Bilgisayar Bilimleri ve Mühendisliği: Teori, Metodoloji ve Uygulama

Editör
Doç. Dr. Arafat Şentürk

DOI: 10.5281/zenodo.17396194

Bilgisayar Bilimleri ve Mühendisliği: Teori, Metodoloji ve Uygulama

Editör

Doç. Dr. Arafat Şentürk

İmtiyaz Sahibi
Platanus Publishing®

Editör
Doç. Dr. Arafat Şentürk

Kapak & Mizanpaj & Sosyal Medya
Platanus Yayın Grubu

Birinci Basım
Ekim, 2025

Yayımcı Sertifika No
45813

ISBN
978-625-6517-00-4

Adres: Natoyolu Cad. Fahri Korutürk Mah. 157/B, 06480, Mamak, Ankara, Türkiye.

Telefon: +90 312 390 1 118
web: www.platanuspublishing.com
e-mail: platanuskitap@gmail.com

Platanus Publishing®

İÇİNDEKİLER


BÖLÜM 1

Kriptoloji ve Modern Şifreleme Sistemleri ve Gelecek Perspektifleri

Tarık Yerlikaya[1]

Kriptolojinin Temelleri ve Tarihsel Gelişimi

Kriptoloji, enformasyon çağının temelini oluşturan en kritik bilim dallarından biridir. Bu bölüm, kriptoloji bilimini tanımlayarak, temel terminolojiyi ve tarihsel bağlamı oluşturarak sonraki bölümler için bir zemin hazırlamaktadır. Bilginin korunması ihtiyacının antik çağlardan günümüze nasıl evrildiği ve bu evrimin modern kriptografik ilkeleri nasıl şekillendirdiği incelenecektir.

Kriptoloji, Kriptografi ve Kriptoanaliz: Tanımlar ve İlişkiler

Kriptoloji, en geniş tanımıyla, haberleşen iki veya daha fazla tarafın bilgi alışverişini güvenli bir ortamda yapmasını sağlayan, temelleri matematiksel olarak zor problemlere dayanan tekniklerin ve uygulamaların bütününü ifade eder (Acar, 2025). Bu bilim dalı, birbirini tamamlayan ve aynı zamanda birbiriyle sürekli bir rekabet içinde olan iki ana alt daldan oluşur: kriptografi ve kriptoanaliz.

Kriptografi, Yunanca "gizli" anlamına gelen *kryptos* ve "yazı" anlamına gelen *graphein* kelimelerinden türemiştir (Kaya, 2021). Temel amacı, bir bilgiyi veya veriyi, yetkisiz kişilerin okuyamayacağı veya anlayamayacağı bir forma dönüştürmektir (Kaya, 2021). Bu işlem, yani şifreleme, matematiksel fonksiyonlar ve algoritmalar kullanılarak gerçekleştirilir. Kriptanaliz ise, bu şifrelenmiş metinleri, şifreleme sırasında kullanılan anahtara sahip olmaksızın analiz ederek orijinal metni elde etme bilimidir (Avaroğlu, 2017). Kriptografi "kalkan yapma" sanatıysa, kriptoanaliz "kalkan kırma" sanatıdır. Bu iki alan arasındaki diyalektik ilişki, kriptolojinin tarihsel evrimindeki en temel itici güç olmuştur. Kriptograflar tarafından geliştirilen her yeni ve daha güçlü şifreleme algoritması, kriptoanalistler için yeni bir hedef teşkil etmiş; kırılan her şifre ise kriptografları daha da karmaşık ve güvenli sistemler tasarlamaya itmiştir (Kaya, 2021) .

[1] Dr. Öğretim Üyesi, Trakya Üniversitesi, ORCID: 0000-0002-9888-0151

Kriptografik Sistemlerin Temel Bileşenleri: Düz Metin, Şifreli Metin, Anahtar

Herhangi bir kriptografik sistemin işleyişini anlamak için beş temel bileşeni tanımak gerekir. Bu bileşenler, şifreleme sürecinin evrensel yapı taşlarıdır (Acar, 2025):

1. Düz Metin (Plaintext, P): Şifrelenmemiş, orijinal, okunabilir ve anlaşılır haldeki veridir.
2. Şifreleme Algoritması (Encryption, E): Düz metni anlaşılamaz bir forma dönüştüren matematiksel kurallar ve işlemler bütünüdür.
3. Şifreli Metin (Ciphertext, C): Düz metnin şifreleme algoritması uygulandıktan sonraki anlamsız ve okunamaz halidir.
4. Deşifreleme Algoritması (Decryption, D): Şifreli metni tekrar orijinal düz metne dönüştürme işlemidir.
5. Anahtar (Key, K): Şifreleme ve deşifreleme algoritmalarının davranışını kontrol eden kritik parametredir. Aynı algoritma, farklı anahtarlarla tamamen farklı sonuçlar üretir.

Modern kriptografinin en temel paradigmalarından biri, 19. yüzyılda Auguste Kerckhoffs tarafından ortaya konan ve daha sonra Claude Shannon tarafından yeniden formüle edilen Kerckhoffs Prensibi'dir. Bu prensibe göre, bir kriptosistemin güvenliği, şifreleme algoritmasının gizliliğine değil, yalnızca ve yalnızca anahtarın gizliliğine bağlı olmalıdır (Acar, 2025). Bu, AES gibi modern standartların neden herkes tarafından bilinip incelenebildiğini açıklar. Algoritmanın kendisi kamuya açıktır, bu sayede dünya çapındaki uzmanlar tarafından sürekli olarak analiz edilebilir ve zayıflıkları test edilebilir. Sistemin tüm güvenliği, sadece yetkili tarafların bildiği anahtarın gizli kalmasına dayanır. Bu yaklaşım, sistemlerin güvenilirliğinin matematiksel olarak kanıtlanabilmesini ve kamusal denetime açık olmasını sağlayarak "güvenlik için gizlilik" (security through obscurity) gibi zayıf bir yaklaşımdan kaçınılmasını sağlar.

