi, donanım uyumluluğunun test edilmesi, güvenli biçimde dağıtılması, gerçek cihaz üzerindeki performansının izlenmesi ve gerektiğinde güncellenmesi gerekir. Ayrıca veri dağılımının zaman içinde değişmesi (concept drift) model doğruluğunu düşürebilir. Bu nedenle sistem performansı ile model performansı birlikte gözlemlenmelidir.



Şekil 4. Edge AI modelinin eğitimden saha güncellemesine uzanan yaşam döngüsü.

Sürümleme ve geri alma mekanizmaları edge ortamında özellikle önemlidir. Binlerce saha cihazına hatalı model dağıtılması, merkezi bir sunucudaki hatadan daha zor geri alınabilir. Güvenli güncelleme, model bütünlüğü doğrulama, staged rollout, A/B testleri ve telemetri; Edge AI MLOps süreçlerinin temel bileşenleri olarak değerlendirilmelidir.

Uygulama Alanları

Akıllı Şehirler

Akıllı şehirlerde trafik kameraları, hava kalitesi sensörleri, akıllı aydınlatma, toplu taşıma ve altyapı izleme sistemleri geniş coğrafi alana dağılmıştır. Edge katmanında trafik olayı veya nesne tespiti yapılabilir; fog/MEC katmanında belirli bir bölgedeki veriler birleştirilerek kavşak koordinasyonu gerçekleştirilebilir; bulutta ise şehir genelindeki uzun dönemli örüntüler analiz edilebilir. Bu dağılım, hem bant genişliğini kontrol altında tutar hem bölgesel kararları hızlandırır.

Endüstriyel IoT

Endüstriyel IoT uygulamalarında titreşim, sıcaklık, basınç, motor akımı, ses ve görüntü gibi sinyaller üretim ekipmanının durumunu izlemek için kullanılır. Ekipman arızasını gösterebilecek bir anomaliyi önce buluta gönderip sonra alarm üretmek yerine, yerel edge bilgisayarda çalışan model erken uyarı sağlayabilir. Bulut ise uzun dönemli kestirimci bakım analizi, filo karşılaştırması ve model eğitimi için kullanılabilir.

Endüstriyel ve otomotiv uygulamalarında görüntü tabanlı yapay zekâ yöntemleri, ekipman ve bileşen durumunun otomatik olarak

değerlendirilmesinde de kullanılabilmektedir. Gürfidan vd. (2023), araç lastiği yüzey görüntülerini ResNet50, DenseNet121, AlexNet ve CNN tabanlı modellerle analiz ederek lastiklerin yüzey aşınması ve dayanıklılık durumlarının makine öğrenmesi yoluyla sınıflandırılabileceğini göstermiştir. Benzer görsel denetim modellerinin üretim hattına veya veri kaynağına yakın edge düğümlerinde çalıştırılması, görüntülerin sürekli olarak merkezi buluta aktarılması yerine yerel ve düşük gecikmeli durum değerlendirmesine olanak sağlayabilir.

Gerçek Zamanlı Video Analitiği

Video, edge computing açısından en zorlu veri türlerinden biridir. Yüksek veri oranı ve derin öğrenme çıkarım maliyeti, bütün kamera akışlarının merkezi sisteme taşınmasını pahalı hâle getirir. Ananthanarayanan vd. (2017), coğrafi olarak dağıtık cloud-edge mimarisinin büyük ölçekli gerçek zamanlı video analitiği için önemli olduğunu belirtmektedir. Kamera yakınında nesne tespiti, yüz bulanıklaştırma veya olay algılama yapılarak yalnızca seçilmiş görüntüler ve metaveri üst katmana gönderilebilir.

Sağlık ve Giyilebilir Sistemler

Giyilebilir cihazlarda kalp hızı, hareket, oksijen düzeyi, uyku ve diğer fizyolojik sinyaller sürekli üretilebilir. Uçta çalışan modeller, belirli olayları hızlı biçimde tespit edebilir ve ham kişisel verinin sürekli aktarımını azaltabilir. Bununla birlikte sağlık uygulamalarında yalnızca düşük gecikme değil; klinik doğruluk, açıklanabilirlik, kalibrasyon, güvenlik, mevzuat ve yanlış alarm maliyeti de değerlendirilmelidir.

Otonom Sistemler ve Robotik

Otonom araç ve robotlarda çevresel algılama ile kontrol döngülerinin önemli kısmı yerel olmak zorundadır. Kamera, radar ve LiDAR verisinin uzak buluta sürekli gönderilmesi, bağlantı belirsizliği nedeniyle güvenli değildir. Buna karşılık harita güncellemesi, filo analizi ve büyük model eğitimi bulut kaynaklarında yürütülebilir. Böylece kritik kararlar edge'de, küresel öğrenme ve koordinasyon daha üst katmanlarda gerçekleştirilir.

Tablo 4. Başlıca edge/fog uygulama alanlarında gereksinimler

Uygulama	Öncelikli gereksinim	Edge görevi	Üst katman görevi
Akıllı şehir	Ölçek, bant genişliği, bölgesel tepki	Olay/nesne tespiti	Şehir geneli analitik ve planlama
Endüstriyel IoT	Düşük gecikme, güvenilirlik	Anomali algılama, yerel kontrol	Uzun dönem bakım modeli
Video analitiği	Çok yüksek veri hacmi	Filtreleme ve DNN çıkarımı	Arşivleme, küresel korelasyon
Sağlık/giyilebilir	Mahremiyet, enerji	Yerel sinyal analizi	Uzun dönem sağlık analitiği
Otonom sistem	Çok düşük gecikme, dayanıklılık	Algılama ve kontrol	Filo/harita/model güncelleme

Güvenlik, Mahremiyet ve Güvenilir Edge Intelligence

Veriyi kaynağa yakın işlemek mahremiyet açısından olumlu görünse de edge mimarisi saldırı yüzeyini azaltmak yerine çoğu zaman dağıtır. Merkezi veri merkezleri kontrollü fiziksel ortamlarda bulunurken edge cihazlar saha dolaplarında, araçlarda, fabrikalarda veya kullanıcıların fiziksel erişebildiği ortamlarda olabilir. Bu nedenle güvenlik yaklaşımı ağ iletişiminin şifrelenmesiyle sınırlı tutulamaz.

- Güvenli önyükleme ve donanım tabanlı güven kökü
- İmzalı firmware, yazılım ve model güncellemeleri
- Cihaz/servis kimlik doğrulaması ve en az ayrıcalık ilkesi
- Model bütünlüğü ve model hırsızlığına karşı koruma
- Adversarial örnekler ve veri zehirleme risklerinin izlenmesi
- Hassas verinin yerel saklanması, şifrelenmesi ve yaşam döngüsü yönetimi
- Edge düğümün fiziksel ele geçirilmesine karşı dayanıklılık

Güncel Edge Intelligence literatürü güvenilirliği yalnızca siber güvenlik boyutunda ele almamaktadır. Wang vd. (2025), güvenilir Edge Intelligence kavramını güvenlik ve mahremiyetin yanında güvenilirlik, şeffaflık ve sürdürülebilirlik boyutlarıyla incelemektedir. Bu yaklaşım, edge AI sisteminde model doğruluğu kadar kararın açıklanabilirliği, hizmet

sürekliliği, kaynak tüketimi ve sistem davranışının denetlenebilir olmasını da önemsemektedir.

Özellikle kritik uygulamalarda edge düğümü arızalandığında sistemin nasıl davranacağı önceden tasarlanmalıdır. Yerel fail-safe davranış, yedek düğüme geçiş, buluta geri dönüş (fallback) veya azaltılmış işlev modları hizmet sürekliliğinin parçalarıdır. Bu nedenle güvenilir Edge AI, algoritmik güvenilirlik ile dağıtık sistem güvenilirliğini birlikte ele almalıdır.

Enerji, Kaynak Kısıtları ve TinyML

Edge cihazların önemli bir kısmı enerji ve bellek bakımından kısıtlıdır. On-device inference ağ iletişimini azaltarak toplam enerji tüketimini düşürebilir; ancak büyük bir modelin sürekli çalıştırılması işlem enerjisini ve sıcaklığı artırabilir. Bu nedenle yerel işleme ile offloading arasında enerji açısından tek yönlü bir üstünlük yoktur. Enerji maliyeti; sensör, işlemci, bellek erişimi ve kablosuz haberleşmenin ortak sonucu olarak değerlendirilmelidir.

TinyML, makine öğrenmesi çıkarımını mikrodenetleyici sınıfı çok düşük güçlü cihazlara taşımayı amaçlayan yaklaşımı ifade eder. Warden ve Situnayake (2019), mikrodenetleyici üzerinde ses, hareket ve benzeri algılama görevlerinin küçük modellerle uygulanabileceğini göstermiştir. Han ve Siebert (2022) ise TinyML literatürünü sistematik biçimde inceleyerek ultra düşük güç, düşük maliyet ve yerel mahremiyetin alanın temel motivasyonları arasında bulunduğunu göstermektedir.

TinyML ortamında yalnızca model boyutunu küçültmek yeterli değildir. Flash/RAM sınırı, operatör desteği, sabit noktalı aritmetik, sensör örnekleme enerjisi ve real-time işletim sistemi görevleri birlikte düşünülmelidir. Uygun durumda wake-word, titreşim anomali tespiti, basit görüntü sınıflandırma ve hareket tanıma gibi görevler sürekli bulut bağlantısı olmadan gerçekleştirilebilir.

Edge AI Sistemlerinin Değerlendirilmesi

Edge AI çalışmalarında yalnızca doğruluk, F1-score veya AUC raporlanması sistemin gerçek kullanım uygunluğunu göstermeye yetmez. Model, gerçek hedef cihaz üzerinde ve temsil edici ağ koşullarında değerlendirilmelidir. Bir modelin doğruluğu daha yüksek olmasına rağmen P99 gecikmesi veya enerji tüketimi uygulama sınırlarını aşıyorsa pratikte daha kötü seçenek olabilir.

Tablo 5. Edge AI değerlendirmesinde önerilen ölçütler

Boyut	Örnek ölçütler	Neden önemlidir?
ML performansı	Accuracy, precision, recall, F1, AUC, hata türleri	Karar kalitesini gösterir
Gecikme	Ortalama, medyan, P95/P99, deadline ihlal oranı	Gerçek zamanlı uygunluğu gösterir
Throughput	İstek/sn, kare/sn, örnek/sn	Kapasiteyi gösterir
Kaynak	RAM/Flash, CPU/GPU/NPU kullanımı	Cihaz uygunluğunu gösterir
Enerji	J/inference, ortalama güç, batarya ömrü	Saha sürdürülebilirliğini gösterir
Ağ	Gönderilen bayt, RTT, paket kaybı, jitter	Offloading ve bant genişliği maliyetini gösterir
Güvenilirlik	Uptime, hata kurtarma, başarısız inference	Hizmet sürekliliğini gösterir
Mahremiyet/güvenlik	Veri yerelliği, şifreleme, saldırı dayanımı	Risk ve uyumluluğu gösterir

Karşılaştırmalar aynı veri kümesi ve aynı donanım koşullarında yapılmalı; mümkünse cihazın ısınması ve uzun süreli yük altındaki davranışı da ölçülmelidir. Bir edge hızlandırıcının ilk birkaç inference süresi ile termal dengeye ulaştıktan sonraki süreleri farklı olabilir. Bu nedenle tekrarlı ölçüm, güven aralığı veya standart sapma ve kuyruk yüzdelikleri raporlamak sonuçların güvenilirliğini artırır.

Güncel Eğilimler ve Gelecek Araştırma Yönleri

Edge-Cloud Continuum ve Akıllı Orkestrasyon

Gelecek edge sistemlerinin sabit katmanlardan çok dinamik bir kaynak sürekliliği biçiminde çalışması beklenmektedir. İş yükleri; gecikme, maliyet, enerji, mahremiyet, karbon yoğunluğu ve anlık kaynak kullanılabilirliği gibi ölçütlere göre cihaz, edge, MEC ve cloud arasında taşınabilir. Bu problem, klasik görev zamanlamanın ötesinde çok amaçlı ve bağlama duyarlı orkestrasyon gerektirir.

5G/6G ve MEC

ETSI MEC mimarisi, telekom erişim ağına yakın bilişim kaynaklarının uygulamalara sunulmasını standartlaştıran önemli çerçevelerden biridir. 2024 tarihli ETSI GS MEC 003 V3.2.1, MEC platformu, uygulamalar, yönetim ve orkestrasyon işlevleri arasındaki referans mimariyi güncel biçimde tanımlamaktadır (ETSI, 2024). 5G/6G ile edge entegrasyonunda radyo ağı bilgisi, kullanıcı hareketliliği ve servis yerleşimi kararlarının birlikte ele alınması beklenmektedir.

Güvenilir ve Sürdürülebilir Edge Intelligence

Edge Intelligence olgunlaştıkça araştırma odağı yalnızca daha hızlı çıkarımdan güvenilir, denetlenebilir ve sürdürülebilir yapay zekâya doğru genişlemektedir. Wang vd. (2025), güvenilir Edge Intelligence için güvenlik, güvenilirlik, şeffaflık ve sürdürülebilirliği birlikte ele alan çok katmanlı bir çerçeve önermektedir. Bu eğilim, enerji/karbon maliyeti ve açıklanabilirliğin gelecekte edge performans ölçütlerine daha fazla dahil olacağını göstermektedir.

Uçta Büyük Dil Modelleri

Büyük dil modellerinin (LLM) yerel veya edge ortamında çalıştırılması, Edge AI'ın yeni araştırma alanlarından biridir. Büyük modellerin bellek ve işlem gereksinimleri tam cihaz içi kullanımı zorlaştırsa da nicemleme, KV-cache optimizasyonu, model parçalama, uzman model yerleştirme ve MEC tabanlı hizmetler bu sınırı genişletmektedir. Qu vd. (2025), mobile edge intelligence yaklaşımını LLM önbellekleme, dağıtım, eğitim ve çıkarım boyutlarıyla incelemekte; edge yerleşiminin gecikme ve mahremiyet açısından bulut ile cihaz arasında ara bir çözüm sunduğunu belirtmektedir.

Uçta LLM kullanımında geleneksel Edge AI'dan farklı olarak bellek kapasitesi ve bellek bant genişliği daha kritik hâle gelir. Token üretim gecikmesi, time-to-first-token, token/s, model yükleme süresi ve enerji/tok gibi ölçütler önem kazanır. Ayrıca istemlerin hassas veri içerebilmesi, yerel çıkarımın mahremiyet avantajını güçlendirebilir; ancak model güvenliği ve güncelleme yönetimi daha karmaşık hâle gelir.

Donanım-Yazılım Ortak Tasarımı

Edge AI sistemlerinde CPU, GPU, DSP, NPU ve FPGA gibi heterojen hızlandırıcılar yaygınlaşmaktadır. Gelecek araştırmalarda model mimarisi, derleyici, bellek düzeni, veri tipi ve hızlandırıcı zamanlamasının birlikte

optimize edilmesi beklenmektedir. Bu yaklaşım, aynı modelin farklı donanımlarda çok farklı gerçek performansa sahip olduğu gerçeğini doğrudan tasarım sürecine dahil eder.

Tartışma

Edge computing'in en önemli katkısı, hesaplamayı fiziksel olarak dağıtmak değil, uygulamanın zaman, ağ ve veri yerelliği gereksinimlerini mimarinin birincil tasarım değişkenleri hâline getirmektir. Bu nedenle edge dönüşümü yalnızca bir sunucunun tesise kurulmasıyla tamamlanmaz. Uygulama veri akışının yeniden tasarlanması, ham verinin hangi noktada azaltılacağı, kararın hangi katmanda üretileceği ve bulutun hangi işlevleri sürdüreceği açık biçimde belirlenmelidir.

Fog computing ise bu dağıtık yapıya hiyerarşi ve bölgesel koordinasyon ekler. Kavramların literatürde kısmen örtüşmesi, pratik sistem tasarımında bir sorun olarak görülmemelidir. Asıl önemli olan cihaz-edge-fog-cloud arasında işlevlerin tanımlanmasıdır. Örneğin 'edge' sözcüğünün kullanılması sistemin otomatik olarak düşük gecikmeli olduğu anlamına gelmez; uygulamanın uçtan uca ölçülmesi gerekir.

Edge AI açısından benzer bir nokta model doğruluğu ile sistem performansının ayrılmaz oluşudur. Bulut ortamında yüzde birlik doğruluk artışı anlamlı olabilirken, aynı artış modeli edge cihazda deadline dışına çıkarıyorsa sistem hedefini bozabilir. Bu nedenle Edge AI optimizasyonu çok amaçlıdır: doğruluk, P99 gecikme, enerji, model boyutu, bellek, ağ trafiği, mahremiyet ve güvenilirlik birlikte değerlendirilmelidir.

Son olarak edge ve cloud arasında keskin bir ikilik yerine dinamik süreklilik yaklaşımı giderek daha gerçekçidir. Kritik ve hızlı kararlar veriye yakın; ağır analitik ve küresel öğrenme merkezi kaynaklarda; bölgesel koordinasyon ise fog/MEC katmanında yürütülebilir. Ağ veya kaynak koşulu değiştiğinde bu dağılımın dinamik olarak yeniden düzenlenmesi, yeni nesil akıllı dağıtık sistemlerin temel araştırma problemlerinden biri olacaktır.

SONUÇ VE ÖNERİLER

IoT ve yapay zekâ destekli siber-fiziksel uygulamaların yaygınlaşması, verinin yalnızca nerede depolandığını değil nerede ve ne kadar sürede işlendiğini de bilişim mimarisinin temel sorusu hâline getirmiştir. Merkezi bulut bilişim yüksek hesaplama kapasitesi ve küresel ölçek sağlar; ancak gecikme, bant genişliği, bağlantı sürekliliği ve veri mahremiyeti

gereksinimleri belirli iş yüklerinin veri kaynağına daha yakın yürütülmesini gerekli kılar. Edge computing bu yerelliği sağlarken fog computing ve MEC, edge ile bulut arasında bölgesel ve hiyerarşik kaynaklar sunmaktadır.

Gerçek zamanlı sistemler açısından edge'in değeri yalnızca ortalama gecikmeyi düşürmesi değildir. Ağ yolunu kısaltması, yerel karar verme sağlaması ve bağlantı kesintilerinde belirli işlevleri sürdürebilmesi önemlidir. Bununla birlikte gerçek zaman garantisi; jitter, kuyruk gecikmesi, deadline, en kötü durum yürütme süresi ve kaynak planlama gibi ölçütlerin ayrıca ele alınmasını gerektirir.

Edge AI, yapay zekâ modellerini bu dağıtık sürekliliğin etkin bir bileşeni hâline getirmektedir. Hafif mimariler, nicemleme, budama, bilgi damıtma ve donanıma duyarlı tasarım, cihaz üzerinde çıkarımı mümkün kılarken split inference ve federated learning cihaz ile üst katmanlar arasında yeni işbirliği modelleri sunmaktadır. Başarılı bir Edge AI sistemi yalnızca yüksek doğruluklu değil, aynı zamanda gecikme, enerji, bellek, mahremiyet, güvenlik ve güncellenebilirlik bakımından da dengeli olmalıdır.

Güncel eğilimler, alanın MEC ve 5G/6G entegrasyonu, güvenilir ve sürdürülebilir Edge Intelligence, TinyML, edge-cloud continuum ve uçta büyük dil modelleri yönünde genişlediğini göstermektedir. Bu nedenle edge ve fog computing, bulut bilişimin yerine geçen teknolojilerden çok, bulutu gerçek zamanlı ve bağlama duyarlı hizmetlerle tamamlayan dağıtık bir hesaplama sürekliliğinin temel katmanları olarak değerlendirilmelidir.

KAYNAKÇA

Ananthanarayanan, G., Bahl, P., Bodík, P., Chintalapudi, K., Philipose, M., Ravindranath, L., & Sinha, S. (2017). Real-time video analytics: The killer app for edge computing. Computer, 50(10), 58–67. https://doi.org/10.1109/MC.2017.3641638

Aksoy, B., Özsoy, K., Yücel, M., Ekrem, Ö., & Salman, O. K. M. (2022). Motion control of the robot arm manufactured with a three-dimensional printer and hardness detection of objects. Bilişim Teknolojileri Dergisi, 15(3), 289-300.

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (pp. 13–16). Association for Computing Machinery. https://doi.org/10.1145/2342509.2342513

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H. B., Van Overveldt, T., Petrou, D., Ramage, D., & Roselander, J. (2019). Towards federated learning at scale: System design. Proceedings of Machine Learning and Systems, 1.

Buttazzo, G. (2024). Hard real-time computing systems: Predictable scheduling algorithms and applications (4th ed.). Springer. https://doi.org/10.1007/978-3-031-45410-3

Cai, H., Gan, C., Wang, T., Zhang, Z., & Han, S. (2020). Once for all: Train one network and specialize it for efficient deployment. In International Conference on Learning Representations (ICLR).

Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. Proceedings of the IEEE, 107(8), 1655–1674. https://doi.org/10.1109/JPROC.2019.2921977

European Telecommunications Standards Institute. (2024). Multi-access Edge Computing (MEC); Framework and reference architecture (ETSI GS MEC 003 V3.2.1). ETSI.

Gürfidan, R., Kilim, O., Yiğit, T., & Aksoy, B. (2023). Robustness Classification by Machine Learning from Vehicle Tire Surface Abrasions. *Gazi Journal of Engineering Sciences, 9*(4), 151–157. https://doi.org/10.30855/gmbd.0705S15

Han, H., & Siebert, J. (2022). TinyML: A systematic review and synthesis of existing research. In 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (pp. 269–274). IEEE. https://doi.org/10.1109/ICAIIC54071.2022.9722636

Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In International Conference on Learning Representations (ICLR).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. NIPS Deep Learning and Representation Learning Workshop. https://doi.org/10.48550/arXiv.1503.02531

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv. https://doi.org/10.48550/arXiv.1704.04861

Iorga, M., Feldman, L., Barton, R., Martin, M. J., Goren, N. S., & Mahmoudi, C. (2018). Fog computing conceptual model (NIST Special Publication 500-325). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.500-325

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2704–2713).

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., El Rouayheb, S., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., ... Zhao, S. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1–2), 1–210. https://doi.org/10.1561/2200000083

Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J., & Tang, L. (2017). Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. In Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems (pp. 615–629). Association for Computing Machinery. https://doi.org/10.1145/3037697.3037698

Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 19(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials, 19(4), 2322–2358. https://doi.org/10.1109/COMST.2017.2745201

McMahan, B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 54, 1273–1282.

Qu, G., Chen, Q., Wei, W., Lin, Z., Chen, X., & Huang, K. (2025). Mobile edge intelligence for large language models: A contemporary survey. IEEE Communications Surveys & Tutorials, 27(6), 3820–3860. https://doi.org/10.1109/COMST.2025.3527641

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510–4520).

Satyanarayanan, M. (2017). The emergence of edge computing. Computer, 50(1), 30–39. https://doi.org/10.1109/MC.2017.9

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning, 97, 6105–6114.

Wang, X., Wang, B., Wu, Y., Ning, Z., Guo, S., & Yu, F. R. (2025). A survey on trustworthy edge intelligence: From security and reliability to transparency and sustainability. IEEE Communications Surveys & Tutorials, 27(3), 1729–1757. https://doi.org/10.1109/COMST.2024.3446585

Warden, P., & Situnayake, D. (2019). TinyML: Machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers. O'Reilly Media.

Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, 107(8), 1738–1762. https://doi.org/10.1109/JPROC.2019.2918951

IOT GÜVENLİĞİ VE MAHREMİYET

IoT Cihaz Güvenliği, Ağ Saldırıları, Güvenli Yazılım, Kimlik Doğrulama, Şifreleme ve Veri Mahremiyeti

Recep Çolak*, Cem Deniz Kumral**

GİRİŞ

Nesnelerin İnterneti (Internet of Things-IoT), fiziksel cihazların sensörler, haberleşme teknolojileri ve yazılım bileşenleri aracılığıyla birbirleriyle ve uzak sistemlerle sürekli veri alışverişi gerçekleştirdiği çok katmanlı bir ekosistemdir. Akıllı evlerden endüstriyel otomasyona, sağlık uygulamalarından çevresel izlemeye kadar geniş bir kullanım alanına sahip olan bu yapı, sağladığı işlevsel avantajların yanında farklı güvenlik ve mahremiyet risklerini de beraberinde getirmektedir. IoT cihazlarının sınırlı işlem gücü, bellek ve enerji kaynakları, fiziksel olarak erişilebilir ortamlarda bulunmaları ve farklı haberleşme protokollerini kullanmaları, geleneksel bilgi sistemlerinden farklı güvenlik yaklaşımlarının geliştirilmesini gerekli kılmaktadır.[1]

IoT güvenliğinin etkili biçimde sağlanabilmesi için cihaz, ağ, yazılım, kimlik doğrulama, kriptografi ve veri mahremiyeti katmanlarının birlikte değerlendirilmesi gerekmektedir. Fiziksel müdahale ve donanımsal saldırılar cihaz düzeyinde tehdit oluştururken; ZigBee, BLE, LoRaWAN, MQTT ve yönlendirme protokollerine yönelik saldırılar haberleşme güvenliğini doğrudan etkileyebilmektedir. Benzer biçimde güvenli olmayan firmware, yetersiz güncelleme mekanizmaları ve yazılım geliştirme sürecindeki hatalar sistem genelinde önemli zafiyetler oluşturabilmektedir. Kimlik doğrulama ve yetkilendirme mekanizmaları güvenilir erişimin temelini oluştururken, kaynakları sınırlı cihazlara uygun kriptografik çözümler veri bütünlüğü ve gizliliğinin korunmasını sağlamaktadır. Bunun yanında IoT sistemlerinde sürekli üretilen verilerin yaşam döngüsü boyunca korunması, kullanıcı

* Öğr. Gör. Dr., Isparta Uygulamalı Bilimler Üniversitesi, Uzaktan Eğitim Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Isparta, recepcolak@isparta.edu.tr, ORCID: orcid.org/ 0000-0002-7119-6202
** Öğr. Gör. Dr., Isparta Uygulamalı Bilimler Üniversitesi, Uzaktan Eğitim Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Isparta, cemkumral@isparta.edu.tr, ORCID: orcid.org/ 0000-0002-1326-4537

mahremiyeti ve kişisel verilerin korunması açısından ayrı bir önem taşımaktadır.

Bu bölümde IoT güvenliği ve mahremiyeti, cihaz güvenliğinden ağ saldırılarına, güvenli yazılım geliştirmeden kimlik doğrulamaya, hafif sıklet kriptografiden veri gizliliğine kadar bütüncül bir yaklaşımla ele alınmaktadır. Böylece IoT sistemlerinde güvenliğin tek bir teknoloji veya koruma mekanizmasıyla sağlanamayacağı; farklı katmanlarda uygulanan tamamlayıcı güvenlik önlemlerinin birlikte değerlendirilmesi gerektiği ortaya konulmaktadır.

IOT SİSTEMLERİNDE CİHAZ GÜVENLİĞİ

Internet of Things (IoT) ekosisteminin en kritik ve aynı zamanda en zayıf halkası, verinin ilk defa üretildiği uç cihazlardır (Shukla vd., 2026). Sanayide, tıbbi cihazlarda, akıllı şehirlerde, akıllı evlerde ve daha birçok alanda yaygın kullanım alanı olan IoT cihazlar, bilgisayar sistemlerinden tamamen farklı koşullarda ve farklı güvenlik tehditleri altında çalışmaktadır (Avcı, 2022). Bu bölümde; cihaz düzeyindeki donanımsal kısıtlar, fiziksel müdahale riskleri, yan kanal saldırıları ve donanım tabanlı güvenlik çözümleri teknik boyutuyla incelenmektedir.

Donanımsal Kapasite Kısıtları

IoT cihazları doğası gereği, maliyeti, boyutu ve enerji tüketimi optimize edilmiş mikrodenetleyiciler (MCU) ile çalışır (Gbadega vd., 2026). Bu donanımsal kısıtlar geleneksel katmanlı güvenlik mekanizmalarının uygulanmasını zorlaştırmaktadır (Latif vd., 2026). Bu donanımsal kısıtlar:

- İşlemci Gücü (Processing Power): IoT cihazlarda klasik bilgisayar sistemlerine göre çok daha düşük frekanslarda (MHz seviyesinde) çalışan işlemciler kullanılır. Bu işlemciler karmaşık asimetrik kriptografik algoritmaları (örneğin 4096-bit RSA vb.) çalıştırmakta yetersiz kalır.
- Bellek Kısıtları (RAM / Flash): Kilobayt seviyesindeki SRAM ve megabayt altı Flash bellekler, yoğun güvenlik protokol verilerini, kapsamlı antivirüs/IDS yazılımlarını veya büyük anahtar depolama alanlarını saklamaya yetmez.
- Güç Tüketimi (Power Consumption): IoT cihazları genellikle şebekeye bağlı olmadan pil gibi kapasitesi oldukça düşük bir enerji kaynağı ile çalışmaktadır. Bu durum güç tüketiminin optimize edilmesini zorunlu kılar. Pil gücünü optimize edebilmek için hem işlem kapasitesi kısıtlanır hem de sürekli kontrol edilmesi gereken işlemler çalıştırılamaz.

- Yazılım Güncelleme Kısıtları: Yukarıda bahsedilen fiziksel ve yapısal güçlükler nedeniyle IoT cihazlar donanımsal olarak kısıtlanmış şekilde üretilirler. Bu nedenle cihazlarda genellikle RTOS (Real-Time Operating System) veya hiçbir işletim sistemi olmaksızın doğrudan bare-metal kodlar çalışır. Donanımsal kısıtlardan dolayı karmaşık erişim kontrol mekanizmaları (DAC/MAC) desteklenmeyebilir.

Fiziksel Müdahale (Tampering) Tehditleri

IoT cihazları doğası gereği genelde denetimsiz ve açık alanlarda kullanılmaktadır. Bu durum cihazları, doğrudan saldırganların fiziksel müdahalesine karşı savunmasız bırakmaktadır. Bu cihazlar saldırganlar tarafından kolayca müdahale edilebilecek konumlarda yer almaktadır. Bu durum cihazların fiziksel olarak kurcalanmasına sebep olabilmektedir (Sharma vd., 2025). Bu da aşağıdaki riskleri ortaya çıkarır:

JTAG ve UART arayüzü İstismarı

Birçok IoT cihazında geliştirme ve test amacıyla JTAG (Joint Test Action Group) ve UART (Universal Asynchronous Receiver-Transmitter) gibi seri haberleşme arayüzleri kullanılmaktadır. Bu arayüzler mikrodenetleyici veya SoC birimlerine doğrudan erişim sağlayarak firmware yükleme, hata ayıklama ve donanım testi işlemlerini kolaylaştırmaktadır. Ancak seri üretim sonrasında devre dışı bırakılmamaları veya fiziksel olarak korunmamaları ciddi bir güvenlik açığı oluşturabilmektedir (Pearson vd., 2020).

Saldırgan, PCB üzerindeki JTAG/UART pinlerini veya test noktalarını tespit ederek özel donanımlar yardımıyla bu arayüzlere erişebilir. Böyle bir erişim sonucunda:

- Firmware dışa aktarımı: Flash bellekteki firmware okunarak tersine mühendislik yoluyla gömülü kimlik bilgileri, şifreleme anahtarları veya kodlama hataları analiz edilebilir.
- Bellek ve kayıt erişimi: JTAG üzerinden işlemci kayıtları ve bellek alanları incelenerek sistemin çalışma yapısı hakkında bilgi elde edilebilir.
- Yetkili sistem erişimi: UART üzerinden bootloader veya shell oturumuna erişilmesi durumunda, zayıf ya da varsayılan kimlik doğrulama mekanizmaları saldırgana yüksek yetkili erişim sağlayabilir.

Bu saldırılar fiziksel erişim gerektirse de tedarik zinciri, çalınan cihazlar ve iç tehdit senaryolarında önemli risk oluşturmaktadır. Bu nedenle JTAG/UART arayüzlerinin üretim sonrasında devre dışı bırakılması, fiziksel olarak korunması veya kimlik doğrulama mekanizmalarıyla sınırlandırılması önerilmektedir (Pearson vd., 2020).

Probing ve bus grabbing

IoT cihazlarındaki gömülü sistemlerde işlemci; EEPROM, flash bellek, sensörler ve diğer çevresel birimlerle SPI, I2C ve CAN gibi seri veri yolları üzerinden haberleşmektedir (Ojo vd., 2018). Bu veri yollarında şifreleme veya kimlik doğrulamanın bulunmaması, iletilen verilerin fiziksel erişim yoluyla dinlenmesine ve probing/bus grabbing saldırılarına açık hâle gelmesine neden olabilmektedir (Hernández-Serrano vd., 2017).

Saldırgan veri yolu hatlarına prob veya lojik analizör bağlayarak:

- Sensör verilerini doğrudan okuyabilir.
- Bellek trafiğini izleyerek kimlik bilgileri veya şifreleme anahtarlarını tespit edebilir.
- Komut ve kontrol mesajlarını analiz ederek sistemin çalışma mantığını çıkarabilir ve veri enjeksiyonu gibi saldırılara zemin hazırlayabilir.
- Özellikle CAN tabanlı araç ve endüstriyel sistemlerde, sahte mesaj enjeksiyonu ile fiziksel aktüatörlerin yanlış yönlendirilmesine neden olabilir (Buscemi vd., 2023).

Bu saldırılar fiziksel erişim gerektirse de çoğu zaman cihaz üzerinde kalıcı değişiklik oluşturmadan gerçekleştirilebildiğinden tespit edilmeleri zordur. Koruma amacıyla veri yolu düzeyinde şifreleme, hassas PCB hatlarının iç katmanlarda konumlandırılması ve kurcalama algılama mekanizmalarının kullanılması önerilmektedir.

Bellek dökümü (Memory dumping)

Bazı IoT cihazlarında hassas veriler, işlemciden bağımsız harici flash bellek veya EEPROM bileşenlerinde saklanmaktadır (De vd., 2017). Yazılımsal güvenlik mekanizmaları çoğunlukla ağ, uygulama veya API gibi normal erişim yollarını koruduğundan, belleğe doğrudan fiziksel erişim bu kontrollerin dışında kalabilmektedir.

Cihaza fiziksel erişim sağlayan saldırgan, bellek yongasını sökerek veya uygun test klipsleri ve programlayıcılar kullanarak ham bellek içeriğini dışa aktarabilir. Elde edilen bellek dökümünde:

- Yapılandırma verileri ve cihaz kimlik bilgileri,
- Sabit kodlanmış API anahtarları veya erişim token'ları,
- Şifreleme anahtarları ve sertifikalar

tespit edilebilir. Ortak anahtarların birden fazla cihazda kullanılması durumunda bu risk yalnızca tek cihazla sınırlı kalmayarak tüm cihaz filosunu etkileyebilir.

Bellek dökümü saldırılarının en önemli özelliği, kimlik doğrulama ve erişim kontrolü gibi çalışma zamanı güvenlik mekanizmalarını atlayarak veriye doğrudan erişim sağlamasıdır. Bu nedenle hassas bilgilerin HSM veya güvenli element içinde saklanması, harici belleğin şifrelenmesi ve her cihaz için benzersiz kriptografik anahtarların kullanılması önerilmektedir.

Donanım Tabanlı Güvenlik Mekanizmaları

Önceki bölümlerde ele alınan JTAG/UART istismarı, prob'lama, bellek dökümü gibi tehditler göz önüne alındığında, salt yazılımsal güvenlik önlemlerinin (şifreleme protokolleri, erişim kontrolü, kimlik doğrulama vb.) fiziksel erişime sahip bir saldırgana karşı yeterli koruma sağlayamadığını açıkça ortaya koymaktadır. Çünkü yazılım, nihayetinde donanım üzerinde çalışır. Donanıma doğrudan erişimi olan bir saldırgan, yazılım katmanındaki tüm korumaları teorik olarak atlayabilir. Bu sorunu temelden çözmek amacıyla geliştirilen yaklaşım, güvenliğin donanımın kendisinde, değiştirilemez ve sökülemez bir temelde (Root of Trust – Güven Kökü) inşa edilmesidir. RoT (Zhao vd., 2014), bir sistemdeki tüm güvenlik işlemlerinin (önyükleme doğrulama, anahtar yönetimi, şifreleme) dayandığı, kendisi doğrulanmaya ihtiyaç duymayan, doğası gereği güvenilir kabul edilen donanımsal temel bileşeni ifade eder.

Donanım güvenlik modülü (HSM) ve güvenli element (SE)

HSM ve SE, kriptografik anahtarların üretilmesi, saklanması ve kullanılması işlemlerini ana işlemciden (main CPU/SoC) tamamen ayrı, kendi başına korumalı bir donanım sınırı içinde gerçekleştiren özel entegre devrelerdir. Bu bileşenlerin temel güvenlik avantajı, anahtarların hiçbir zaman ana işlemcinin belleğine veya sistem veri yoluna açık (plaintext) biçimde çıkmamasıdır — şifreleme/deşifreleme işlemleri SE/HSM'nin kendi

içinde yürütülür ve yalnızca işlem sonucu dışarı iletilir. Ayrıca bu modüller genellikle fiziksel kurcalamayı algılayan (tamper-detection) sensörlerle donatılmıştır; bir sıcaklık, voltaj veya paket bütünlüğü anomalisi tespit edildiğinde, içerdiği anahtarları otomatik olarak silecek (zeroization) şekilde tasarlanabilirler. Bu özellik, daha önce bahsedilen bellek dökümü ve prob'lama saldırılarına karşı doğrudan bir savunma hattı oluşturabilir. Özellikle akıllı sayaçlar, ödeme cihazları ve kritik altyapı sensörlerinin korunmasında tercih edilen yöntemlerdir.