Bilgi Güvenliğinin Temel Hedefleri: Gizlilik, Bütünlük, Kimlik Doğrulama ve İnkâr Edilemezlik

Modern kriptografi, sadece mesajları gizlemekten çok daha fazlasını hedefler. Günümüz dijital dünyasında güvenli bir etkileşim ortamı yaratmak için dört temel güvenlik ilkesini sağlamayı amaçlar (Avaroğlu, 2017):

- **Gizlilik (Confidentiality):** Bilginin içeriğinin yetkisiz kişiler tarafından anlaşılmasını engeller. Bu, şifrelemenin en temel ve en bilinen amacıdır. Bir e-postanın içeriğinin sadece alıcı tarafından okunabilmesi gizlilik ilkesine bir örnektir (Avaroğlu, 2017).

- **Bütünlük (Integrity):** Verinin, iletim sırasında veya depolanırken yetkisiz kişilerce değiştirilmediğini garanti eder. Alınan bir dosyanın, gönderilen dosyanın birebir aynısı olduğundan emin olmak bütünlük sayesinde mümkündür (AWS, 2025).
- **Kimlik Doğrulama (Authentication):** İletişimdeki tarafların kimliklerinin doğru olduğunu teyit eder. Bir web sitesine bağlandığınızda, gerçekten doğru sunucuyla iletişim kurduğunuzdan emin olmanızı sağlar(Avaroğlu, 2017) .
- **İnkâr Edilemezlik (Non-repudiation):** Bir işlemi gerçekleştiren tarafın, daha sonra bu işlemi yaptığını inkâr etmesini önler. Dijital bir sözleşmeyi imzalayan bir kişinin, imzasını sonradan reddedememesi bu ilkeye dayanır (Avaroğlu, 2017).

Bu dört hedef, dijital dünyadaki güvenli etkileşimlerin temel direkleridir. Örneğin, bir online banka transferi hem gizli (transfer detayı korunmalı) hem bütünlüğe sahip (miktar değiştirilememeli) olmalı, hem banka ve müşteri kimlikleri doğrulanmalı, hem de işlem inkâr edilememelidir. Farklı kriptografik araçlar bu hedefleri farklı şekillerde karşılar. Simetrik şifreleme algoritmaları öncelikli olarak gizlilik sağlarken, asimetrik şifreleme algoritmaları ve onlara dayalı dijital imzalar, bu dört hedefin tamamını, özellikle de kimlik doğrulama ve inkâr edilemezliği etkin bir şekilde sağlar (Avaroğlu, 2017).

Kriptografinin Tarihsel Serüveni: Antik Çağlardan Modern Döneme

Bilgiyi gizleme ihtiyacı insanlık tarihi kadar eskidir. Kriptografinin kökleri, M.Ö. 2000'lere, antik Mısır, Mezopotamya ve Hindistan'a kadar uzanmaktadır (Bingöl, 2022). İlk örneklerden biri, M.Ö. 1900'lerde Mısır'da bulunan ve standart hiyerogliflerin bilinçli olarak değiştirildiği kitabelerdir (Avaroğlu, 2017). En bilinen erken dönem şifreleme yöntemlerinden biri, M.Ö. 60-50 yıllarında Roma İmparatoru Jül Sezar tarafından kullanılan ve her harfin alfabede kendisinden üç harf sonraki harfle değiştirilmesine dayanan Sezar şifresidir (Kaya, 2021). Bu basit "yerine koyma" (substitution) şifresi, askeri ve idari haberleşmede uzun süre kullanılmıştır.

Orta Çağ'da, özellikle Arap-İslam medeniyetinde kriptoloji önemli bir gelişme göstermiştir. 9. yüzyılda yaşayan büyük alim El-Kindi, yazdığı "Şifreli Mesajların Çözümü Üzerine Risale" adlı eseriyle, şifreli metinlerdeki harf frekanslarını analiz ederek şifre kırma yöntemini, yani kriptoanalizin temellerini atmıştır (Bingöl, 2022). Bu, kriptografi tarihinde bir dönüm noktasıdır. Rönesans İtalya'sında gelişen diplomasi, şehir devletleri arasındaki gizli haberleşme ihtiyacını artırmış ve daha karmaşık polialfabetik şifrelerin (örneğin Vigenère şifresi) geliştirilmesine yol açmıştır (Kaya, 2021).

Osmanlı İmparatorluğu'nda ise diplomatik kriptografi uygulamaları 18. yüzyılın sonlarında, III. Selim döneminde ortaya çıkmıştır. Başlangıçta basit sayısal kodlamalar kullanılırken, 19. yüzyılda telgrafın icadıyla birlikte daha karmaşık, kelime tabanlı ve numerikleştirilmiş "Kod Defterleri" geliştirilmiştir (Bingöl, 2022). Bu, Osmanlı bürokrasisinin modernleşme çabalarının bir yansıması olarak görülebilir. 20.yüzyıl, kriptografinin mekanikleştiği ve endüstrileştiği bir dönem oldu. II. Dünya Savaşı sırasında Almanların kullandığı Enigma makinesi, karmaşık rotor mekanizmasıyla o dönem için kırılması neredeyse imkansız görünen şifreler üretiyordu. Ancak İngiliz matematikçi Alan Turing ve Bletchley Park'taki ekibinin, Enigma şifrelerini kırmak için geliştirdiği "Bombe" adlı elektromekanik cihaz, savaşın seyrini değiştiren en önemli olaylardan biri olarak tarihe geçmiştir (Kaya, 2021).

Kriptografinin evrimi, sadece teknolojik bir ilerleme değil, aynı zamanda güvenin merkezsizleşmesi sürecidir. Antik ve orta çağlarda kriptografi, mutlak güce (imparator, kral) hizmet eden kapalı bir sistemdi ve güven, bu merkezi otoriteye duyulurdu (Avaroğlu, 2017). Enigma gibi mekanik cihazlar, güvenliği insanlardan makinelere kaydırsa da, bu makinelerin tasarımı hala devlet sırrıydı ve ulus-devletlerin kontrolündeydi (Kaya, 2021). 1970'lerde yaşanan iki devrimci gelişme bu yapıyı temelden sarstı. İlk olarak, IBM'in geliştirdiği ve 1976'da ABD hükümeti tarafından standartlaştırılan DES (Veri Şifreleme Standardı), kriptografiyi ticari alana taşıdı (Avaroğlu, 2017). İkinci ve daha önemli gelişme ise, aynı yıl Whitfield Diffie ve Martin Hellman tarafından "açık anahtarlı şifreleme" konseptinin ortaya atılmasıydı (Avaroğlu, 2017). Bu paradigma kayması, iki tarafın birbirlerine veya herhangi bir merkezi otoriteye güvenmek zorunda kalmadan, sadece matematiğin zor problemlerine (örneğin büyük sayıları çarpanlarına ayırma zorluğu) güvenerek güvenli bir iletişim kanalı kurabilmesini sağladı. Bu devrim, güvenin artık bir kurumdan veya gizli bir tasarımdan değil, herkes tarafından denetlenebilen, açık ve matematiksel olarak kanıtlanabilir algoritmalardan kaynaklandığı yeni bir çağ başlattı. Bu merkezsizleşmiş güven modeli, günümüzün e-ticaret, online bankacılık, kripto paralar ve sayısız diğer dijital uygulamasının temelini atmıştır (Kaya, 2021).