PUF (Physical unclonable function – Fiziksel kopyalanamaz fonksiyon)

PUF, yarı iletken üretim sürecinin doğası gereği barındırdığı, tasarımcının bile kontrol edemediği mikroskobik düzeydeki rastgele fiziksel varyasyonlardan (transistör eşik gerilimi farkları, kapı gecikme süreleri, kablolama direnç farkları gibi) faydalanan bir mekanizmadır. Aynı üretim hattından, aynı maskeyle üretilen iki çip bile bu mikroskobik farklılıklar nedeniyle birbirinden ayırt edilebilir, benzersiz bir "dijital parmak izi" üretir. PUF'un güvenlik açısından en kritik özelliği, bu benzersiz değerin cihaz kapatıldığında ortadan kalkması ve yalnızca cihaz her açıldığında (challenge-response mekanizmasıyla) yeniden üretilmesidir. Bu sayede kriptografik anahtar, flash bellek gibi kalıcı bir ortamda hiçbir zaman saklanmaz; bu da anahtarın söküm, bellek dökümü veya prob'lama yoluyla doğrudan okunma riskini ortadan kaldırır.

TPM (Trusted platform module – Güvenilir platform modülü)

TPM (Trusted Platform Module), Trusted Computing Group (TCG) tarafından standartlaştırılmış ve önyükleme sürecinin bütünlüğünü donanımsal olarak doğrulayan bir güvenlik yongasıdır. Sistem açılırken bootloader, işletim sistemi çekirdeği ve kritik sürücüler gibi bileşenlerin kriptografik özetleri alınarak Platform Configuration Registers (PCR) içerisinde saklanır.

Secure Boot, bir sonraki bileşenin dijital imzasını doğrulayarak yetkisiz veya değiştirilmiş yazılımın yüklenmesini engeller. Measured Boot ise önyüklemeyi durdurmak yerine her aşamanın hash değerini kaydederek sistemin beklenen durumda olup olmadığının uzaktan doğrulanmasına imkân sağlar.

TPM içerisindeki anahtarlar donanımsal olarak korunur ve yalnızca uygun PCR değerleri sağlandığında kullanılabilir. Kaynak ve donanım gereksinimleri nedeniyle klasik düşük güçlü IoT cihazlarından çok, daha güçlü merkezi IoT bileşenlerinde tercih edilebilecek bir çözümdür. IoT cihazlarındaki temel güvenlik açıkları ve bunlara yönelik önlemler Tablo 1'de özetlenmiştir.

Tablo 1. IoT cihazlarında güvenlik açıkları ve çözüm önerileri

Tehdit / Zafiyet Alanı	Saldırı Yöntemi	Etki Seviyesi	Donanımsal / Mimari Çözüm Önerisi
Açık Debug Arayüzleri	JTAG/UART bağlantısı ile bellenim okuma ve yetki yükseltme	Kritik	Üretim aşamasında JTAG/UART pinlerinin eFuse ile fiziksel olarak kapatılması, debug kilitlerinin aktif edilmesi
Harici Bellek Dinleme	SPI/I2C veri yollarına prob bağlayarak şifresiz veri sızdırma	Yüksek	Şifreli bellek denetleyicileri (In-Line Memory Encryption) kullanımı, SoC entegre dahili bellek tercihi
Anahtar Depolama Riski	Flash bellek dökümü (dumping) ile sabit anahtarların ele geçirilmesi	Kritik	Kriptografik anahtarların PUF veya Secure Element (SE) içinde saklanması
Kötü Amaçlı Yazılım / Yazılım Değişimi	Sahte veya manipüle edilmiş bellenim (firmware) yüklenmesi	Kritik	Secure Boot ve RoT tabanlı dijital imza doğrulaması
Kapasite Kısıtları (CPU/RAM)	Ağır şifreleme yükü nedeniyle güvenlik protokollerinin devre dışı bırakılması	Orta	Hafif Sıklet Kriptografi (Lightweight Cryptography — ASCON vb.) ve donanımsal AES/ECC hızlandırıcılar

IOT AĞ SALDIRILARI VE İLETİŞİM ZAFİYETLERİ

IoT sistemleri, farklı özelliklere sahip çok sayıda cihazın kablolu ve kablosuz teknolojiler üzerinden haberleştiği heterojen ağ yapılarıdır. Düşük güç ve sınırlı işlem kapasitesi, haberleşme protokollerinde düşük ek yük gerektirirken bazı geleneksel güvenlik mekanizmalarının doğrudan uygulanmasını güçleştirmektedir. Kablosuz iletişim, geniş coğrafi dağılım ve farklı protokollerin birlikte kullanılması saldırı yüzeyini artırmakta; MQTT, ZigBee, Bluetooth/BLE ve LoRaWAN gibi protokoller farklı güvenlik özellikleri ve zafiyetler barındırmaktadır (Dritsas ve Trigka, 2025).

IoT ağ saldırıları veri gizliliği, bütünlük, kimlik doğruluğu ve hizmet sürekliliğini etkileyebilmektedir. Trafiğin dinlenmesi, paketlerin değiştirilmesi veya tekrar gönderilmesi, sahte cihazların ağa katılması ve ağ kaynaklarının tüketilmesi bu tehditlere örnektir. Bu nedenle IoT ağ güvenliği; şifreleme yanında kimlik doğrulama, anahtar yönetimi, mesaj

tazeliği kontrolü, güvenli yönlendirme ve anomali tespiti gibi tamamlayıcı mekanizmalara dayanmalıdır (Dritsas ve Trigka, 2025).

ZigBee ve BLE Tabanlı Ağlarda Güvenlik

ZigBee, IEEE 802.15.4 tabanlı, düşük güç ve düşük veri hızlı IoT cihazları için kullanılan bir kablosuz haberleşme teknolojisidir. Akıllı evler, sensör ağları ve otomasyon uygulamalarında yaygın olarak kullanılmaktadır. AES-128 tabanlı güvenlik mekanizmaları bulunmasına rağmen anahtar yönetimi, ağa katılım süreci ve cihazlar arası güven ilişkisi farklı saldırı yüzeyleri oluşturabilmektedir. ZigBee ağlarında anahtar ele geçirme, trafik analizi, cihaz taklidi ve tekrar oynatma saldırıları öne çıkmaktadır (Zohourian vd., 2023; Dritsas ve Trigka, 2025).

Tekrar oynatma (Replay) saldırısında saldırgan daha önce geçerli bir cihaz tarafından gönderilen paketi kaydedip yeniden iletir. Alıcı taraf mesajın güncelliğini doğrulamazsa eski bir komutu yeniymiş gibi kabul edebilir. Bu nedenle paket sıra numaraları, çerçeve sayaçları ve mesaj tazeliği kontrolleri replay saldırılarına karşı temel önlemler arasında yer almaktadır (Zohourian vd., 2023).

Bluetooth Low Energy (BLE) ise giyilebilir cihazlar, sağlık sensörleri ve kısa menzilli IoT uygulamalarında yaygın biçimde kullanılmaktadır. Düşük enerji tüketimi avantaj sağlarken güvenlik seviyesi protokol sürümüne, eşleştirme yöntemine ve yapılandırmaya bağlıdır. Özellikle zayıf eşleştirme süreçleri dinleme, yetkisiz eşleştirme, replay ve Ortadaki Adam saldırılarına zemin hazırlayabilmektedir. Bu nedenle BLE ve Bluetooth Mesh sistemlerinde eşleştirme ve bağlantı güvenliğinin doğru yapılandırılması kritik önem taşımaktadır (Lacava vd., 2022).

Ortadaki Adam ve Tekrar Oynatma Saldırıları

Ortadaki Adam (Man-in-the-Middle-MitM) saldırısında saldırgan, iki meşru taraf arasındaki iletişime dâhil olarak trafiği dinleyebilir, paketleri değiştirebilir veya sahte mesajlar üretebilir. Zayıf kimlik doğrulama, güvensiz eşleştirme ve şifrelenmemiş bağlantılar bu saldırıları kolaylaştırmaktadır. BLE ve MQTT tabanlı sistemlerde MitM saldırıları önemli iletişim tehditleri arasında değerlendirilmektedir (Lacava vd., 2022; Dritsas ve Trigka, 2025).

MQTT, IoT cihazları ile sunucu veya bulut servisleri arasında kullanılan hafif sıklet bir yayınla–abone ol (publish-subscribe) protokolüdür. MQTT

broker, yayıncı cihazlardan aldığı mesajları ilgili abonelere iletir. TLS veya uygun kimlik doğrulama kullanılmadığında MQTT sistemleri MitM, replay ve yetkisiz erişim saldırılarına açık hâle gelebilmektedir. TLS, istemci/broker kimlik doğrulaması ve erişim kontrolü bu riskleri azaltırken, oluşturduğu ek işlem maliyeti kaynakları kısıtlı cihazlarda dikkate alınmalıdır (Dritsas ve Trigka, 2025).

DoS ve DDoS Saldırıları

Hizmet Dışı Bırakma (Denial of Service-DoS) saldırılarında amaç, cihazın veya ağ servisinin işlem, bellek, bağlantı ya da enerji kaynaklarını tüketerek meşru erişimi engellemektir. Saldırının çok sayıda ele geçirilmiş cihaz üzerinden eş zamanlı yürütülmesi ise Dağıtık Hizmet Dışı Bırakma (DDoS) olarak adlandırılmaktadır. IoT cihazlarının sınırlı kaynaklara sahip olması, özellikle uç cihazlar ve ağ geçitlerini bu saldırılara karşı daha hassas hâle getirmektedir (Dritsas ve Trigka, 2025).

MQTT gibi broker tabanlı yapılarda yoğun bağlantı veya mesaj istekleri broker kaynaklarını tüketebilirken, kablosuz IoT sistemlerinde gereksiz trafik cihazların enerji ve iletişim kapasitesini azaltabilmektedir. LoRaWAN üzerinde de replay tabanlı seçici DoS, mesaj değiştirme ve batarya tüketimine yönelik saldırılar gösterilmiştir (Yang vd., 2018). Bu durum, IoT ağlarında erişilebilirliğin yalnızca yüksek hacimli DDoS saldırılarıyla değil, sınırlı cihaz kaynaklarını hedefleyen düşük hacimli saldırılarla da bozulabileceğini göstermektedir.

LoRaWAN Haberleşmesindeki Zafiyetler

LoRaWAN, düşük güç tüketimiyle uzun mesafeli haberleşmeye ihtiyaç duyan sensör ağları için kullanılan LPWAN teknolojilerinden biridir. Akıllı tarım, çevresel izleme, lojistik ve akıllı şehir uygulamaları gibi geniş alanlara dağıtılmış sistemlerde kullanılabilmektedir. LoRaWAN güvenliğinde cihaz aktivasyonu, oturum anahtarları, mesaj sayaçları ve mesaj doğrulama mekanizmaları önemli rol oynamaktadır. Bu bileşenlerin hatalı yapılandırılması veya uygun biçimde yönetilmemesi replay ve mesaj manipülasyonu gibi saldırılara zemin hazırlayabilmektedir (Yang vd., 2018).

Yang vd. (2018) tarafından gerçekleştirilen güvenlik analizinde LoRaWAN üzerinde beş farklı saldırı senaryosu ele alınmış; replay saldırısının belirli cihazlara yönelik seçici DoS'a dönüşebildiği, mesaj içeriklerinin değiştirilebildiği, teslim bilgileri üzerinde sahtecilik yapılabildiği ve cihazların batarya kaynaklarının tüketilebildiği gösterilmiştir. Bu nedenle LoRaWAN uygulamalarında cihaz başına güvenli

anahtar üretimi, çerçeve sayaçlarının doğru yönetimi, güvenli aktivasyon süreçleri ve olağandışı haberleşme davranışlarının izlenmesi önem taşımaktadır (Yang vd., 2018).

IoT Yönlendirme Saldırıları

Düşük güçlü ve kayıplı ağlarda kullanılan RPL (Routing Protocol for Low-Power and Lossy Networks) gibi yönlendirme protokolleri, sınırlı kaynaklara sahip IoT düğümleri arasında verimli veri aktarımı sağlamak üzere geliştirilmiştir. Ancak yönlendirme bilgilerinin kötü niyetli bir düğüm tarafından değiştirilmesi ağ topolojisini ve veri akışını etkileyebilmektedir. IoT yönlendirme güvenliği üzerine yapılan çalışmalar wormhole, sinkhole ve DoS saldırılarını RPL tabanlı ağlarda önemli tehditler arasında değerlendirmektedir (Kumar vd., 2023).

Sinkhole saldırısında kötü niyetli düğüm kendisini ağ açısından avantajlı bir yönlendirme noktası olarak göstererek trafiğin önemli bölümünü üzerine çekmeye çalışırken, wormhole saldırısında ağın birbirinden uzak noktaları arasında saldırgana ait sahte veya beklenmeyen bir iletişim yolu oluşturulabilmektedir. Bu tür yönlendirme manipülasyonları paketlerin düşürülmesine, gecikmelerin artmasına veya verilerin saldırgan tarafından kontrol edilen güzergâhlardan geçirilmesine neden olabilir. RPL güvenliğinde kriptografik doğrulamanın yanında güven tabanlı yönlendirme, eşik tabanlı denetim ve saldırı tespit mekanizmaları gibi yaklaşımların da değerlendirildiği görülmektedir (Kumar vd., 2023).

Tablo 2. Yaygın IoT iletişim protokollerinde başlıca güvenlik tehditleri ve koruma yaklaşımları (Dritsas ve Trigka, 2025; Zohourian vd., 2023; Lacava vd., 2022; Yang vd., 2018; Kumar vd., 2023)

Protokol / Ağ Yapısı	Başlıca saldırılar	Temel etki	Başlıca koruma yaklaşımı
ZigBee	Replay, cihaz taklidi, anahtar ele geçirme, trafik analizi	Gizlilik / bütünlük	Güvenli anahtar yönetimi, çerçeve sayaçları, güvenli cihaz katılımı
BLE	Dinleme, MitM, replay, yetkisiz eşleştirme	Gizlilik / kimlik doğrulama	Güvenli eşleştirme, şifreli bağlantı, güncel BLE güvenlik özellikleri
LoRaWAN	Replay, seçici DoS, mesaj değiştirme, batarya tüketimi	Bütünlük / erişilebilirlik	Güvenli aktivasyon, benzersiz anahtarlar, sayaç yönetimi
MQTT	MitM, replay, yetkisiz erişim, DoS	Gizlilik / erişilebilirlik	TLS, istemci kimlik doğrulama, erişim kontrolü, trafik sınırlama
RPL	Sinkhole, wormhole, DoS	Yönlendirme bütünlüğü / erişilebilirlik	Güvenli yönlendirme doğrulaması, güven/IDS tabanlı izleme

Tablo 2'de görüldüğü üzere IoT haberleşme protokollerindeki güvenlik tehditleri protokolün çalışma biçimine göre farklılaşmaktadır. ZigBee ve BLE gibi kısa menzilli teknolojilerde eşleştirme, anahtar yönetimi ve yerel kablosuz trafiğin korunması öne çıkarken, LoRaWAN'da uzun menzilli iletişim ve mesaj sayaçları; MQTT'de broker güvenliği ve erişim kontrolü; RPL'de ise yönlendirme bilgilerinin güvenilirliği kritik hâle gelmektedir. Bu nedenle IoT ağlarında tek bir güvenlik mekanizmasının bütün haberleşme katmanlarını koruması beklenmemeli; protokole özgü önlemler katmanlı bir güvenlik yaklaşımı içerisinde birlikte uygulanmalıdır.

Sonuç olarak IoT ağ saldırıları; iletişimin dinlenmesi, mesajların yeniden kullanılması veya değiştirilmesi, ağ kaynaklarının tüketilmesi ve yönlendirme altyapısının manipüle edilmesi gibi farklı biçimlerde ortaya çıkabilmektedir. ZigBee, BLE, LoRaWAN, MQTT ve RPL farklı amaçlarla tasarlanmış olsa da güvenli kimlik doğrulama, anahtar yönetimi, mesaj tazeliği kontrolü, şifreleme ve trafik izleme mekanizmalarına ortak biçimde ihtiyaç duymaktadır. Ağ güvenliğinin protokol özellikleri ve cihazların kaynak kısıtları birlikte değerlendirilerek tasarlanması, iletişim zafiyetlerinin sistem genelinde daha büyük güvenlik ihlallerine dönüşmesini önlemek açısından önem taşımaktadır.

IOT İÇİN GÜVENLİ YAZILIM GELİŞTİRME

IoT sistemlerinin güvenliği yalnızca donanım bileşenlerinin veya haberleşme protokollerinin korunmasıyla sağlanamamaktadır. Cihaz üzerinde çalışan firmware, işletim sistemi veya RTOS bileşenleri, haberleşme kütüphaneleri, API'ler ve bulut servisleri sistemin yazılım katmanını oluşturmakta ve bu katmanda ortaya çıkan bir zafiyet cihazın tamamının ele geçirilmesine neden olabilmektedir. Özellikle IoT cihazlarının uzun süre sahada çalışması, farklı üreticilere ait yazılım bileşenlerini kullanması ve bazı cihazlarda güncelleme olanaklarının sınırlı olması, geliştirme aşamasında yapılan hataların uzun süre güvenlik riski olarak kalmasına yol açabilmektedir. Bu nedenle güvenlik, yazılım tamamlandıktan sonra eklenen bir özellik yerine ürünün tüm yaşam döngüsüne yayılan temel bir tasarım gereksinimi olarak değerlendirilmelidir (Bakhshi vd., 2024; Ali vd., 2025).

Güvenli Yazılım Geliştirme Yaşam Döngüsü

Güvenli Yazılım Geliştirme Yaşam Döngüsü (Secure Software Development Life Cycle-S-SDLC), güvenlik gereksinimlerinin yazılım

geliştirme sürecinin başlangıcından itibaren tanımlanmasını ve tasarım, gerçekleştirme, test, dağıtım ve bakım aşamalarının tamamında sürdürülmesini amaçlamaktadır. IoT sistemlerinde bu yaklaşım daha da önemlidir; çünkü sonradan giderilmesi kolay olan bir web uygulaması zafiyeti, fiziksel olarak uzak veya güncelleme kapasitesi sınırlı binlerce IoT cihazında çok daha maliyetli bir probleme dönüşebilmektedir. IoT için güvenli SDLC yaklaşımlarında gereksinim analizi, yazılım tasarımı, geliştirme/uygulama, test ve kabul, dağıtım ve entegrasyon ile bakım ve kullanım dışı bırakma aşamalarının güvenlik açısından birlikte ele alınması önerilmektedir (ENISA, 2019).

Gereksinim aşamasında sistemin işleyeceği veri türleri, erişim yetkileri, kimlik doğrulama gereksinimleri, güncelleme yöntemi ve olası saldırı yüzeyleri belirlenmelidir. Tasarım aşamasında ise bu gereksinimlerin mimariye nasıl aktarılacağı değerlendirilerek tehdit modellemesi yapılmalıdır. STRIDE gibi tehdit modelleme yöntemleri; kimliğe bürünme, verinin değiştirilmesi, bilgi ifşası, hizmet dışı bırakma ve yetki yükseltme gibi risklerin yazılım geliştirilmeden önce sistematik biçimde incelenmesine yardımcı olabilmektedir. IoT ve IIoT sistemlerinde tehdit modellemesinin geliştirme yaşam döngüsüne erken aşamada eklenmesi, güvenlik problemlerinin üretim ortamına taşınmadan belirlenmesi açısından önemli görülmektedir (Ali vd., 2025).



Şekil 1. IoT sistemleri için güvenli yazılım geliştirme yaşam döngüsü

Şekil 1'de gösterilen yaşam döngüsünde güvenlik gereksinimlerinin belirlenmesi ve tehdit modellemesinden sonra güvenli tasarım ve kodlama, güvenlik testleri, güvenli dağıtım, OTA güncelleme ve sürekli izleme aşamaları birbirini takip etmektedir. Döngüsel yapı özellikle önemlidir; çünkü dağıtım sonrasında tespit edilen yeni bir zafiyetin güncelleme ve yama yönetimi aracılığıyla tekrar geliştirme sürecine aktarılması gerekmektedir. Böylece güvenlik yalnızca geliştirme öncesinde gerçekleştirilen tek seferlik bir kontrol olmaktan çıkarılarak cihazın kullanım ömrü boyunca sürdürülen bir süreç hâline gelmektedir (ENISA, 2019; Bahaa vd., 2021).

Güvenli Kodlama ve Yazılım Bileşenlerinin Yönetimi

IoT yazılımlarında güvenli kodlama, kaynakları sınırlı bir ortamda mümkün olan en küçük saldırı yüzeyinin oluşturulmasını amaçlamaktadır. Kullanıcı veya ağ üzerinden alınan girdilerin doğrulanmaması, güvenli olmayan bellek işlemleri, varsayılan ya da sabit kodlanmış kimlik bilgilerinin kullanılması, gereksiz servislerin açık bırakılması ve hataların güvenli biçimde yönetilmemesi cihazın saldırıya açık hâle gelmesine neden olabilmektedir. Bu nedenle kod geliştirme sırasında girdi doğrulama, en az ayrıcalık ilkesi, güvenli hata yönetimi ve hassas bilgilerin kaynak kod içerisine sabit olarak yerleştirilmemesi gibi temel prensiplerin uygulanması gerekmektedir. Firmware güvenliği üzerine yapılan incelemelerde zayıf erişim kontrolleri, yanlış yapılandırmalar, yeniden kullanılan yazılım bileşenleri ve güvensiz ağ arayüzleri önemli zafiyet kaynakları arasında gösterilmektedir (Bakhshi vd., 2024).

Modern IoT yazılımları yalnızca geliştirici tarafından yazılan kodlardan oluşmamaktadır. Açık kaynak kütüphaneler, üçüncü taraf sürücüler, protokol yığınları ve işletim sistemi bileşenleri de firmware içerisine dâhil edilmektedir. Bu bileşenlerden birinde ortaya çıkan güvenlik açığı doğrudan IoT ürününü etkileyebildiğinden bağımlılıkların sürümlerinin izlenmesi ve bilinen zafiyetler açısından düzenli olarak kontrol edilmesi gerekmektedir. Güvenli geliştirme sürecinde kaynak kod incelemesi, statik analiz, bağımlılık taraması ve kritik bileşenlerin güncel sürümlerinin takip edilmesi bu nedenle önem taşımaktadır. ENISA'nın IoT güvenli yazılım geliştirme yaklaşımında da üçüncü taraf yazılım bağımlılıklarının üretime alınmadan önce test edilmesi ve bakım döneminde yeni güvenlik açıkları açısından izlenmesi önerilmektedir (ENISA, 2019).

Firmware ve OTA Güncelleme Güvenliği

IoT cihazlarının sahaya dağıtılmasından sonra ortaya çıkan güvenlik açıklarının giderilebilmesi için firmware güncelleme yeteneği önemli bir gereksinimdir. Ancak güncelleme mekanizmasının kendisi güvenli biçimde tasarlanmadığında, güvenlik açığını gidermek amacıyla kullanılan süreç yeni bir saldırı vektörüne dönüşebilmektedir. Saldırgan tarafından değiştirilmiş veya sahte bir firmware dosyasının cihaza yüklenmesi; zararlı kod çalıştırılmasına, cihazın kontrolünün kaybedilmesine veya cihazın bağlı olduğu diğer sistemlere erişim sağlanmasına neden olabilir. IoT firmware güncellemelerine ilişkin çalışmalar güncelleme sürecinde manipülasyon, Ortadaki Adam, kimliğe bürünme ve hizmet dışı bırakma gibi saldırıların ortaya çıkabileceğini göstermektedir (Bettayeb vd., 2019).

Bu nedenle Over-the-Air (OTA) güncelleme mekanizmasının temel gereksinimleri firmware dosyasının kaynağının doğrulanması ve bütünlüğünün korunmasıdır. Güncelleme paketinin üretici tarafından dijital olarak imzalanması ve cihazın kuruluma başlamadan önce bu imzayı doğrulaması, değiştirilmiş veya yetkisiz firmware'in yüklenmesini engelleyen temel mekanizmalardan biridir. Güncelleme dosyasının güvenli bir iletişim kanalı üzerinden aktarılması ve cihazın yalnızca güvenilir güncelleme kaynağıyla iletişim kurması da sürecin güvenliğini artırmaktadır. Güncellemenin kesintiye uğraması durumunda cihazın çalışamaz hâle gelmesini önlemek amacıyla güvenli geri dönüş ve kurtarma mekanizmalarının da tasarımda dikkate alınması gerekmektedir. Firmware güncellemelerinin zafiyetlerin giderilmesinde kritik olduğu kadar, güncelleme sürecinin doğrulanmaması hâlinde saldırganlar tarafından istismar edilebilecek önemli bir güvenlik noktası olduğu literatürde vurgulanmaktadır (Bettayeb vd., 2019; Bakhshi vd., 2024).

Güncelleme yönetiminin yalnızca yeni firmware'in cihaza gönderilmesi olarak değerlendirilmemesi gerekir. Üretici tarafından hangi cihazın hangi yazılım sürümünü kullandığının bilinmesi, kritik güvenlik açıklarından etkilenen cihazların belirlenebilmesi ve gerekli yamaların kontrollü biçimde dağıtılabilmesi önemlidir. Güncellenemeyen veya desteği sona ermiş cihazların ağ içerisinde kontrolsüz biçimde çalışmaya devam etmesi ise uzun vadeli bir güvenlik riski oluşturabilmektedir. Bu nedenle bakım ve yama yönetimi, güvenli yazılım yaşam döngüsünün dağıtım sonrasındaki devamı olarak ele alınmalıdır (ENISA, 2019; Bakhshi vd., 2024).

Güvenlik Testleri ve Sürekli İzleme

IoT yazılımının dağıtımdan önce güvenlik testlerinden geçirilmesi, geliştirme sırasında fark edilmeyen hataların belirlenmesini sağlamaktadır. Statik analiz, kaynak kod veya firmware ikili dosyasını çalıştırmadan olası zafiyetlerin incelenmesine; dinamik analiz ise yazılımın çalışması sırasında davranışının ve dış girdilere verdiği tepkilerin değerlendirilmesine olanak sağlamaktadır. Firmware güvenliği çalışmalarında statik ve dinamik analiz ile bunların birlikte kullanıldığı hibrit yöntemler, zafiyet denetiminin temel yaklaşımları arasında değerlendirilmektedir. Bununla birlikte farklı işlemci mimarileri, kapalı kaynak firmware ve cihaz çeşitliliği IoT güvenlik testlerinin ölçeklenmesini güçleştirebilmektedir (Bakhshi vd., 2024).

Güvenlik testinin ürünün piyasaya çıkışıyla sona ermemesi gerekir. Yeni zafiyetlerin ortaya çıkması, üçüncü taraf bağımlılıkların zaman içerisinde güvensiz hâle gelmesi ve cihazların gerçek kullanım koşullarında beklenmeyen saldırılara maruz kalması sürekli izleme ihtiyacını ortaya çıkarmaktadır. DevSecOps yaklaşımı güvenlik kontrollerinin geliştirme, entegrasyon ve dağıtım süreçlerine otomatik olarak dâhil edilmesini amaçlamaktadır. IoT güvenliği üzerine gerçekleştirilen sistematik incelemelerde, güvenlik izleme ve saldırı tespit mekanizmalarının DevSecOps tabanlı yazılım hatlarıyla ilişkilendirilmesinin geliştirme sürecindeki güvenlik görünürlüğünü artırabileceği belirtilmektedir (Bahaa vd., 2021).

Sonuç olarak IoT için güvenli yazılım geliştirme, yalnızca kod içerisinde belirli güvenlik önlemlerinin uygulanmasıyla sınırlı değildir. Güvenlik gereksinimlerinin ve tehditlerin tasarım öncesinde belirlenmesi, güvenli kodlama ilkelerinin uygulanması, üçüncü taraf bileşenlerin izlenmesi, yazılımın statik ve dinamik yöntemlerle test edilmesi ve cihazların güvenilir OTA mekanizmalarıyla güncel tutulması birbirini tamamlayan süreçlerdir. Dağıtım sonrasında gerçekleştirilen güvenlik izleme ve yama yönetimi de bu yaşam döngüsünün devamını oluşturmaktadır. Böylece güvenlik, IoT cihazının geliştirilmesinden kullanım dışı bırakılmasına kadar sürdürülen bütüncül bir yazılım mühendisliği yaklaşımına dönüşmektedir.

IOT EKOSİSTEMİNDE KİMLİK DOĞRULAMA

Nesnelerin İnterneti (Internet of Things-IoT), fiziksel nesnelerin algılama, veri üretme, haberleşme ve belirli durumlarda otonom karar verme yetenekleriyle ağ ortamına dâhil edildiği geniş ölçekli bir teknolojik

ekosistemi ifade etmektedir. Sensörler, aktüatörler, giyilebilir cihazlar, akıllı ev bileşenleri, endüstriyel kontrol birimleri ve çeşitli gömülü sistemlerden oluşan bu yapı içerisinde çok sayıda heterojen cihazın birbirleriyle ve uzak servislerle iletişim kurması söz konusudur. Bu nedenle IoT güvenliğinin temel gereksinimlerinden biri, iletişime katılan cihazın veya kullanıcının iddia ettiği kimliğe sahip olduğunun doğrulanmasıdır. Yetkisiz veya ele geçirilmiş bir cihazın sisteme güvenilir bir bileşen gibi dâhil olması, yalnızca ilgili cihazı değil, aynı ağ üzerindeki diğer bileşenleri ve üretilen verilerin güvenilirliğini de etkileyebilmektedir. Bu yönüyle kimlik doğrulama, erişim kontrolü, veri bütünlüğü ve gizliliğin sağlanmasından önce oluşturulması gereken temel güvenlik katmanlarından biri olarak değerlendirilmektedir (El-hajj vd., 2019; Trnka vd., 2022).

IoT ortamlarında kimlik doğrulama, geleneksel bilgi sistemlerindeki kullanıcı adı ve parola doğrulamasından daha geniş bir kapsam taşımaktadır. Sisteme erişen kişinin kimliğinin belirlenmesinin yanı sıra cihazların, ağ geçitlerinin ve servislerin de karşılıklı olarak doğrulanması gerekebilmektedir. Bu nedenle IoT güvenlik mimarilerinde cihazların, kullanıcıların ve servislerin kimliklerinin farklı iletişim ve erişim senaryolarında doğrulanması gerekebilmektedir. Kimlik doğrulama ile yetkilendirme kavramlarının da birbirinden ayrılması gerekmektedir. Kimlik doğrulama bir varlığın kim olduğunun doğrulanmasına odaklanırken, yetkilendirme doğrulanmış varlığın hangi kaynaklara erişebileceğini ve hangi işlemleri gerçekleştirebileceğini belirlemektedir. IoT güvenliğine ilişkin sistematik çalışmalar, kimlik doğrulama ve yetkilendirmenin güvenli erişim kontrolünün birbirini tamamlayan unsurları olduğunu göstermektedir (Trnka vd., 2022).

M2M ve M2C İletişiminde Kimlik Doğrulama

IoT ekosistemindeki temel iletişim modellerinden biri makineden makineye iletişim (Machine-to-Machine-M2M) yaklaşımıdır. Bu yapıda cihazlar insan müdahalesi olmadan veri alışverişinde bulunarak önceden tanımlanmış işlemleri gerçekleştirebilmektedir. Sensör düğümlerinin ağ geçidine veri göndermesi veya akıllı cihazların birbirleriyle haberleşmesi bu modele örnek verilebilir. M2M yapılarında temel güvenlik gereksinimlerinden biri karşılıklı kimlik doğrulamadır. Taraflardan yalnızca birinin doğrulanması, saldırganın güvenilir bir cihaz veya servis gibi davranmasına imkân verebileceğinden, kimliğe bürünme, tekrar oynatma ve

Ortadaki Adam saldırılarına karşı karşılıklı doğrulama önem taşımaktadır (Ferrag vd., 2017).

Bir diğer yaygın model cihazdan buluta iletişim (Machine-to-Cloud/Device-to-Cloud-M2C) yapısıdır. Bu modelde IoT cihazlarının ürettiği veriler doğrudan veya ağ geçidi üzerinden bulut altyapısına aktarılmaktadır. Yetkisiz cihazların sisteme veri göndermesini ve sahte sunuculara hassas veri aktarılmasını önlemek için cihaz ile bulut servisinin karşılıklı olarak doğrulanması gerekmektedir. Cihaz, ağ geçidi ve uzak sunucu arasında gerçekleştirilen karşılıklı kimlik doğrulama ile güvenli oturum anahtarı oluşturma süreçleri IoT güvenliğinin temel bileşenleri arasında değerlendirilmektedir (El-hajj vd., 2019; Kumar vd., 2022).

Sertifika ve Token Tabanlı Kimlik Doğrulama

IoT ortamlarında güçlü cihaz kimliği oluşturmak için kullanılan yaklaşımlardan biri sertifika tabanlı kimlik doğrulamadır. Açık Anahtar Altyapısı (Public Key Infrastructure-PKI) kapsamında kullanılan X.509 sertifikaları, cihaz kimliğini açık anahtarla ilişkilendirerek TLS/DTLS gibi güvenli iletişim mekanizmalarında doğrulama amacıyla kullanılabilmektedir. Ancak sertifikaların üretimi, dağıtımı, yenilenmesi ve iptali yaşam döngüsü yönetimi gerektirmekte; geleneksel PKI işlemlerinin hesaplama ve iletişim yükü kaynakları sınırlı IoT cihazlarında önemli bir tasarım faktörü oluşturmaktadır (El-hajj vd., 2019).

Token tabanlı yaklaşımlarda ise kimlik bilgilerinin her işlemde doğrudan aktarılması yerine, sınırlı süreli ve belirli yetkilere sahip belirteçler kullanılmaktadır. Bu yöntem özellikle web servisleri, mobil uygulamalar ve bulut platformlarıyla bütünleşen IoT sistemlerinde esneklik sağlamaktadır. Buna karşın token'ın ele geçirilmesi, uzun geçerlilik süreleri veya aşırı geniş yetkiler güvenlik riski oluşturabilir. Bu nedenle belirteçlerin güvenli saklanması, sürelerinin sınırlandırılması ve iletişim kanalının korunması gerekmektedir. OAuth tabanlı modellerin güvenliği de protokolün yanı sıra uygulama biçimine ve cihaz kaynaklarına bağlıdır (El-hajj vd., 2019; Oh ve Kim, 2020).

Çok Faktörlü ve Hafif Sıklet Kimlik Doğrulama

Çok faktörlü kimlik doğrulamada parola veya PIN, güvenilir cihaz/akıllı kart ve biyometrik özellik gibi birden fazla unsur birlikte kullanılabilmektedir. Bu yaklaşım güvenliği artırmakla birlikte hesaplama, haberleşme ve kullanıcı etkileşimi maliyetlerini de yükseltebildiğinden,

uygulanacağı katman cihazın donanım özellikleri ve sistemin risk düzeyi dikkate alınarak belirlenmelidir (El-hajj vd., 2019; Çetintav ve Sandikkaya, 2025).

IoT kimlik doğrulama mekanizmalarında temel sorunlardan biri güvenlik ile kaynak tüketimi arasındaki dengedir. Sınırlı işlem gücü, bellek, bant genişliği ve enerjiye sahip cihazlarda geleneksel doğrulama mekanizmaları yüksek gecikme ve kaynak tüketimine yol açabilmektedir. Bu nedenle yöntemlerin yalnızca güvenlik açısından değil, hesaplama, iletişim ve donanım maliyetleri bakımından da değerlendirilmesi gerekmektedir (Çetintav ve Sandikkaya, 2025).

Bu gereksinimler hafif sıklet kimlik doğrulama (lightweight authentication) yaklaşımlarını öne çıkarmıştır. Karma fonksiyonları, XOR işlemleri, simetrik anahtarlar, eliptik eğri tabanlı yöntemler ve donanımsal güven bileşenlerinden yararlanan bu protokollerde düşük maliyet ile saldırılara karşı dayanıklılık birlikte değerlendirilmelidir. Bu nedenle tek bir yöntemin tüm IoT uygulamaları için en uygun çözüm olduğu kabul edilmemekte; güvenlik gereksinimleri ile cihaz kısıtlarının birlikte ele alınması önerilmektedir (Çetintav ve Sandikkaya, 2025). Başlıca kimlik doğrulama yaklaşımlarının güvenlik ve kaynak gereksinimleri açısından karşılaştırması Tablo 3'te sunulmuştur.

Tablo 3. IoT ortamlarında temel kimlik doğrulama yaklaşımlarının karşılaştırılması (El-hajj vd., 2019; Kumar vd., 2022; Çetintav ve Sandikkaya, 2025)

Kimlik doğrulama yaklaşımı	Hesaplama yükü	İletişim yükü	Ölçeklenebilirlik	Güvenlik düzeyi	IoT açısından temel değerlendirme
Önceden paylaşılan anahtar (PSK)	Düşük	Düşük	Düşük-Orta	Orta	Basit, ancak anahtar yönetimi zordur.
Sertifika/PKI tabanlı	Orta-Yüksek	Orta-Yüksek	Yüksek	Yüksek	Güçlü kimlik doğrulama sağlar.
Token tabanlı	Düşük-Orta	Düşük-Orta	Yüksek	Orta-Yüksek	Bulut ve servis erişiminde esnektir.
Çok faktörlü doğrulama	Orta-Yüksek	Orta	Orta	Yüksek	Kritik erişimlerde güvenliği artırır.

Sonuç olarak IoT ekosisteminde kimlik doğrulama, yalnızca kullanıcıların sisteme girişinin denetlenmesi olarak değil, cihazlar, ağ geçitleri, bulut servisleri ve kullanıcılar arasında güven ilişkisinin kurulmasını sağlayan bütüncül bir güvenlik mekanizması olarak ele

alınmalıdır. Kullanılacak yaklaşımın belirlenmesinde yüksek güvenlik seviyesi tek başına yeterli bir ölçüt değildir. Cihazın hesaplama kapasitesi, enerji kaynağı, ağ topolojisi, ölçeklenebilirlik gereksinimi ve korunacak verinin hassasiyeti birlikte değerlendirilmelidir. Özellikle kaynakları sınırlı IoT sistemlerinde etkin bir kimlik doğrulama mimarisi, güvenlik gereksinimlerini mümkün olan en düşük işlem ve iletişim maliyetiyle karşılayacak biçimde tasarlanmalıdır.

KISITLI CİHAZLAR İÇİN ŞİFRELEME TEKNOLOJİLERİ

IoT sistemlerinde verinin güvenli biçimde iletilmesi ve saklanması, yalnızca iletişim kanalının korunmasına değil, kullanılan kriptografik mekanizmaların cihazın donanımsal özelliklerine uygun biçimde seçilmesine de bağlıdır. Geleneksel bilgisayar sistemlerinde yüksek işlem gücü ve geniş bellek kapasitesi sayesinde birçok kriptografik algoritma önemli bir performans problemi oluşturmadan çalıştırılabilirken, IoT cihazlarında aynı durum her zaman geçerli değildir. Sensör düğümleri, RFID bileşenleri, düşük güçlü mikrodenetleyiciler ve batarya ile çalışan gömülü cihazlar sınırlı işlem gücü, RAM/Flash kapasitesi ve enerji bütçesi ile çalışabilmektedir. Bu nedenle IoT ortamlarında kriptografik güvenlik değerlendirilirken algoritmanın sağladığı güvenlik düzeyinin yanı sıra hesaplama süresi, bellek kullanımı, enerji tüketimi ve iletişim maliyeti gibi ölçütlerin de dikkate alınması gerekmektedir (Rana vd., 2022; Radhakrishnan vd., 2024).

Kriptografik işlemlerin maliyeti özellikle uzun süre bağımsız biçimde çalışması beklenen IoT düğümlerinde önem kazanmaktadır. Karmaşık matematiksel işlemler işlemcinin daha uzun süre aktif kalmasına, daha fazla enerji harcanmasına ve bazı uygulamalarda veri iletim gecikmesinin artmasına neden olabilmektedir. Benzer biçimde, algoritmaya ait kodun ve geçici verilerin bellek gereksinimi düşük kapasiteli mikrodenetleyiciler üzerinde uygulama alanını sınırlayabilmektedir. Bu durum, IoT sistemlerinde "en güçlü algoritmanın" doğrudan seçilmesinden ziyade, uygulamanın gerektirdiği güvenlik düzeyi ile cihaz kaynakları arasında dengeli bir seçim yapılmasını gerekli kılmaktadır. Hafif sıklet kriptografi çalışmalarının temel çıkış noktası da bu güvenlik-performans dengesidir (Rana vd., 2022).

Simetrik ve Asimetrik Şifreleme Yaklaşımları

Simetrik kriptografide verinin şifrelenmesi ve çözülmesi için aynı gizli anahtar kullanılmaktadır. Düşük hesaplama maliyeti ve yüksek işlem hızı

sayesinde özellikle sensör verilerinin ve büyük veri bloklarının korunmasında kullanılmaktadır. AES, IoT sistemlerinde yaygın kullanılan simetrik algoritmalardan biridir. Ancak ortak anahtarın güvenli biçimde dağıtılması, korunması, yenilenmesi ve gerektiğinde iptal edilmesi, özellikle çok sayıda cihazın bulunduğu IoT ortamlarında önemli bir yönetim problemidir (Rao ve Deebak, 2023).

Asimetrik kriptografide ise açık ve özel olmak üzere iki ilişkili anahtar kullanılmaktadır. Bu yapı, önceden ortak anahtar paylaşmayan cihazlar arasında güven oluşturmayı kolaylaştırırken, RSA gibi yöntemler kaynakları sınırlı cihazlarda yüksek hesaplama maliyeti oluşturabilmektedir. Eliptik Eğri Kriptografisi (ECC), benzer güvenlik düzeyini daha kısa anahtarlarla sağlayabildiğinden bellek, işlem ve enerji gereksinimleri açısından IoT sistemleri için avantajlıdır (Adeniyi vd., 2024).

Bu nedenle IoT mimarilerinde simetrik ve asimetrik yöntemler çoğunlukla tamamlayıcı biçimde kullanılmaktadır. Asimetrik yöntemler kimlik doğrulama ve güvenli oturum anahtarı oluşturma amacıyla kullanılırken, sürekli veri şifrelemede daha düşük maliyetli simetrik algoritmalar tercih edilebilmektedir.

Hafif Sıklet Kriptografi

Kaynakları sınırlı cihazların yaygınlaşmasıyla birlikte hafif sıklet kriptografi (Lightweight Cryptography-LWC) IoT güvenliğinin önemli araştırma alanlarından biri hâline gelmiştir. Hafif sıklet tasarımlar yalnızca algoritmanın matematiksel olarak basitleştirilmesi anlamına gelmemekte; bellek gereksinimi, devre alanı, işlem süresi, enerji tüketimi ve iletişim performansı gibi farklı ölçütlerin birlikte optimize edilmesini amaçlamaktadır. Literatürde simetrik blok şifreler, akış şifreleri, hafif sıklet karma fonksiyonları ve kimlik doğrulamalı şifreleme mekanizmaları bu kapsamda değerlendirilmektedir (Rana vd., 2022).

AES, SPECK ve ASCON gibi algoritmaların kaynakları sınırlı geliştirme kartlarında karşılaştırıldığı deneysel çalışmalar, algoritma seçiminin tek bir performans ölçütüne göre yapılamayacağını göstermektedir. Radhakrishnan vd. (2024) tarafından Arduino Nano ve Arduino Micro platformlarında gerçekleştirilen deneylerde RAM/ROM kullanımı, yürütme süresi, gecikme ve throughput değerlerinin algoritmalara göre önemli ölçüde değiştiği belirlenmiştir. Örneğin Arduino Nano üzerinde ASCON daha düşük RAM kullanımı gösterirken, SPECK'in şifreleme süresinin daha düşük olduğu

görülmüştür. Bu sonuçlar, düşük bellek tüketimi açısından avantajlı bir algoritmanın aynı zamanda bütün performans ölçütlerinde en iyi seçenek olmak zorunda olmadığını göstermektedir (Radhakrishnan vd., 2024)

Tablo 4. Arduino Nano üzerinde seçilmiş kriptografik algoritmaların kaynak kullanımı ve şifreleme süresi (Radhakrishnan vd., 2024)

Algoritma	RAM kullanımı (Bayt)	ROM kullanımı (Bayt)	Şifreleme süresi (µs)
AES-128	1570	10.928	42,39
SPECK	1164	8.848	9,74
ASCON	940	1.108	42,48

Tablo 4'teki değerler algoritmaların yalnızca güvenlik özelliklerinin değil, çalıştırıldıkları donanım platformunun da seçim sürecinde dikkate alınması gerektiğini ortaya koymaktadır. Aynı çalışmada farklı platformlarda bellek kullanım değerlerinin değişebildiği gösterildiğinden, bu sonuçların bütün IoT donanımlarına doğrudan genellenmesi doğru olmayacaktır. Bunun yerine algoritma seçiminin hedef mikrodenetleyici üzerinde ölçülen bellek kullanımı, işlem süresi, throughput ve enerji tüketimi gibi ölçütler üzerinden yapılması daha uygun bir yaklaşımdır. Ayrıca SPECK gibi performans açısından avantajlı sonuç veren bir algoritmanın yalnızca hız ölçütüne dayanılarak tercih edilmemesi; güvenlik analizi ve uygulamanın risk düzeyinin de birlikte değerlendirilmesi gerekmektedir (Radhakrishnan vd., 2024).

Güvenli Anahtar Üretimi ve Yönetimi

Kriptografik algoritmanın güvenilirliği kadar anahtar yönetimi de önemlidir; çünkü anahtarın ele geçirilmesi durumunda güçlü bir şifreleme algoritması tek başına yeterli koruma sağlayamaz. Bu nedenle IoT sistemlerinde anahtar yönetimi, cihazın sisteme kaydedilmesinden kullanım dışı bırakılmasına kadar uzanan bir yaşam döngüsü olarak ele alınmalıdır. Anahtarların güvenli biçimde oluşturulması, dağıtılması, saklanması, güncellenmesi ve gerektiğinde iptal edilmesi bu sürecin temel aşamalarını oluşturmaktadır (Rao ve Deebak, 2023).

Geniş ölçekli IoT ağlarında tek bir anahtarın uzun süre veya çok sayıda cihaz tarafından kullanılması güvenlik riskini artırırken, her cihaz için bağımsız anahtar yönetimi de depolama ve dağıtım maliyetini yükseltebilmektedir. Bu nedenle anahtar yönetimi yaklaşımı; cihaz sayısı, ağ topolojisi, hesaplama kapasitesi ve fiziksel erişim riski dikkate alınarak tasarlanmalıdır (Rao ve Deebak, 2023). Kısıtlı IoT cihazlarında anahtar

oluşturma, güvenli saklama, kimlik doğrulama, veri şifreleme, yenileme ve iptal süreçlerini kapsayan yaşam döngüsü Şekil 2'de özetlenmiştir.



Şekil 2. Kısıtlı IoT cihazlarında kriptografik koruma ve anahtar yaşam döngüsü

Sonuç olarak kısıtlı IoT cihazlarında şifreleme teknolojilerinin seçimi yalnızca kriptografik dayanıklılığa göre yapılmamalıdır. İşlem kapasitesi, bellek kullanımı, enerji tüketimi ve iletişim gecikmesi, güvenlik gereksinimleriyle birlikte değerlendirilmelidir. Simetrik algoritmalar sürekli veri aktarımının korunmasında düşük maliyetli çözümler sunarken, ECC gibi açık anahtarlı mekanizmalar güvenli anahtar oluşturma ve taraflar arasında güven tesis edilmesinde önemli avantaj sağlayabilmektedir. Hafif sıklet kriptografi ise bu iki gereksinim arasındaki dengeyi kaynakları sınırlı cihazlara uyarlamayı amaçlamaktadır. Bununla birlikte algoritmanın güvenliği, anahtarların güvenli biçimde oluşturulması, saklanması, yenilenmesi ve iptal edilmesini kapsayan bir anahtar yönetimi mekanizmasıyla desteklenmediği sürece tek başına yeterli değildir.

IOT UYGULAMALARINDA VERİ GİZLİLİĞİ

Nesnelerin İnterneti uygulamalarında sensörler, giyilebilir cihazlar, akıllı ev bileşenleri, sağlık sistemleri, konum tabanlı uygulamalar ve endüstriyel izleme birimleri tarafından sürekli olarak büyük miktarda veri üretilmektedir. Toplanan verilerin bir bölümü doğrudan kişisel bilgi içermese dahi konum, zaman, kullanım alışkanlığı, hareket örüntüsü veya diğer veri kümeleriyle ilişkilendirildiğinde kullanıcılar hakkında ayrıntılı çıkarımlar yapılmasına imkân verebilmektedir. Bu nedenle IoT ortamlarında veri gizliliği yalnızca verinin yetkisiz kişiler tarafından okunmasının

engellenmesiyle sınırlı değildir. Verinin hangi amaçla toplandığı, nerede işlendiği, kimlerle paylaşıldığı ve ne kadar süreyle saklandığı da mahremiyet açısından önem taşımaktadır. IoT sistemlerinde yaygın veri toplama, kullanıcıların izlenmesi, profillenmesi ve farklı veri kaynaklarının birbirleriyle ilişkilendirilmesi başlıca mahremiyet riskleri arasında değerlendirilmektedir (Ziegeldorf vd., 2014).

IoT ortamlarını geleneksel bilgi sistemlerinden ayıran önemli özelliklerden biri veri toplama sürecinin çoğu zaman arka planda, sürekli ve çok sayıda cihaz tarafından gerçekleştirilmesidir. Tek bir veri akışına sensör düğümleri, ağ geçitleri, edge sistemleri, bulut platformları, mobil uygulamalar ve üçüncü taraf hizmet sağlayıcılar dâhil olabilmektedir. Bu yapı, kişisel veya hassas verinin yaşam döngüsü boyunca farklı sistemlerden geçmesine neden olmaktadır. Dolayısıyla veri gizliliğinin yalnızca merkezi bir sunucuda uygulanacak güvenlik mekanizmasıyla sağlanması yeterli değildir. Verinin üretildiği noktadan kullanım dışı bırakıldığı ana kadar bütün sürecin gizlilik açısından değerlendirilmesi gerekmektedir (Majeed vd., 2025).

Veri Yaşam Döngüsü ve Gizlilik Gereksinimleri

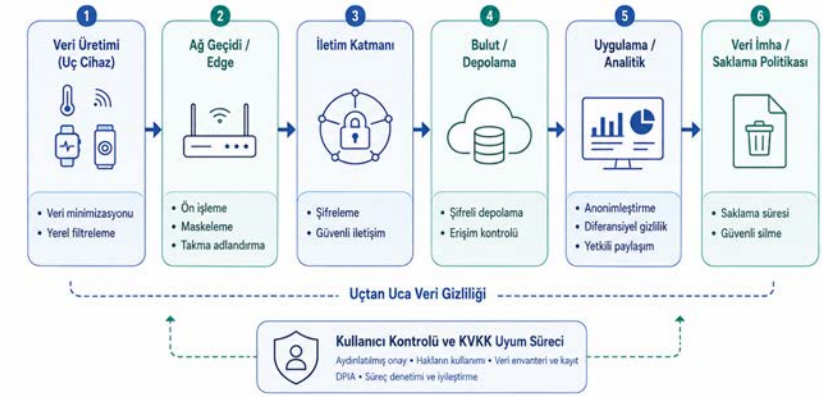
IoT sistemlerinde veri yaşam döngüsü; verinin üretilmesi, iletilmesi, işlenmesi, depolanması, kullanılması veya paylaşılması ve kullanım amacı sona erdiğinde silinmesi aşamalarından oluşmaktadır. Her aşama farklı gizlilik riskleri barındırmaktadır. Gereğinden fazla veri toplanması, iletişim sırasında verinin ele geçirilmesi, kullanıcı profili çıkarılması, yetkisiz erişim ve verinin gereğinden uzun süre saklanması bu risklere örnek verilebilir. Bu nedenle mahremiyet, veri yaşam döngüsünün tamamını kapsayan bir güvenlik gereksinimi olarak ele alınmalıdır.

Veri üretiminde temel yaklaşımlardan biri veri minimizasyonudur. Bu yaklaşım, uygulamanın işlevi için gerekli olmayan bilgilerin toplanmamasını veya işlenmemesini amaçlamaktadır. Böylece hem işlenen veri miktarı hem de olası bir ihlalde açığa çıkabilecek bilgi azaltılmaktadır. Kullanıcı merkezli IoT veri yönetiminde, kullanıcı kontrolünün artırılması ve verinin hangi amaçla toplandığına ilişkin şeffaflık önemli unsurlar olarak değerlendirilmektedir (Pinto vd., 2026).

Veri iletiminde iletişim kanalının şifrelenmesi, veri bütünlüğünün sağlanması ve tarafların kimliklerinin doğrulanması temel güvenlik mekanizmalarıdır. Ancak güvenli iletim, verinin sonraki aşamalarda

gereğinden fazla işlenmesini veya farklı veri kümeleriyle birleştirilerek kullanıcı hakkında yeni bilgiler çıkarılmasını engellememektedir. Özellikle veri kümelerinin ilişkilendirilmesi yoluyla yeniden tanımlama ve kullanıcı profili oluşturulması önemli mahremiyet tehditleri arasındadır (Ziegeldorf vd., 2014).

Veri işleme süreçlerinin cihaz veya edge katmanında gerçekleştirilmesi, hassas ham verinin buluta aktarılmasını azaltarak mahremiyet açısından avantaj sağlayabilmektedir. Güncel çalışmalarda federated learning, split learning, diferansiyel gizlilik ve anonimleştirme gibi yöntemlerin farklı IoT mimarilerinde uygulanabildiği belirtilmektedir (Majeed vd., 2025). IoT verisinin üretimden kullanım dışı bırakılmasına kadar izlediği akış ile veri minimizasyonu, ön işleme, şifreli iletişim, güvenli depolama, anonimleştirme ve güvenli silme gibi koruma noktaları Şekil 3'te özetlenmiştir.



Şekil 3. IoT sistemlerinde uçtan uca veri yaşam döngüsü ve gizlilik koruma noktaları

Anonimleştirme, Maskeleme ve Takma Adlandırma

IoT verilerinin araştırma, analitik veya hizmet geliştirme amacıyla kullanılması sırasında kişisel ve hassas bilgilerin görünürlüğünün azaltılması gerekebilmektedir. Veri maskeleme, hassas alanların değiştirilmesi, gizlenmesi veya farklı değerlerle temsil edilmesi yoluyla gerçek bilginin doğrudan görülmesini engellemeyi amaçlar. Ancak veri kümesindeki diğer alanlar kullanıcıyı tanımlamaya yeterliyse maskeleme tek başına tam mahremiyet sağlamayabilir (Ren vd., 2024).

Anonimleştirme, verinin belirli bir kişiyle ilişkilendirilebilirliğini azaltmayı amaçlamaktadır. Doğrudan tanımlayıcıların kaldırılması, genelleştirme, pertürbasyon ve k-anonimlik bu amaçla kullanılabilecek

yöntemler arasındadır. IoT ortamlarında sürekli veri akışlarının bulunması ve farklı veri kaynaklarının birleştirilebilmesi, anonimleştirme süreçlerini daha karmaşık hâle getirmektedir (Ren vd., 2024).

Takma adlandırmada (pseudonymization) ise gerçek kimlik, doğrudan kişiyi göstermeyen başka bir tanımlayıcıyla değiştirilir. Bu yöntem kimliğin görünürlüğünü azaltmakla birlikte, gerçek kimlik ile takma değer arasındaki eşleştirme bilgisinin ayrıca korunmasını gerektirir. Maskeleme, takma adlandırma, genelleştirme ve pertürbasyon IoT mahremiyetini destekleyen tamamlayıcı yöntemler olarak değerlendirilmektedir (Ren vd., 2024).

Tablo 5. IoT sistemlerinde temel veri gizliliği yöntemlerinin karşılaştırılması (Ren vd., 2024; Majeed vd., 2025)

Yöntem	Temel amaç	Uygulama noktası	Temel sınırlılık
Veri minimizasyonu	Gereksiz veri toplamayı azaltmak	Veri üretimi	Veri gereksinimlerinin doğru belirlenmesini gerektirir
Şifreleme	Yetkisiz erişimi önlemek	İletim / depolama	Çözülmüş verinin kullanımını tek başına denetlemez
Veri maskeleme	Hassas alanları gizlemek	İşleme / paylaşım	Yeniden tanımlama riski tamamen ortadan kalkmayabilir
Anonimleştirme	Kişiyle ilişkilendirilebilirliği azaltmak	İşleme / paylaşım	Veri kullanılabilirliğinde azalmaya yol açabilir
Diferansiyel gizlilik	Analizlerde bireysel mahremiyeti korumak	Analitik / modelleme	Mahremiyet ve veri faydası arasında denge gerektirir

Tablo 5'te görüldüğü üzere IoT uygulamalarında kullanılan veri gizliliği yöntemleri birbirlerinin doğrudan alternatifi olmaktan çok, veri yaşam döngüsünün farklı aşamalarına yönelik tamamlayıcı koruma mekanizmalarıdır. Şifreleme verinin iletişim veya depolama sırasında yetkisiz kişiler tarafından okunmasını engellemeye odaklanırken, maskeleme ve anonimleştirme özellikle verinin işlenmesi ve paylaşılması sırasında kimliğin ortaya çıkma olasılığını azaltmaktadır. Veri minimizasyonu ise hassas verinin sisteme hiç alınmamasını sağlayarak riski kaynağında azaltmaktadır.

Diferansiyel gizlilik (Differential Privacy-DP) özellikle veri analizi ve istatistiksel işlemler açısından farklı bir yaklaşım sunmaktadır. Bu yöntemde çıktılara kontrollü miktarda gürültü eklenerek tek bir kullanıcıya ait verinin analiz sonucundan çıkarılmasının zorlaştırılması amaçlanmaktadır. Bunun karşılığında mahremiyet düzeyi ile elde edilen verinin faydası arasında bir

denge kurulması gerekmektedir. Güncel IoT mahremiyet literatüründe diferansiyel gizliliğin anonimleştirme, federated learning ve diğer mahremiyet koruyucu yöntemlerle birlikte kullanılabildiği görülmektedir (Majeed vd., 2025).

Mahremiyet Odaklı Tasarım ve Kullanıcı Kontrolü

IoT sistemlerinde veri gizliliğinin, sistem tamamlandıktan sonra eklenen bağımsız bir güvenlik özelliği yerine tasarımın başlangıcından itibaren ele alınması gerekmektedir. Privacy by Design (PbD) yaklaşımında veri minimizasyonu, kullanıcı kontrolü, erişim sınırlandırması ve mahremiyet koruyucu tekniklerin sistem mimarisine başlangıç aşamasından itibaren dâhil edilmesi amaçlanmaktadır. Majeed vd. (2025), IoT mahremiyet çözümlerini Privacy by Design kapsamında federated learning ve split learning gibi yöntemlerle, privacy engineering kapsamında ise diferansiyel gizlilik ve anonimleştirme gibi tekniklerle sınıflandırmaktadır. Ayrıca birden fazla mahremiyet koruyucu mekanizmanın birlikte kullanıldığı hibrit yapıların da IoT sistemleri açısından önem kazandığı görülmektedir.

Kullanıcıların kendi verileri üzerinde kontrol sahibi olması bu yaklaşımın önemli bir diğer boyutudur. Kullanıcının hangi bilgilerin toplandığını, nerede tutulduğunu ve hangi amaçlarla kullanıldığını anlayabilmesi sistemin şeffaflığını artırmaktadır. Pinto vd. (2026) tarafından kullanıcı merkezli IoT veri mahremiyeti üzerine gerçekleştirilen deneysel çalışma, kişisel veriler üzerinde daha fazla kontrol sağlayan yaklaşımların kullanıcıların algıladığı kontrol, şeffaflık ve mahremiyet farkındalığı bakımından önem taşıdığını göstermektedir. Bu durum IoT veri gizliliğinin yalnızca arka planda çalışan teknik mekanizmalardan oluşmadığını; kullanıcıya verisi üzerinde görünür ve anlaşılabilir kontrol imkânlarının sunulmasının da önemli olduğunu ortaya koymaktadır.

KVKK Bağlamında IoT Veri Gizliliği

IoT cihazlarından elde edilen verilerin gerçek kişilerle ilişkilendirilebilir olması, kişisel verilerin korunmasına ilişkin hukuki gereksinimleri de sistem tasarımının bir parçası hâline getirmektedir. Türkiye'de kişisel verilerin işlendiği IoT uygulamalarının 6698 sayılı Kişisel Verilerin Korunması Kanunu (KVKK) kapsamında değerlendirilmesi gerekmektedir. Bu kapsamda veri işleme faaliyetleri, mahremiyet riskleri ve IoT ekosistemindeki farklı paydaşların sorumlulukları önem taşımaktadır (Ertan, 2022).

Cihaz üreticisi, platform sağlayıcısı, uygulama geliştiricisi ve hizmet sunucusu gibi farklı aktörlerin aynı veri üzerinde işlem yapabilmesi, verinin kim tarafından ve hangi amaçla işlendiğinin açık biçimde belirlenmesini gerekli kılmaktadır. Bu nedenle yalnızca teknik güvenlik yeterli olmayıp veri minimizasyonu, erişim yetkilerinin sınırlandırılması ve kullanım amacı sona eren verilerin uygun biçimde yönetilmesi de gerekmektedir (Ertan, 2022).

Sonuç olarak IoT uygulamalarında veri gizliliği; şifreleme, veri minimizasyonu, maskeleme, anonimleştirme, diferansiyel gizlilik ve erişim kontrolü gibi birbirini tamamlayan mekanizmalarla sağlanmalıdır. Kullanıcı kontrolünün artırılması ve kişisel veri işleme süreçlerinin KVKK ile uyumlu yürütülmesi de bu yaklaşımın önemli bileşenleridir. Bu nedenle veri gizliliği, cihazdan buluta kadar tüm IoT katmanlarını kapsayan bütüncül bir tasarım anlayışıyla ele alınmalıdır.

SONUÇ VE ÖNERİLER

IoT sistemlerinin yaygınlaşması, güvenlik ve mahremiyetin cihaz tasarımından veri kullanımının sona ermesine kadar uzanan bütün yaşam döngüsünde ele alınmasını gerekli hâle getirmiştir. Bu bölümde incelenen tehditler, IoT güvenliğinin yalnızca ağ trafiğinin şifrelenmesi veya kullanıcı kimliğinin doğrulanmasıyla sınırlandırılamayacağını göstermektedir. Fiziksel müdahale, debug arayüzleri, bellek dökümü ve veri yolu dinleme gibi donanımsal saldırılar; MitM, replay, DoS/DDoS ve yönlendirme saldırıları gibi ağ tehditleri; güvenli olmayan firmware ve yazılım bileşenleri ile veri mahremiyetine yönelik riskler birbirleriyle ilişkili bir saldırı yüzeyi oluşturmaktadır.

Güvenli IoT sistemlerinin geliştirilmesinde öncelikle security by design ve privacy by design yaklaşımlarının benimsenmesi gerekmektedir. Donanımsal düzeyde güven kökü, güvenli element, PUF ve secure boot gibi mekanizmalar kullanılmalı; üretim sonrasında gereksiz debug arayüzleri sınırlandırılmalıdır. Ağ katmanında protokole özgü güvenlik kontrolleri, güçlü kimlik doğrulama, güvenli anahtar yönetimi ve saldırı tespit mekanizmaları uygulanmalıdır. Yazılım tarafında güvenli geliştirme yaşam döngüsü, tehdit modelleme, statik ve dinamik analiz, bağımlılık yönetimi ve güvenilir OTA güncelleme süreçleri temel gereksinimler hâline getirilmelidir.

Kaynakları sınırlı IoT cihazlarında güvenlik mekanizmaları seçilirken yalnızca teorik güvenlik seviyesi değil; işlem maliyeti, bellek kullanımı,

enerji tüketimi ve iletişim yükü de dikkate alınmalıdır. Bu nedenle hafif sıklet kimlik doğrulama ve kriptografi yöntemlerinin uygun kullanım senaryolarında değerlendirilmesi önemlidir. Veri mahremiyeti açısından ise veri minimizasyonu, anonimleştirme, maskeleme, şifreleme, erişim kontrolü ve kullanıcı kontrolü birlikte ele alınmalı; kişisel veri işleme süreçleri ilgili hukuki düzenlemelerle uyumlu şekilde yürütülmelidir. Sonuç olarak sürdürülebilir IoT güvenliği, cihazdan buluta kadar bütün sistem katmanlarını kapsayan, sürekli güncellenen ve yaşam döngüsü temelli bütüncül bir güvenlik yaklaşımıyla sağlanabilir.

KAYNAKÇA

Adeniyi, A. E., Jimoh, R. G. ve Awotunde, J. B. (2024). A systematic review on elliptic curve cryptography algorithm for Internet of Things: Categorization, application areas, and security. Computers & Electrical Engineering, 118, 109330. DOI: 10.1016/j.compeleceng.2024.109330.

Ali, M., Arif, H., Raza, A. ve Nazir, M. (2025). Secure software engineering for Industrial IoT: Integrating threat modeling into the development lifecycle. ICCK Journal of Software Engineering, 1(2), 63-74. DOI: 10.62762/JSE.2025.729568.

Avcı, İ. (2022). Akıllı evlerde IoT teknolojileri ve siber güvenlik. Avrupa Bilim ve Teknoloji Dergisi, 34, 226-233. DOI: 10.31590/ejosat.1080228.

Bahaa, A., Abdelaziz, A., Sayed, A., Elfangary, L. ve Fahmy, H. (2021). Monitoring real time security attacks for IoT systems using DevSecOps: A systematic literature review. Information, 12(4), 154. DOI: 10.3390/info12040154.

Bakhshi, T., Ghita, B. ve Kuzminykh, I. (2024). A review of IoT firmware vulnerabilities and auditing techniques. Sensors, 24(2), 708. DOI: 10.3390/s24020708.

Bettayeb, M., Nasir, Q. ve Abu Talib, M. (2019). Firmware update attacks and security for IoT devices: Survey. Proceedings of the ArabWIC 6th Annual International Conference Research Track içinde (ss. 1-6). ACM. DOI: 10.1145/3333165.3333169.

Buscemi, A., Turcanu, I., Castignani, G., Panchenko, A., Engel, T. ve Shin, K. G. (2023). A survey on Controller Area Network reverse engineering. IEEE Communications Surveys & Tutorials, 25(3), 1445-1481. DOI: 10.1109/COMST.2023.3264928.

Çetintav, I. ve Sandikkaya, M. T. (2025). A review of lightweight IoT authentication protocols from the perspective of security requirements, computation, communication, and hardware costs. IEEE Access, 13, 37703-37723. DOI: 10.1109/ACCESS.2025.3546147.

De, A., Khan, M. N. I., Park, J. ve Ghosh, S. (2017). Replacing eFlash with STTRAM in IoTs: Security challenges and solutions. Journal of Hardware and Systems Security, 1(4), 328-339. DOI: 10.1007/s41635-017-0026-x.

Dritsas, E. ve Trigka, M. (2025). A survey on cybersecurity in IoT. Future Internet, 17(1), 30. DOI: 10.3390/fi17010030.

El-hajj, M., Fadlallah, A., Chamoun, M. ve Serhrouchni, A. (2019). A survey of Internet of Things (IoT) authentication schemes. Sensors, 19(5), 1141. DOI: 10.3390/s19051141.

Ertan, A. (2022). Nesnelerin İnternetinin kişisel verilerin korunması kapsamında incelenmesi. Kişisel Verileri Koruma Dergisi, 4(2), 48-68.

European Union Agency for Cybersecurity (ENISA). (2019). Good practices for security of IoT: Secure software development lifecycle. European Union Agency for Cybersecurity.

Ferrag, M. A., Maglaras, L. A., Janicke, H., Jiang, J. ve Shu, L. (2017). Authentication protocols for Internet of Things: A comprehensive survey. Security and Communication Networks, 2017, 6562953. DOI: 10.1155/2017/6562953.

Gbadega, P. A., Imoize, A. L., Joshua, R. G. ve Tofade, S. O. (2026). TinyML power consumption and memory in IoT MCUs. Tiny Machine Learning içinde (ss. 163-204). John Wiley & Sons, Ltd.

Hernández-Serrano, J., Muñoz, J. L., Bröring, A., Esparza, O., Mikkelsen, L., Schwarzott, W., León, O. ve Zibuschka, J. (2017). On the road to secure and privacy-preserving IoT ecosystems. I. Podnar Žarko, A. Broering, S. Soursos ve M. Serrano (Ed.), Interoperability and Open-Source Solutions for the Internet of Things içinde (ss. 107-122). Springer International Publishing.

Kumar, A., Saha, R., Conti, M., Kumar, G., Buchanan, W. J. ve Kim, T. H. (2022). A comprehensive survey of authentication methods in Internet-of-Things and its conjunctions. Journal of Network and Computer Applications, 204, 103414. DOI: 10.1016/j.jnca.2022.103414.

Kumar, R., Malik, A. ve Ranga, V. (2023). Security concerns over IoT routing using emerging technologies: A review. Transactions on Emerging Telecommunications Technologies, 34(7), e4798. DOI: 10.1002/ett.4798.

Lacava, A., Zottola, V., Bonaldo, A., Cuomo, F. ve Basagni, S. (2022). Securing Bluetooth Low Energy networking: An overview of security procedures and threats. Computer Networks, 211, 108953. DOI: 10.1016/j.comnet.2022.108953.

Latif, S., Djenouri, D., Idrees, Z., Ahmad, J., Mastoi, Q.-U.-A. ve Zou, Z. (2026). Hardware security modules for secure communications in the Industrial Internet of Things. IEEE Communications Surveys & Tutorials, 28, 64-108. DOI: 10.1109/COMST.2025.3600161.

Majeed, A., Patni, S. ve Hwang, S. O. (2025). A comprehensive analysis of privacy-preserving solutions developed for IoT-based systems and applications. Electronics, 14(11), 2106. DOI: 10.3390/electronics14112106.

Oh, S. R. ve Kim, Y. G. (2020). AFaaS: Authorization framework as a service for Internet of Things based on interoperable OAuth. International Journal of Distributed Sensor Networks, 16(2). DOI: 10.1177/1550147720906388.

Ojo, M. O., Giordano, S., Procissi, G. ve Seitanidis, I. N. (2018). A review of low-end, middle-end, and high-end IoT devices. IEEE Access, 6, 70528-70554. DOI: 10.1109/ACCESS.2018.2879615.

Pearson, B., Luo, L., Zou, C., Crain, J., Jin, Y. ve Fu, X. (2020). Building a low-cost and state-of-the-art IoT security hands-on laboratory. A. Casaca, S. Katkoori, S. Ray ve L. Strous (Ed.), Internet of Things: A Confluence of Many Disciplines içinde (ss. 289-306). Springer International Publishing.

Pinto, G. P., Sousa, N. R. ve Prazeres, C. V. S. (2026). My data, my rules: An experimental study on a user-centric approach to data privacy in the Internet of Things. Computing, 108, 33. DOI: 10.1007/s00607-025-01611-y.

Radhakrishnan, I., Jadon, S. ve Honnavalli, P. B. (2024). Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices. Sensors, 24(12), 4008. DOI: 10.3390/s24124008.

Rana, M., Mamun, Q. ve Islam, R. (2022). Lightweight cryptography in IoT networks: A survey. Future Generation Computer Systems, 129, 77-89. DOI: 10.1016/j.future.2021.11.011.

Rao, P. M. ve Deebak, B. D. (2023). A comprehensive survey on authentication and secure key management in Internet of Things: Challenges, countermeasures, and future directions. Ad Hoc Networks, 146, 103159. DOI: 10.1016/j.adhoc.2023.103159.

Ren, W., Tong, X., Du, J., Wang, N., Li, S., Min, G. ve Zhao, Z. (2024). Privacy enhancing techniques in the Internet of Things using data anonymisation. Information Systems Frontiers, 26, 2227-2238. DOI: 10.1007/s10796-021-10116-w.

Sharma, H., Kumar, P. ve Sharma, K. (2025). Advanced security for IoT and smart devices. Emerging Threats and Countermeasures in Cybersecurity içinde (ss. 191-216). John Wiley & Sons, Ltd.

Shukla, R. S., Ather, D. ve Huda, M. (2026). An Internet of Things-based intelligent network traffic analysis framework for anomaly detection. Engineered Science, 41(0), 2355. https://www.espublisher.com/journals/articledetails/2355/.

Trnka, M., Abdelfattah, A. S., Shrestha, A., Coffey, M. ve Cerny, T. (2022). Systematic review of authentication and authorization advancements for the Internet of Things. Sensors, 22(4), 1361. DOI: 10.3390/s22041361.

Yang, X., Karampatzakis, E., Doerr, C. ve Kuipers, F. (2018). Security vulnerabilities in LoRaWAN. 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI) içinde (ss. 129-140). IEEE. DOI: 10.1109/IoTDI.2018.00022.

Zhao, S., Zhang, Q., Hu, G., Qin, Y. ve Feng, D. (2014). Providing root of trust for ARM TrustZone using on-chip SRAM. Proceedings of the 4th International Workshop on

Trustworthy Embedded Devices (TrustED ’14) içinde (ss. 25-36). Association for Computing Machinery.

Ziegeldorf, J. H., Garcia Morchon, O. ve Wehrle, K. (2014). Privacy in the Internet of Things: Threats and challenges. Security and Communication Networks, 7(12), 2728-2742. DOI: 10.1002/sec.795.

Zohourian, A., Dadkhah, S., Neto, E. C. P., Mahdikhani, H., Danso, P. K., Molyneaux, H. ve Ghorbani, A. A. (2023). IoT Zigbee device security: A comprehensive review. Internet of Things, 22, 100791. DOI: 10.1016/j.iot.2023.100791.