Tablo 1, kriptolojinin bu uzun ve zengin tarihindeki bazı önemli kilometre taşlarını özetlemektedir.

Tablo 1: Kriptolojinin Tarihsel Gelişimindeki Önemli Kilometre Taşları

Tarih/Dönem	Olay/Gelişme	Önemi/Etkisi
M.Ö. 1900	Mısır'da Hiyeroglif Şifreleme	Bilinen en eski kriptografik uygulamalardan biri(Avaroğlu, 2017) .
M.Ö. 50	Sezar Şifresi	Basit yerine koyma şifresinin askeri alanda sistematik kullanımı (Kaya, 2021).
9. Yüzyıl	El-Kindi ve Frekans Analizi	Kriptoanaliz biliminin temelleri atıldı (Bingöl, 2022).
18. Yüzyıl Sonu	Osmanlı Diplomatik Kriptografisi	Modern devlet bürokrasisinde kriptografinin kullanımı başladı (Bingöl, 2022).
1945	Enigma Makinesinin Kırılması	Kriptoanalizin savaşın seyrini değiştirebilecek stratejik bir güç olduğu kanıtlandı (Avaroğlu, 2017).
1976	DES Standardının Kabulü	Kriptografinin ticari ve sivil kullanıma açılmasında ilk büyük adım (Avaroğlu, 2017).
1976	Diffie-Hellman Anahtar Değişimi	Asimetrik (açık anahtarlı) kriptografi devrimini başlattı (Avaroğlu, 2017).
1978	RSA Algoritmasının Geliştirilmesi	İlk pratik ve yaygın olarak kullanılan açık anahtarlı şifreleme sistemi (Beşkirli, Özdemir, & Beşkirli, 2019).
1985	ECC'nin (Eliptik Eğri Kriptografisi) Önerilmesi	Daha verimli bir asimetrik şifreleme alternatifi sunuldu(Avaroğlu, 2017) .
2001	AES Standardının Kabulü	Modern, güvenli ve verimli simetrik şifreleme standardı belirlendi (Avaroğlu, 2017).
2022-2024	PQC Standardizasyon Süreci	Kuantum bilgisayar tehdidine karşı yeni nesil algoritmaların seçimi(Newhouse & Regenscheid, 2025) .

Simetrik Şifreleme Algoritmaları

Simetrik Şifrelemenin Çalışma Prensibi: Paylaşılan Gizli Anahtar

Simetrik anahtarlı kriptografide, hem gönderici hem de alıcı, veriyi şifrelemek ve deşifre etmek için aynı gizli anahtarı kullanır (Dijital Güvenlik Platformu – Aksigorta, t.y.). Süreç oldukça basittir: Gönderici, düz metni paylaşılan gizli anahtar ile şifreleyerek şifreli metni oluşturur ve alıcıya gönderir. Alıcı, aynı gizli

anahtarı kullanarak şifreli metni deşifre eder ve orijinal düz metni elde eder. Bu yöntemlerin en büyük avantajı, asimetrik yöntemlere kıyasla çok daha hızlı olmalarıdır. Bu hız, büyük boyutlu verilerin (dosyalar, video akışları, veritabanları vb.) şifrelenmesi gerektiğinde onları ideal bir çözüm haline getirir (Precisely, t.y.).

Ancak simetrik şifrelemenin doğasında kritik bir zorluk bulunur: anahtar yönetimi. Tarafların, şifreli iletişime başlamadan önce, kullanacakları gizli anahtarı güvenli bir kanal üzerinden birbirlerine iletmeleri gerekir. Eğer bu anahtar iletim sırasında üçüncü bir tarafın eline geçerse, tüm şifreli iletişim tehlikeye girer. Bu "anahtar dağıtım problemi", asimetrik şifrelemenin ve günümüzdeki modern güvenlik protokollerinde kullanılan hibrit sistemlerin geliştirilmesindeki ana motivasyon kaynaklarından biri olmuştur.

Veri Şifreleme Standardı (DES): Bir Zamanların Devi

Feistel Ağ Yapısı ve Çalışma Mekanizması

Veri Şifreleme Standardı (DES), 1970'lerde IBM tarafından "Lucifer" adıyla geliştirilen bir algoritmanın Ulusal Güvenlik Ajansı (NSA) tarafından modifiye edilmesiyle ortaya çıkmış ve 1976 yılında ABD federal standardı olarak kabul edilmiştir (Avaroğlu, 2017). DES, bir blok şifreleme algoritmasıdır; yani veriyi sabit boyutlu bloklara ayırarak şifreler. DES için bu blok boyutu 64 bittir ve algoritma, 64 bitlik bir anahtar almasına rağmen, bunun sadece 56 bitini efektif olarak kullanır (8 bit parite kontrolü için ayrılmıştır) (Şahin, 2015) .