IOT' DE YAPAY ZEKA VE VERİ ANALİTİĞİ

Ali Topal[**], İbrahim Arda Çankaya[***]

GİRİŞ

Nesnelerin İnterneti (Internet of Things – IoT), fiziksel dünyadaki nesnelerin sensörler, haberleşme teknolojileri ve gömülü bilgi işlem sistemleri aracılığıyla birbirleriyle ve merkezi ya da dağıtık bilgi sistemleriyle etkileşim kurmasını sağlayan önemli bir teknolojik paradigma haline gelmiştir. Akıllı şehirlerden endüstriyel otomasyona, sağlık sistemlerinden tarıma, ulaşımdan enerji yönetimine kadar genişleyen IoT uygulamaları, fiziksel ortamdan sürekli ve yüksek hacimli veri üretilmesine olanak sağlamaktadır. Buna ek olarak, IoT sistemlerinin sağladığı temel değer yalnızca verinin toplanmasından değil, toplanan verinin anlamlandırılarak karar verme süreçlerinde kullanılabilir bilgiye dönüştürülmesinden kaynaklanmaktadır. Bu nedenle veri analitiği ve yapay zekâ yöntemleri, modern IoT sistemlerinin ayrılmaz birer omurgası durumundadır.

Geleneksel IoT mimarilerinde sensörlerden elde edilen verilerin merkezi sunuculara veya bulut sistemlerine aktarılması, burada işlenmesi ve elde edilen sonuçların ilgili cihazlara geri gönderilmesi yaygın bir yaklaşım olmuştur. Ancak bağlı cihaz sayısının ve üretilen veri miktarının artması; iletişim bant genişliği, gecikme, enerji tüketimi, ölçeklenebilirlik ve veri gizliliği açısından önemli kısıtlar ortaya çıkarmaktadır. Özellikle gerçek zamanlı karar verilmesi gereken endüstriyel kontrol, sağlık izleme, otonom sistemler ve güvenlik uygulamalarında bütün verinin uzak bir bulut altyapısına aktarılması çoğunlukla sürdürülebilir bir çözüm sunmamaktadır. Bu durum, verinin üretildiği noktaya daha yakın işlenmesini sağlayan uç bilişim (edge computing) yaklaşımının ve buna bağlı olarak yapay zekâ işlemlerinin uç cihazlara taşındığı Uç Yapay Zekâ (Edge AI) paradigmasının önemini artırmıştır. Güncel çalışmalar, cihaz üzerinde gerçekleştirilen yapay zekâ işlemlerinin özellikle düşük gecikme, iletişim yükünün azaltılması ve

[**] Öğretim Görevlisi, Isparta Uygulamalı Bilimler Üniversitesi, Uzaktan Eğitim Meslek Yüksekokulu, Bilgisayar Teknolojileri, Isparta, alitopal@isparta.edu.tr, ORCID: orcid.org/0000-0001-5975-5932
[***] Dr. Öğr. Üyesi, Süleyman Demirel Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, Isparta, ardacankaya@sdu.edu.tr, ORCID: https://orcid.org/0000-0001-5375-3573

veri gizliliğinin geliştirilmesi açısından önemli avantajlar sunduğunu göstermektedir (Wang et al., 2025).

Yapay zekâ ile IoT teknolojilerinin bütünleşmesi literatürde giderek daha fazla Nesnelerin Yapay Zekası (Artificial Intelligence of Things, AIoT) kavramı altında ele alınmaktadır. AIoT yaklaşımında IoT altyapısı yalnızca veri üreten ve ileten pasif bir yapı olmaktan çıkarak, veriyi analiz edebilen, örüntüleri belirleyebilen ve belirli ölçüde otonom kararlar verebilen akıllı bir sisteme dönüşmektedir. Makine öğrenmesi ve derin öğrenme yöntemleri; sensör verilerinin sınıflandırılması, anomali tespiti, kestirimci bakım, zaman serisi tahmini, insan aktivitesi tanıma ve çevresel durumların belirlenmesi gibi çok sayıda IoT probleminde kullanılabilmektedir. Özellikle anomali tespitine yönelik güncel çalışmalar, makine öğrenmesi ve derin öğrenme algoritmalarının IoT sistemlerinde yalnızca siber güvenlik amacıyla değil, endüstriyel süreçlerin ve fiziksel sistemlerin durumunun izlenmesinde de önemli bir araç haline geldiğini göstermektedir (Trilles et al., 2024).

IoT sistemlerinde yapay zekâ kullanımını geleneksel veri analitiği uygulamalarından ayıran temel unsurlardan biri, verinin yapısı ve üretildiği ortamdır. IoT verileri sıcaklık, nem, basınç, ivme, açısal hız, konum, ses ve görüntü gibi farklı fiziksel büyüklüklerden elde edilebilmekte; zaman serisi, görüntü, ses veya çok değişkenli sensör akışı biçiminde oluşabilmektedir. Ayrıca farklı örnekleme hızları, eksik gözlemler, sensör gürültüsü ve cihazlar arasındaki donanım farklılıkları veri analizini zorlaştırmaktadır. Bu nedenle başarılı bir yapay zekâ uygulaması yalnızca uygun bir öğrenme algoritmasının seçilmesini değil; veri temizleme, normalizasyon, zaman pencerelerine ayırma, özellik çıkarımı ve gerektiğinde farklı sensörlerden gelen bilgilerin birleştirilmesi gibi veri işleme aşamalarının da birlikte tasarlanmasını gerektirmektedir.

IoT sistemlerinin diğer önemli özelliği ise uç cihazların hesaplama ve enerji kaynaklarının çoğu zaman sınırlı olmasıdır. Geleneksel derin öğrenme modellerinin yüksek bellek ve işlem gereksinimleri, bu modellerin mikrodenetleyiciler ve düşük güçlü IoT cihazları üzerinde doğrudan çalıştırılmasını güçleştirmektedir. Bu gereksinim, sınırlı donanım üzerinde makine öğrenmesi modellerinin çalıştırılmasını amaçlayan TinyML yaklaşımını ön plana çıkarmıştır. TinyML ve Uç Yapay Zekâ sayesinde sensör verilerinin tamamının buluta aktarılması yerine, belirli analizlerin doğrudan veri kaynağında gerçekleştirilmesi mümkün hale gelmektedir. Güncel çalışmalar özellikle anomali tespitinde TinyML tabanlı

yaklaşımların kullanılabildiğini ve sınıflandırma tabanlı yöntemler ile evrişimsel sinir ağlarının bu alanda yaygın biçimde değerlendirildiğini göstermektedir (Trilles et al., 2024).

Buna ek olarak, verinin uç cihazlarda işlenmesi yeni bir problemi beraberinde getirmektedir. Farklı IoT cihazları tarafından üretilen verilerin merkezi bir veri havuzunda toplanması; kişisel verilerin korunması, iletişim maliyeti ve güvenlik açısından her zaman uygun olmayabilir. Bu nedenle Federe Öğrenme (Federated Learning, FL), IoT sistemlerinde dağıtık yapay zekâ için önemli bir yaklaşım olarak ortaya çıkmıştır. Federe öğrenme, ham verinin cihazdan çıkarılmasına gerek kalmadan farklı cihazların ortak bir modelin geliştirilmesine katkıda bulunmasına olanak sağlamaktadır. Böylece dağıtık IoT cihazlarında bulunan verilerden yararlanılırken veri gizliliğinin korunması ve merkezi veri aktarımının azaltılması hedeflenmektedir. Bununla birlikte cihaz heterojenliği, iletişim maliyeti, sınırlı işlem kaynakları ve güvenlik sorunları, IoT ortamlarında federe öğrenme uygulamalarının temel araştırma problemleri arasında yer almaktadır (Li et al., 2024).

IoT, Uç Yapay Zekâ, TinyML ve Federe Öğrenme teknolojilerinin birlikte gelişmesi, veri toplama merkezli geleneksel IoT yaklaşımını giderek dağıtık zekâ anlayışına dönüştürmektedir. Bu dönüşümde amaç yalnızca daha yüksek sınıflandırma veya tahmin başarısı elde etmek değil; aynı zamanda gecikme, enerji tüketimi, iletişim maliyeti, gizlilik ve donanım kaynakları gibi operasyonel kısıtlar altında uygulanabilir yapay zekâ sistemleri geliştirmektir. Nitekim son çalışmalar TinyML ile Federe öğrenmenin birlikte değerlendirilmesinin kaynak kısıtlı IoT cihazlarında gizlilik korumalı ve ölçeklenebilir yapay zekâ uygulamaları açısından önemli bir araştırma yönü oluşturduğunu göstermektedir (Ramadan et al., 2025).

Bu bölümün amacı, IoT sistemlerinde üretilen verilerin yapay zekâ ve veri analitiği yöntemleriyle nasıl işlenebileceğini uçtan uca incelemeyi hedeflemektedir. Bu kapsamda öncelikle IoT veri kaynaklarının özellikleri, veri ön işleme ve özellik çıkarımı süreçleri incelenecek; ardından makine öğrenmesi, derin öğrenme, Uç Yapay Zekâ ve TinyML yaklaşımları değerlendirilecektir. Anomali tespiti, tahminleme, sınıflandırma ile görüntü, ses ve sensör verilerinin analizi temel uygulama alanları üzerinden ele alınacaktır. Ayrıca dağıtık IoT sistemlerinde veri gizliliğini koruyarak ortak model geliştirilmesine olanak sağlayan Federe Öğrenme yaklaşımı

incelenecek ve IoT tabanlı yapay zekâ sistemlerinin mevcut sınırlılıkları ile gelecekteki araştırma yönelimleri tartışılacaktır.

IOT VERİLERİ VE VERİ ANALİTİĞİ

IoT sistemlerinin temel işlevlerinden biri, fiziksel çevrede gerçekleşen olayların sensörler ve bağlı cihazlar aracılığıyla sayısal verilere dönüştürülmesidir. Bu sistemlerde sıcaklık, nem, basınç, hareket, konum, görüntü ve ses gibi farklı niteliklere sahip veriler sürekli veya belirli zaman aralıklarında elde edilebilmektedir. Toplanan veriler, sistemin kullanım amacına bağlı olarak yerel cihazlarda, uç bilgi işlem birimlerinde veya bulut altyapılarında işlenerek anlamlı bilgiye dönüştürülmektedir. Bu açıdan değerlendirildiğinde IoT sistemlerinde veri analitiği; ham verinin toplanmasından başlayarak işlenmesi, analiz edilmesi ve karar verme süreçlerinde kullanılmasına kadar uzanan temel bir süreçtir. IoT'nin yaygınlaşması ve bağlı cihaz sayısının artmasıyla birlikte veri analitiği, yalnızca geçmiş verilerin incelenmesi için değil, gerçek zamanlı olayların belirlenmesi ve sistemlerin değişen koşullara tepki verebilmesi açısından da önem kazanmıştır (Siow et al., 2018).

IoT ortamlarında üretilen veriler, geleneksel veri kümelerinden farklı olarak yüksek hacim, yüksek üretim hızı ve veri çeşitliliği gibi özellikler gösterebilmektedir. Farklı üreticilere ve donanım özelliklerine sahip cihazların aynı sistem içerisinde bulunması ise veri yapılarının, ölçüm hassasiyetlerinin ve örnekleme sıklıklarının farklılaşmasına neden olabilmektedir. Ayrıca fiziksel ortamdan gerçekleştirilen ölçümlerde sensör hataları, haberleşme problemleri ve çevresel etkiler nedeniyle gürültülü, eksik, hatalı veya aykırı gözlemler oluşabilmektedir. IoT sensör verilerinin heterojen, dinamik ve gerçek zamanlı yapısı, büyük miktarda verinin doğrudan analiz edilmesini zorlaştırmaktadır. Bu durum geleneksel veri işleme yaklaşımlarının yanında IoT ortamlarının özelliklerini dikkate alan yöntemlere ihtiyaç doğurmaktadır (Krishnamurthi et al., 2020).

Bu özellikler nedeniyle IoT tabanlı veri analitiği, yalnızca uygun analiz veya yapay zekâ algoritmasının seçilmesinden oluşan bir süreç değildir. Ham verinin analiz öncesinde uygun bir forma dönüştürülmesi; gürültünün azaltılması, eksik değerlerin ele alınması, aykırı gözlemlerin belirlenmesi ve farklı veri kaynaklarının uyumlu hale getirilmesi gibi işlemleri gerekli kılmaktadır. Özellikle makine öğrenmesi tabanlı uygulamalarda verinin temsil biçimi ve veriden çıkarılan özellikler, oluşturulan modelin başarısını doğrudan etkileyebilmektedir. Bu nedenle veri toplama, ön işleme ve özellik

çıkarımı aşamaları, IoT sistemlerinde yapay zekâ tabanlı analizin temelini oluşturmaktadır. IoT verilerinin etkin biçimde analiz edilebilmesi için öncelikle bu verilerin hangi kaynaklardan ve hangi yapılarda elde edildiğinin incelenmesi, ardından ham verinin analiz edilebilir hale getirilmesini sağlayan ön işleme ve özellik çıkarımı yöntemlerinin değerlendirilmesi gerekmektedir (Krishnamurthi et al., 2020).

IoT veri kaynakları ve veri türleri

IoT sistemlerinde veri, fiziksel ortam ile sayısal sistemler arasındaki etkileşimin temelini oluşturmaktadır. Sistemin kullanım amacına bağlı olarak farklı fiziksel büyüklüklerin ölçülmesi için sıcaklık, nem, basınç, ışık, gaz, hareket, yakınlık ve konum sensörleri kullanılabilmektedir. Bunun yanında ivmeölçer ve jiroskop gibi eylemsizlik sensörleri hareket ve yönelim bilgilerinin elde edilmesini sağlarken; mikrofonlar, kameralar ve diğer görüntüleme sistemleri ses, görüntü ve video gibi daha yüksek boyutlu verilerin IoT altyapılarına dahil edilmesine olanak sağlamaktadır. RFID ve benzeri tanımlama teknolojileri ise nesnelerin kimlik ve durum bilgilerinin izlenmesinde kullanılabilmektedir. Dolayısıyla IoT sistemlerinde veri kaynağı yalnızca çevresel ölçüm gerçekleştiren sensörlerden oluşmamakta, fiziksel ortam ve bağlı nesneler hakkında farklı özelliklerde bilgi sağlayan çok çeşitli cihazları kapsamaktadır (Krishnamurthi et al., 2020).

IoT cihazlarından elde edilen veriler yapısal özelliklerine göre yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış veri biçimlerinde bulunabilmektedir. Sıcaklık, nem veya basınç sensörlerinden belirli zaman aralıklarında elde edilen sayısal ölçümler genellikle zaman damgası ve sensör kimliği gibi bilgilerle birlikte yapılandırılmış biçimde saklanabilirken, cihaz günlükleri ve farklı haberleşme sistemlerinden elde edilen kayıtlar yarı yapılandırılmış veri özelliği gösterebilmektedir. Kamera, mikrofon ve video tabanlı IoT cihazlarından elde edilen veriler ise yüksek boyutlu ve yapılandırılmamış veri kaynakları arasında değerlendirilmektedir. Bu çeşitlilik, IoT sistemlerinde kullanılacak depolama ve analiz yöntemlerinin verinin yapısına göre belirlenmesini gerekli kılmaktadır. Özellikle büyük ölçekli IoT uygulamalarında aynı sistem içerisinde yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış verilerin birlikte bulunabilmesi veri yönetimi ve analizi açısından önemli bir gereksinim oluşturmaktadır (Naghib et al., 2023).

Sensör tabanlı IoT uygulamalarında en yaygın veri biçimlerinden biri zaman serisi verileridir. Bir fiziksel büyüklüğün belirli zaman aralıklarında

ölçülmesi sonucunda oluşan bu veriler, gözlemler arasındaki zamansal ilişkinin korunmasını gerektirmektedir. Endüstriyel bir makineden elde edilen titreşim değerleri, bir giyilebilir cihazın ivmeölçer çıktıları, ortam sıcaklığının sürekli izlenmesi veya enerji tüketiminin ölçülmesi bu veri yapısına örnek olarak gösterilebilir. Bunun yanında bazı IoT uygulamalarında sensörler sürekli veri üretmek yerine yalnızca belirli bir olay gerçekleştiğinde veri oluşturabilmektedir. Hareket algılama, kapı açılma bilgisi, belirli bir eşik değerinin aşılması veya cihaz durumundaki değişiklikler bu tür olay tabanlı verilere örnek oluşturmaktadır. Bu nedenle IoT uygulamalarında veri toplama süreci, uygulamanın gereksinimlerine göre sürekli, periyodik veya olay tabanlı biçimde gerçekleştirilebilmektedir.

Gelişmiş IoT sistemlerinde tek bir veri kaynağının kullanılması yerine farklı sensörlerden elde edilen verilerin birlikte değerlendirilmesi de yaygınlaşmaktadır. Örneğin bir insan aktivitesinin belirlenmesinde ivmeölçer ve jiroskop verileri birlikte kullanılabilirken, akıllı ulaşım sistemlerinde kamera, konum, hız ve çevresel sensörlerden elde edilen bilgiler aynı analiz sürecine dahil edilebilmektedir. Farklı kaynaklardan gelen verilerin bir arada kullanılması, fiziksel olayların daha kapsamlı biçimde temsil edilmesini sağlayabilmektedir. Bununla birlikte farklı örnekleme hızları, veri biçimleri, ölçüm birimleri ve sensör doğrulukları nedeniyle heterojen veri kaynaklarının birlikte değerlendirilmesi ek veri işleme gereksinimleri ortaya çıkarmaktadır. Bu nedenle sensör verilerinin yapısının ve özelliklerinin doğru biçimde belirlenmesi, sonraki aşamalarda uygulanacak ön işleme, özellik çıkarımı ve yapay zekâ yöntemlerinin seçimi açısından önemli bir başlangıç noktasıdır (Krishnamurthi et al., 2020).

IoT verilerinin ön işlenmesi ve özellik çıkarımı

IoT cihazlarından elde edilen ham veriler çoğu zaman doğrudan veri analizi veya yapay zekâ algoritmalarında kullanılabilecek nitelikte değildir. Sensörlerin ölçüm hassasiyeti, çevresel etkiler, haberleşme problemleri, cihaz arızaları ve farklı örnekleme özellikleri; veriler içerisinde gürültü, eksik gözlem, aykırı değer ve tekrarlı kayıtların oluşmasına neden olur. Özellikle gerçek zamanlı ve sürekli veri üreten IoT sistemlerinde bu problemlerin zaman içerisinde birikmesi, veri kalitesinin düşmesine ve geliştirilen yapay zekâ modellerinin başarısının olumsuz etkilenmesine yol açabilir. Bu nedenle ham sensör verilerinin analiz öncesinde temizlenmesi ve uygun bir veri yapısına dönüştürülmesi, IoT tabanlı veri analitiğinin temel aşamalarından biridir (Krishnamurthi et al., 2020).

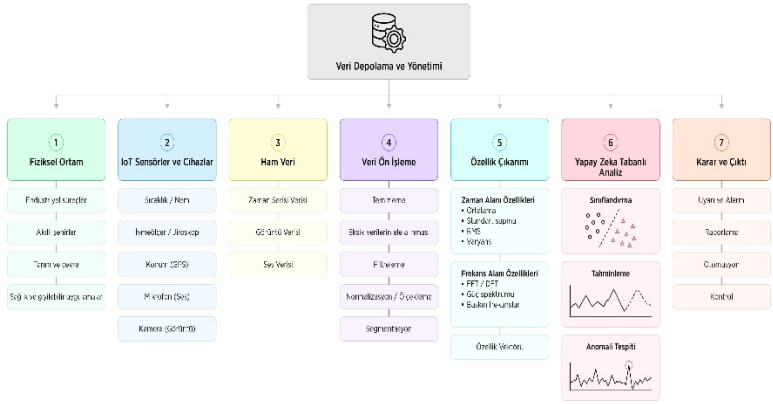
IoT verilerinin ön işlenmesinde gerçekleştirilecek işlemler verinin türüne ve uygulamanın gereksinimlerine göre değişmekle birlikte; gürültünün giderilmesi, eksik verilerin ele alınması, aykırı değerlerin belirlenmesi, veri dönüşümü ve ölçeklendirme yaygın olarak kullanılan işlemler arasında yer almaktadır. Sensör ölçümlerindeki istenmeyen değişimlerin azaltılması amacıyla hareketli ortalama ve farklı sayısal filtreleme yöntemlerinden yararlanılabilirken, eksik gözlemler silme veya uygun değerlerle tamamlama yöntemleri kullanılarak ele alınabilmektedir. Bununla birlikte sensörlerden elde edilen değerlerin farklı ölçüm aralıklarına sahip olması, özellikle uzaklık veya gradyan temelli makine öğrenmesi algoritmalarında bazı değişkenlerin diğerlerine göre daha baskın hale gelmesine neden olabilmektedir. Bu nedenle verilerin belirli bir aralığa dönüştürülmesini sağlayan normalizasyon veya ortalama ve standart sapma dikkate alınarak gerçekleştirilen standardizasyon işlemleri, yapay zekâ modellerinin eğitimi öncesinde uygulanabilmektedir.

Sürekli veri akışı oluşturan sensörlerde önemli ön işleme aşamalarından biri de verinin belirli zaman aralıklarına bölünmesidir. Özellikle ivmeölçer, jiroskop, titreşim, biyomedikal sinyal ve benzeri zaman serilerinde tek bir ölçüm değeri çoğu zaman incelenen fiziksel olayın belirlenmesi için yeterli bilgi içermemektedir. Bu nedenle sürekli sensör akışı, sabit veya örtüşen zaman pencerelerine ayrılarak her pencerenin belirli bir olay veya aktiviteyi temsil etmesi sağlanır. Kullanılan pencerenin uzunluğu ve ardışık pencereler arasındaki örtüşme oranı, elde edilen veri örneklerinin yapısını ve sonraki sınıflandırma veya tahminleme performansını etkileme ihtimali oluşturur. Böylece sürekli biçimde üretilen ham sensör akışı, makine öğrenmesi algoritmalarının işleyebileceği örneklere dönüşüm sağlar.

Ön işleme sonrasında gerçekleştirilen özellik çıkarımı, ham verinin incelenen problemi temsil edebilecek daha anlamlı değişkenlere dönüştürülmesini amaçlamaktadır. Sensör tabanlı zaman serilerinde ortalama, standart sapma, minimum, maksimum, medyan, varyans ve kök ortalama kare (Root Mean Square – RMS) gibi zaman alanı özellikleri kullanılmaktadır. Bunun yanında sinyalin frekans bileşenlerinin önemli olduğu titreşim, hareket veya ses tabanlı uygulamalarda Fourier dönüşümü gibi yöntemler aracılığıyla frekans alanı özellikleri elde edebilir. Bu işlemler sonucunda yüksek miktarda ham sensör örneği, incelenen olayın karakteristik özelliklerini temsil eden daha sınırlı bir özellik kümesine dönüştürülebilmektedir. Özellik çıkarımı özellikle geleneksel makine

öğrenmesi yöntemlerinde model başarısını etkileyen önemli aşamalardan biri olurken, derin öğrenme yaklaşımlarında ilgili özelliklerin bir bölümü doğrudan ham veya sınırlı ölçüde işlenmiş veriden model tarafından öğrenmeye çalışır.

IoT veri analitiğinde uygulanacak ön işleme ve özellik çıkarımı adımlarının tüm uygulamalar için sabit bir işlem dizisi olarak değerlendirilmesi uygun değildir. Kullanılan sensörün özellikleri, örnekleme frekansı, analiz edilecek fiziksel olay ve uygulanacak yapay zekâ yöntemi, veri işleme sürecinin tasarımını doğrudan etkilemektedir. Bunun yanında uç cihazlarda gerçekleştirilen analizlerde bellek, işlem gücü ve enerji tüketimi gibi donanımsal kısıtların da dikkate alınması gerekmektedir. Bu nedenle IoT sistemlerinde veri ön işleme ve özellik çıkarımı, yalnızca veri kalitesini artıran bir hazırlık aşaması değil, geliştirilecek yapay zekâ sisteminin doğruluğunu, hesaplama maliyetini ve gerçek zamanlı uygulanabilirliğini etkileyen temel bir tasarım bileşeni olarak değerlendirilmelidir. IoT tabanlı veri işleme ve yapay zekâ destekli analiz sürecinin genel mimarisi Şekil 1'de gösterilmiştir.



Şekil 1. IoT tabanlı veri işleme ve yapay zekâ destekli analiz sürecinin genel mimarisi

IOT'DE YAPAY ZEKÂ VE MAKİNE ÖĞRENMESİ

IoT sistemlerinde üretilen verinin hacmi, çeşitliliği ve sürekliliği arttıkça, bu verilerin yalnızca geleneksel istatistiksel yöntemlerle analiz edilmesi yerine veriden otomatik olarak örüntü öğrenebilen yapay zekâ yöntemlerinden yararlanılması önem kazanmıştır. Makine öğrenmesi, IoT

cihazlarından elde edilen veriler içerisindeki ilişkilerin ve örüntülerin belirlenmesini sağlayarak sistemlerin yalnızca veri toplayan yapılardan, çevresindeki değişimleri algılayabilen ve elde ettiği bilgiler doğrultusunda karar verebilen akıllı sistemlere dönüşmesine katkı sağlamaktadır. Bu kapsamda makine öğrenmesi yöntemleri; sensör verilerinin sınıflandırılması, gelecekteki değerlerin tahmin edilmesi, normal davranıştan sapmaların belirlenmesi ve farklı fiziksel durumların tanınması gibi çok sayıda IoT probleminde kullanılabilmektedir. IoT verilerinin akıllı biçimde işlenmesinin, ham veriden daha yüksek düzeyde bilgi çıkarılmasında temel bir rol oynadığı literatürde uzun süredir vurgulanmaktadır (Mahdavinejad et al., 2018).

IoT sistemlerinde kullanılabilecek yapay zekâ yaklaşımının belirlenmesi, yalnızca elde edilecek tahmin veya sınıflandırma başarısına bağlı değildir. Verinin türü ve miktarı, etiketli örneklerin bulunabilirliği, modelin eğitileceği ve çalıştırılacağı donanım, gerçek zamanlı çalışma gereksinimi, enerji tüketimi ve iletişim maliyeti gibi faktörler de yöntem seçiminde etkilidir. Örneğin sınırlı miktarda etiketli sensör verisinin bulunduğu bir uygulamada geleneksel makine öğrenmesi yöntemleri yeterli olabilirken, görüntü, ses veya yüksek boyutlu zaman serilerinin analiz edildiği uygulamalarda derin öğrenme yöntemleri daha uygun sonuçlar üretebilir. Ayrıca modelin doğrudan kaynakları sınırlı bir IoT cihazında çalıştırılması gerektiğinde, model doğruluğunun yanı sıra bellek kullanımı, işlem yükü ve enerji tüketiminin de değerlendirilip sonuca ulaştırılması gerekir. Özellikle cihaz üzerinde gerçekleştirilen makine öğrenmesi uygulamalarında bu kaynak kısıtları önemli bir tasarım problemi oluşturmaktadır (Tekin et al., 2024).

Yapay zekâ işlemlerinin IoT mimarisi içerisinde gerçekleştirildiği konum da sistemin tasarımını doğrudan etkilemektedir. Geleneksel yaklaşımda sensörlerden toplanan veriler bulut veya merkezi sunucu altyapılarına aktarılırken, uç cihazların işlem kapasitelerinin gelişmesiyle birlikte bazı yapay zekâ modellerinin verinin üretildiği noktada veya bu noktaya yakın uç bilgi işlem cihazlarında çalıştırılması mümkün hale gelmiştir. Böylece IoT tabanlı yapay zekâ sistemleri, merkezi bulut tabanlı analizden Uç Yapay Zekâ ve kaynakları daha sınırlı cihazları hedefleyen TinyML uygulamalarına kadar farklı hesaplama katmanlarında gerçekleştirilebilmektedir. Güncel çalışmalar, IoT ve kablosuz sensör ağlarında makine öğrenmesi ve derin öğrenmenin kullanımının yalnızca model başarısı açısından değil; hesaplama

maliyeti, enerji verimliliği, gecikme ve donanım kısıtlarıyla birlikte değerlendirilmesi gerektiğini göstermektedir (Egwuche et al., 2026).

Bu doğrultuda IoT sistemlerinde yapay zekâ tabanlı veri analizinin değerlendirilmesi için öncelikle geleneksel makine öğrenmesi yaklaşımlarının ve derin öğrenme yöntemlerinin IoT verileri üzerindeki kullanımının incelenmesi gerekmektedir. Bunun ardından yapay zekâ modellerinin merkezi sistemlerden uç cihazlara taşınmasını sağlayan Uç Yapay Zekâ ve TinyML yaklaşımları, IoT sistemlerinin gerçek zamanlılık ve kaynak kısıtları açısından ele alınacaktır.

Makine öğrenmesi yaklaşımları

Makine öğrenmesi, IoT sistemlerinden elde edilen veriler içerisindeki örüntülerin ve ilişkilerin belirlenerek sınıflandırma, tahminleme veya karar verme amacıyla kullanılmasını sağlayan temel yapay zekâ yaklaşımlarından biridir. IoT uygulamalarında kullanılacak öğrenme yöntemi; verinin etiketli olup olmaması, çözülmek istenen problemin niteliği ve elde edilmek istenen çıktıya göre değişmektedir. Makine öğrenmesi yöntemleri genel olarak denetimli, denetimsiz ve pekiştirmeli öğrenme olmak üzere üç temel grupta ele alınabilmektedir. IoT veri analitiğinde özellikle denetimli ve denetimsiz öğrenme yöntemleri yaygın olarak kullanılmakta ve farklı sensör verilerinden daha yüksek düzeyde bilgi elde edilmesine olanak sağlamaktadır (Mahdavinejad et al., 2018).

Denetimli öğrenmede model, giriş verileri ile bu verilere karşılık gelen etiketlerin birlikte bulunduğu bir eğitim veri kümesi kullanılarak geliştirilmektedir. Modelin amacı, eğitim sürecinde öğrendiği ilişkilerden yararlanarak daha önce karşılaşmadığı verilerin sınıfını veya sayısal çıktı değerini tahmin etmektir. Çıktının belirli kategorilerden oluştuğu problemler sınıflandırma, sürekli bir sayısal değerin tahmin edildiği problemler ise regresyon kapsamında değerlendirilmektedir. IoT uygulamalarında insan aktivitesinin sensör verilerinden belirlenmesi, cihazların çalışma durumlarının sınıflandırılması veya ağ trafiğinin normal ve anormal olarak ayrılması sınıflandırma problemlerine; enerji tüketimi, çevresel ölçümler veya gelecekteki sensör değerlerinin kestirilmesi ise regresyon problemlerine örnek olarak verilebilir. Destek Vektör Makineleri (Support Vector Machines – SVM), karar ağaçları, Random Forest ve k-En Yakın Komşu (k-Nearest Neighbors – k-NN) gibi algoritmalar IoT verilerinin denetimli öğrenme analizinde kullanılabilmektedir. Mahdavinejad et al. (2018), IoT veri analitiğinde farklı makine öğrenmesi yöntemlerini sınıflandırmakta ve

SVM' nin akıllı şehir trafik verileri üzerindeki kullanımını da örnek bir uygulama olarak ele almaktadır.

Denetimsiz öğrenmede ise eğitim verileri için önceden belirlenmiş sınıf etiketlerine ihtiyaç duyulmaz. Bunun yerine algoritma, veriler arasındaki benzerlikleri, grupları veya olağan veri davranışından farklılaşan örnekleri belirlemeye çalışır. K-means ve hiyerarşik kümeleme gibi yöntemlerle benzer sensör gözlemlerinin gruplandırılması, büyük miktarda etiketlenmemiş IoT verisinin yapısının incelenmesine olanak sağlar. Bu özellik, sürekli veri üreten ve bütün gözlemlerin insanlar tarafından etiketlenmesinin pratik olmadığı IoT ortamları açısından önemlidir. Denetimsiz öğrenme yöntemleri özellikle veri içerisindeki doğal grupların keşfedilmesi, veri temsilinin oluşturulması ve olağandışı davranışların belirlenmesi gibi amaçlarla uygulamalarda yer edinir. Denetimli ve denetimsiz yöntemlerin farklı IoT uygulamalarında kullanılması, makine öğrenmesinin heterojen IoT verilerinden anlamlı bilgi çıkarılması açısından geniş bir uygulama alanına sahip olduğunu göstermektedir (Gherbi et al., 2023).

Pekiştirmeli öğrenme (Reinforcement Learning – RL), denetimli ve denetimsiz öğrenmeden farklı olarak bir ajanın bulunduğu ortamla etkileşim kurarak gerçekleştirdiği eylemler sonucunda aldığı ödül veya ceza üzerinden uygun davranış stratejisini öğrenmesine dayanmaktadır. Bu yaklaşım, yalnızca mevcut verinin analiz edilmesinden ziyade sistemin zaman içerisinde hangi eylemleri gerçekleştirmesi gerektiğinin öğrenilmesini gerektiren IoT uygulamalarında kullanılabilmektedir. Kaynak tahsisi, haberleşme yönetimi, enerji optimizasyonu ve otonom kontrol gibi dinamik karar verme problemleri bu kapsamda değerlendirilebilir. Ayrıca pekiştirmeli öğrenme, etkileşimli öğrenme süreci ve hesaplama gereksinimleri nedeniyle klasik sınıflandırma ve regresyon yöntemlerinden farklı tasarım gereksinimlerine sahiptir. Makine öğrenmesinin üç temel öğrenme kategorisinden biri olmakla birlikte, IoT verilerinin analizine yönelik çalışmalarda denetimli ve denetimsiz öğrenmenin daha yaygın biçimde ele alındığı görülmektedir (Mahdavinejad et al., 2018).

IoT sistemlerinde makine öğrenmesi algoritmasının seçimi yalnızca tahmin başarısına göre yapılması doğru bir yaklaşım değildir. IoT cihazlarının sınırlı işlem gücü, bellek kapasitesi ve enerji kaynakları nedeniyle modelin hesaplama karmaşıklığı, eğitim ve çıkarım süresi ile kaynak tüketimi de dikkate alınmalıdır. Modelin bulut altyapısı yerine

özellikle doğrudan uç cihaz üzerinde çalıştırılacağı uygulamalarda daha düşük hesaplama maliyetine sahip algoritmalar önemli avantajlar sağlayabilir. Buna ek olarak IoT verilerinin zaman içerisinde değişmesi, farklı cihazlardan elde edilen verilerin aynı dağılıma sahip olmaması ve etiketli veri elde etmenin güçlüğü, geliştirilecek makine öğrenmesi sistemlerinin gerçek ortam koşullarındaki performansını değiştirebilir. Bu nedenle IoT için uygun makine öğrenmesi yaklaşımının belirlenmesiyle model doğruluğu ile sistemin operasyonel gereksinimleri arasında optimize bir eşitlik sağlanmalıdır.

Derin öğrenme yaklaşımları

Derin öğrenme, çok katmanlı yapay sinir ağları aracılığıyla veriler içerisindeki karmaşık ve doğrusal olmayan ilişkilerin öğrenilmesini sağlayan makine öğrenmesi yaklaşımıdır. Geleneksel makine öğrenmesi yöntemlerinde model başarısı çoğu zaman veri üzerinden uygun özelliklerin önceden belirlenmesine ve çıkarılmasına bağlıyken, derin öğrenme modelleri yüksek boyutlu verilerden probleme ilişkin temsilleri öğrenme yeteneğine sahiptir. Bu özellik; görüntü, ses ve çok değişkenli sensör akışları gibi karmaşık veri türlerinin üretildiği IoT uygulamalarında derin öğrenmenin kullanımını daha değerli yapmıştır. Yüksek miktarda eğitim verisine ve hesaplama kaynağına duyulan gereksinim, özellikle kaynakları sınırlı IoT ortamlarında derin öğrenmenin uygulanmasında dikkate alınması gereken temel unsurlardan kabul edilir (Mohammadi et al., 2018).

Evrişimsel Sinir Ağları (Convolutional Neural Networks – CNN), özellikle uzamsal veya yerel örüntülerin belirlenmesinde kullanılan derin öğrenme mimarilerindendir. CNN modelleri görüntü verilerinin analizinde yaygın olarak kullanılmakla birlikte, tek boyutlu evrişim işlemleri aracılığıyla sensör ve zaman serisi verilerine de uygulanabilmektedir. Akıllı kamera sistemlerinde nesne veya durumların belirlenmesi, endüstriyel IoT uygulamalarında makine durumunun izlenmesi ve giyilebilir sensörlerden elde edilen hareket verilerinin sınıflandırılması CNN tabanlı yaklaşımların kullanılabileceği uygulamalara örnektir. Ham veya sınırlı ölçüde işlenmiş veriden ayırt edici özelliklerin öğrenilmesine olanak sağlaması bu modellerin önemli özelliklerinden biridir. Böylelikle geleneksel makine öğrenmesi süreçlerinde gerçekleştirilen bazı manuel özellik çıkarımı işlemleri modelin öğrenme sürecinin bir parçası haline getirilebilmektedir (Mohammadi et al., 2018).

IoT sistemlerinde sensör verilerinin önemli bir bölümü zamana bağlı olarak üretildiğinden, ardışık gözlemler arasındaki ilişkinin modellenmesi de önem taşımaktadır. Tekrarlayan Sinir Ağları (Recurrent Neural Networks – RNN), önceki zaman adımlarından elde edilen bilgilerin sonraki hesaplamalarda kullanılmasına olanak sağlayarak sıralı verilerin analiz edilmesini sağlar. Klasik RNN yapılarında uzun süreli bağımlılıkların öğrenilmesinde ortaya çıkabilen problemlerin azaltılması amacıyla Long Short-Term Memory (LSTM) ve Gated Recurrent Unit (GRU) gibi mimariler geliştirilmiştir. Bu modeller; enerji tüketiminin tahmin edilmesi, endüstriyel sensör değerlerinin izlenmesi, çevresel ölçümlerin tahmin edilmesi ve insan aktivitelerinin zamansal sensör verilerinden belirlenmesi gibi IoT uygulamalarında kullanılmaktadır. Özellikle geçmiş durumların gelecekteki sistemin davranışı üzerinde etkili olduğu problemlerde zamansal ilişkilerin modellenmesi, yalnızca bağımsız gözlemlerin değerlendirilmesine kıyasla daha kapsamlı veri temsillerinin oluşturulmasına olanak sağlaması onu daha da önemli kılmaktadır.

Son yıllarda Transformer tabanlı mimariler de zaman serisi analizi ve IoT verilerinin işlenmesinde değerlendirilmeye başlanmıştır. Transformer mimarisinin temelinde bulunan dikkat mekanizması, bir veri dizisinin farklı bölümleri arasındaki ilişkilerin öğrenilmesine olanak sağlamaktadır. RNN ve LSTM gibi tekrarlayan yapılardan farklı olarak dizinin ardışık biçimde işlenmesi zorunluluğunu azaltması ve uzun dönemli ilişkileri modelleyebilmesi, Transformer tabanlı yöntemlerin özellikle uzun zaman serilerinde kullanılmasını olanak sağlamaktadır. IoT açısından bu yaklaşımlar; çok değişkenli sensör verilerinin analizi, zaman serisi tahmini ve anomali tespiti gibi problemlerde değerlendirilebilmektedir. Bununla birlikte Transformer modellerinin hesaplama ve bellek gereksinimleri, kaynakları sınırlı IoT cihazlarında doğrudan uygulanmalarını güçleştirebilmektedir (Wen et al., 2022).

Derin öğrenme yöntemlerinin IoT sistemlerine uygulanmasında kullanılacak mimarinin veri türü ve çözülmek istenen problemle uyumlu olması gerekmektedir. Görüntü ve yerel örüntülerin önemli olduğu sensör verilerinde CNN tabanlı modeller, zamansal bağımlılıkların bulunduğu veri akışlarında RNN, LSTM ve GRU yapıları, uzun süreli ilişkilerin ve karmaşık zaman serilerinin modellenmesinde ise Transformer tabanlı yaklaşımlar tercih edilir. Bununla birlikte daha karmaşık bir modelin her IoT problemi için daha başarılı veya daha uygun olacağını düşünmek doğru değildir.

Modelin doğruluğunun yanı sıra çıkarım süresi, bellek gereksinimi, enerji tüketimi ve çalıştırılacağı donanım platformu da sistem tasarımının bir parçası olarak değerlendirilmeli ve ona göre karar verilmelidir. Bu gereksinimler, derin öğrenme modellerinin bulut altyapılarından veri kaynağına daha yakın hesaplama birimlerine taşınmasını ve kaynak kısıtlı cihazlarda çalışabilecek daha küçük modellerin geliştirilmesini önemli hale getirmekte; böylece Uç Yapay Zekâ ve TinyML yaklaşımlarına devam edebilmek için altyapı oluşturur.

Edge AI ve TinyML

IoT sistemlerinde yapay zekâ tabanlı veri analizinin geleneksel uygulama biçiminde, sensörlerden elde edilen veriler ağ üzerinden merkezi sunuculara veya bulut altyapılarına aktarılmakta ve yapay zekâ modellerinin çalıştırılması bu sistemler üzerinde gerçekleştirilmektedir. Bulut sistemlerinin yüksek hesaplama kapasitesi karmaşık modellerin çalıştırılmasına olanak sağlar. Fakat sürekli veri üreten IoT cihazlarının bütün veriyi merkezi sistemlere aktarması iletişim yükü, gecikme, bağlantı bağımlılığı ve veri gizliliği gibi problemlere neden olur. Özellikle gerçek zamanlı cevaba ihtiyaç duyulmayan uygulamalarda sensör verisinin uzak bir sisteme gönderilmesi, işlenmesi ve elde edilen sonucun tekrar cihaza geri gönderilmesi sonuç süresini uzatabilir. Bu kısıtlar, yapay zekâ işlemlerinin verinin üretildiği noktaya daha yakın gerçekleştirilmesini amaçlayan Uç Yapay Zekâ yaklaşımının gelişmesinde etkili olmuştur (Wang et al., 2025).

Uç Yapay Zekâ, yapay zekâ modellerinin merkezi bulut altyapılarına tamamen bağımlı olmadan, veri kaynağına yakın uç bilgi işlem sistemlerinde çalıştırılması anlamına gelmektedir. Bu uç sistemler IoT cihazının kendisi olabileceği gibi ağ geçitleri, tek kartlı bilgisayarlar, akıllı telefonlar veya farklı gömülü bilgi işlem platformları da olabilmektedir. Verinin yerel olarak işlenmesi sayesinde yalnızca analiz sonucunun veya gerekli bilgilerin ağ üzerinden aktarılması mümkün olmakta, böylece iletişim trafiği ve bulut sistemlerine olan bağımlılık azaltılabilmektedir. Ayrıca görüntü, ses veya kişisel sensör verileri gibi hassas bilgilerin cihazdan çıkarılmadan analiz edilebilmesi veri gizliliği açısından da avantaj sağlamaktadır. Düşük gecikme, gerçek zamanlı çalışma ve bağlantının sınırlı veya kesintili olduğu ortamlarda belirli işlevlerin sürdürülebilmesi de Uç Yapay Zekânın IoT sistemleri açısından önemli avantajları arasında bulunmaktadır (Wang et al., 2025).

Tiny Machine Learning (TinyML) ise yapay zekâ işlemlerinin daha sınırlı işlem gücü, bellek ve enerji kapasitesine sahip mikrodenetleyici tabanlı cihazlarda çalışmasına odaklanmıştır. Bu açıdan TinyML, Uç Yapay Zekâ yaklaşımının kaynak kısıtlarının daha çok olduğu uç noktalara çalıştırılması olarak düşünülebilir. Uç Yapay Zekâ kapsamında görece yüksek hesaplama kapasitesine sahip uç sunucular veya tek kartlı bilgisayarlar kullanılabilirken, TinyML uygulamalarında yapay zekâ modeli çoğunlukla sensöre doğrudan bağlı mikrodenetleyiciler üzerinde işlemlerine devam etmektedir. TinyML'nin IoT ile bütünleşmesi özellikle düşük gecikme, bant genişliğinin etkin kullanımı, gizlilik ve sürekli bulut bağlantısına duyulan gereksinimin azaltılması açısından önemli olanaklar sunmaktadır (Dutta & Bharali, 2021).

Ayrıca, bulut ortamında veya güçlü uç sistemlerde çalıştırılabilen yapay zekâ modellerinin mikrodenetleyicilere doğrudan aktarılması zordur. Mikrodenetleyiciler sınırlı RAM ve program belleğinin yanı sıra düşük işlem kapasitesi ve enerji tüketimi gereksinimlerine sahiptir. Bu nedenle TinyML uygulamalarında model boyutunun ve hesaplama maliyetinin azaltılması önemli bir tasarım problemi oluşturur. Model kuantizasyonu, budama ve farklı model sıkıştırma yöntemleri bu amaçla kullanılabilmektedir. Kuantizasyon, model parametrelerinin daha düşük sayısal hassasiyetle temsil edilerek bellek ve hesaplama gereksiniminin azaltılmasını amaçlarken; budama, model içerisindeki etkisi düşük bağlantı veya parametrelerin kaldırılması yoluyla daha küçük modeller oluşturulmasını sağlamaktadır. Böylece model doğruluğunda kabul edilebilir düzeyde bir performans korunurken kaynak tüketiminin azaltılması hedeflenmektedir. TinyML literatüründe model tasarımı ile donanım ve yazılım özelliklerinin birlikte değerlendirilmesi, kaynak kısıtlı cihazlarda yapay zekânın uygulanabilirliği açısından temel gereksinimlerden biri olarak ele alınmaktadır (Capogrosso et al., 2024).

Uç Yapay Zekâ ve TinyML yaklaşımları, farklı IoT uygulamalarında verinin üretildiği noktada analiz gerçekleştirilmesine olanak sağlamaktadır. Örneğin endüstriyel bir sistemde titreşim veya ses sensörlerinden elde edilen veriler yerel olarak analiz edilerek makinedeki olağandışı davranışlar belirlenebilir; giyilebilir bir cihazda hareket sensörleri kullanılarak fiziksel aktiviteler sınıflandırılabilir veya çevresel izleme sistemlerinde sensör ölçümlerindeki anomaliler merkezi sisteme sürekli ham veri gönderilmeden tespit edilebilir. TinyML tabanlı anomali tespit çalışmalarının sistematik olarak incelendiği güncel araştırmalar, sınıflandırma yöntemlerinin ve CNN

tabanlı modellerin bu amaçla kullanılabildiğini ve mikrodenetleyici tabanlı platformlarda gerçek uygulamaların geliştirildiğini göstermektedir (Trilles et al., 2024).

IoT sistemlerinde kullanılan yapay zekâ yöntemleri; işlenen veri türü, çözülmek istenen problem ve hesaplama gereksinimleri açısından aynı özelliklere sahip değildir. Geleneksel makine öğrenmesi yöntemleriyle yapılandırılmış sensör verilerinde daha düşük hesaplama maliyetleri uygulanabilirken, derin öğrenme yöntemleriyle görüntü, ses ve yüksek boyutlu zaman serilerindeki karmaşık örüntülerin öğrenilmesinde önemli avantajlar sağlayabilmektedir. Bununla birlikte modelin uç cihazlarda çalıştırılması sadece tahmin başarısı olarak değerlendirilmemektedir. Ek olarak, bellek kullanımı, hesaplama maliyeti ve enerji tüketiminin de dikkate alınmaktadır. Bu bölümde ele alınan temel yapay zekâ yaklaşımlarının IoT sistemleri açısından genel özellikleri Tablo 1'de özetlenmiştir.

Tablo 1. IoT sistemlerinde yapay zekâ yaklaşımlarının genel karşılaştırılması

Yaklaşım	Uygun veri/problem	Temel avantaj	IoT açısından kullanım
SVM	Sensör verisi, sınıflandırma	Küçük ve orta verilerde etkili	Aktivite/durum sınıflandırma
Random Forest	Yapılandırılmış sensör verisi	Dayanıklı	Anomali ve durum tespiti
k-NN	Sınıflandırma	Eğitim maliyeti düşük	Sensör örüntüsü sınıflandırma
CNN	Görüntü, ses, sensör sinyali	Yerel örüntüleri otomatik öğrenme	Görüntü/sinyal analizi
LSTM/GRU	Zaman serileri	Zamansal bağımlılıkları modelleme	Tahminleme, aktivite tanıma
Transformer	Çok değişkenli zaman serileri	Uzun dönemli ilişkileri modelleme	Tahminleme, anomali tespiti
TinyML Modelleri	Kaynak kısıtlı sensör sistemleri	Düşük güçte yerel çıkarım	Cihaz üzerinde AI

IOT'DE YAPAY ZEKÂ TABANLI VERİ ANALİZİ UYGULAMALARI

IoT sistemlerinde yapay zekâ ve veri analitiği yöntemlerinin kullanımı, sensörler ve bağlı cihazlar tarafından üretilen ham verilerin fiziksel sistemlerin durumunu temsil eden anlamlı bilgilere dönüştürülmesine olanak sağlar. Geçmiş bölümlerde ele alınan makine öğrenmesi ve derin öğrenme yöntemleri, IoT uygulamalarında farklı amaçlarla kullanılabilmekte; kullanılacak yaklaşım verinin türüne, sistemden beklenen çıktıya ve uygulamanın operasyonel gereksinimlerine göre değişmektedir. Bu çerçevede yapay zekâ tabanlı IoT uygulamaları yalnızca belirli bir durumun sınıflandırılmasıyla sınırlı değildir. Ayrıca olağandışı davranışların belirlenmesi, gelecekteki sistem durumlarının tahmin edilmesi ve farklı veri kaynaklarından elde edilen bilgilerin birlikte değerlendirilmesi gibi çeşitli analiz görevlerini kapsamaktadır (Mahdavinejad et al., 2018).

IoT sistemlerinde gerçekleştirilen analizler, toplanacak bilgi açısından farklı kategoriler altında değerlendirilir. Sınıflandırma yöntemleri sensör gözlemlerinin önceden tanımlanmış sınıflara atanmasını sağlarken, tahminleme yöntemleri geçmiş ve mevcut gözlemlerden yararlanarak gelecekteki değerlerin veya sistem durumlarının kestirilmesini amaçlar. Anomali tespiti ise sistemin normal davranışından farklılaşan gözlem veya örüntülerin belirlenmesine odaklanır. Bu görevler birbirlerinden tamamen bağımsız değildir. Örneğin endüstriyel bir IoT sisteminde sensör verileri kullanılarak mevcut makine durumu sınıflandırılabilir, olağandışı çalışma davranışı tespit edilebilir ve geçmiş gözlemlerden yararlanılarak gelecekte meydana gelebilecek bir arıza tahmin edilebilir. Bu nedenle aynı IoT altyapısı içerisinde farklı yapay zekâ görevlerinin birbirini tamamlayacak biçimde kullanılması mümkündür.

IoT veri analitiğinin kapsamı kullanılan veri kaynaklarının çeşitlenmesiyle daha da genişlemektedir. Sayısal sensör ölçümlerinin yanı sıra kameralar, mikrofonlar, giyilebilir cihazlar ve farklı algılama sistemlerinden elde edilen görüntü, ses ve çok değişkenli sensör akışları yapay zekâ modelleriyle analiz edilebilmektedir. Ayrıca tek bir sensörün sağlayamadığı bilginin elde edilmesi amacıyla farklı veri kaynaklarının birlikte değerlendirilmesi, sensör füzyonu ve çok modlu veri analizi yaklaşımlarının IoT sistemlerindeki önemini artırmaktadır (Krishnamurthi et al., 2020). Bu doğrultuda IoT sistemlerinde yapay zekâ tabanlı veri analizinin başlıca uygulamaları; anomali tespiti, tahminleme, sınıflandırma,

görüntü, ses ve sensör verilerinin analizi ile farklı veri kaynaklarının birlikte değerlendirildiği çok modlu analiz ve sensör füzyonu başlıkları altında incelenebilir.

Anomali tespiti

Anomali tespiti, IoT sistemlerinden elde edilen veriler içerisinde sistemin normal çalışma davranışından önemli ölçüde farklılaşan gözlem, olay veya örüntülerin belirlenmesini amaçlamaktadır. IoT cihazlarının fiziksel sistemlerle sürekli etkileşim halinde olması, anomali tespitini hem sistem güvenliği hem de operasyonel durumların izlenmesi açısından önemli hale getirmektedir. Bir sensör değerindeki beklenmeyen değişim; cihaz arızası, sensör hatası, çevresel koşullardaki olağandışı bir değişiklik, endüstriyel süreçte meydana gelen bozulma veya siber saldırı gibi farklı nedenlerden kaynaklanabilir. Bu nedenle IoT ortamlarında anomali tespiti yalnızca ağ güvenliği kapsamında değil, fiziksel süreçlerin ve cihaz durumlarının izlenmesinde de kullanılmaktadır. Güncel çalışmalar makine öğrenmesi ve derin öğrenme yöntemlerinin IoT tabanlı anomali tespitinde yaygın biçimde kullanıldığını göstermektedir (Trilles et al., 2024).

Anomaliler, veride ortaya çıkış biçimlerine göre farklı kategorilerde değerlendirilebilmektedir. Noktasal anomaliler (point anomalies), tek bir gözlemin veri kümesinin genel davranışından belirgin biçimde farklılaşması durumunu ifade etmektedir. Örneğin normal çalışma sırasında belirli bir aralıkta ölçülen makine sıcaklığının tek bir zaman noktasında olağan dışı yüksek bir değere ulaşması noktasal anomali olarak değerlendirilebilir. Bağlamsal anomaliler (contextual anomalies) ise bir gözlemin yalnızca belirli bir bağlam içerisinde anormal kabul edildiği durumları gösterir. Bir ortam sıcaklığının belirli bir mevsim veya zaman diliminde normal kabul edilirken farklı bir dönemde olağandışı olması buna örnek gösterilebilir. Topluluk anomalilerinde (collective anomalies) ise tek tek bakılınca normal görünebilecek bir gözlem grubunun birlikte bakıldığında anormal olmayan bir örüntü oluşturması söz konusudur. Bu ayrım özellikle zaman bağımlılığının bulunduğu IoT sensör verilerinde anomali tespit yönteminin belirlenmesi açısından önem taşır.

IoT ortamlarında anomali tespiti için istatistiksel yöntemlerin yanı sıra denetimli, denetimsiz ve yarı denetimli makine öğrenmesi yaklaşımlarından yararlanılır. Denetimli öğrenmede normal ve anormal durumları temsil eden etiketli örnekler kullanılarak SVM, Random Forest ve benzeri sınıflandırma modelleri geliştirilir. Ancak gerçek IoT ortamlarında meydana gelebilecek

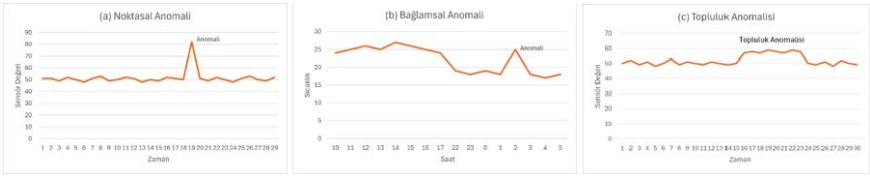
bütün anomali türlerinin önceden bilinmesi ve yeterli miktarda etiketli anomali verisinin elde edilmesi çoğu zaman olmamaktadır. Ayrıca normal çalışma verilerinin miktarı, anormal durumlara ilişkin verilerden önemli ölçüde fazla olabilmektedir. Bu durum anomali tespitinde sınıf dengesizliği problemini ortaya çıkarmakta ve yalnızca denetimli öğrenmeye dayanan yaklaşımların uygulanabilirliğini sınırlandırmaktadır. (Adhikari et al., 2024).

Etiketli anomali verisinin sınırlı olduğu durumlarda denetimsiz veya yarı denetimli yöntemler önemli bir alternatif oluşturmaktadır. Bu yaklaşımlarda sistemin normal davranışını temsil eden veri dağılımının veya örüntülerinin öğrenilmesi ve bu davranıştan belirgin biçimde uzaklaşan yeni gözlemlerin anomali olarak değerlendirilmesi gerçekleşebilen bir durumdur. Kümeleme tabanlı yöntemler, Isolation Forest ve One-Class SVM gibi geleneksel makine öğrenmesi yöntemlerinin yanı sıra autoencoder tabanlı derin öğrenme modelleri de bu amaçla kullanımı mevcuttur. Autoencoder yaklaşımında model normal veriyi yeniden oluşturmayı öğrenmekte; normal davranıştan farklılaşan gözlemlerde yeniden oluşturma hatasının yükselmesi anomali göstergesi olarak kullanılabilmektedir. Özellikle yüksek boyutlu ve çok değişkenli IoT sensör verilerinde derin öğrenme yöntemleri, karmaşık ilişkilerin doğrudan veriden öğrenilebilmesine olanak sağlar (Adhikari et al., 2024).

IoT tabanlı anomali tespitinde modelin doğruluk performansı tek başına yeterli değildir. Anomalinin ne kadar kısa sürede belirlenebildiği de önem taşır. Endüstriyel makinelerde meydana gelen arızalar, sağlık izleme sistemlerindeki kritik değişimler veya güvenlik uygulamalarındaki olağandışı durumlar mümkün olduğunca erken belirlenmesi gerekir. Bununla birlikte sürekli sensör akışlarının merkezi sisteme aktarılması iletişim yükünü ve gecikmeyi artırabileceğinden, anomali tespit modellerinin Uç Yapay Zekâ veya TinyML yaklaşımıyla veri kaynağına yakın noktalarda çalıştırılması önemli avantajlar sağlayabilir. TinyML tabanlı anomali tespit çalışmalarının sistematik incelemesi, mikrodenetleyici gibi kaynakları sınırlı platformlarda da çeşitli makine öğrenmesi ve derin öğrenme modellerinin bu amaçla uygulanabildiğini göstermektedir (Trilles et al., 2024).

IoT sistemlerinde anomali tespitinin önemli zorlukları vardır. Bunlardan birisi de normal davranışın zaman içerisinde değişebilmesi durumudur. Sensörlerin zamanla eskiyip yanlış veriler verme durumu, çevresel koşulların değişmesi, kullanıcı davranışlarındaki farklılaşmalar veya sistemin çalışma biçimindeki değişiklikler nedeniyle eğitim aşamasında öğrenilen veri

dağılımı zamanla gerçek sistem davranışını temsil etmez. Kavram kayması olarak adlandırılan bu durum, başlangıçta başarılı olan bir anomali tespit modelinin zaman içerisinde yanlış alarm üretmesine veya gerçek anomalileri belirleyememesine neden olabilir. Dolayısıyla gerçek IoT ortamlarında geliştirilen anomali tespit sistemlerinin yalnızca başlangıç performansına göre değil, değişen veri dağılımlarına uyum sağlayabilme ve düşük yanlış alarm oranını sürdürebilme yetenekleri açısından da değerlendirilmesi gerekmektedir. IoT sensör verilerinde karşılaşılabilecek noktasal, bağlamsal ve topluluk anomalilerinin kavramsal gösterimi Şekil 2'de verilmiştir.



Şekil 2. IoT sensör verilerinde temel anomali türlerinin kavramsal gösterimi.

Tahminleme

Tahminleme, IoT sistemlerinde geçmiş ve mevcut verilerden yararlanılarak gelecekte sensörlerden gelen verileri, sistem durumlarının veya belirli olayların gerçekleşme olasılığının tahminini sağlayan önemli veri analizi uygulamalarıdır. IoT cihazlarının sürekli ve zaman damgalı veri üretmesi, geçmiş gözlemler içerisindeki zamansal belirteçlerin belirlenerek geleceğe yönelik çıkarımlar yapılmasına olanak sağlar. Bu özellik sayesinde IoT sistemleri yalnızca mevcut fiziksel durumun izlenmesinde değil, gelecekte ortaya çıkabilecek değişimlerin önceden belirlenmesinde de kullanılabilmektedir. Enerji tüketiminin tahmin edilmesi, çevresel koşulların öngörülmesi, trafik yoğunluğunun kestirilmesi ve endüstriyel ekipmanların gelecekteki durumlarının belirlenmesi IoT tabanlı tahminleme uygulamalarına örnek olarak gösterilebilir (Mahdavinejad et al., 2018).

IoT tabanlı tahminleme problemleri birçok farklı çeşitte olsa da önemli bir bölümü zaman serisi analizi kapsamında değerlendirilir. Bir sensörden belirli zaman aralıklarıyla elde edilen $x_1, x_2, \dots, x_t$gözlemleri kullanılarak gelecekteki x_{t+h} değerinin tahmin edilmesi bu problemin temel yapısını oluşturmaktadır. Burada t mevcut zaman noktasını, h ise tahmin ufkunu ifade etmektedir. Tahmin yalnızca bir sonraki zaman adımı için gerçekleştirilebileceği gibi, gelecekteki birden fazla zaman noktasını

kapsayacak biçimde de yapılır. IoT uygulamasının niteliğine göre kısa dönemli tahminler gerçek zamanlı kontrol ve kaynak yönetimi açısından önemli olabilirken, buna ek olarak daha uzun dönemli tahminler bakım planlaması, enerji yönetimi ve kapasite planlaması gibi karar süreçlerinde kullanılır.

Tahminleme amacıyla doğrusal regresyon ve farklı geleneksel zaman serisi yöntemleri genel olarak yaygın kullanılır. Fakat bunların yanı sıra makine öğrenmesi ve derin öğrenme algoritmalarından da yararlanılır. Random Forest ve Support Vector Regression (SVR) gibi makine öğrenmesi yöntemleri geçmiş sensör gözlemlerinden çıkarılan ayraçlar kullanılarak gelecekteki değerlerin kestirilmesinde uygulanırken, LSTM ve GRU gibi tekrarlayan sinir ağı mimarileri verideki zamansal bağımlılıkların doğrudan modellenmesine olanak sağlar. Özellikle uzun süreli bağımlılıkların ve doğrusal olmayan ilişkilerin bulunduğu zaman serilerinde derin öğrenme yöntemleri önemli bir alternatif oluşturmaktadır. Son yıllarda dikkat mekanizmasına dayanan Transformer modelleri de uzun dönemli ilişkilerin modellenmesi ve çok değişkenli zaman serilerinin tahmin edilmesi amacıyla yaygın biçimde araştırılmaktadır (Wen et al., 2022).

IoT sistemlerinde tahminleme yalnızca gelecekteki sayısal sensör değerlerinin kestirilmesiyle sınırlı değildir. Endüstriyel IoT uygulamalarında sensörlerden elde edilen titreşim, sıcaklık, basınç, akım veya ses verilerinin zaman içerisindeki değişimi analiz edilerek ekipmanların gelecekteki çalışma durumu hakkında çıkarım yapılabilir. Bu yaklaşım, arızanın meydana gelmesinden sonra gerçekleştirilen bakım yerine arıza olasılığının veya ekipmanın kalan faydalı ömrünün önceden tahmin edilmesine dayanır. Bu tahminleme işlemine kestirimci bakım (predictive maintenance) da denilir. Böylece bakım faaliyetlerinin gerçek sistem durumuna göre planlanması, beklenmeyen duruşların azaltılması ve ekipmanların daha etkin kullanılması hedeflenmektedir. IoT, sensör teknolojileri ve yapay zekâ yöntemlerinin birlikte kullanılması kestirimci bakımın Endüstri 4.0 ortamlarındaki temel uygulama alanlarından biri olarak yaygınlaşmıştır (Zonta et al., 2020).

Tahmin modellerinin değerlendirilmesinde kullanılacak performans ölçütü problemin yapısına göre belirlenmektedir. Sayısal değerlerin tahmin edildiği regresyon problemlerinde Ortalama Mutlak Hata (Mean Absolute Error – MAE), Ortalama Kare Hata (Mean Squared Error – MSE) ve Kök Ortalama Kare Hata (Root Mean Squared Error – RMSE) gibi ölçütlerden

yararlanılır. IoT ortamlarında başarılı bir tahmin sisteminin yalnızca düşük tahmin hatasına sahip olması yeterli değildir. Gerçek zamanlı uygulamalarda modelin çıkarım süresi, yeni sensör verilerine uyum sağlayabilmesi, eksik veya gürültüsüz olmayan gözlemlere karşı dayanıklılığı ve hesaplama maliyeti de sistem performansını etkiler. Ayrıca sensör verilerinin zaman içerisinde değişen dağılımı, önceki bölümde ele alınan kavram kayması problemini tahminleme sistemleri açısından da önemli bir kriterdir.

IoT tabanlı tahminleme, geçmiş sensör verilerinin geleceğe yönelik karar süreçlerinde kullanılmasını sağlayarak sistemlerin tepkisel yapılardan daha önleyici yapılara dönüşmesine katkı sağlamaktadır. Kullanılacak modelin karmaşıklığı ile tahmin başarısı ve kaynak tüketimi arasında uygun bir denge kurulması önemli bir problemdir. Özellikle tahminlerin uç cihazlarda gerçekleştirilmesi gereken uygulamalarda model boyutu, bellek gereksinimi ve enerji tüketimi gibi faktörler daha belirleyici haldedir. Bu nedenle IoT tahminleme sistemlerinde veri özellikleri, tahmin ufku, uygulamanın gerçek zamanlılık gereksinimi ve çalıştırılacak donanım platformunun birlikte değerlendirilmesi gerekmektedir.

Sınıflandırma

Sınıflandırma, IoT sistemlerinden elde edilen sensör verilerinden önceden tanımlanmış sınıflardan birinin anlamını taşıyacak şekilde atanmasını amaçlayan temel yapay zekâ uygulamalarından biridir. Sensör verilerinin belirli fiziksel durumları, aktiviteleri veya sistem koşullarını temsil etmesi, sınıflandırma yöntemlerinin IoT uygulamalarında geniş bir kullanım alanına sahip olmasını sağlar. Bir makinenin çalışma durumunun normal veya arızalı olarak belirlenmesi, çevresel koşulların kategorilere ayrılması, ağ trafiğinin farklı davranış sınıflarına atanması ve giyilebilir sensörlerden insan aktivitelerinin tanınması bu problemlere örnek olarak gösterilebilir. IoT verilerinin sınıflandırılmasında SVM, k-NN, karar ağaçları, Random Forest ve yapay sinir ağları gibi farklı yöntemlerden yararlanılabilmektedir (Mahdavinejad et al., 2018).

Zaman serisi biçimindeki IoT verilerinde sürekli sensör akışının belirli zaman pencerelerine ayrılması ve her pencerenin ilgili fiziksel durumu temsil eden bir sınıf etiketiyle ilişkilendirilmesi yaygın bir yaklaşımdır. Örneğin insan aktivitesi tanıma (Human Activity Recognition – HAR) uygulamalarında ivmeölçer ve jiroskop verileri kullanılarak “yürüme”, “oturma”, “ayakta durma” veya “koşma” gibi aktiviteler sınıflandırılabilmektedir. Geleneksel makine öğrenmesi yöntemlerinde bu

zaman pencerelerinden zaman veya frekans alanında özellikler çıkarılırken, CNN ve LSTM gibi derin öğrenme modelleri sensör verilerinden ayırt edici temsilleri otomatik olarak öğrenebilmektedir (Wang et al., 2019).

IoT tabanlı sınıflandırma sistemlerinin değerlendirilmesinde yalnızca doğruluk (accuracy) değerinin kullanılması her zaman yeterli değildir. Özellikle sınıfların dengesiz olduğu veri kümelerinde kesinlik (precision), duyarlılık (recall), F1-skoru ve karışıklık matrisi gibi ölçütlerin birlikte değerlendirilmesi gerekmektedir. Ayrıca aynı fiziksel olayın farklı sensörler, cihaz yerleşimleri, kullanıcılar veya çevresel koşullar altında farklı veri dağılımları oluşturabilmesi, geliştirilen modellerin genelleştirilebilmesi önemlidir. Sınıflandırmanın uç cihaz üzerinde gerçekleştirildiği IoT uygulamalarında ise model başarısının yanı sıra sonuç çıkarabilme süresi, bellek gereksinimi ve enerji tüketimi de model seçiminde dikkate alınacak unsurlardır.

Görüntü, ses ve sensör verilerinin analizi

IoT sistemlerinde kullanılan veri kaynakları yalnızca sayısal sensör ölçümleriyle sınırlı değildir. Kameralar, mikrofonlar, ivmeölçerler, jiroskoplar ve çevresel sensörler aracılığıyla görüntü, ses ve çok değişkenli zaman serileri gibi farklı veri türleri üretilebilir. Bu verilerin yapısal özelliklerinin birbirinden farklı olması, uygulanacak yapay zekâ yöntemlerinin de veri türüne göre belirlenmesini şart koşar. Görüntü verilerinin analizinde nesne tespiti, görüntü sınıflandırma ve segmentasyon gibi bilgisayarlı görü yöntemleri kullanılırken; ses verileri konuşma, çevresel ses veya makine sesi gibi akustik ayraçların belirlenmesi amacıyla analiz edilir. Sensör verileri ise fiziksel sistemlerin durumunun, çevresel değişimlerin veya kullanıcı aktivitelerinin belirlenmesinde önemli bir veri kaynağı oluşturur.

Görüntü tabanlı IoT uygulamalarında CNN ve benzeri derin öğrenme mimarileri görüntülerden ayırt edici özelliklerin otomatik olarak öğrenilmesini sağlayarak akıllı gözetim, endüstriyel kalite kontrol ve çevresel izleme gibi alanlarda kullanılır. Buna paralel olarak mikrofonlardan elde edilen ses sinyalleri zaman veya frekans alanında işlenerek çevresel seslerin sınıflandırılması, makine durumlarının belirlenmesi ve olağandışı akustik olayların tespit edilmesi mümkün olmaktadır. İvmeölçer, jiroskop, sıcaklık ve basınç gibi sensörlerden elde edilen zaman serilerinde ise istatistiksel özellikler ve frekans bileşenleri kullanılabileceği gibi CNN,

LSTM ve benzeri derin öğrenme modelleriyle ham veya ön işlenmiş sinyaller doğrudan analiz edilebilir.

Bu veri türlerinin uç cihazlarda analiz edilmesi, IoT sistemlerinde gerçek zamanlı karar verme açısından önemli avantajlar gösterir. Özellikle kamera ve mikrofon gibi yüksek miktarda veri üreten kaynaklarda bütün ham verinin buluta aktarılması yerine gerekli çıkarımların Edge AI yaklaşımıyla cihaz üzerinde gerçekleştirilmesi iletişim yükünü ve gecikmeyi azaltabilmekte, aynı zamanda hassas verilerin cihaz dışına çıkarılmamasına katkı sağlar. Buna ek olarak görüntü ve ses modellerinin görece yüksek hesaplama gereksinimleri, kaynakları sınırlı IoT cihazlarında model sıkıştırma, nicemleme (quantization) ve uygun donanım seçimi gibi optimizasyonları gerekli hale getirmektedir. Bu nedenle IoT sistemlerinde veri analizi yaklaşımının belirlenmesinde yalnızca verinin türü değil, analiz işleminin gerçekleştirileceği donanım ve uygulamanın gerçek zamanlılık gereksinimleri de birlikte değerlendirmek daha doğru olur.

Çok modlu veri analizi ve sensör füzyonu

IoT sistemlerinde fiziksel bir olayın veya ortamın yalnızca tek bir sensör üzerinden değerlendirilmesi her zaman yeterli bilgi sağlamayabilir. Farklı sensörlerden elde edilen verilerin birlikte kullanılması, sistemin izlenen durum hakkında daha kapsamlı ve güvenilir bilgi üretmesine olanak sağlamaktadır. Sensör füzyonu (sensor fusion), birden fazla sensörden elde edilen bilgilerin birleştirilerek tek bir veri kaynağından elde edilebilecek bilgiden daha anlamlı bir temsil oluşturulmasını amaçlamaktadır. Örneğin bir giyilebilir IoT sisteminde ivmeölçer ve jiroskop verileri birlikte kullanılarak hareketlerin daha ayrıntılı biçimde belirlenmesi mümkünken, akıllı bir araçta kamera, radar ve konum sensörlerinden elde edilen bilgiler çevrenin daha güvenilir biçimde algılanmasına katkı sağlayabilmektedir. Sensör füzyonu veri, özellik veya karar düzeyinde gerçekleştirilebilmekte ve kullanılacak yaklaşım sensörlerin özelliklerine ve uygulamanın gereksinimlerine göre değişmektedir.

Çok modlu veri analizi ise farklı sensör ölçümlerinin ötesinde görüntü, ses, konum ve zaman serisi gibi farklı veri biçimlerinin ortak bir analiz sürecinde değerlendirilmesini kapsamaktadır. Örneğin akıllı sağlık uygulamalarında giyilebilir sensörlerden elde edilen fizyolojik ölçümler ile görüntü veya ses verileri; akıllı şehir uygulamalarında ise kamera görüntüleri, çevresel sensörler ve konum bilgileri birlikte değerlendirilebilir. Farklı veri kaynaklarının bir arada kullanılması tek bir veri türünde

bulunmayan tamamlayıcı bilgilerin modele aktarılmasını sağlayabilmektedir. Bununla birlikte farklı örnekleme hızları, zaman senkronizasyonu, eksik sensör verileri ve veri kaynaklarının farklı boyutlarda olması çok modlu IoT analizinin temel zorlukları arasındadır. Bu nedenle veri kaynaklarının uygun biçimde hizalanması ve birleştirilmesi, kullanılacak yapay zekâ modelinin başarısını doğrudan etkileyen önemli bir aşamadır.