DES'in temel mimarisi, Feistel Ağı olarak bilinen zarif bir yapıya dayanır. Bu yapı, şifreleme sürecini yönetilebilir ve tersine çevrilebilir turlara böler. DES'in çalışma mekanizması şu adımları içerir (Şeker, t.y.):

1. 64 bitlik düz metin bloğu, bir başlangıç permütasyonundan (Initial Permutation - IP) geçirilerek karıştırılır.
2. Karıştırılmış blok, 32 bitlik iki eşit parçaya ayrılır: Sol Parça (L0) ve Sağ Parça (R0).
3. Bu iki parça, 16 tur boyunca bir dizi işleme tabi tutulur. Her bir i turunda (1'den 16'ya kadar):
    - Sol parça değişmez: $L_i = R_{i-1}$
    - Sağ parça, önceki sağ parça ile bir fonksiyonun sonucunun XOR'lanmasıyla elde edilir: $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
4. f fonksiyonu, DES'in kalbidir ve karıştırma işlemini gerçekleştirir. Bu fonksiyon, 32 bitlik R_{i-1} parçasını önce bir **genişletme permütasyonu** (Expansion Permutation) ile 48 bite genişletir. Daha sonra bu 48 bitlik

veri, o tura özel 48 bitlik tur anahtarı (Ki) ile XOR'lanır. Sonuç, **S-Box** (Substitution Box) adı verilen sekiz farklı arama tablosuna gönderilir. Her S-Box, 6 bitlik girdiyi 4 bitlik bir çıktıya dönüştürerek doğrusal olmayan bir ikame işlemi gerçekleştirir. Son olarak, S-Box'lardan çıkan 32 bitlik veri, bir **P-Box** (Permutation Box) ile tekrar karıştırılır (Şeker, t.y.).

5. 16 tur tamamlandıktan sonra, sol ve sağ parçalar birleştirilir ve başlangıç permütasyonunun tersi olan bir son permütasyondan (Final Permutation) geçirilerek 64 bitlik şifreli metin elde edilir.

Feistel yapısının en önemli avantajı, şifreleme ve deşifreleme algoritmalarının neredeyse tamamen aynı olmasıdır. Deşifreleme işlemi için, şifreli metne aynı algoritma uygulanır, ancak 16 tur boyunca kullanılan tur anahtarları (K1,K2 ,...,K16) ters sırada (K16,K15,...,K1) kullanılır. Bu simetri, DES'in donanım ve yazılım implementasyonlarını büyük ölçüde basitleştirmiş ve 1970'ler ve 80'lerde hızla yaygınlaşmasında kritik bir rol oynamıştır.

DES'in Zayıflıkları ve 3DES Çözümü

DES, tasarlandığı dönem için oldukça güçlü bir algoritma olmasına rağmen, zamanla en büyük zayıflığı ortaya çıkmıştır: 56 bitlik anahtar boyutu. Bilgisayarların işlem gücünün Moore Yasası uyarınca katlanarak artmasıyla, 56 bitlik anahtar uzayını (256 olası anahtar) kaba kuvvet (brute-force) saldırısıyla, yani tüm olası anahtarları deneyerek taramak, pratik olarak mümkün hale gelmiştir (Şahin, 2015). 1990'ların sonunda, özel donanımlar kullanılarak DES şifresinin birkaç gün içinde kırılabildiği gösterilmiştir.

Bu kritik güvenlik açığını kapatmak ve mevcut DES tabanlı altyapıyı kullanmaya devam etmek için bir ara çözüm olarak **3DES (Triple DES)** geliştirilmiştir (Şahin, 2015). 3DES, adından da anlaşılacağı gibi, DES algoritmasının üç kez ardışık olarak uygulanmasıyla çalışır. Genellikle iki veya üç farklı anahtar kullanılır:

- **İki Anahtarlı 3DES:** Şifreleme (K1 ile), Deşifreleme (K2 ile), Şifreleme (K1 ile). Bu, etkin bir şekilde 112 bitlik bir anahtar güvenliği sağlar.
- **Üç Anahtarlı 3DES:** Şifreleme (K1 ile), Deşifreleme (K2 ile), Şifreleme (K3 ile). Bu da 168 bitlik bir anahtar güvenliği sunar.

3DES, DES'e göre çok daha güvenli bir alternatif sunarak ömrünü bir süre daha uzatmıştır. Ancak bu güvenlik artışının bir bedeli vardı: 3DES, standart DES'e göre yaklaşık üç kat daha yavaştır (Şahin, 2015). Bu yavaşlık ve DES'in 64 bitlik blok boyutunun modern uygulamalar için artık verimsiz kalması,

kriptografi dünyasını tamamen yeni, daha hızlı ve daha güvenli bir standarda yöneltmiştir. Bu ihtiyaç, AES'in doğuşuna zemin hazırlamıştır.

Gelişmiş Şifreleme Standardı (AES): Günümüzün Altın Standardı

Kriptografik standartların evrimi, "yeterince iyi" güvenlikten "ölçeklenebilir ve verimli" güvenliğe doğru bir kaymayı yansıtmaktadır. DES, tasarlandığı dönem için yeterliydi. 3DES, acil bir güvenlik açığını performanstan ödün vererek kapattı. Ancak Gelişmiş Şifreleme Standardı (AES), modern kriptografinin sadece kırılmaz olması gerekmediğini, aynı zamanda hızlı, verimli ve esnek olması gerektiğini de ortaya koyan bir devrimdir.

NIST Standardizasyon Süreci ve Rijndael'in Seçimi

1997 yılında, ABD Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), artık güvensiz kabul edilen DES'in yerini alacak yeni bir standart belirlemek amacıyla küresel ve halka açık bir yarışma başlattı (Avaroğlu, 2017). Bu süreç, DES'in daha kapalı geliştirme sürecinin aksine, şeffaflık, titiz kamusal analiz ve uluslararası katılım ilkeleri üzerine kuruluydu.[8] Yarışmaya dünyanın dört bir yanından 15 aday algoritma katıldı. Adaylar, sadece güvenliğe (bilinen tüm kriptoanaliz saldırılarına karşı direnç) göre değil, aynı zamanda çeşitli donanım ve yazılım platformlarındaki performans, verimlilik (düşük bellek ve işlemci kullanımı), implementasyon kolaylığı ve esneklik (farklı anahtar ve blok boyutlarını destekleme yeteneği) gibi çok yönlü kriterlere göre değerlendirildi.[8]

Yıllar süren yoğun analiz ve birçok uluslararası konferansta yapılan tartışmaların ardından, Ekim 2000'de NIST, Belçikalı iki kriptograf, Joan Daemen ve Vincent Rijmen tarafından geliştirilen **Rijndael** (okunuşu: "Rayn-dal") algoritmasını kazanan olarak ilan etti (National Institute of Standards and Technology [NIST], t.y.). Rijndael, güvenlik, performans, verimlilik ve esneklik kombinasyonunda rakiplerine göre en dengeli ve üstün profili sunmuştu. Özellikle düşük bellek gereksinimleri, onu kısıtlı kaynaklara sahip cihazlar için bile uygun kılıyordu (NIST, t.y.). Rijndael, Kasım 2001'de **FIPS 197** standardı olarak resmi olarak yayınlandı ve **AES** adını aldı (National Institute of Standards and Technology [NIST], 2001). Bu şeffaf ve rekabetçi süreç, AES'in küresel çapta hızla güvenilirlik kazanmasını ve endüstri standardı olarak benimsenmesini sağlamıştır.