IOT'DE FEDERE ÖĞRENME

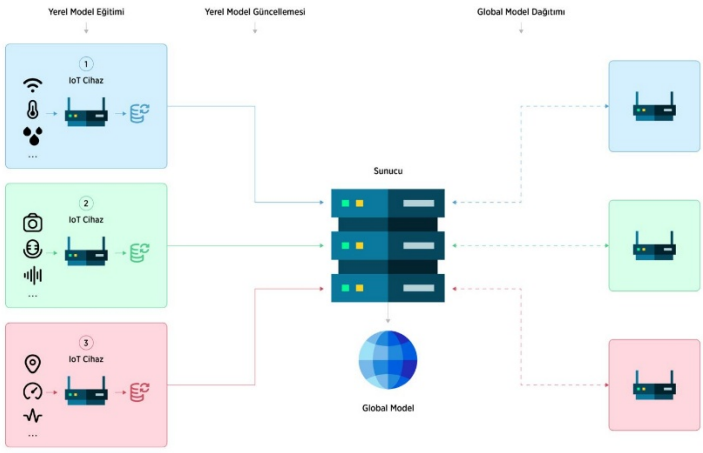
IoT sistemlerinde üretilen verilerin merkezi bir sunucuda toplanarak yapay zekâ modellerinin eğitilmesi, özellikle çok sayıda dağıtık cihazın bulunduğu ortamlarda iletişim maliyeti, veri gizliliği ve güvenlik açısından çeşitli sorunlar ortaya çıkarabilmektedir. Federe Öğrenme (FL), ham verilerin üretildiği cihazlardan merkezi bir veri havuzuna aktarılması yerine, model eğitiminin dağıtık biçimde gerçekleştirilmesini sağlayan bir makine öğrenmesi yaklaşımıdır. Bu yapıda cihazlar kendi yerel verileri üzerinde model eğitimi gerçekleştirirken, merkezi sunucuya ham veriler yerine model güncellemeleri gönderilmektedir. Böylece farklı cihazlarda bulunan veriler ortak bir modelin geliştirilmesinde kullanılabilmektedir (McMahan et al., 2017).

Federe Öğrenme yaklaşımı, verilerin farklı cihazlarda dağıtık biçimde bulunduğu IoT ortamları açısından özellikle önem taşımaktadır. uç cihazların hesaplama yeteneklerinin artması, model eğitim işlemlerinin bir bölümünün veri kaynağına yakın noktalarda gerçekleştirilmesini mümkün hale getirmektedir. Bununla birlikte FL, verilerin cihazda tutulması nedeniyle gizlilik açısından avantaj sağlasa da tek başına tam bir güvenlik veya gizlilik çözümü değildir; paylaşılan model parametreleri veya gradyanlar üzerinden bilgi sızıntısı oluşabilmesi, cihaz ve veri heterojenliği ile iletişim ve hesaplama maliyetleri IoT ortamlarında çözülmesi gereken önemli problemler arasında yer almaktadır (Li et al., 2024).

Federe öğrenme mimarisi

Federe Öğrenme'nin yaygın istemci-sunucu mimarisinde öğrenme süreci merkezi bir sunucu ile eğitime katılan istemci cihazlar arasında tekrarlanan iletişim turları şeklinde gerçekleştirilmektedir. İlk aşamada merkezi sunucu global modeli oluşturmakta ve seçilen istemcilere göndermektedir. Her istemci modeli kendi yerel veri kümesi üzerinde belirli sayıda adım veya epoch boyunca eğitmekte ve elde edilen model güncellemesini sunucuya iletmektedir. Sunucu, farklı istemcilerden gelen güncellemeleri birleştirerek

global modeli güncellemekte ve yeni model tekrar istemcilere dağıtılmaktadır. Bu işlem belirlenen durdurma kriterine ulaşılıncaya kadar tekrarlanmaktadır. Federe ortalama (FedAvg), istemciler tarafından gerçekleştirilen yerel eğitim sonucunda elde edilen model güncellemelerinin birleştirilmesine dayanan temel FL algoritmalarındandır (McMahan et al., 2017). Federe öğrenme kapsamında gerçekleştirilen yerel model eğitimi, model güncellemelerinin birleştirilmesi ve global model dağıtımı süreci Şekil 3'te gösterilmiştir.



Şekil 3. IoT ortamlarında federe öğrenme tabanlı yerel model eğitimi, model güncellemelerinin birleştirilmesi ve global model dağıtım süreci.

IoT sistemlerinde kullanımı ve avantajları

Federe Öğrenme, çok sayıda cihazın farklı konumlarda veri ürettiği IoT sistemlerinin dağıtık yapısıyla doğal olarak uygunluk verir. Geleneksel merkezi öğrenmede sensörlerden elde edilen verilerin eğitim amacıyla merkezi bir sunucuya aktarılması gerekirken, FL yaklaşımında veriler üretildikleri cihazlarda veya uç düğümlerde tutulur. Bu yetenek özellikle kişisel, endüstriyel veya çevresel nitelikte hassas verilerin işlendiği IoT uygulamalarında önemli bir avantaj sağlar. Cihazların yerel verileri üzerinde gerçekleştirdiği eğitim sonucunda elde edilen model güncellemelerinin birleştirilmesiyle, farklı kaynaklardan elde edilen bilgiler ham veriler paylaşılmadan ortak bir global modele aktarılır (Li et al., 2024).

FL'nin IoT sistemlerindeki bir diğer avantajı, büyük miktardaki ham verinin sürekli olarak merkezi sunucuya gönderilmesi gereksinimini azaltabilmesidir. Bu durum Uç bilgisayar ile birlikte kullanıldığında hesaplama işlemlerinin veri kaynağına yakın noktalarda gerçekleştirilmesi; merkezi bulut altyapısı üzerindeki yükün azaltılması ve dağıtık cihazların hesaplama kapasitelerinden yararlanılması açısından önem taşımaktadır. Bunun yanı sıra FL'nin sağladığı gizlilik avantajı, sistemin tamamen güvenli olduğu anlamına gelmemektedir. Model güncellemelerinden bilgi çıkarılması, cihazların farklı hesaplama kapasitelerine sahip olması ve cihazlarda üretilen verilerin dağılımlarının birbirinden farklı olması gibi problemler FL'nin gerçek IoT ortamlarında uygulanmasını zorlaştırır (Li et al., 2024).

Sınırlılıklar ve Zorluklar

Federe Öğrenmenin IoT sistemlerinde uygulanmasındaki temel zorluklardan biri veri ve cihaz heterojenliğidir. Farklı IoT cihazlarında üretilen verilerin miktarı, dağılımı ve niteliği birbirinden önemli ölçüde farklı olabilmektedir. Özellikle cihazlardaki verilerin bağımsız ve aynı dağılımdan gelmediği non-ID durumlar, yerel modeller arasında farklılaşmaya neden olarak global modelin yakınsamasını ve başarısını olumsuz etkiler. Bunun yanında IoT cihazlarının işlem gücü, bellek kapasitesi, enerji seviyesi ve ağ bağlantısı açısından farklı özelliklere sahip olması eğitim sürecine katılımı ve model güncellemelerinin zamanında iletilmesini zorlaştırmaktadır (Li et al., 2024).

Kısıtlardan diğeri de iletişim maliyeti ve güvenliktir. Federe Öğrenme ham verilerin merkezi sunucuya gönderilmesini önlese de yerel ve global model parametrelerinin çok sayıda eğitim turunda cihazlar ile sunucu arasında aktarılması gerekmektedir. Bu durum özellikle düşük bant genişliğine veya sınırlı enerji kapasitesine sahip IoT cihazlarında önemli bir maliyet oluşturabilmektedir. Ayrıca verinin cihazda kalması gizliliği artırmasına rağmen model güncellemelerinin paylaşılması risk oluşturur. Model veya gradyan güncellemelerinden yerel verilere ilişkin bilgilerin çıkarılması ve kötü niyetli istemcilerin model zehirleme gibi saldırılar gerçekleştirmesi mümkündür. Bu nedenle IoT ortamlarında Federe Öğrenmenin uygulanmasında model başarısının yanı sıra iletişim verimliliği, cihaz heterojenliği, enerji tüketimi, güvenlik ve gizlilik gereksinimlerinin birlikte değerlendirilmesi gerekmektedir (Li et al., 2024).

ZORLUKLAR VE GELECEK ARAŞTIRMA YÖNELİMLERİ

Yapay zekâ tabanlı IoT sistemlerinin yaygınlaşmasında karşılaşılan temel zorlukların başında sınırlı hesaplama kaynakları, enerji tüketimi ve gerçek zamanlı çalışma gereksinimleri gelmektedir. Özellikle sensör düğümleri ve gömülü IoT cihazları, bulut sistemleriyle karşılaştırıldığında daha düşük işlem gücü ve bellek kapasitesine sahiptir. Buna karşılık derin öğrenme modellerinin artan hesaplama gereksinimleri, bu modellerin doğrudan uç cihazlarda çalıştırılmasını zorlaştırabilmektedir. Gelecekte model sıkıştırma, nicemleme ve donanıma duyarlı model tasarımı gibi yöntemlerle daha küçük ve enerji verimli yapay zekâ modellerinin geliştirilmesi, Uç yapay Zekâ ve TinyML tabanlı IoT sistemleri açısından önemli bir araştırma alanı olmaya devam edecektir (Wang et al., 2025).

IoT ortamlarının dinamik yapısı, yapay zekâ modellerinin uzun süre aynı başarıyla çalışmasını da güçleştirmektedir. Sensörlerin yaşlanması, cihazların veya kullanıcı davranışlarının değişmesi ve çevresel koşullardaki farklılaşmalar zaman içerisinde verinin dağılımını değiştirme potansiyeline sahiptir. Kavram kayması olarak ifade edilen bu durum, eğitim sırasında öğrenilen veri dağılımı ile gerçek kullanım sırasında karşılaşılan dağılımın farklılaşmasına ve model performansının azalmasına neden olabilmektedir. Bu nedenle gelecekteki IoT sistemlerinin yalnızca çıkarım yapan statik modeller yerine, veri dağılımındaki değişimleri tespit edebilen ve gerektiğinde yeni koşullara uyum sağlayabilen yapılar olarak geliştirilmesi önem taşımaktadır (Hinder et al., 2024).

Bir diğer önemli araştırma alanı ise yapay zekâ modellerinin açıklanabilirliği, güvenilirliği ve dağıtık biçimde çalıştırılmasıdır. Özellikle sağlık, endüstriyel kontrol ve kritik altyapılar gibi alanlarda modelin yalnızca yüksek doğrulukla karar vermesi değil, bu kararın hangi verilere veya özelliklere dayanılarak üretildiğinin anlaşılabilmesi de önemlidir. Buna ek olarak SHAP ve LIME gibi açıklanabilir yapay zekâ yöntemlerinin ek hesaplama yükü oluşturması, kaynakları sınırlı IoT cihazlarında uygulanmalarını zorlaştırabilmektedir. Bu nedenle düşük kaynak tüketen açıklanabilir yapay zekâ yöntemlerinin Uç Yapay Zekâ, TinyML ve Federe öğrenme yaklaşımlarıyla bütünleştirilmesi; gelecekte daha güvenilir, gizliliği koruyan ve dağıtık IoT sistemlerinin geliştirilmesi açısından önemli bir araştırma yönü oluşturmaktadır (Moss et al., 2025).

SONUÇ

IoT sistemlerinde üretilen büyük miktardaki ve farklı yapıdaki verilerin anlamlı bilgiye dönüştürülmesi, yapay zekâ ve veri analitiği yöntemlerini akıllı IoT sistemlerinin temel bileşenlerinden biri haline getirmiştir. Sensörlerden, kameralardan ve mikrofonlardan elde edilen veriler; uygun ön işleme ve özellik çıkarımı aşamalarının ardından sınıflandırma, tahminleme ve anomali tespiti gibi yöntemlerle analiz edilebilmektedir. Makine öğrenmesi ve derin öğrenme yöntemleri bu analizlerin gerçekleştirilmesini sağlarken, Uç Yapay Zekâ ve TinyML yaklaşımları yapay zekâ modellerinin veri kaynağına daha yakın ve kaynakları sınırlı cihazlarda çalıştırılmasına olanak sağlamaktadır.

IoT sistemlerinin dağıtık yapısı, yapay zekâ modellerinin geliştirilmesi ve uygulanmasında yeni yaklaşımları da gerekli kılmaktadır. Federe Öğrenme, ham verilerin merkezi bir noktada toplanmasına gerek kalmadan farklı cihazların ortak bir modelin geliştirilmesine katkı sağlamasına imkân verirken; sensör füzyonu ve çok modlu veri analizi farklı veri kaynaklarından elde edilen tamamlayıcı bilgilerin birlikte değerlendirilmesini mümkün kılmaktadır. Bununla birlikte cihaz ve veri heterojenliği, enerji ve hesaplama kısıtları, iletişim maliyetleri, güvenlik ve gizlilik, kavram kayması ve açıklanabilirlik gibi konular güncel IoT sistemlerinin önemli araştırma problemleri olmaya devam etmektedir. Bu doğrultuda yapay zekâ, uç bilişim ve dağıtık öğrenme yaklaşımlarının birlikte ele alınması, daha uyarlanabilir, güvenilir ve verimli IoT sistemlerinin geliştirilmesi açısından önem taşımaktadır.

KAYNAKÇA

Adhikari, D., Jiang, W., Zhan, J., Rawat, D. B., & Bhattarai, A. (2024). Recent advances in anomaly detection in Internet of Things: Status, challenges, and perspectives. *Computer Science Review*, *54*, 100665. https://doi.org/10.1016/j.cosrev.2024.100665

Capogrosso, L., Cunico, F., Cheng, D. S., Fummi, F., & Cristani, M. (2024). A machine learning-oriented survey on tiny machine learning. *IEEE access*, *12*, 23406-23426. https://doi.org/10.1109/ACCESS.2024.3365349

Dutta, L., & Bharali, S. (2021). Tinyml meets iot: A comprehensive survey. *Internet of Things*, *16*, 100461. https://doi.org/10.1016/j.iot.2021.100461

Egwuche, O. S., Greeff, J., & Ezugwu, A. E. (2026). A comprehensive review of machine learning and deep learning applications for intelligent data processing in the Internet of Things and wireless sensor networks. *Discover Applied Sciences*, *8*(5), 515. https://doi.org/10.1007/s42452-026-08445-z

Gherbi, C., Senouci, O., Harbi, Y., Medani, K., & Aliouat, Z. (2023). A systematic literature review of machine learning applications in IoT. *International Journal of Communication Systems*, *36*(11), e5500. https://onlinelibrary.wiley.com/doi/10.1002/dac.5500

Hinder, F., Vaquet, V., & Hammer, B. (2024). One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A: detecting concept drift. *Frontiers in Artificial Intelligence*, *7*, 1330257. https://doi.org/10.3389/frai.2024.1330257

Krishnamurthi, R., Kumar, A., Gopinathan, D., Nayyar, A., & Qureshi, B. (2020). An overview of IoT sensor data processing, fusion, and analysis techniques. Sensors, 20(21), 6076. https://doi.org/10.3390/s20216076

Li, H., Ge, L. & Tian, L. Survey: federated learning data security and privacy-preserving in edge-Internet of Things. Artif Intell Rev 57, 130 (2024). https://doi.org/10.1007/s10462-024-10774-7

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., & Sheth, A. P. (2018). Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks*, *4*(3), 161-175. https://doi.org/10.1016/j.dcan.2017.10.002

Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, *20*(4), 2923-2960. https://doi.org/10.1109/COMST.2018.2844341

Moss, J., Gordon, J., Duclos, W., Liu, Y., Wang, Q., & Wang, J. (2025). Explainable AI in IoT: A Survey of Challenges, Advancements, and Pathways to Trustworthy Automation. *Electronics*, *14*(23), 4622. https://doi.org/10.3390/electronics14234622

Naghib, A., Jafari Navimipour, N., Hosseinzadeh, M., & Sharifi, A. (2023). A comprehensive and systematic literature review on the big data management techniques in the internet of things. Wireless Networks, 29(3), 1085-1144. https://doi.org/10.1007/s11276-022-03177-5

Ramadan, M. N., Ali, M. A., Khoo, S. Y., & Alkhedher, M. (2025). Federated learning and TinyML on IoT edge devices: Challenges, advances, and future directions. *ICT Express*. https://doi.org/10.1016/j.icte.2025.06.008

Siow, E., Tiropanis, T., & Hall, W. (2018). Analytics for the Internet of Things: A survey. *ACM Computing Surveys, 51*(4), Article 74, 1–36. DOI: 10.1145/3204947.

Tekin, N., Aris, A., Acar, A., Uluagac, S., & Gungor, V. C. (2024). A review of on-device machine learning for IoT: An energy perspective. *Ad Hoc Networks*, *153*, 103348. https://doi.org/10.1016/j.adhoc.2023.103348

Trilles, S., Hammad, S. S., & Iskandaryan, D. (2024). Anomaly detection based on artificial intelligence of things: A systematic literatüre mapping. Internet of Things, 25, 101063. https://doi.org/10.1016/j.iot.2024.101063

Wang, X., Tang, Z., Guo, J., Meng, T., Wang, C., Wang, T., & Jia, W. (2025). Empowering edge intelligence: A comprehensive survey on on-device AI models. *ACM Computing Surveys*, *57*(9), 1-39. https://doi.org/10.1145/3724420

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2022). Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*. https://doi.org/10.48550/arXiv.2202.07125

Zonta, T., Da Costa, C. A., da Rosa Righi, R., De Lima, M. J., Da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & industrial engineering*, *150*, 106889. https://doi.org/10.1016/j.cie.2020.106889

AKILLI IOT UYGULAMALARI VE GELECEK EĞİLİMLERİ

Muzaffer Tatlı*, İbrahim Arda Çankaya**

GİRİŞ

Nesnelerin İnterneti (Internet of Things–IoT), fiziksel çevrede bulunan nesnelerin algılama, haberleşme ve veri işleme yetenekleriyle sayısal sistemlere dâhil edilmesini sağlayarak birçok uygulama alanında önemli bir dönüşüm oluşturmuştur. Sensör teknolojilerindeki gelişmeler, düşük güçlü haberleşme altyapılarının yaygınlaşması, gömülü sistemlerin işlem kapasitesindeki artış ve bulut bilişim hizmetlerinin erişilebilir hâle gelmesi; çok sayıda fiziksel sistemin sürekli olarak gözlemlenmesini ve üretilen verilerin farklı bilgi işlem platformlarında değerlendirilmesini mümkün kılmıştır. Bunun sonucunda IoT, yalnızca tüketici elektroniğiyle sınırlı bir bağlantı teknolojisi olmaktan çıkarak şehir altyapıları, sağlık hizmetleri, tarımsal üretim (Aksoy vd., 2023), endüstriyel süreçler (Aksoy vd., 2022), enerji sistemleri, anında afet müdahale (Yücel vd., 2024) ve ulaşım gibi kritik alanların sayısal dönüşümünde kullanılan temel teknolojilerden biri hâline gelmiştir.

Bununla birlikte, bağlantılı cihaz sayısındaki artış IoT sistemlerinin ürettiği veri miktarını ve çeşitliliğini önemli ölçüde artırmıştır. Fiziksel ortamdan sürekli olarak elde edilen verilerin yalnızca toplanması ve depolanması, akıllı sistemlerden beklenen işlevlerin yerine getirilmesi açısından yeterli değildir. Güncel IoT uygulamalarında verinin gerçek veya gerçeğe yakın zamanda değerlendirilmesi, anlamlı örüntülerin belirlenmesi, olağan dışı durumların saptanması, gelecekteki sistem davranışının tahmin edilmesi ve gerektiğinde fiziksel ortama yönelik uygun kararların oluşturulması beklenmektedir. Bu gereksinimler, IoT altyapılarının veri aktarımına dayalı pasif sistemlerden analitik ve karar verme yeteneklerine sahip daha akıllı yapılara doğru gelişmesine neden olmuştur.

* Öğretim Görevlisi, Isparta Uygulamalı Bilimler Üniversitesi, Isparta Teknik Bilimler Meslek Yüksekokulu, Bilgisayar Teknolojileri, Isparta, muzaffertatli@isparta.edu.tr, ORCID: https://orcid.org/0000-0001-9075-3791

** Dr. Öğr. Üyesi, Süleyman Demirel Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, Isparta, ardacankaya@sdu.edu.tr, ORCID: https://orcid.org/0000-0001-5375-3573

Bu dönüşümün merkezinde yapay zekâ, makine öğrenmesi, edge ve bulut bilişim, gelişmiş haberleşme altyapıları ve dijital ikiz teknolojilerinin IoT ile bütünleşmesi bulunmaktadır. Özellikle yapay zekâ ile IoT arasındaki yakınsama, Artificial Intelligence of Things (AIoT) olarak tanımlanan yeni bir yaklaşımın gelişmesine yol açmıştır. AIoT sistemlerinde sensörlerden elde edilen veriler yalnızca kullanıcıya iletilmemekte; öğrenme algoritmaları aracılığıyla işlenerek tahmin, sınıflandırma, anomali tespiti, optimizasyon ve operasyonel karar verme süreçlerinde kullanılabilmektedir. ITU-T Y.4612 standardının AIoT için kavramsal bir çerçeve tanımlaması da bu yaklaşımın güncel IoT ekosistemi içerisindeki önemini göstermektedir (ITU-T, 2025).

IoT'nin akıllı sistemlerle bütünleşmesi farklı uygulama alanlarında farklı gereksinimler ortaya çıkarmaktadır. Akıllı şehirlerde ölçeklenebilirlik ve heterojen altyapıların birlikte çalışabilirliği ön plana çıkarken, sağlık uygulamalarında veri güvenilirliği ve mahremiyet daha kritik bir nitelik taşımaktadır. Tarımsal IoT sistemlerinde enerji tüketimi, kapsama alanı ve çevresel dayanıklılık belirleyici olurken, endüstriyel sistemlerde düşük gecikme, güvenilirlik ve kesintisiz çalışma gereksinimi ağırlık kazanmaktadır. Enerji ve ulaşım sistemlerinde ise gerçek zamanlı veri analizi ile karar mekanizmalarının fiziksel sistemlerle güvenli biçimde bütünleştirilmesi önem taşımaktadır. Dolayısıyla akıllı IoT uygulamaları ortak bir teknolojik altyapıyı paylaşmakla birlikte, her uygulama alanı kendine özgü mühendislik gereksinimleri ve kısıtları içermektedir.

Bu bölümde IoT'nin akıllı sistemlere doğru geçirdiği dönüşüm, öncelikle AIoT kavramı üzerinden ele alınacaktır. Ardından akıllı şehirler, sağlık, tarım, Endüstri 4.0, enerji ve ulaşım alanlarındaki IoT uygulamaları teknik açıdan incelenecektir. Fiziksel ve sayısal sistemler arasındaki ilişkinin daha ileri bir biçimini temsil eden dijital ikiz yaklaşımı ayrıca değerlendirilecek; akıllı IoT sistemlerinin ortaya çıkardığı etik, mahremiyet, veri yönetişimi ve sürdürülebilirlik sorunları tartışılacaktır. Son bölümde ise edge intelligence, federated learning, TinyML, yeni nesil haberleşme sistemleri ve otonom IoT gibi yaklaşımlar çerçevesinde gelecekteki teknolojik eğilimler değerlendirilecektir.

AKILLI IOT KAVRAMI VE IOT'DEN AIOT'YE DÖNÜŞÜM

Klasik IoT sistemlerinin temel işlevi fiziksel çevrenin algılanması, üretilen verinin sayısallaştırılması ve iletişim ağları üzerinden ilgili bilgi sistemlerine aktarılması üzerine kuruludur. Bu yapı, uzaktan izleme ve belirli

düzeyde kontrol yeteneği sağlamasına rağmen, sistemin ürettiği büyük miktardaki verinin anlamlandırılması için ek analitik mekanizmalara ihtiyaç duyulmaktadır. Özellikle sağlık, ulaşım, endüstri ve enerji gibi zaman açısından kritik uygulamalarda yalnızca mevcut durumun belirlenmesi yeterli değildir; sistemin gelecekteki davranışının tahmin edilmesi ve değişen koşullara uygun kararların oluşturulması gerekir. AIoT yaklaşımı, yapay zekâ yeteneklerini IoT sisteminin veri üretme, iletişim ve hesaplama süreçleriyle bütünleştirerek bu gereksinime yanıt vermektedir.

ITU-T Y.4612'ye göre AIoT, IoT operasyonlarının etkinliğinin artırılması, insan-makine etkileşiminin geliştirilmesi ve veri yönetimi ile analitik yeteneklerin güçlendirilmesi amacıyla yapay zekâ teknolojilerinin IoT altyapısıyla bütünleştirilmesini ifade etmektedir (ITU-T, 2025). Standardın AIoT'nin kavramları, karakteristikleri, teknik özellikleri, teknik yaklaşımları ve kavramsal modelini ayrı başlıklar altında ele alması, AIoT'nin yalnızca yapay zekâ algoritmalarının sensör verilerine uygulanması şeklinde değerlendirilmemesi gerektiğini göstermektedir.

AIoT'nin temel farklılığı, IoT tarafından üretilen verinin sistem içindeki rolünün değişmesidir. Geleneksel bir IoT uygulamasında sıcaklık, nem, basınç, konum, titreşim veya fizyolojik sinyaller gibi ölçümler uzaktaki bir sisteme aktarılıp kullanıcı tarafından izlenebilir. AIoT sisteminde ise aynı ölçümler bir makine öğrenmesi veya yapay zekâ modelinin girdisi olarak kullanılarak olay tanıma, sınıflandırma, tahmin, anomali tespiti veya karar üretme gibi daha ileri işlevler gerçekleştirilebilir. Khadam vd. (2025) tarafından 103 AIoT çalışması üzerinde gerçekleştirilen sistematik literatür incelemesi, tahmin, nesne ve olay tanıma ile operasyonel karar vermenin IoT sistemlerinde yapay zekânın en sık kullanıldığı görevler arasında olduğunu göstermektedir. Aynı incelemede sağlık, tarım ve ulaşımın öne çıkan uygulama alanları olduğu belirlenmiştir.

Siam vd. (2025), AIoT sistemlerinin teknolojik yapısını algılama (sensing), hesaplama (computing) ve ağ/iletişim (networking and communication) olmak üzere üç temel bileşen üzerinden açıklamaktadır. Algılama bileşeni fiziksel ortamdaki olayların ve değişkenlerin gözlemlenmesini, hesaplama bileşeni elde edilen veriden bilgi ve karar üretilmesini, ağ ve iletişim bileşeni ise verilerin cihazlar ve farklı hesaplama kaynakları arasında taşınmasını sağlamaktadır. Bu yaklaşımda yapay zekâ yalnızca merkezi veri analizi amacıyla kullanılan ayrı bir yazılım bileşeni değildir. Yapay zekâ yöntemlerinden algılama süreçlerinin iyileştirilmesi,

haberleşme kaynaklarının yönetilmesi ve hesaplama görevlerinin farklı düğümler arasında dağıtılması gibi sistem düzeyindeki işlevlerde de yararlanılabilmektedir (Siam vd., 2025).

AIoT sistemlerinin davranışı fiziksel ortamdan elde edilen verinin yapay zekâ tabanlı bir modele aktarılması üzerinden matematiksel olarak gösterilebilir. $t$anında $n$farklı sensörden elde edilen ölçümlerin oluşturduğu gözlem vektörü aşağıdaki şekilde tanımlanabilir:

$$x_t = [x_{1,t} \quad x_{2,t} \quad \cdots \quad x_{n,t}]^T \qquad (1)$$

Burada x_t, sistemin $t$anındaki gözlem durumunu; $x_{i,t}$ise $i.$ sensörden elde edilen ölçüm değerini göstermektedir. Ancak bir sensör değerinin tek başına kullanılması, özellikle bağlama duyarlı sistemlerde yeterli olmayabilir. Ölçümün gerçekleştirildiği konum, zaman, kullanıcı durumu, cihaz çalışma koşulları veya çevresel değişkenler gibi bağlamsal bilgilerin de modele dâhil edilmesi gerekebilir. Bu durumda yapay zekâ modelinin çıktısı genel olarak aşağıdaki biçimde ifade edilebilir:

$$\hat{y}_t = f_\theta(x_t, c_t) \qquad (2)$$

Burada f_θ, parametreleri θile tanımlanan öğrenilmiş yapay zekâ modelini; c_t, $t$anındaki bağlamsal bilgiyi; $\hat{y}_t$ise model tarafından üretilen tahmin veya çıkarımı göstermektedir. $\hat{y}_t$, uygulamanın niteliğine bağlı olarak bir arıza olasılığı, sağlık riski, trafik yoğunluğu, enerji talebi, bitki hastalığı sınıfı veya anomali skoru biçiminde olabilir.

Akıllı IoT sistemlerinin önemli özelliklerinden biri, model çıktısının doğrudan bir bilgi olarak kalmayıp operasyonel bir karara dönüştürülebilmesidir. Bunun için karar mekanizması aşağıdaki genel yapı ile ifade edilebilir:

$$a_t = \pi(\hat{y}_t, q_t) \qquad (3)$$

Burada a_t, sistem tarafından gerçekleştirilecek eylemi; π, karar politikasını; $q_t$ise güvenlik sınırları, kullanılabilir enerji, kullanıcı tercihleri ve işletim koşulları gibi karar sürecindeki kısıtları göstermektedir. Bu ayrım önemlidir. Yapay zekâ modelinin ürettiği tahmin ile fiziksel sisteme uygulanacak karar aynı kavram değildir. Örneğin bir model endüstriyel

ekipmanda yüksek arıza olasılığı belirleyebilir; buna karşılık sistemin vereceği karar makinenin hemen durdurulması, çalışma yükünün azaltılması veya bakım ekibine bildirim gönderilmesi biçiminde farklılaşabilir.

Eylemin fiziksel ortam üzerinde oluşturduğu değişiklik yeni sensör ölçümleriyle tekrar gözlemlendiğinde AIoT sistemi geri beslemeli bir yapıya dönüşmektedir. Bu süreç kavramsal olarak; algılama, veri, yapay zekâ tabanlı çıkarım, karar, eylem, fiziksel sistem, yeniden algılama

biçiminde ifade edilebilir. Böyle bir yapı, IoT'nin yalnızca fiziksel sistemi izleyen bir teknoloji olmaktan çıkarak fiziksel sistemin davranışını etkileyebilen siber-fiziksel bir karar mekanizmasının parçası hâline gelmesini sağlar.

AIoT sistemlerinin gelişiminde yapay zekâ işlemlerinin nerede gerçekleştirileceği de önemli bir mimari karar hâline gelmiştir. Yüksek işlem kapasitesi ve büyük ölçekli depolama olanağı nedeniyle bulut bilişim IoT verilerinin analizinde önemli avantajlar sunmaktadır. Bununla birlikte, sensörlerden elde edilen tüm verinin uzak veri merkezlerine gönderilmesi özellikle düşük gecikmenin gerekli olduğu uygulamalarda uygun olmayabilir. Sürekli veri aktarımı bant genişliği kullanımını ve haberleşmeye bağlı enerji tüketimini artırabileceği gibi, ağ bağlantısındaki kesintiler karar mekanizmasının çalışmasını da etkileyebilir. Sağlık veya kullanıcı davranışı gibi hassas verilerin uzak sistemlere aktarılması ise ilave güvenlik ve mahremiyet gereksinimleri doğurmaktadır.

Bu sınırlamalar, yapay zekâ işlemlerinin veri kaynağına yaklaştırıldığı Edge Intelligence ve Edge AI yaklaşımlarının gelişimini hızlandırmıştır. Edge Intelligence, edge computing ile yapay zekâ yeteneklerini birleştirerek analitik ve karar işlemlerinin cihazlara yakın hesaplama kaynaklarında gerçekleştirilmesini amaçlamaktadır. Wang vd. (2025), edge intelligence'ın gerçek zamanlı ve verimli akıllı karar süreçlerini desteklediğini; buna karşılık kaynak kısıtları, heterojen ağ ortamları, güvenlik, güvenilirlik, şeffaflık ve sürdürülebilirlik gibi sorunların bu sistemlerin güvenilirliği açısından birlikte değerlendirilmesi gerektiğini belirtmektedir.

Modern AIoT mimarilerinde bu nedenle cihaz–edge–bulut (device–edge–cloud) yapısının birlikte kullanılması giderek önem kazanmaktadır. Sensör veya gömülü cihaz üzerinde temel veri temizleme ve özellik çıkarımı gerçekleştirilebilirken düşük gecikme gerektiren çıkarımlar edge düğümünde

yapılabilir. Yüksek hesaplama maliyeti taşıyan model eğitimi, geçmiş verinin uzun süreli analizi veya büyük veri kümelerinin işlenmesi ise bulut altyapısına bırakılabilir. Khadam vd. (2025), mevcut AIoT çalışmalarında hem edge hem de bulut tabanlı yapay zekâ konuşlandırmalarının yaygın olduğunu, buna karşılık hibrit konuşlandırmaların görece daha az incelendiğini ortaya koymaktadır.

Bu yapı AIoT tasarımının yalnızca en yüksek doğruluğu sağlayan yapay zekâ modelini seçmekten ibaret olmadığını göstermektedir. Gerçek bir IoT sisteminde tahmin başarımı; gecikme, enerji tüketimi, işlem yükü, haberleşme maliyeti, veri kalitesi, güvenilirlik ve güvenlik gibi sistem düzeyindeki ölçütlerle birlikte değerlendirilmelidir. Çok yüksek sınıflandırma doğruluğu sağlayan ancak kararını kabul edilemeyecek sürede üreten bir model, gerçek zamanlı bir endüstriyel kontrol sisteminde işlevsel olmayabilir. Benzer şekilde sürekli bulut bağlantısına bağımlı olan bir yaklaşım, düşük güçlü ve sınırlı haberleşme altyapısına sahip tarımsal sensör düğümlerinde sürdürülebilir bir çözüm oluşturmayabilir.

AIoT'deki “akıllılık” kavramının bu nedenle yalnızca sisteme bir makine öğrenmesi algoritmasının eklenmesiyle açıklanması doğru değildir. Akıllı davranış; fiziksel ortamın güvenilir biçimde algılanması, elde edilen verinin doğru bağlam içerisinde yorumlanması, uygun çıkarımın oluşturulması, kararın operasyonel kısıtlar dikkate alınarak verilmesi ve bu kararın gerektiğinde fiziksel sisteme güvenli şekilde uygulanması sonucunda ortaya çıkmaktadır. Bu bakımdan AIoT, algoritma odaklı bir yapıdan ziyade algılama, haberleşme, hesaplama ve karar mekanizmalarının birlikte ele alındığı uçtan uca bir sistem mühendisliği yaklaşımı olarak değerlendirilmelidir.

Sonuç olarak IoT'den AIoT'ye geçiş, yalnızca bağlantılı cihaz sayısındaki artışı değil, fiziksel ve sayısal sistemler arasındaki ilişkinin niteliğinde meydana gelen değişimi ifade etmektedir. IoT fiziksel çevrenin algılanabilir ve bağlantılı hâle gelmesini sağlarken AIoT bu altyapıya veriyi yorumlama, örüntü öğrenme, gelecekteki durumları tahmin etme ve belirli koşullar altında karar üretme yeteneği kazandırmaktadır. Bu dönüşüm, sonraki bölümlerde incelenecek akıllı şehir, sağlık, tarım, Endüstri 4.0, enerji ve ulaşım sistemlerinin ortak teknolojik temelini oluşturmaktadır.

AKILLI ŞEHIRLERDE IOT UYGULAMALARI

Kentleşmenin hızlanması; ulaşım, enerji, su, atık yönetimi, çevresel kalite ve kamu hizmetleri üzerinde giderek artan bir baskı oluşturmaktadır. Bu koşullar altında akıllı şehir yaklaşımı, kent kaynaklarının yalnızca fiziksel altyapı yatırımlarıyla değil, veri temelli ve bütünleşik yönetim mekanizmalarıyla daha etkin kullanılmasını amaçlamaktadır. International Telecommunication Union (ITU), akıllı sürdürülebilir şehri; bilgi ve iletişim teknolojilerinden yararlanarak yaşam kalitesini, kentsel operasyonların ve hizmetlerin verimliliğini ve rekabet gücünü geliştiren, bunu gerçekleştirirken mevcut ve gelecek nesillerin ekonomik, sosyal, çevresel ve kültürel gereksinimlerini gözeten yenilikçi bir şehir olarak tanımlamaktadır (ITU-T, 2022). Bu tanım, akıllı şehir kavramının yalnızca kentsel altyapının sayısallaştırılmasıyla sınırlandırılamayacağını; teknoloji, hizmet kalitesi, sürdürülebilirlik ve kent yaşamının birlikte değerlendirilmesi gerektiğini göstermektedir.

IoT, bu yaklaşımın fiziksel altyapı ile sayısal hizmetler arasındaki bağlantısını sağlayan temel teknolojilerden biridir. Kent içerisinde dağıtılan sensörler, kameralar, akıllı sayaçlar, çevresel izleme istasyonları, bağlı ulaşım sistemleri ve diğer algılama birimleri fiziksel çevrenin sürekli gözlemlenmesine olanak tanımaktadır. Elde edilen veriler iletişim ağları üzerinden kentsel veri platformlarına aktarılmakta; burada depolama, akış işleme, analitik ve görselleştirme işlemlerinden geçirildikten sonra karar destek veya otomatik kontrol sistemlerinin girdisi hâline gelmektedir. Bellini vd. (2022), IoT destekli akıllı şehir mimarilerinin algılama ve aktüasyondan veri iletimine, depolamadan edge, fog ve cloud tabanlı işlemeye kadar uzanan bütünleşik bir yapı sergilediğini belirtmektedir. Bu nedenle modern akıllı şehir, birbirinden bağımsız "akıllı" uygulamalardan çok, çok sayıda veri kaynağının ve hizmetin ortak bir kentsel bilgi ekosistemi içerisinde çalıştığı bir sistemler sistemi (system of systems) olarak değerlendirilmelidir.