AES'in Yapısı: İkame-Permütasyon Ağı (SPN)

AES, DES'in Feistel ağ yapısından farklı olarak, İkame-Permütasyon Ağı (Substitution-Permutation Network - SPN) adı verilen daha modern bir mimari kullanır (Kula, 2009). AES, sabit 128 bitlik veri blokları üzerinde çalışır ve üç farklı anahtar boyutunu destekler: 128, 192 ve 256 bit (NIST, 2001).

Şifreleme süreci boyunca veri, **State** (Durum) adı verilen 4x4'lük bir bayt matrisi üzerinde işlenir (NIST, 2001). SPN yapısının temel avantajı, her turda tüm veri bloğunun tamamının paralel olarak işlenmesidir. Bu, Feistel ağının her turda bloğun sadece yarısını değiştirmesine kıyasla, girdideki küçük bir değişikliğin çıktıya çok daha hızlı yayılmasını (daha iyi **difüzyon**) sağlar. Bu verimlilik, AES'in DES'e göre daha az turla daha yüksek bir güvenlik seviyesine ulaşmasına olanak tanır.

Temel Operasyonlar: SubBytes, ShiftRows, MixColumns, AddRoundKey

AES şifreleme süreci, kullanılan anahtar boyutuna göre değişen sayıda turdan oluşur. Bu turların sayısı, algoritmanın farklı versiyonlarını tanımlar.

Tablo 2: AES Anahtar Boyutları ve Tur Sayıları (NIST, 2001)

AES Versiyonu	Anahtar Boyutu (bit)	Blok Boyutu (bit)	Tur Sayısı (Nr)
AES-128	128	128	10
AES-192	192	128	12
AES-256	256	128	14

Her tur (son tur hariç) dört temel dönüşümden oluşur. Bu dönüşümler, Claude Shannon'ın modern kriptografinin temelini oluşturan "karışıklık" (confusion) ve "yayılma" (diffusion) ilkelerini uygulamak üzere tasarlanmıştır (NIST, 2001):

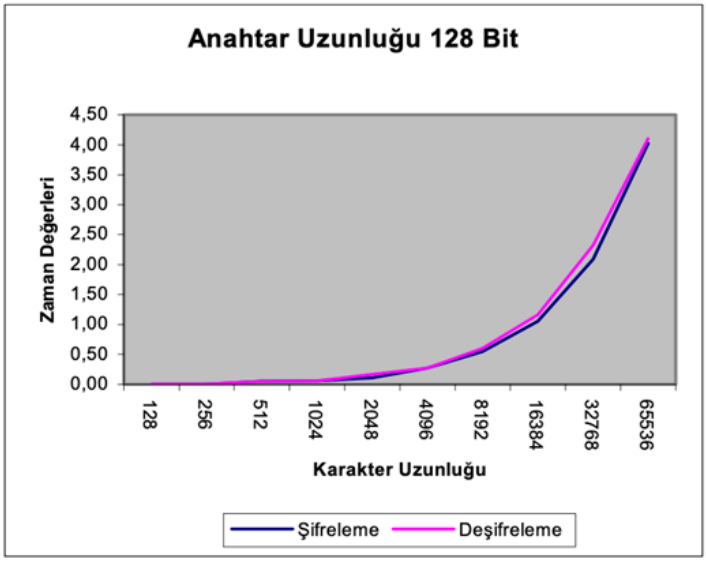
1. **SubBytes (Bayt Değiştirme):** Bu, doğrusal olmayan bir ikame adımıdır ve **karışıklık** sağlar. State matrisindeki her bayt, **S-Box** adı verilen önceden tanımlanmış bir arama tablosu kullanılarak bağımsız olarak başka bir bayt ile değiştirilir. Bu işlem, şifreli metin ile anahtar arasındaki ilişkiyi oldukça karmaşık hale getirir (NIST, 2001) .
2. **ShiftRows (Satırları Kaydırma):** Bu, bir permütasyon adımıdır ve **yayılma** sağlar. State matrisinin her satırı (ilk satır hariç) farklı miktarlarda döngüsel olarak sola kaydırılır. Bu, bir sütundaki baytların farklı sütunlara dağılmasını sağlayarak sütunlar arasındaki bağımlılığı kırar(NIST, 2001) .
3. **MixColumns (Sütunları Karıştırma):** Bu adım, **yayılma** özelliğini daha da güçlendirir. State matrisinin her sütunu, sonlu cisim (GF(28)) aritmetiği kullanılarak önceden tanımlanmış sabit bir matrisle çarpılır. Bu işlem, bir bayttaki tek bir bitlik değişikliğin bile o sütundaki tüm baytları etkilemesini sağlar (Güney Bilişim, t.y.).

4. **AddRoundKey (Tur Anahtarını Ekleme):** Her tur için anahtar çizelgesinden türetilen 128 bitlik tur anahtarı, basit bir bit-bit XOR işlemiyle State matrisine eklenir. Bu, anahtarı şifreleme sürecine dahil eden tek adımdır (Güney Bilişim, t.y.).

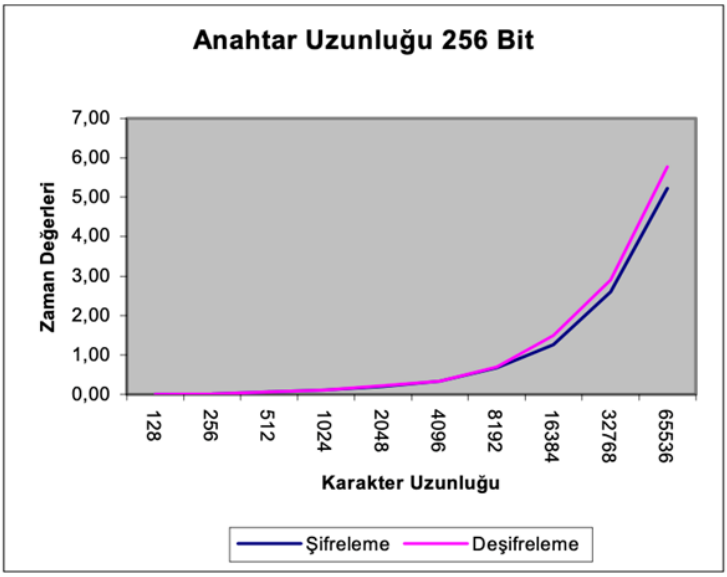
Deşifreleme işlemi, bu adımların tersi (InvSubBytes, InvShiftRows, InvMixColumns) ve ters sırada uygulanan tur anahtarları ile gerçekleştirilir. Bu dört operasyonun birleşimi, AES'i diferansiyel ve lineer kriptoanaliz gibi bilinen güçlü saldırı tekniklerine karşı son derece dirençli kılar ve onu günümüzün en güvenilir ve en yaygın simetrik şifreleme standardı yapar.

AES Performans Karşılaştırması

AES algoritmasının anahtar uzunluğu ve karakter uzunluğuna göre değişen performans grafikleri Şekil 1, Şekil 2 ve Şekil 3'te gösterilmiştir.



Şekil 1: AES 128 bit performans grafiği



Şekil 3: AES 256bit performs grafiği

Asimetrik (Açık Anahtarlı) Şifreleme Algoritmaları

Asimetrik kriptografi, 1970'lerdeki icadıyla dijital iletişimde bir devrim yaratmıştır. Bu bölümde, şifreleme ve deşifreleme için iki farklı anahtar (biri genel, diğeri özel) kullanan bu sistemlerin temelleri incelenecektir. Bu yaklaşımın, simetrik şifrelemenin doğasında var olan anahtar dağıtım sorununu nasıl çözdüğü ve dijital imzalar gibi tamamen yeni güvenlik olanaklarını nasıl ortaya çıkardığı ele alınacaktır. Bu alandaki iki temel yapı taşı olan RSA ve Eliptik Eğri Kriptografisi'nin (ECC) dayandığı soyut matematiksel problemlerin, nasıl somut ve dünya değiştiren teknolojilere dönüştüğü detaylandırılacaktır.

Asimetrik Şifrelemenin Temelleri: Açık ve Özel Anahtar Kavramı

1976 yılında Whitfield Diffie ve Martin Hellman tarafından kamuoyuna duyurulan açık anahtarlı şifreleme konsepti, kriptografi tarihinde bir dönüm noktasıdır (Avaroğlu, 2017). Bu sistemin temelinde, matematiksel olarak birbirine bağlı ancak birbirinden farklı bir **anahtar çifti** yatar: **açık anahtar (public key)** ve **özel anahtar (private key)** (AWS, 2025).

- **Açık Anahtar:** Adından da anlaşılacağı gibi, bu anahtar gizli değildir ve herkesle güvenli bir şekilde paylaşılabilir. Genellikle bir sunucuda veya herkese açık bir dizinde saklanır. Veriyi şifrelemek veya bir dijital imzayı doğrulamak için kullanılır.

- **Özel Anahtar:** Bu anahtar, sahibi tarafından mutlak bir gizlilikle korunmalıdır. Açık anahtarla şifrelenmiş veriyi deşifre etmek veya dijital imza oluşturmak için kullanılır.

Bu sistemin çalışma prensibi şöyledir: Bir kişi (örneğin Ali), başka bir kişiye (örneğin Veli) güvenli bir mesaj göndermek istediğinde, Veli'nin herkese açık olan açık anahtarını kullanarak mesajı şifreler. Şifrelenmiş bu mesaj, artık sadece ve sadece Veli'nin sahip olduğu özel anahtar ile çözülebilir (Kocaturk, t.y.). Bu yapı, simetrik şifrelemenin en büyük handikabı olan "anahtar dağıtım problemini" zarif bir şekilde ortadan kaldırır. Tarafların, iletişime başlamadan önce güvenli bir kanal üzerinden gizli bir anahtar paylaşmalarına gerek kalmaz. Ali, güvenilmeyen bir ağ (internet gibi) üzerinden Veli'nin açık anahtarını alabilir ve mesajını güvenle şifreleyebilir. Bu devrim niteliğindeki özellik, internet üzerinde güvenli iletişimin, e-ticaretin, online bankacılığın ve daha sayısız uygulamanın temelini atmıştır.

RSA Algoritması: Çarpanlara Ayırmanın Gücü

Asimetrik kriptografi konseptinin ilk pratik ve başarılı uygulaması, 1978 yılında Massachusetts Teknoloji Enstitüsü'nden (MIT) Ron **R**ivest, Adi **S**hamir ve Leonard **A**dleman tarafından geliştirilen ve isimlerinin baş harflerini taşıyan RSA algoritmasıdır(Beşkirli, Özdemir, & Beşkirli, 2019).

Matematiksel Temelleri: Büyük Sayıların Çarpanlarına Ayrılmasının Zorluğu

RSA'nın güvenliği, sayı teorisindeki temel bir probleme dayanır: **büyük tam sayıların asal çarpanlarına ayrılmasının hesaplamasal zorluğu** (Beşkirli, Özdemir, & Beşkirli, 2019). Bu problem şu şekilde özetlenebilir: İki çok büyük asal sayıyı (örneğin yüzlerce basamaklı) birbiriyle çarpmak, modern bir bilgisayar için saniyeler içinde yapılabilecek kolay bir işlemdir. Ancak, bu çarpım sonucu elde edilen çok büyük sayı verildiğinde, orijinal iki asal sayıyı bulmak (yani sayıyı çarpanlarına ayırmak), bilinen en iyi algoritmalarla bile klasik bir bilgisayar için milyarlarca yıl sürebilecek, pratik olarak imkansız bir iştir (Kocaturk, t.y.) .

Bu asimetri, RSA'nın temelindeki **tek yönlü fonksiyon (one-way function)** yapısını oluşturur. Çarpma işlemi kolayca yapılabilirken (tek yön), çarpanlara ayırma işlemi (ters yön) son derece zordur. Özel anahtar, bu tek yönlü fonksiyonu kolayca tersine çevirmeyi sağlayan "gizli kapı" (trapdoor) bilgisini, yani orijinal asal sayıları içerir.