Akıllı şehirlerde IoT'nin önemli kullanım alanlarından biri çevresel izleme sistemleridir. Hava sıcaklığı, bağıl nem, partikül madde, karbon monoksit, azot oksitler, gürültü düzeyi, su kalitesi ve meteorolojik değişkenler gibi göstergeler dağıtık sensör ağları aracılığıyla sürekli izlenebilir. Geleneksel çevresel gözlem altyapıları sınırlı sayıdaki sabit ölçüm istasyonlarına dayanırken, düşük maliyetli IoT sensörlerinin yaygınlaştırılması kentsel çevrenin daha yüksek zamansal ve mekânsal çözünürlükte gözlemlenmesine imkân sağlamaktadır. Bununla birlikte sensör

yoğunluğunun artırılması tek başına daha güvenilir çevresel bilgi anlamına gelmez. Sensör kalibrasyonu, ölçüm belirsizliği, cihazlar arası farklılıklar, iletişim kayıpları ve eksik gözlemler, elde edilen verinin güvenilirliğini doğrudan etkileyebilir. Bu nedenle çevresel IoT sistemlerinde veri kalitesi kontrolünün algılama sürecinin ayrılmaz bir parçası olarak ele alınması gerekir.

Çevresel ölçümlerin gerçek değeri, farklı veri kaynaklarıyla birleştirildiğinde daha belirgin hâle gelmektedir. Örneğin hava kalitesi sensörlerinden elde edilen ölçümlerin trafik yoğunluğu, meteorolojik koşullar, yol ağı ve arazi kullanım verileriyle bütünleştirilmesi, belirli bölgelerde ortaya çıkan kirleticilerin yalnızca gözlemlenmesini değil, olası nedenlerinin değerlendirilmesini de sağlayabilir. AIoT tabanlı sistemlerde geçmiş ve gerçek zamanlı verilerin birlikte işlenmesiyle kısa dönemli hava kalitesi tahminleri üretilebilir, yüksek risk taşıyan bölgeler belirlenebilir ve belediye yönetimleri için erken uyarı mekanizmaları oluşturulabilir. Bu yapı, akıllı şehirlerde IoT'nin basit ölçüm teknolojisinden karar destek altyapısına dönüşmesinin tipik bir örneğidir.

Akıllı atık yönetimi de IoT'nin kent hizmetlerinde somut fayda sağlayabildiği uygulama alanlarından biridir. Geleneksel sistemlerde atık toplama araçlarının sabit güzergâh ve zaman çizelgelerine göre çalışması, bazı konteynerlerin henüz dolmadan boşaltılmasına karşılık diğerlerinin toplama zamanından önce kapasitesini aşmasına neden olabilir. IoT tabanlı sistemlerde konteynerlere yerleştirilen doluluk, ağırlık veya çevresel durum sensörleri atık birikimini gerçek zamanlı veya periyodik olarak izleyebilir. Bu bilgiler coğrafi konum ve trafik verileriyle birleştirildiğinde toplama güzergâhlarının mevcut duruma göre yeniden planlanması mümkün hâle gelir. Güncel akıllı atık yönetimi çalışmaları, IoT algılama sistemlerinin AI tabanlı tahmin, rota optimizasyonu ve otomatik sınıflandırma teknolojileriyle giderek daha fazla bütünleştiğini göstermektedir.

Bu sistemlerde karar yalnızca “konteyner dolu veya boş” biçiminde ikili bir duruma indirgenmemelidir. Toplama önceliğinin belirlenmesinde doluluk oranı, atık türü, çevresel sıcaklık, bekleme süresi, aracın mevcut konumu, trafik durumu ve toplama kapasitesi gibi çok sayıda değişken birlikte kullanılabilir. Böylece IoT tabanlı algılama, optimizasyon algoritmalarıyla bütünleştirilerek araçların toplam yol mesafesinin, yakıt tüketiminin ve gereksiz toplama işlemlerinin azaltılmasına katkı sağlayabilir. Akıllı atık

yönetimi bu yönüyle IoT'nin çevresel sürdürülebilirlik ile operasyonel verimliliği aynı uygulamada bir araya getirebildiği alanlardan biridir.

Akıllı şehirlerin diğer önemli bileşenini akıllı binalar ve kentsel altyapılar oluşturmaktadır. Binalarda sıcaklık, nem, iç ortam hava kalitesi, aydınlatma, doluluk, enerji tüketimi ve ekipman durumu gibi değişkenlerin izlenmesi, bina yönetim sistemlerinin çevresel ve kullanım koşullarına daha duyarlı hâle gelmesini sağlamaktadır. Örneğin bir ofis binasında yalnızca sabit zaman çizelgesine göre çalışan iklimlendirme sistemi yerine, doluluk sensörleri ve iç ortam ölçümlerine göre hava akışını ve sıcaklığı dinamik biçimde düzenleyen bir kontrol yaklaşımı uygulanabilir. Böyle bir yapı hem kullanıcı konforunun korunmasına hem de gereksiz enerji tüketiminin azaltılmasına katkı sağlayabilir.

Benzer yaklaşım köprüler, yollar, tüneller, su şebekeleri ve diğer fiziksel altyapılara da uygulanabilir. Titreşim, deformasyon, basınç, debi veya sızıntı gibi değişkenlerin sürekli izlenmesi; altyapı bileşenlerindeki bozulmaların belirli aralıklarla gerçekleştirilen manuel denetimler arasında da takip edilebilmesine imkân verir. Özellikle kritik altyapılarda IoT verisinin geçmiş bakım kayıtları ve yapısal modellerle birleştirilmesi, bakım faaliyetlerinin yalnızca sabit takvimlere göre değil, gerçek sistem durumuna göre planlanmasına yardımcı olabilir. Böylece arıza sonrasında gerçekleştirilen düzeltici bakım yaklaşımından durum temelli veya kestirimci bakım stratejilerine geçiş mümkün hâle gelmektedir.

IoT altyapıları kamu hizmetleri ve kentsel operasyonların izlenmesinde de kullanılmaktadır. Su dağıtım sistemlerinde akış ve basınç ölçümleri, şebekedeki olağan dışı durumların veya kaçak olasılıklarının belirlenmesini kolaylaştırabilir. Kentsel aydınlatma altyapılarında ortam ışığı, zaman, yaya veya araç hareketliliği gibi değişkenlere bağlı dinamik kontrol uygulanabilir. Kamuya ait fiziksel varlıkların konumu ve çalışma durumu izlenerek bakım, envanter ve kaynak tahsis süreçleri veri temelli hâle getirilebilir. Bu tür uygulamalarda IoT'nin sağladığı temel değer, fiziksel altyapının durumunun belediye bilgi sistemlerine sürekli ve otomatik olarak aktarılmasıdır.

Akıllı şehir uygulamalarının ölçeği büyüdükçe temel problem, bağımsız IoT uygulamalarının geliştirilmesinden çok heterojen sistemlerin bütünleştirilmesine dönüşmektedir. Bir şehirde çevresel izleme, atık yönetimi, aydınlatma, su, ulaşım, enerji ve kamu güvenliği sistemleri farklı üreticilerin donanımları, haberleşme protokolleri ve veri modelleriyle

çalışabilir. Bu sistemlerin tamamen bağımsız olarak işletilmesi, verinin dikey bilgi silolarında kalmasına neden olur. Bellini vd. (2022), akıllı şehirlerde farklı veri sağlayıcılarının, protokollerin ve veri biçimlerinin desteklenmesinin yanı sıra birlikte çalışabilirlik ve ölçeklenebilirliğin temel teknik gereksinimler olduğunu vurgulamaktadır.

Bu bağlamda akıllı şehir platformları, farklı IoT sistemleri arasında ortak veri ve hizmet altyapısı oluşturmayı amaçlamaktadır. ITU-T Y.4200, akıllı şehir platformları arasındaki birlikte çalışabilirliğin kent hizmetlerinin sayısını ve kalitesini artırabileceğini, aynı zamanda ölçeklenebilirlik ve entegrasyonu destekleyebileceğini belirtmektedir (ITU-T, 2018). ITU-T Y.4223 ise akıllı şehir ve topluluklar açısından IoT ve ICT tabanlı ortak gereksinimleri ve yetenekleri tanımlamakta; akıllı şehir platformunu farklı kent sistemlerini ve üçüncü taraf bileşenlerini açık arayüzler üzerinden bütünleştirebilen bir altyapı olarak ele almaktadır (ITU-T, 2023).

Birlikte çalışabilirlik yalnızca cihazların aynı ağa bağlanabilmesi anlamına gelmemektedir. Teknik birlikte çalışabilirlik, sistemlerin ortak arayüz ve iletişim mekanizmaları üzerinden veri alışverişi gerçekleştirebilmesini; semantik birlikte çalışabilirlik ise aktarılan verinin farklı sistemler tarafından aynı anlamla yorumlanabilmesini gerektirir. Örneğin iki ayrı sistem sıcaklık verisi üretiyor olsa bile kullanılan birim, zaman damgası, sensör konumu veya ölçüm bağlamı farklı biçimlerde temsil edildiğinde bu veri kümelerinin doğrudan bütünleştirilmesi mümkün olmayabilir. OECD'nin akıllı şehir veri yönetişimi incelemesi de birlikte çalışabilirliği farklı sektörlerden ve kurumlardan gelen verilerin anlamını kaybetmeden birleştirilebilmesi ve hizmetler arasında kullanılabilmesi açısından temel bir unsur olarak değerlendirmektedir.

Güncel akıllı şehir platformlarının karmaşıklığı bu gereksinimi daha da belirgin hâle getirmektedir. Bellini vd. (2024), çok kiracılı (multi-tenant) büyük ölçekli akıllı şehir altyapılarında gerçek zamanlı veri akışlarının, olay tabanlı işleme süreçlerinin, analitik bileşenlerin, API'lerin ve farklı uygulamaların aynı veri havuzunu paylaşmasının yönetim açısından önemli bir karmaşıklık oluşturduğunu göstermektedir. Çalışmada geliştirilen birleşik bilgi modeli, farklı veri, süreç ve hizmetler arasındaki ilişkilerin semantik olarak temsil edilmesinin ölçeklenebilirlik, bakım yapılabilirlik ve hata teşhisi bakımından önemini ortaya koymaktadır.

Bu durum akıllı şehir mimarisinde verinin artık yalnızca sensörlerden elde edilen bir çıktı olmadığını, farklı kentsel hizmetlerin birbirine bağlanmasını sağlayan ortak bir altyapı kaynağı hâline geldiğini göstermektedir. Trafik sensörlerinden elde edilen bir veri yalnızca trafik yönetim sisteminde değil, çevresel emisyonların değerlendirilmesinde veya acil durum güzergâhlarının belirlenmesinde de kullanılabilir. Benzer biçimde hava durumu, etkinlik takvimi veya nüfus hareketliliği gibi veriler birden fazla belediye hizmetinin planlanmasında kullanılabilir. Dikey uygulama silolarından ortak kentsel veri platformlarına geçiş bu nedenle modern akıllı şehirlerin temel mimari eğilimlerinden biridir.

Bununla birlikte, kent ölçeğinde gerçekleştirilen sürekli algılama ve veri birleştirme süreçleri önemli güvenlik, mahremiyet ve yönetişim sorunları doğurmaktadır. Kamera, konum, ulaşım kartı, mobil cihaz veya bina kullanım verilerinin farklı kaynaklardan birleştirilmesi, tek başına hassas görünmeyen verilerden bireylerin davranışları hakkında ayrıntılı sonuçların çıkarılmasına yol açabilir. Akıllı şehir sistemlerinin teknik olarak gerçekleştirebildiği veri toplama ile hukuken ve etik olarak gerçekleştirilmesi uygun olan veri işleme faaliyetleri aynı değildir. Bu nedenle veri minimizasyonu, erişim kontrolü, amaç sınırlaması, kimliksizleştirme ve hesap verebilirlik gibi ilkelerin sistem tasarımının başlangıç aşamasında dikkate alınması gerekir. OECD de başarılı akıllı şehir veri yönetiminde mahremiyet, kişisel verilerin korunması ve birlikte çalışabilirliğin ortak bir yönetişim yapısı içerisinde değerlendirilmesinin gerekliliğine dikkat çekmektedir.

Sonuç olarak akıllı şehirlerde IoT'nin temel işlevi kente daha fazla sensör yerleştirmek değil, fiziksel kentsel süreçlerin ölçülebilir hâle getirilmesini ve bu ölçümlerin güvenilir karar süreçlerine dönüştürülmesini sağlamaktır. Çevresel izleme, atık yönetimi, akıllı binalar, altyapı izleme ve kamu hizmetleri bunun farklı uygulama örneklerini oluşturmaktadır. Uygulamaların gerçek anlamda akıllı şehir ekosistemine dönüşebilmesi ise cihaz bağlantısının ötesinde veri kalitesi, ortak veri modelleri, birlikte çalışabilirlik, ölçeklenebilirlik, güvenlik ve yönetişim gereksinimlerinin birlikte karşılanmasına bağlıdır. Ulaşım, enerji ve sağlık gibi kent ekosisteminin diğer önemli bileşenleri bu bölümün ilerleyen başlıklarında kendi operasyonel ve teknik gereksinimleri çerçevesinde ayrıca ele alınacaktır.

Tablo 1. Akıllı şehirlerde başlıca IoT uygulamaları ve teknik işlevleri

Uygulama alanı	Başlıca veri kaynakları	IoT'nin temel işlevi	Akıllı sistem çıktısı	Kritik gereksinim
Çevresel izleme	Hava kalitesi, sıcaklık, nem, gürültü, su sensörleri	Sürekli çevresel gözlem	Risk haritalama, tahmin, erken uyarı	Veri kalitesi ve kalibrasyon
Atık yönetimi	Doluluk, ağırlık, konum sensörleri	Atık oluşumunun izlenmesi	Dinamik toplama ve rota optimizasyonu	Enerji verimliliği ve bağlantı
Akıllı binalar	Doluluk, sıcaklık, hava kalitesi, enerji sensörleri	Bina durumunun izlenmesi	HVAC ve aydınlatma optimizasyonu	Gerçek zamanlı kontrol
Kentsel altyapı	Titreşim, deformasyon, basınç, debi	Yapısal ve operasyonel durum izleme	Bakım ve arıza uyarıları	Güvenilirlik
Su sistemleri	Akış, basınç, kalite sensörleri	Şebekenin sürekli izlenmesi	Kaçak ve anomali belirleme	Sensör doğruluğu
Akıllı aydınlatma	Işık, hareket, trafik sensörleri	Kullanım durumunun belirlenmesi	Dinamik aydınlatma kontrolü	Enerji ve gecikme
Kentsel veri platformları	Çoklu IoT ve kurumsal veri kaynakları	Veri bütünleştirme	Çok alanlı karar desteği	Birlikte çalışabilirlik ve güvenlik

SAĞLIKTA IOT VE INTERNET OF MEDİCAL THİNGS (IOMT)

Sağlık hizmetlerinin sayısallaşması, hasta verilerinin yalnızca hastane ortamında ve belirli muayene zamanlarında elde edildiği geleneksel yaklaşımdan sürekli ve uzaktan izlemeye dayalı sistemlere geçişi hızlandırmıştır. Bu dönüşümün önemli bileşenlerinden biri Internet of Medical Things (IoMT) olarak adlandırılan tıbbi nesnelerin internetidir. IoMT; sensörler, giyilebilir cihazlar, tıbbi ölçüm sistemleri, mobil sağlık uygulamaları ve hastane ekipmanları gibi bileşenlerin iletişim ağları üzerinden sağlık bilgi sistemleriyle bütünleştirilmesini ifade etmektedir. Bu yapı sayesinde fizyolojik ve operasyonel veriler sürekli veya belirli aralıklarla toplanabilmekte, işlenebilmekte ve sağlık profesyonellerinin değerlendirmesine sunulabilmektedir. Güncel çalışmalar IoMT'nin özellikle

uzaktan hasta izleme, kişiselleştirilmiş sağlık hizmetleri, akıllı hastaneler ve klinik karar desteği açısından önemli bir altyapı oluşturduğunu göstermektedir (Sharma vd., 2024; Sun vd., 2026).

IoMT sistemlerinin temel çalışma yapısı algılama, iletişim, veri işleme ve sağlık hizmeti arasındaki bütünleşmeye dayanmaktadır. Hastadan veya çevreden elde edilen kalp hızı, elektrokardiyogram (ECG), kan oksijen satürasyonu (SpO_2), vücut sıcaklığı, kan basıncı, glikoz düzeyi veya fiziksel aktivite gibi veriler sensörler aracılığıyla sayısallaştırılır. Toplanan veriler Bluetooth, Wi-Fi, hücresel ağlar veya diğer kablosuz haberleşme teknolojileri üzerinden mobil cihazlara, edge düğümlerine veya bulut tabanlı sağlık platformlarına aktarılabilir. Veri işleme aşamasında filtreleme, özellik çıkarımı, anomali tespiti veya yapay zekâ tabanlı tahmin gerçekleştirildikten sonra elde edilen sonuçlar sağlık profesyonellerine, hastalara veya ilgili klinik bilgi sistemlerine iletilir. Bu nedenle IoMT, yalnızca "internete bağlı tıbbi cihaz" kavramından daha geniş bir uçtan uca sağlık bilgi sistemi olarak değerlendirilmelidir (Sun vd., 2023).

IoMT'nin en belirgin uygulamalarından biri uzaktan hasta izleme (Remote Patient Monitoring–RPM) sistemleridir. Özellikle kronik hastalıkların izlenmesi, yaşlı bireylerin ev ortamında takip edilmesi ve hastane sonrası bakım süreçlerinde hastanın fizyolojik durumunun sürekli gözlemlenmesi mümkündür. Giyilebilir cihazlardan alınan verilerin sağlık platformlarına aktarılması, hastanın yalnızca sağlık kuruluşuna başvurduğu anda değil günlük yaşamı sırasında da değerlendirilmesine olanak sağlar. Böylece zamana bağlı değişimler veya olağan dışı fizyolojik durumlar daha erken fark edilebilir. Giyilebilir IoT sistemlerinin kalp hızı, kan basıncı, vücut sıcaklığı ve oksijen düzeyi gibi yaşamsal göstergelerin sürekli izlenmesinde giderek daha fazla kullanılması, sağlık hizmetlerinin episodik ölçüm modelinden sürekli izlemeye doğru geliştiğini göstermektedir.

IoMT yalnızca bireysel hasta izlemede değil, akıllı hastane uygulamalarında da önemli işlevler üstlenmektedir. Hastane içerisindeki tıbbi cihazların ve ekipmanların konumlarının takip edilmesi, yatak doluluk durumunun izlenmesi, ilaç ve sarf malzemelerinin yönetilmesi ve kritik ekipmanların çalışma durumunun gözlemlenmesi IoMT tabanlı çözümlerle gerçekleştirilebilir. Böylece IoMT verileri klinik verilerin yanı sıra hastane operasyonlarının yönetiminde de kullanılabilir. Bununla birlikte sağlık alanındaki temel amaç otomasyonun kendisi değil, güvenilir ve zamanında elde edilen bilgilerin klinik iş akışını desteklemesidir.

Yapay zekânın IoMT altyapısına dâhil edilmesi, sağlık verisinin kullanım biçimini daha ileri bir düzeye taşımaktadır. Sensörlerden sürekli elde edilen zaman serileri üzerinde çalışan makine öğrenmesi ve derin öğrenme modelleri; anomali tespiti, hastalık riski tahmini, hasta durumunun sınıflandırılması ve erken uyarı oluşturulması gibi görevlerde kullanılabilir. Özellikle gerçek zamanlı karar gerektiren uygulamalarda verilerin tamamının uzak bulut sistemlerine aktarılması yerine bazı analizlerin edge üzerinde gerçekleştirilmesi gecikmenin azaltılması açısından önemlidir. Edge AI tabanlı IoMT çalışmalarında, işlem kapasitesinin veri kaynağına yaklaştırılmasının gerçek zamanlı çıkarımı kolaylaştırdığı; buna karşılık cihazların bellek, enerji ve hesaplama kısıtlarının model tasarımında dikkate alınması gerektiği vurgulanmaktadır (Abbas vd., 2024).

IoMT sistemlerinde teknik başarım tek başına yeterli değildir. Üretilen verilerin doğrudan bireyin sağlık durumuna ilişkin olması nedeniyle güvenlik, mahremiyet, veri bütünlüğü ve erişim kontrolü temel tasarım gereksinimleridir. Bir tıbbi sensörden elde edilen hatalı veya değiştirilmiş veri yalnızca bilgi güvenliği problemi oluşturmaz; klinik değerlendirmeyi etkileyebileceği için doğrudan hasta güvenliği sorununa dönüşebilir. Benzer şekilde sağlık verilerinin yetkisiz kişilerin erişimine açılması ciddi mahremiyet riskleri doğurmaktadır. Güncel IoMT literatüründe veri güvenliği ve mahremiyetin yanı sıra birlikte çalışabilirlik, düzenleyici uyumluluk, güvenilirlik, maliyet ve kullanıcı kabulü de yaygın uygulamanın önündeki temel sorunlar arasında gösterilmektedir.

IoMT açısından özellikle birlikte çalışabilirlik kritik önem taşımaktadır. Farklı üreticilere ait giyilebilir cihazların, tıbbi sensörlerin, mobil uygulamaların ve hastane bilgi sistemlerinin aynı sağlık ekosistemi içerisinde veri alışverişi yapabilmesi gerekir. Ancak veri formatlarının, iletişim protokollerinin ve klinik terminolojilerin farklı olması, cihaz düzeyindeki bağlantının sağlık bilgi sistemleri arasında anlamlı veri bütünleşmesine dönüşmesini güçleştirebilir. Bu nedenle gelecekteki IoMT sistemlerinde yalnızca bağlantı kurulmasına değil, sağlık verisinin güvenilir ve semantik olarak tutarlı biçimde paylaşılmasına yönelik standartların geliştirilmesi önemini koruyacaktır.

Sonuç olarak IoMT, sağlık hizmetlerini belirli bir zaman ve mekânda gerçekleştirilen ölçümlerden sürekli, bağlantılı ve veri temelli sağlık izleme sistemlerine doğru dönüştürmektedir. Giyilebilir sensörler ve uzaktan hasta izleme çözümleri bireyin sağlık durumunun günlük yaşam içerisinde takip

edilmesini sağlarken, akıllı hastane sistemleri IoMT'yi operasyonel süreçlere taşımaktadır. AI ve edge computing ile bütünleşme ise bu sistemlere gerçek zamanlı analiz ve erken karar desteği kazandırmaktadır. Bununla birlikte sağlık alanının doğrudan insan yaşamıyla ilişkili olması nedeniyle IoMT uygulamalarında güvenilirlik, veri kalitesi, mahremiyet, güvenlik ve birlikte çalışabilirlik teknik performans kadar belirleyici unsurlar olarak değerlendirilmelidir.

Tablo 2. Sağlık alanında başlıca IoMT uygulamaları

Uygulama	**Temel veri/bileşen**	**Sağlanan işlev**
Uzaktan hasta izleme	Kalp hızı, SpO_2, kan basıncı, sıcaklık	Sürekli sağlık takibi ve erken uyarı
Giyilebilir sağlık sistemleri	Akıllı saat, biyosensör, aktivite sensörü	Günlük fizyolojik ve davranışsal izleme
Kronik hastalık yönetimi	Glikoz, kardiyak ve diğer uzun dönem verileri	Hastalığın sürekli takibi
Akıllı hastane	Hasta, yatak ve tıbbi cihaz verileri	Kaynak ve operasyon yönetimi
AIoT tabanlı sağlık	Çoklu sensör ve klinik veriler	Tahmin, sınıflandırma ve anomali tespiti
Edge-IoMT	Yerel sensör akışları	Düşük gecikmeli gerçek zamanlı analiz

AKILLI TARIM VE HASSAS TARIM

Tarımda üretim verimliliğinin artırılması, su ve enerji gibi sınırlı kaynakların daha etkin kullanılması ve iklim değişkenliğinin neden olduğu belirsizliklerin yönetilmesi, veri temelli üretim yaklaşımlarına olan gereksinimi artırmıştır. Bu kapsamda hassas tarım (precision agriculture), tarımsal alan içerisindeki mekânsal ve zamansal değişkenliği dikkate alarak girdilerin ihtiyaç duyulan yerde, zamanda ve miktarda uygulanmasını hedeflemektedir. Akıllı tarım (smart agriculture) ise bu yaklaşımı IoT, yapay zekâ, uzaktan algılama, robotik, insansız hava araçları ve veri analitiği gibi teknolojilerle genişleterek izleme, karar verme ve kontrol süreçlerini bütünleştirmektedir. IoT bu yapı içerisinde fiziksel tarım ortamından sürekli veri sağlayan temel bağlantı katmanını oluşturmaktadır (Soussi vd., 2024; Kumar vd., 2024).

Tarımsal IoT sistemlerinde veri üretimi çoğunlukla toprak, bitki ve atmosfer koşullarının ölçülmesine dayanır. Toprak nemi, sıcaklık, elektriksel iletkenlik ve pH; hava sıcaklığı, bağıl nem, yağış, güneş ışınımı ve rüzgâr; bitki gelişimine ilişkin görüntüsel veya spektral göstergeler farklı sensörler aracılığıyla izlenebilir. Bu ölçümlerin aynı veri platformunda birleştirilmesi, üretim alanındaki değişkenliğin daha ayrıntılı biçimde değerlendirilmesini sağlar. Akıllı sensörlerin IoT, büyük veri analitiği ve yapay zekâ ile bütünleştirilmesi; sulama, gübreleme ve bitki sağlığı yönetiminin sabit uygulama programları yerine gerçek tarla koşullarına göre düzenlenmesine olanak vermektedir (Soussi vd., 2024).

IoT'nin tarımdaki en belirgin uygulamalarından biri akıllı sulama sistemleridir. Geleneksel sulama yaklaşımlarında su uygulaması çoğunlukla zaman çizelgesi veya üreticinin deneyimine göre gerçekleştirilirken, IoT tabanlı sistemlerde toprak nemi, meteorolojik koşullar ve bitki su gereksinimi eş zamanlı olarak değerlendirilebilir. Ölçümler belirlenen sınırların dışına çıktığında sulama sistemleri otomatik olarak etkinleştirilebilir veya uygulanacak su miktarı dinamik biçimde düzenlenebilir. Makine öğrenmesinin bu sürece eklenmesi, yalnızca mevcut nem durumuna göre tepki vermek yerine gelecekteki sulama gereksiniminin tahmin edilmesini de mümkün kılmaktadır. Nsoh vd. (2024), IoT ve makine öğrenmesinin bütünleştirildiği uçtan uca sulama sistemlerinin su kullanım etkinliği ve otomasyon açısından önemli potansiyele sahip olduğunu; ancak birlikte çalışabilirlik, standardizasyon ve siber güvenlik sorunlarının henüz tam olarak çözülemediğini belirtmektedir.

Akıllı tarım uygulamalarında bitki sağlığının izlenmesi ve hastalıkların erken belirlenmesi diğer önemli kullanım alanlarından biridir. Tarla sensörlerinden elde edilen çevresel veriler, RGB veya multispektral görüntüler ve insansız hava araçlarından sağlanan uzaktan algılama verileri birlikte kullanılabilir. Yapay zekâ modelleri bu veri kümelerinde hastalık, zararlı, yabancı ot veya su stresiyle ilişkili örüntüleri belirlemek amacıyla kullanılmaktadır. Bu yaklaşım, tarımsal müdahalelerin bütün araziye eşit biçimde uygulanması yerine yalnızca risk bulunan bölgelerde gerçekleştirilmesine imkân verebilir. IoT, AI ve UAV teknolojilerinin birlikte kullanılması özellikle bitki fenotipleme, hastalık ve yabancı ot belirleme, sulama ve girdi yönetiminde veri temelli karar mekanizmalarının geliştirilmesini desteklemektedir.

IoT tabanlı tarım uygulamalarının kapsamı bitkisel üretimle sınırlı değildir. Akıllı seralarda sıcaklık, nem, ışık, CO_2 düzeyi ve toprak veya yetiştirme ortamına ilişkin parametreler sürekli izlenerek havalandırma, sulama ve iklimlendirme sistemleri otomatik olarak kontrol edilebilir. Hayvancılıkta ise giyilebilir sensörler ve konum belirleme sistemleri kullanılarak hayvanların hareketlilikleri, vücut sıcaklıkları, beslenme davranışları ve konumları takip edilebilir. Bunun yanında IoT çözümleri tarım makinelerinin izlenmesi, ürün depolama koşullarının kontrolü ve tarımsal tedarik zincirlerinin yönetimi gibi üretim sonrası süreçlerde de kullanılmaktadır. Güncel incelemeler akıllı sulama, hassas tarım, bitki ve toprak takibi, akıllı sera, hayvancılık, tarımsal drone sistemleri ve tedarik zinciri yönetimini IoT'nin başlıca tarımsal kullanım alanları arasında göstermektedir (Kumar vd., 2024).

Yapay zekânın IoT altyapısıyla daha sıkı bütünleşmesi, tarım alanında da AIoT yaklaşımını öne çıkarmaktadır. Muhammed vd. (2024), AIoT'nin akıllı tarımda ürün verimi tahmini, toprak ve su kaynaklarının yönetimi, hastalık ve zararlı belirleme ile tedarik zinciri yönetiminde kullanılabildiğini ortaya koymaktadır. Böyle bir sistemde IoT sensörleri fiziksel çevreden veri sağlarken makine öğrenmesi modelleri bu verilerden tahmin veya karar üretmektedir. Bu nedenle tarımsal AIoT'nin temel değeri, algılama ile karar verme arasındaki sürenin kısaltılması ve tarımsal uygulamaların değişen çevre koşullarına uyarlanabilmesidir.

Tarım alanlarının önemli bir bölümünün kırsal bölgelerde bulunması ise genel IoT sistemlerinden farklı bazı mühendislik kısıtları ortaya çıkarmaktadır. Geniş alanlarda haberleşme kapsaması, sensörlerin enerji tüketimi, cihazların zorlu çevre koşullarına dayanıklılığı ve saha bakım maliyetleri sistem tasarımında kritik öneme sahiptir. Bunun yanında farklı sensörlerden elde edilen verilerin kalitesi, cihazlar arasındaki birlikte çalışabilirlik, tarımsal verilerin sahipliği ve kullanıcıların teknolojik altyapıya erişimi de uygulamaların ölçeklenebilirliğini etkileyebilir. Güncel değerlendirmelerde bağlantı sorunları, ilk yatırım maliyeti, veri yönetimi, güvenlik, mahremiyet ve teknik uzmanlık gereksinimi akıllı tarımın yaygınlaşmasının önündeki temel sınırlamalar arasında gösterilmektedir.

Sonuç olarak akıllı tarım, tarımsal süreçlerin yalnızca sayısallaştırılmasını değil, ölçülen koşullara göre kaynak kullanımının uyarlanmasını ve kararların veri temelli hâle getirilmesini amaçlamaktadır. IoT tabanlı algılama altyapısı, AI ve otomatik kontrol sistemleriyle bütünleştirildiğinde

sulama, gübreleme, hastalık yönetimi ve verim tahmini gibi süreçler daha hassas biçimde gerçekleştirilebilir. Bununla birlikte teknolojinin tarımsal sürdürülebilirliğe gerçek katkısı; yüksek model başarımının ötesinde enerji tüketimi, bağlantı olanakları, maliyet, bakım yapılabilirlik ve çiftçi tarafından uygulanabilirlik gibi saha koşullarının da dikkate alınmasına bağlıdır.

Tablo 3. Akıllı tarımda başlıca IoT uygulamaları

Uygulama	Temel veri kaynakları	Temel işlev
Hassas sulama	Toprak nemi, hava durumu, bitki verileri	Su ihtiyacının belirlenmesi ve otomatik sulama
Toprak izleme	Nem, sıcaklık, pH, iletkenlik	Toprak durumunun değerlendirilmesi
Bitki sağlığı	Görüntü, spektral veri, çevresel sensörler	Hastalık ve stres belirleme
Akıllı sera	Sıcaklık, nem, ışık, CO_2	Kontrollü yetiştirme ortamı
Hayvancılık	Konum, hareket, sıcaklık	Hayvan sağlığı ve davranış takibi
Verim tahmini	Sensör, iklim ve geçmiş üretim verileri	Üretim planlama ve karar desteği

ENDÜSTRİ 4.0, ENDÜSTRİYEL IOT VE DİJİTAL İKİZLER

Endüstri 4.0, üretim sistemlerinin sayısallaştırılması ve fiziksel üretim süreçlerinin bilgi teknolojileriyle bütünleştirilmesi üzerine kurulu bir dönüşüm yaklaşımıdır. Bu dönüşümün temel teknolojik bileşenlerinden biri Endüstriyel Nesnelerin İnterneti (Industrial Internet of Things–IIoT) olup üretim makineleri, sensörler, kontrol birimleri ve bilgi sistemleri arasında sürekli veri alışverişinin gerçekleştirilmesini sağlamaktadır. IIoT sayesinde fiziksel üretim süreçleri yalnızca otomasyon sistemleri tarafından kontrol edilen kapalı yapılar olmaktan çıkarak, durumları sürekli gözlemlenebilen ve üretim verileri diğer kurumsal sistemlerle paylaşılabilen bağlantılı sistemlere dönüşmektedir.

IIoT altyapılarında makinelerden titreşim, sıcaklık, basınç, akım, hız ve üretim miktarı gibi operasyonel veriler elde edilebilir. Bu veriler üretim performansının izlenmesinin yanı sıra ekipman sağlığının değerlendirilmesi, enerji tüketiminin analiz edilmesi, ürün kalitesindeki değişimlerin

belirlenmesi ve üretim planlamasının iyileştirilmesi amacıyla kullanılabilir. Özellikle IoT verilerinin yapay zekâ ve makine öğrenmesi yöntemleriyle bütünleştirilmesi, üretim sistemlerinde yalnızca mevcut durumun izlenmesine değil, gelecekte ortaya çıkabilecek sorunların tahmin edilmesine de olanak sağlamaktadır.

Bu yaklaşımın en yaygın uygulamalarından biri kestirimci bakım (predictive maintenance) sistemleridir. Geleneksel düzeltici bakımda ekipmana arıza gerçekleştikten sonra müdahale edilirken, periyodik bakımda bakım faaliyetleri ekipmanın gerçek durumundan bağımsız olarak belirli zaman aralıklarında yürütülmektedir. Kestirimci bakımda ise sensörlerden elde edilen durum verileri analiz edilerek ekipmanın bozulma eğilimi veya arıza olasılığı tahmin edilmeye çalışılır. Böylece bakım faaliyetleri gerçek ekipman durumuyla ilişkilendirilebilir ve plansız üretim duruşlarının azaltılması hedeflenebilir. Güncel sistematik çalışmalar, IoT ve makine öğrenmesinin gerçek zamanlı arıza belirleme ve kestirimci bakım uygulamalarının temel teknolojileri arasında bulunduğunu göstermektedir (Velmurugan ve Dhingra, 2025).

IIoT'nin diğer önemli kullanım alanları arasında üretim hattının gerçek zamanlı izlenmesi, otomatik kalite kontrol, varlık takibi ve üretim kaynaklarının optimizasyonu bulunmaktadır. Görüntüleme sistemleri ve yapay zekâ modelleri üretim hatlarında hatalı ürünlerin belirlenmesinde kullanılabilirken, bağlı sensörler ekipmanların çalışma süreleri ve performans göstergeleri hakkında sürekli veri sağlayabilir. Böylece üretim yönetimi geçmiş kayıtlara dayalı değerlendirmeden, fiziksel sistemin güncel durumuna dayanan veri temelli karar mekanizmalarına doğru ilerlemektedir.

Endüstri 4.0 ile yakından ilişkili diğer bir teknoloji dijital ikiz (digital twin) yaklaşımıdır. Dijital ikiz, fiziksel bir varlık, makine, üretim hattı veya sürecin verilerle sürekli olarak güncellenen sayısal temsilidir. Basit bir üç boyutlu modelden farklı olarak dijital ikizin temel özelliği fiziksel sistemle veri alışverişinin sürdürülmesidir. Bu nedenle IoT, fiziksel varlık ile onun dijital karşılığı arasındaki bağlantının kurulmasında temel rol üstlenmektedir.

Dijital ikiz yapısındaki ilişki genel olarak şu biçimde ifade edilebilir; fiziksel sistem, IoT sensörleri, veri işleme, dijital ikiz, analiz/simülasyon, karar, fiziksel sistem.

Bu döngü içerisinde IoT sensörleri fiziksel varlığın mevcut durumunu dijital modele aktarırken, dijital ikiz üzerinde gerçekleştirilen simülasyon veya yapay zekâ tabanlı analizler fiziksel sistem hakkında tahmin ve optimizasyon sonuçları üretebilir. Örneğin bir üretim makinesinin sıcaklık, titreşim ve yük verileri dijital ikiz üzerinde sürekli güncellenerek makinenin çalışma davranışı izlenebilir ve olası arıza senaryoları fiziksel sisteme müdahale edilmeden önce sayısal ortamda değerlendirilebilir. Bu yönüyle dijital ikiz, IIoT tarafından sağlanan gerçek zamanlı veriyi simülasyon ve karar destek süreçleriyle bütünleştiren önemli bir Endüstri 4.0 teknolojisidir.