Anahtar Üretimi, Şifreleme ve Deşifreleme Süreçleri

RSA algoritmasının işleyişi üç ana adımdan oluşur (Kocaturk, t.y.):

1. **Anahtar Üretimi:**

   - Birbirinden farklı, çok büyük iki asal sayı, p ve q, rastgele seçilir.
   - Modül $n=p\times q$ olarak hesaplanır. Bu n değeri, hem açık hem de özel anahtarın bir parçasıdır ve anahtar boyutunu (örneğin 2048-bit) belirler.
   - Euler'in totient fonksiyonu, $\phi(n)=(p-1)\times(q-1)$, hesaplanır. Bu değer, n'den küçük ve n ile aralarında asal olan pozitif tam sayıların adedini verir ve özel anahtarın hesaplanmasında kritik rol oynar.
   - $1<e<\phi(n)$ koşulunu sağlayan ve $\phi(n)$ ile aralarında asal olan (yani $ebob(e,\phi(n))=1$) bir e tam sayısı seçilir. e (encryption exponent), **açık üs** olarak adlandırılır. Genellikle 65537 gibi sabit bir değer seçilir.
   - Genişletilmiş Öklid algoritması kullanılarak $d\times e\equiv 1 (mod \phi(n))$ denklemini sağlayan d tam sayısı bulunur. d (decryption exponent), **özel üs** olarak adlandırılır.
   - Sonuç olarak, **Açık Anahtar** (n,e) ikilisi ve **Özel Anahtar** (n,d) ikilisi oluşturulur. p, q ve $\phi(n)$ değerleri, d'yi hesaplamak için kullanıldıktan sonra güvenli bir şekilde imha edilmeli veya gizli tutulmalıdır.

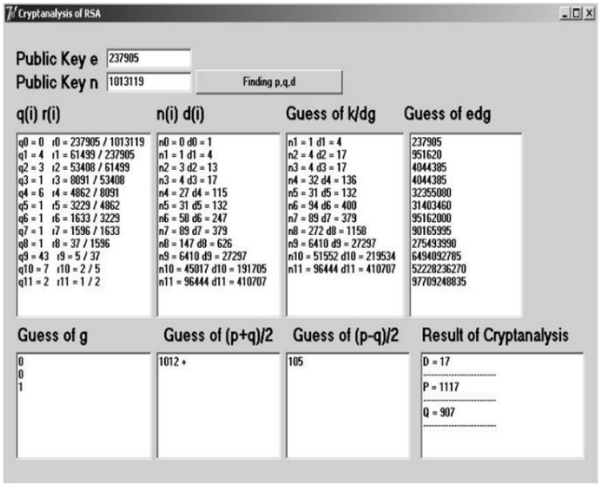
2. **Şifreleme:**

   - Düz metin M (bir sayı olarak temsil edilir), alıcının açık anahtarı (n,e) kullanılarak aşağıdaki modüler üs alma işlemiyle şifreli metin C'ye dönüştürülür:
     $C\equiv Me(modn)$ Bu işlem, mesajı gönderen herhangi bir kişi tarafından yapılabilir.

3. **Deşifreleme:**

   - Alıcı, elindeki şifreli metin C'yi kendi özel anahtarı (n,d) ile deşifre eder:
     $M\equiv Cd(modn)$ Euler teoreminin bir sonucu olarak, bu işlem orijinal M metnini verir.
   - Güvenlik, n ve e biliniyor olsa bile, p ve q'yu (yani n'nin çarpanlarını) bilmeden $\phi(n)$'yi ve dolayısıyla d'yi hesaplamanın imkansız olmasına dayanır.

Kriptanaliz işlemi için geliştirilen bir uygulama Şekil 5'te gösterilmiştir.



Şekil 5: RSA Uygulaması

Eliptik Eğri Kriptografisi (ECC): Verimlilik ve Güvenliğin Buluşması

RSA'nın yayınlanmasından yaklaşık on yıl sonra, 1985'te Neal Koblitz ve Victor Miller adlı matematikçiler, asimetrik kriptografi için tamamen farklı bir matematiksel yapı önerdiler: eliptik eğriler (Avaroğlu, 2017). Eliptik Eğri Kriptografisi (ECC), RSA ile aynı güvenlik hedeflerini (anahtar değişimi, dijital imza) çok daha verimli bir şekilde gerçekleştirebilmesiyle öne çıkmaktadır.

Matematiksel Temelleri: Eliptik Eğri Ayrık Logaritma Problemi (ECDLP)

ECC'nin güvenliği, **Eliptik Eğri Ayrık Logaritma Problemi (Elliptic Curve Discrete Logarithm Problem - ECDLP)** olarak bilinen farklı bir zor matematiksel probleme dayanır (Elektrik Mühendisleri Odası [EMO], t.y.). Bu problem, RSA'nın dayandığı çarpanlara ayırma probleminden kavramsal olarak farklıdır.

Bir eliptik eğri üzerinde bir taban noktası P ve bu noktanın kendisiyle k defa toplanmasıyla elde edilen başka bir nokta olan Q verildiğinde ($Q=k \cdot P$ olacak şekilde), Q ve P noktaları bilinirken k tamsayısını bulma işlemine ECDLP denir (EITCA Academy, t.y.).

Bu problemin klasik ayrık logaritma probleminden (DLP) ve çarpanlara ayırma probleminden daha "zor" olduğuna inanılmaktadır. Bu, aynı güvenlik seviyesini sağlamak için, ECDLP'yi çözmenin bilinen en iyi algoritmalarının, RSA'yı kırmak için kullanılan algoritmalardan daha fazla hesaplama gücü

gerektirdiği anlamına gelir. Bu "zorluk farkı", ECC'nin neden çok daha küçük anahtar boyutlarıyla RSA ile eşdeğer güvenlik sağlayabildiğinin temel nedenidir (EITCA Academy, t.y.).