IIoT ve dijital ikiz uygulamalarının yaygınlaşması bazı teknik sorunları da beraberinde getirmektedir. Endüstriyel ortamlarda düşük gecikme, yüksek güvenilirlik ve kesintisiz çalışma gereksinimleri tüketici IoT sistemlerine kıyasla daha kritik olabilir. Bunun yanında eski üretim makinelerinin yeni IoT altyapılarıyla bütünleştirilmesi, heterojen protokoller, veri kalitesi, siber güvenlik ve yüksek miktardaki endüstriyel verinin yönetimi önemli mühendislik sorunlarıdır. Kestirimci bakım literatüründe veri eksikliği, sistemlerin gerçek üretim ortamlarında uygulanma güçlükleri ve yatırım geri dönüşünün ölçülmesi de uygulamanın önündeki önemli kısıtlar arasında gösterilmektedir (Velmurugan ve Dhingra, 2025).

Endüstriyel dönüşümün güncel tartışmalarında Endüstri 5.0 yaklaşımı da giderek önem kazanmaktadır. Endüstri 4.0 büyük ölçüde otomasyon, bağlantılı üretim ve verimlilik ekseninde gelişirken Endüstri 5.0 bu teknolojik altyapıyı insan merkezlilik, sürdürülebilirlik ve dayanıklılık ilkeleriyle tamamlamayı amaçlamaktadır. IoT ve siber-fiziksel sistemler bu dönüşümde önemini korumakla birlikte, teknolojinin insanı üretim sisteminden tamamen çıkarması yerine insan-makine iş birliğini desteklemesi ön plana çıkmaktadır (Gladysz vd., 2023). Güncel çalışmalar da akıllı üretimin geleceğinde insan-makine etkileşimi, sürdürülebilir üretim ve dayanıklı sistemlerin birlikte değerlendirilmesi gerektiğine dikkat çekmektedir.

Sonuç olarak IIoT, Endüstri 4.0'ın fiziksel üretim süreçleri ile sayısal sistemler arasındaki veri bağlantısını sağlayan temel bileşenlerinden biridir. Sensör verilerinin yapay zekâ, kestirimci analitik ve dijital ikizlerle bütünleştirilmesi; üretim sistemlerinin yalnızca izlenmesini değil, durumlarının tahmin edilmesini ve operasyonların optimize edilmesini de mümkün kılmaktadır. Bu gelişim, Endüstri 5.0 perspektifiyle birlikte değerlendirildiğinde geleceğin endüstriyel IoT sistemlerinin yalnızca verimli

değil, aynı zamanda daha sürdürülebilir, dayanıklı ve insan merkezli olması gerektiğini göstermektedir.

Tablo 4. Endüstriyel IoT ve dijital ikizlerin başlıca uygulamaları

Uygulama	Kullanılan veriler	Temel amaç
Durum izleme	Titreşim, sıcaklık, basınç, akım	Makine sağlığının izlenmesi
Kestirimci bakım	Sensör ve geçmiş arıza verileri	Arızaların önceden tahmin edilmesi
Kalite kontrol	Görüntü ve proses verileri	Hatalı ürünlerin belirlenmesi
Üretim optimizasyonu	Makine, süreç ve üretim verileri	Verimlilik ve kaynak kullanımının iyileştirilmesi
Dijital ikiz	Gerçek zamanlı çoklu sensör verileri	Simülasyon, tahmin ve optimizasyon

AKILLI ENERJİ VE ULAŞIM SİSTEMLERİ

Enerji ve ulaşım, IoT tabanlı akıllı sistemlerin en yoğun kullanıldığı ve kent ölçeğinde birbirleriyle doğrudan etkileşime giren iki temel altyapı alanıdır. Enerji sistemlerinde üretim, iletim, dağıtım ve tüketimin gerçek zamanlı olarak izlenmesi gerekirken, ulaşım sistemlerinde araç, yol, trafik ve kullanıcı hareketliliğine ilişkin verilerin sürekli değerlendirilmesi gerekmektedir. IoT her iki alanda da fiziksel altyapının gözlemlenebilir hâle getirilmesini, verilerin ağ üzerinden paylaşılmasını ve analitik yöntemlerle karar süreçlerine aktarılmasını sağlamaktadır.

Enerji alanında IoT'nin başlıca uygulama ortamlarından biri akıllı şebekelerdir (smart grids). Geleneksel elektrik şebekelerinden farklı olarak akıllı şebekeler, enerji akışının yanında çift yönlü bilgi iletişimini de destekleyen dağıtık yapılardır. Akıllı sayaçlar, gerilim ve akım sensörleri, dağıtım otomasyonu cihazları ve güç kalitesi ölçüm sistemleri aracılığıyla şebekenin farklı noktalarından sürekli veri elde edilebilir. Bu veriler tüketim profilinin belirlenmesi, arızaların tespiti, güç kalitesinin izlenmesi ve şebeke işletiminin iyileştirilmesi amacıyla kullanılmaktadır. IoT'nin akıllı şebekelere entegrasyonu sistemin gözlemlenebilirliğini artırmakla birlikte ölçeklenebilirlik, birlikte çalışabilirlik, veri yönetimi ve güvenlik gibi yeni gereksinimler de oluşturmaktadır (Tabassum vd., 2024).

Akıllı sayaçlar, tüketici ile enerji sistemi arasındaki bu veri ilişkisinin önemli bileşenlerinden biridir. Elektrik tüketiminin belirli zaman aralıklarıyla ölçülmesi, enerji talebinin zamana bağlı olarak analiz edilmesine ve talep tarafı yönetimi uygulamalarının geliştirilmesine imkân verir. Büyük miktarda ölçüm verisinin IoT ve bulut altyapılarıyla bütünleştirilmesi, tüketim örüntülerinin belirlenmesi, yük tahmini ve enerji kullanımının optimize edilmesi gibi uygulamaları desteklemektedir (Al-Ali vd., 2024).

IoT ayrıca güneş ve rüzgâr gibi dağıtık yenilenebilir enerji kaynaklarının şebekeye entegrasyonunda önemli rol üstlenmektedir. Üretim miktarı, meteorolojik koşullar, ekipman sıcaklığı, batarya durumu ve yük talebi gibi değişkenlerin eş zamanlı izlenmesi, enerji üretimi ve tüketimi arasındaki dengenin daha dinamik biçimde yönetilmesini mümkün kılmaktadır. IoT'nin bulut ve edge bilişimle birlikte kullanılması, yenilenebilir enerji sistemlerinde gerçek zamanlı izleme ve tahmin tabanlı enerji yönetiminin geliştirilmesini desteklemektedir. Bununla birlikte enerji sistemlerinin kritik altyapı niteliği taşıması nedeniyle iletişim güvenilirliği ve siber güvenlik temel tasarım koşulları arasında yer almaktadır.

Akıllı enerji sistemlerinin sürdürülebilirlik açısından taşıdığı önem yalnızca yenilenebilir enerji kullanımının artırılmasıyla sınırlı değildir. IoT tabanlı ölçüm ve kontrol sistemleri binalarda, endüstriyel tesislerde ve şebeke düzeyinde gereksiz enerji tüketiminin belirlenmesine ve kaynakların ihtiyaca göre yönetilmesine katkı sağlayabilir. Bununla birlikte IoT cihazlarının kendilerinin de enerji tüketmesi, çok sayıda sensörün ve haberleşme bileşeninin oluşturduğu çevresel yük nedeniyle Green IoT yaklaşımını önemli hâle getirmektedir. Güncel çalışmalar akıllı şebekelerde enerji verimli IoT cihazları, düşük güçlü haberleşme ve kaynak optimizasyonunun sürdürülebilir sistem tasarımının önemli bileşenleri olduğunu vurgulamaktadır.

Benzer bir veri temelli dönüşüm ulaşım sistemlerinde de görülmektedir. Akıllı Ulaşım Sistemleri (Intelligent Transportation Systems–ITS), sensörler, bağlantılı araçlar, kameralar, konum sistemleri ve haberleşme altyapılarından elde edilen verileri kullanarak ulaşımın güvenliğini, verimliliğini ve sürdürülebilirliğini geliştirmeyi amaçlamaktadır. IoT sayesinde trafik yoğunluğu, araç hızı, konum, yol durumu ve park alanlarının doluluk bilgisi gerçek zamanlı olarak elde edilebilir. Bu veriler trafik yönetim merkezlerinde veya edge tabanlı sistemlerde işlenerek trafik akışının

düzenlenmesi, dinamik güzergâh oluşturulması ve sürücülere anlık bilgi sağlanması amacıyla kullanılabilir. Güncel ITS çalışmalarında trafik yönetimi, bağlantılı ve otonom araçlar ile sürdürülebilir mobilite temel uygulama alanları arasında değerlendirilmektedir (Elassy vd., 2024).

Akıllı trafik yönetiminde IoT verisinin yapay zekâ ile bütünleştirilmesi özellikle önemlidir. Geleneksel trafik ışıklarında sinyal süreleri önceden belirlenmiş zaman planlarına göre çalışabilirken, akıllı sistemlerde kavşaklardan elde edilen araç yoğunluğu ve trafik akışı verileri kullanılarak sinyal süreleri mevcut koşullara göre değiştirilebilir. Benzer biçimde geçmiş ve gerçek zamanlı trafik verilerinden yararlanan tahmin modelleri, belirli güzergâhlarda oluşabilecek yoğunluğu önceden belirleyerek trafik yönetim sistemlerinin proaktif karar üretmesini sağlayabilir.

Ulaşım alanındaki diğer önemli gelişme bağlantılı araçlar ve Vehicle-to-Everything (V2X) iletişimidir. V2X yaklaşımı araçların diğer araçlarla, yol altyapısıyla, yayalarla ve iletişim ağlarıyla veri alışverişi gerçekleştirebilmesini ifade eder. Böyle bir yapıda araçlar çevrelerinden yalnızca kendi sensörleri aracılığıyla değil, diğer trafik bileşenleri tarafından paylaşılan veriler aracılığıyla da bilgi edinebilir. Bu durum çarpışma uyarısı, kavşak güvenliği, trafik koordinasyonu ve otonom sürüş gibi uygulamalarda daha geniş bir çevresel farkındalık sağlayabilir.

Enerji ve ulaşım sistemlerinin giderek daha fazla kesiştiği alan ise elektrikli araçlar ve akıllı şarj altyapılarıdır. Elektrikli araç sayısının artması, şarj işlemlerinin elektrik şebekesi açısından yeni bir talep bileşeni hâline gelmesine neden olmaktadır. IoT tabanlı şarj sistemlerinde aracın batarya durumu, şarj istasyonunun kullanılabilirliği, elektrik fiyatı ve şebeke yükü birlikte değerlendirilerek şarj zamanlaması optimize edilebilir. Bu yönüyle geleceğin akıllı ulaşım sistemleri yalnızca yol ve trafik altyapısıyla değil, enerji şebekeleriyle de bütünleşik olarak çalışacaktır.

Her iki uygulama alanında da gerçek zamanlı veri kullanımının artması önemli teknik ve toplumsal sorunlar ortaya çıkarmaktadır. Enerji sistemlerinde yanlış veya değiştirilmiş sensör verileri şebeke operasyonlarını etkileyebilirken, ulaşım sistemlerinde iletişim gecikmeleri veya hatalı kararlar doğrudan fiziksel güvenlik riski oluşturabilir. Konum, araç hareketi ve tüketim alışkanlıkları gibi veriler ise kullanıcı davranışları hakkında ayrıntılı bilgi sağlayabildiğinden mahremiyet açısından hassastır. Bu nedenle akıllı enerji ve ulaşım sistemlerinde düşük gecikme, güvenilir haberleşme, siber güvenlik, veri bütünlüğü ve mahremiyet birlikte değerlendirilmelidir.

Sonuç olarak IoT, enerji ve ulaşım altyapılarını sürekli gözlemlenebilen ve değişen koşullara göre yönetilebilen sistemlere dönüştürmektedir. Akıllı şebekelerde enerji üretimi ve tüketiminin, akıllı ulaşım sistemlerinde ise araç ve trafik hareketlerinin gerçek zamanlı olarak izlenmesi; yapay zekâ ve edge bilişimle birlikte kullanıldığında tahmin ve optimizasyon yeteneklerini geliştirmektedir. Elektrikli araçlar ve akıllı şarj sistemleri ise enerji ile ulaşım arasındaki sınırların giderek ortadan kalktığını ve geleceğin IoT ekosistemlerinin farklı kritik altyapıları ortak veri ve karar mekanizmaları üzerinden bütünleştireceğini göstermektedir.

Tablo 5. Akıllı enerji ve ulaşım sistemlerinde temel IoT uygulamaları

Alan	**IoT uygulaması**	**Temel amaç**
Enerji	Akıllı sayaçlar	Tüketimin gerçek zamanlı izlenmesi
Enerji	Akıllı şebeke	Üretim, dağıtım ve talebin koordinasyonu
Enerji	Yenilenebilir enerji izleme	Üretim tahmini ve kaynak yönetimi
Ulaşım	Akıllı trafik yönetimi	Trafik akışının izlenmesi ve optimizasyonu
Ulaşım	V2X ve bağlantılı araçlar	Araç ve altyapı arasında bilgi paylaşımı
Enerji–Ulaşım	Akıllı şarj sistemleri	Araç şarjı ile şebeke koşullarının koordinasyonu

ETİK, MAHREMİYET VE SÜRDÜRÜLEBİLİRLİK

Akıllı IoT sistemlerinin yaygınlaşması, yalnızca teknik performansla açıklanamayacak yeni sorun alanlarını da beraberinde getirmektedir. Sensörler, giyilebilir cihazlar, kameralar, akıllı sayaçlar ve bağlantılı araçlar aracılığıyla sürekli veri üretilmesi; bireylerin davranışları, konumları, sağlık durumları ve günlük alışkanlıkları hakkında ayrıntılı çıkarımlar yapılabilmesini mümkün kılmaktadır. Bu nedenle akıllı IoT sistemlerinin değerlendirilmesinde mahremiyet, güvenlik, şeffaflık, hesap verebilirlik ve sürdürülebilirlik temel tasarım ölçütleri arasında yer almalıdır.

Mahremiyet açısından temel problem yalnızca kişisel verilerin doğrudan toplanması değildir. Farklı IoT kaynaklarından elde edilen veriler bir araya getirildiğinde, tek başına hassas görünmeyen ölçümlerden kullanıcı hakkında yeni bilgiler çıkarılabilir. Örneğin akıllı sayaç verileri günlük yaşam alışkanlıklarını, konum verileri hareket örüntülerini, giyilebilir cihaz verileri

ise sağlık ve aktivite durumunu gösterebilir. Bu nedenle IoT sistemlerinde veri minimizasyonu, yalnızca hizmetin gerçekleştirilmesi için gerekli verinin toplanması, verinin hangi amaçla işlendiğinin açık biçimde belirlenmesi ve erişim yetkilerinin sınırlandırılması önem taşımaktadır. Akıllı şehir veri yönetişimi açısından da mahremiyet, veri güvenliği ve birlikte çalışabilirlik konularının teknik altyapıdan bağımsız düşünülmemesi gerektiği vurgulanmaktadır (OECD, 2023).

AIoT sistemlerinde mahremiyet sorunlarına ek olarak algoritmik karar süreçleri yeni etik sorular ortaya çıkarmaktadır. Sensör verilerinden elde edilen tahminler sağlık, ulaşım, enerji kullanımı veya kamu hizmetleriyle ilgili karar süreçlerinde kullanıldığında modelin hangi verilerle eğitildiği ve hangi koşullar altında hata yaptığı önem kazanır. Eğitim verisinde bulunan sistematik önyargılar, belirli kullanıcı veya toplulukların hatalı sınıflandırılmasına neden olabilir. Bu nedenle yüksek tahmin doğruluğu tek başına yeterli bir başarı ölçütü değildir; adalet, açıklanabilirlik, insan gözetimi ve hesap verebilirlik de özellikle bireyleri doğrudan etkileyen AIoT uygulamalarında dikkate alınmalıdır. UNESCO'nun yapay zekâ etiğine ilişkin yaklaşımı da insan hakları, şeffaflık, sorumluluk ve insan gözetimini güvenilir yapay zekâ sistemlerinin temel ilkeleri arasında değerlendirmektedir (UNESCO, 2021).

Hesap verebilirlik konusu, AIoT sistemlerinde fiziksel eylem üretildiğinde daha kritik hâle gelir. Bir yapay zekâ modelinin yanlış tahmini sonucunda akıllı binada hatalı kontrol gerçekleştirilmesi ile sağlık veya ulaşım sisteminde yanlış karar verilmesi aynı risk düzeyine sahip değildir. Bu nedenle sistemlerin risk temelli olarak değerlendirilmesi gerekir. Kararın insan tarafından onaylanması, sistem tarafından otomatik olarak uygulanması veya yalnızca öneri şeklinde sunulması uygulamanın kritikliği doğrultusunda belirlenmelidir. İnsan yaşamı ve güvenliği üzerinde yüksek etkisi bulunan sistemlerde tamamen otonom karar mekanizmalarının sınırlandırılması ve gerektiğinde insan müdahalesine olanak sağlayan tasarımların kullanılması daha uygun olabilir.

Etik değerlendirmelerin bir diğer boyutunu sürdürülebilirlik oluşturmaktadır. IoT teknolojileri çevresel sürdürülebilirliği destekleyen önemli uygulamalara sahiptir. Akıllı enerji sistemleri enerji tüketiminin optimize edilmesine, hassas tarım su ve gübre kullanımının azaltılmasına, akıllı ulaşım trafik kaynaklı enerji kayıplarının düşürülmesine ve çevresel sensör ağları doğal kaynakların daha etkin izlenmesine katkı sağlayabilir.

Bununla birlikte IoT'nin sürdürülebilirliğe katkısı yalnızca sağladığı hizmet üzerinden değerlendirilmemelidir. Milyarlarca cihazın üretimi, enerji tüketimi, batarya gereksinimi, haberleşme yükü ve kullanım ömrünü tamamlayan cihazların oluşturduğu elektronik atık da sistemin çevresel maliyetinin bir parçasıdır.

Bu nedenle Green IoT yaklaşımı, IoT sistemlerinin tüm yaşam döngüsünde enerji ve kaynak kullanımının azaltılmasını hedeflemektedir. Düşük güçlü sensörler ve haberleşme teknolojilerinin kullanılması, gereksiz veri aktarımının azaltılması, edge üzerinde veri filtreleme, enerji hasadı yöntemleri, uyku modlarının etkin kullanılması ve cihazların kullanım ömrünün uzatılması bu yaklaşımın temel uygulamaları arasındadır. Akıllı IoT platformlarında enerji verimliliğinin yalnızca cihaz seviyesinde değil, veri toplama, iletişim, edge–cloud iş birliği ve yapay zekâ kaynak yönetimi düzeylerinde birlikte değerlendirilmesi gerektiği ITU-T tarafından da vurgulanmaktadır (ITU-T, 2025).

Sürdürülebilir AIoT açısından yapay zekâ modellerinin hesaplama maliyeti de dikkate alınmalıdır. Daha karmaşık bir modelin küçük bir doğruluk artışı sağlaması, eğer enerji tüketimini veya donanım gereksinimini önemli ölçüde artırıyorsa kaynak kısıtlı IoT sistemlerinde uygun bir tasarım tercihi olmayabilir. Bu nedenle gelecekte model başarımının yalnızca doğruluk üzerinden değil, işlem maliyeti ve enerji tüketimiyle birlikte değerlendirilmesi beklenmektedir. Böyle bir yaklaşım, özellikle uzun süre batarya ile çalışan sensör düğümleri ve büyük ölçekli IoT ağları açısından önemlidir.

Tablo 6. Akıllı IoT sistemlerinde etik ve sürdürülebilirlik açısından temel konular

Boyut	**Temel sorun**	**Tasarım yaklaşımı**
Mahremiyet	Sürekli ve ayrıntılı veri toplama	Veri minimizasyonu ve amaç sınırlaması
Güvenlik	Yetkisiz erişim ve veri manipülasyonu	Kimlik doğrulama, erişim kontrolü ve güvenli iletişim
Şeffaflık	AI kararlarının anlaşılabilir olmaması	Açıklanabilir ve izlenebilir karar süreçleri
Adalet	Veri veya model kaynaklı önyargı	Veri ve model performansının farklı gruplarda değerlendirilmesi
Hesap verebilirlik	Hatalı otomatik kararların sorumluluğu	İnsan gözetimi ve risk temelli otonomi
Enerji	Cihaz ve iletişim tüketimi	Düşük güçlü donanım ve edge işleme
Elektronik atık	Kısa cihaz yaşam döngüsü	Onarılabilirlik ve yaşam döngüsü yönetimi

Sonuç olarak akıllı IoT sistemlerinin başarısı yalnızca daha fazla veri üretmek veya daha yüksek doğrulukta tahmin gerçekleştirmekle ölçülmemelidir. Kullanıcı mahremiyetini koruyan, karar süreçleri denetlenebilir, enerji ve kaynak kullanımını dikkate alan sistemlerin geliştirilmesi uzun dönemli teknoloji kabulü açısından belirleyicidir. Bu nedenle gelecek nesil IoT sistemlerinin akıllı olmanın yanında güvenilir, insan merkezli ve sürdürülebilir olması gerekmektedir.

AKILLI IOT SISTEMLERINDE GELECEK EĞILIMLERI

Akıllı IoT sistemlerinin geleceği, yalnızca daha fazla cihazın internete bağlanmasına değil, hesaplama, iletişim ve yapay zekâ yeteneklerinin fiziksel nesnelere daha yakın biçimde bütünleştirilmesine dayanmaktadır. Mevcut gelişmeler, merkezi bulut tabanlı IoT mimarilerinden cihaz, edge ve bulut katmanlarının birlikte çalıştığı dağıtık yapılara doğru belirgin bir yönelim olduğunu göstermektedir. Bu dönüşümde Edge AI, TinyML, federated learning, yeni nesil haberleşme sistemleri ve üretken yapay zekâ önemli teknolojik eğilimler olarak öne çıkmaktadır.

Gelecek nesil IoT sistemlerinin temel özelliklerinden biri, yapay zekâ tabanlı işlemlerin giderek daha fazla verinin üretildiği noktaya taşınmasıdır. Edge AI yaklaşımında sensörlerden elde edilen verilerin tamamının uzak bulut ortamına gönderilmesi yerine sınıflandırma, tahmin veya anomali tespiti gibi işlemler edge düğümlerinde gerçekleştirilebilir. Bu yapı özellikle düşük gecikmenin gerekli olduğu endüstriyel kontrol, sağlık ve ulaşım uygulamalarında önem taşımaktadır. Edge hesaplamanın daha ileri bir biçimi olan Tiny Machine Learning (TinyML) ise makine öğrenmesi modellerinin mikrodenetleyici sınıfındaki düşük güçlü cihazlarda çalıştırılmasını hedeflemektedir. Oliveira vd. (2024), gömülü sistemler, edge computing ve makine öğrenmesinin yakınsamasının IoT cihazlarını yalnızca veri üreten bileşenlerden yerel çıkarım gerçekleştirebilen akıllı düğümlere dönüştürdüğünü belirtmektedir.

TinyML'nin yaygınlaşması özellikle enerji kısıtlı sistemler açısından önemli sonuçlar doğurabilir. Sensör verisinin cihaz üzerinde işlenmesi, gereksiz veri aktarımını azaltarak haberleşme maliyetini ve enerji tüketimini düşürebilir. Bununla birlikte IoT cihazlarının sınırlı bellek ve işlem kapasitesi, geleneksel yapay zekâ modellerinin doğrudan kullanılmasını güçleştirmektedir. Bu nedenle model niceleme, budama ve diğer model sıkıştırma yöntemlerinin kaynak kısıtlı IoT sistemlerinde daha fazla önem kazanması beklenmektedir.

Diğer önemli eğilim federated learning (federe öğrenme) yaklaşımıdır. Geleneksel merkezi makine öğrenmesi süreçlerinde farklı cihazlardan elde edilen veriler ortak bir sunucuda toplanırken, federated learning cihazların yerel verilerini merkezi sisteme aktarmadan ortak bir modelin geliştirilmesine olanak verir. Cihazlar kendi verileri üzerinde yerel eğitim gerçekleştirdikten sonra model güncellemelerini paylaşır ve bu güncellemeler ortak bir model oluşturmak amacıyla birleştirilir. Bu özellik özellikle sağlık, akıllı şehir ve kişisel IoT uygulamalarında veri mahremiyetinin desteklenmesi açısından önemlidir. Bununla birlikte cihazların farklı işlem kapasitelerine, ağ koşullarına ve veri dağılımlarına sahip olması federated learning'in gerçek IoT ortamlarında uygulanmasını zorlaştırmaktadır. Bu nedenle lightweight federated learning ve TinyML ile federated learning'in birlikte kullanılması güncel araştırma yönelimlerinden biri hâline gelmiştir (Qi vd., 2025).

IoT'nin geleceğini şekillendirecek bir diğer gelişme 5G-Advanced ve 6G iletişim altyapılarıdır. Özellikle çok büyük cihaz yoğunluğu, çok düşük gecikme, yüksek güvenilirlik ve geniş coğrafi kapsama gerektiren IoT uygulamaları mevcut iletişim altyapılarının sınırlarını zorlamaktadır. 6G araştırmalarında yapay zekâ destekli ağ yönetimi, edge intelligence, yeniden yapılandırılabilir akıllı yüzeyler, terahertz haberleşme ve karasal ağların uydu ve hava platformlarıyla bütünleştirilmesi gibi teknolojiler üzerinde durulmaktadır. 6G henüz geleceğe yönelik bir iletişim paradigması olmakla birlikte, IoT açısından temel hedeflerden biri karasal, hava ve uydu sistemlerini kapsayan daha yaygın ve akıllı bağlantı ortamlarının oluşturulmasıdır (Aarti vd., 2025).

Bu gelişmeler aynı zamanda otonom IoT kavramını güçlendirmektedir. Gelecekte IoT düğümlerinin yalnızca merkezi sistemlerden gelen komutları uygulayan cihazlar olmak yerine çevresel koşulları değerlendiren, diğer cihazlarla iş birliği yapan ve belirli sınırlar içerisinde kendi kararlarını verebilen bileşenlere dönüşmesi beklenmektedir. Böyle bir yapıda iletişim ağı da pasif veri taşıma altyapısından çıkarak kaynak tahsisi, bağlantı yönetimi ve görev dağılımını yapay zekâ yöntemleriyle dinamik olarak düzenleyen bir sisteme dönüşebilir. Ancak otonomi düzeyi arttıkça güvenilirlik, açıklanabilirlik, hata toleransı ve insan gözetimi gereksinimleri de daha kritik hâle gelecektir.

Daha yeni bir eğilim ise üretken yapay zekâ ve büyük dil modellerinin IoT ekosistemleriyle bütünleştirilmesidir. Büyük dil modellerinin heterojen

IoT verilerinin anlamlandırılması, sistem yönetimi, kullanıcı–cihaz etkileşimi ve güvenlik gibi alanlarda potansiyel kullanım olanakları bulunmaktadır. Bununla birlikte mevcut sistematik incelemeler yüksek hesaplama gereksinimi, eğitim verisinin niteliği, mahremiyet ve kaynak kısıtları nedeniyle bu modellerin IoT cihazlarında doğrudan kullanımının henüz önemli güçlükler taşıdığını göstermektedir (De Vito vd., 2025). Bu nedenle gelecekte daha küçük, görev odaklı ve edge ortamına uyarlanmış üretken modellerin geliştirilmesi olası bir araştırma yönü olarak değerlendirilebilir.

Tablo 7. Akıllı IoT sistemlerinde öne çıkan gelecek eğilimleri

Eğilim	**Temel yaklaşım**	**IoT açısından beklenen katkı**
Edge AI	AI işlemlerinin edge üzerinde gerçekleştirilmesi	Düşük gecikme ve yerel karar verme
TinyML	ML modellerinin mikrodenetleyicilerde çalıştırılması	Düşük enerji tüketimi ve cihaz üzerinde çıkarım
Federated learning	Veriyi merkezileştirmeden ortak model öğrenme	Mahremiyetin ve dağıtık öğrenmenin desteklenmesi
6G-IoT	Yeni nesil akıllı ve yaygın iletişim	Yüksek cihaz yoğunluğu, düşük gecikme ve geniş kapsama
Otonom IoT	Yerel ve dağıtık karar mekanizmaları	İnsan müdahalesine daha az bağımlı sistemler
Generative AI–IoT	Üretken modellerin IoT ile bütünleştirilmesi	Akıllı etkileşim, veri yorumlama ve sistem yönetimi

Bu eğilimler birlikte değerlendirildiğinde IoT'nin geleceğinin bağlantılı nesnelerden dağıtık zekâya doğru geliştiği görülmektedir. Yapay zekâ modellerinin edge ve cihaz düzeyine taşınması, veriyi merkezileştirmeden öğrenebilen sistemlerin geliştirilmesi ve yeni nesil iletişim altyapılarının devreye girmesi, IoT sistemlerinin daha hızlı ve bağlama duyarlı çalışmasını sağlayabilir. Bununla birlikte gelecekteki başarının yalnızca daha fazla otonomi veya daha karmaşık yapay zekâ modelleriyle değil; enerji verimliliği, güvenilirlik, mahremiyet, birlikte çalışabilirlik ve sürdürülebilirlik gereksinimlerinin aynı sistem içerisinde dengelenmesiyle belirleneceği düşünülmektedir.

KAYNAKÇA

Aarti, Gowroju, S. ve Ansarullah, S. I. (2025). A comprehensive review of developments and challenges in the 6G Internet of Things. *Discover Networks, 1*, 13. DOI: 10.1007/s44354-025-00013-y.

Abbas, A. vd. (2024). Edge AI for Internet of Medical Things: A literature review. *Computers & Electrical Engineering, 116*, 109202. DOI: 10.1016/j.compeleceng.2024.109202.

Aksoy B., Yücel M., Sayın H., Yücel N., And Ekrem Ö., "Hurma Meyvesindeki Kalite Kontrol İşlemlerinin Yapay Zeka İle Tahminlenmesi," Gazi Journal of Engineering Sciences, vol. 9, no. 4, pp. 70–81, Dec. 2023.

Aksoy B., Özsoy K., Yücel M., Ekrem Ö., ve Salman O. K. M., "Motion Control of the Robot Arm Manufactured with a Three-Dimensional Printer and Hardness Detection of Objects ," Bilişim Teknolojileri Dergisi, vol. 15, no. 3, pp. 289–300, Jul. 2022.

Al-Ali, A. R., Gupta, R., Zualkernan, I. ve Das, S. K. (2024). Role of IoT technologies in big data management systems: A review and Smart Grid case study. *Pervasive and Mobile Computing, 100*, 101905. DOI: 10.1016/j.pmcj.2024.101905.

Bellini, P., Bologna, D., Nesi, P. ve Pantaleo, G. (2024). A unified knowledge model for managing smart city/IoT platform entities for multitenant scenarios. *Smart Cities, 7*(5), 2339-2365. DOI: 10.3390/smartcities7050092.

Bellini, P., Nesi, P. ve Pantaleo, G. (2022). IoT-enabled smart cities: A review of concepts, frameworks and key technologies. *Applied Sciences, 12*(3), 1607. DOI: 10.3390/app12031607.

De Vito, G. vd. (2025). The role of large language models in addressing IoT challenges: A systematic literature review. *Future Generation Computer Systems, 171*, 107829. DOI: 10.1016/j.future.2025.107829.

Elassy, M., Al-Hattab, M., Takruri, M. ve Badawi, S. (2024). Intelligent transportation systems for sustainable smart cities. *Transportation Engineering, 16*, 100252. DOI: 10.1016/j.treng.2024.100252.

Gladysz, B., Tran, T. A., Romero, D., van Erp, T., Abonyi, J. ve Ruppert, T. (2023). Industry 5.0 and its technologies: A systematic literature review upon the human place into IoT- and CPS-based industrial systems. *Computers & Industrial Engineering, 184*, 109426. DOI: 10.1016/j.cie.2023.109426.

ITU-T. (2018). *Recommendation ITU-T Y.4200: Requirements for the interoperability of smart city platforms.* International Telecommunication Union, Geneva.

ITU-T. (2022). *Recommendation ITU-T Y.4903: Key performance indicators for smart sustainable cities to assess the achievement of sustainable development goals.* International Telecommunication Union, Geneva.

ITU-T. (2023). *Recommendation ITU-T Y.4223: Common requirements and capabilities of smart cities and communities from IoT and ICT perspectives.* International Telecommunication Union, Geneva.

ITU-T. (2025). *Framework of artificial intelligence of things (Recommendation ITU-T Y.4612).* Geneva: International Telecommunication Union.

ITU-T. (2025). *Functional requirements for energy efficiency in intelligent Internet of Things platforms (Recommendation ITU-T L.1341).* International Telecommunication Union, Geneva.

Khadam, U., Davidsson, P. ve Spalazzese, R. (2025). A systematic literature review on AI in IoT systems: Tasks, applications, and deployment. *Internet of Things, 34*, 101779. DOI: 10.1016/j.iot.2025.101779.

Kumar, V., Sharma, K. V., Kedam, N., Patel, A., Kate, T. R. ve Rathnayake, U. (2024). A comprehensive review on smart and sustainable agriculture using IoT technologies. *Smart Agricultural Technology, 8*, 100487. DOI: 10.1016/j.atech.2024.100487.

Muhammed, D., Ahvar, E., Ahvar, S., Trocan, M., Montpetit, M.-J. ve Ehsani, R. (2024). Artificial Intelligence of Things (AIoT) for smart agriculture: A review of architectures, technologies and solutions. *Journal of Network and Computer Applications, 228*, 103905. DOI: 10.1016/j.jnca.2024.103905.

NIST. (2019). *NISTIR 8228: Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks.* National Institute of Standards and Technology, Gaithersburg.

Nsoh, B., Katimbo, A., Guo, H., Heeren, D. M., Nakabuye, H. N., Qiao, X., Ge, Y., Rudnick, D. R., Wanyama, J., Bwambale, E. ve Kiraga, S. (2024). Internet of Things-based automated solutions utilizing machine learning for smart and real-time irrigation management: A review. *Sensors, 24*(23), 7480. DOI: 10.3390/s24237480.

OECD. (2023). *Smart City Data Governance: Challenges and the Way Forward.* OECD Publishing, Paris.

Oliveira, F., Costa, D. G., Assis, F. ve Silva, I. (2024). Internet of Intelligent Things: A convergence of embedded systems, edge computing and machine learning. *Internet of Things, 26*, 101153. DOI: 10.1016/j.iot.2024.101153.

Qi, P., Chiaro, D. ve Piccialli, F. (2025). Small models, big impact: A review on the power of lightweight federated learning. *Future Generation Computer Systems, 162*, 107484. DOI: 10.1016/j.future.2024.107484.

Sharma, A. vd. (2024). Multirole of the Internet of Medical Things (IoMT) in biomedical systems for managing smart healthcare systems: An overview of current and future innovative trends. *Journal of Infection and Public Health, 17*(4), 559–572.

Siam, S. I., Ahn, H., Liu, L., Alam, S., Shen, H., Cao, Z., Shroff, N., Krishnamachari, B., Srivastava, M. ve Zhang, M. (2025). Artificial intelligence of things: A survey. *ACM Transactions on Sensor Networks, 21*(1), 1-75. DOI: 10.1145/3690639.

Soussi, A., Zero, E., Sacile, R., Trinchero, D. ve Fossa, M. (2024). Smart sensors and smart data for precision agriculture: A review. *Sensors, 24*(8), 2647. DOI: 10.3390/s24082647.

Sun, Y. vd. (2023). Internet of medical things: A systematic review. *Neurocomputing, 557*, 126719. DOI: 10.1016/j.neucom.2023.126719.

Sun, Y. vd. (2026). A survey of Internet of medical things: Technology, application and future directions. *Digital Communications and Networks, 12*(5), 717–742. DOI: 10.1016/j.dcan.2024.11.013.

Tabassum, S., Vijay Babu, A. R. ve Dheer, D. K. (2024). A comprehensive exploration of IoT-enabled smart grid systems: Power quality issues, solutions, and challenges. *Science and Technology for Energy Transition, 79*, 62. DOI: 10.2516/stet/2024056.

UNESCO. (2021). *Recommendation on the Ethics of Artificial Intelligence.* United Nations Educational, Scientific and Cultural Organization, Paris.

Velmurugan, R. S. ve Dhingra, T. (2025). Systematic review of predictive maintenance practices in the manufacturing sector. *Intelligent Systems with Applications, 26*, 200501. DOI: 10.1016/j.iswa.2025.200501.

Wang, X., Wang, B., Wu, Y., Ning, Z. vd. (2025). A survey on trustworthy edge intelligence: From security and reliability to transparency and sustainability. *IEEE Communications Surveys & Tutorials, 27*(3), 1729-1757. DOI: 10.1109/COMST.2024.3446585.

Yücel M., Aksoy B., "İnsansız Hava Aracı ile Yangın Bölgesinin Tespiti ve Müdahalesinin Yapay Zeka ile Tahminlenmesi," Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, vol. 28, no. 3, pp. 559–569, Dec. 2024.

Zhang, X. vd. (2024). Human–machine interaction towards Industry 5.0: Human-centric smart manufacturing. *Digital Engineering, 2*, 100013. DOI: 10.1016/j.dte.2024.100013.