ECC'nin İşleyişi: Nokta Toplama ve Skaler Çarpım

ECC, reel sayılar üzerinde değil, sonlu bir cisim (finite field) üzerinde tanımlanmış ve genellikle $y2=x3+ax+b$ formundaki denklemlerle ifade edilen eliptik eğriler üzerindeki noktalarla çalışır (Yıldızhan, t.y.). Kriptografik işlemler için eğrinin "tekil olmaması" (yani diskriminantının sıfırdan farklı olması gerekir, bu da eğrinin düzgün bir grup yapısına sahip olmasını sağlar (EITCA Academy, t.y.).

Bu eğri üzerindeki noktalar için geometrik temelli özel bir "nokta toplama" kuralı tanımlanmıştır. İki farklı nokta P ve Q'yu toplamak, bu iki noktadan geçen doğrunun eğriyi kestiği üçüncü noktanın x eksenine göre simetriğini almakla elde edilir (Yücelen, Baykal, & Coşkun, 2017). Bir noktayı kendisiyle toplamak ("nokta ikiye katlama") ise o noktadaki teğet doğrusu kullanılarak benzer bir şekilde yapılır.

ECC'nin tek yönlü fonksiyonu bu operasyonlara dayanır:

- **Skaler Çarpım (Kolay Yön):** Bir taban noktası P'yi, gizli bir tam sayı olan k ile çarpmak (yani P'yi kendisiyle k defa toplamak) ve sonuç noktası olan $Q=k\cdot P$'yi bulmak hesaplamasal olarak verimlidir.[28]
- **Ayrık Logaritma (Zor Yön):** Q ve P noktaları bilinirken, skaler çarpan olan k'yı bulmak (ECDLP'yi çözmek) hesaplamasal olarak imkansızdır.

Anahtar üretimi bu prensibe göre yapılır: Sistemin tüm kullanıcıları aynı eliptik eğriyi ve üzerinde bir taban noktası P'yi paylaşır. Her kullanıcı, **özel anahtar** olarak rastgele bir k tamsayısı seçer. Daha sonra skaler çarpım yaparak $Q=k\cdot P$ noktasını hesaplar. Bu Q noktası, kullanıcının **açık anahtarı** olur.

Asimetrik kriptografinin gelişimi, soyut matematiksel problemlerin nasıl somut, dünya değiştiren teknolojilere dönüşebileceğinin en çarpıcı örneklerinden birini sunar. Sayı teorisi ve cebirsel geometri gibi alanlarda on yıllardır çalışılan ve pratik uygulaması olmayan teorik problemler (çarpanlara ayırma, ECDLP), bilgisayar ağlarının yükselişiyle ortaya çıkan güvenli anahtar değişimi ve kimlik doğrulama gibi pratik ihtiyaçları karşılamak için birer temel taşı haline gelmiştir. Bu sentez, tamamen teorik bir matematik dalını, günümüz dijital ekonomisinin ve güvenli iletişim altyapısının vazgeçilmez bir parçasına dönüştürmüş, temel bilimin öngörülemeyen pratik sonuçlar doğurabileceğinin güçlü bir kanıtı olmuştur.

Algoritmaların Karşılaştırmalı Analizi ve Modern Uygulamaları

Performans ve Güvenlik Karşılaştırması: RSA ve ECC

Asimetrik kriptografi alanında RSA ve ECC, aynı güvenlik hedeflerine hizmet etseler de, bunu farklı verimlilik seviyelerinde yaparlar. Aralarındaki temel fark, aynı güvenlik seviyesini sağlamak için gereken anahtar boyutlarından kaynaklanır.

ECC'nin en belirgin ve en önemli avantajı, RSA'ya kıyasla çok daha küçük anahtar boyutlarıyla eşdeğer veya daha yüksek güvenlik seviyeleri sunabilmesidir (YazılımTürk, t.y.). Bu durumun temel nedeni, ECC'nin dayandığı Eliptik Eğri Ayrık Logaritma Problemi'nin (ECDLP), RSA'nın dayandığı çarpanlara ayırma problemine göre bilinen algoritmalarla çözülmesinin daha zor olmasıdır. Bu zorluk farkı, anahtar boyutlarına doğrudan yansır. Aşağıdaki tablo, NIST tarafından önerilen ve endüstride kabul gören eşdeğer anahtar boyutlarını göstermektedir.

Tablo 3: RSA ve ECC Anahtar Boyutlarının Güvenlik Seviyelerine Göre Karşılaştırılması

Simetrik Güvenlik Seviyesi (bit)	Eşdeğer RSA Anahtar Boyutu (bit)	Eşdeğer ECC Anahtar Boyutu (bit)	Anahtar Boyutu Oranı (RSA/ECC)
80	1024	160-223	~6:1
112	2048	224-255	~9:1
128	3072	256-383	~12:1
192	7680	384-511	~20:1
256	15360	512+	~30:1

Kaynaklar: (Sectigo, t.y.) 'dan derlenmiştir.

Tablonun da açıkça gösterdiği gibi, güvenlik seviyesi arttıkça RSA anahtar boyutları katlanarak büyürken, ECC anahtar boyutları çok daha yavaş bir artış gösterir. Bu durumun pratik sonuçları oldukça önemlidir:

- **Performans:** Daha küçük anahtar boyutları, daha az hesaplama gerektirir. Bu nedenle ECC, anahtar üretimi, şifreleme/deşifreleme ve özellikle dijital imza oluşturma gibi işlemlerde RSA'dan önemli ölçüde daha hızlıdır (Cheap SSL Security, t.y.). Örneğin, yapılan testler ECC'nin aynı sayıda isteğe RSA'ya göre yarı sürede yanıt verebildiğini göstermiştir (SSL2BUY, t.y.).

- **Kaynak Kullanımı:** Küçük anahtarlar, daha az depolama alanı, daha az bellek kullanımı ve iletim sırasında daha az ağ bant genişliği anlamına gelir (Mervana, t.y.). Bu, özellikle işlem gücü, bellek ve pil ömrünün kısıtlı olduğu mobil cihazlar, akıllı kartlar ve Nesnelerin İnterneti (IoT) cihazları için ECC'yi bariz bir şekilde üstün kılar (Sectigo, t.y.) .
- **Uyumluluk ve Yaygınlık:** RSA'nın en büyük avantajı, daha eski bir teknoloji olması nedeniyle daha uzun süredir kullanılıyor olması ve bu sayede daha geniş bir eski sistem uyumluluğuna sahip olmasıdır (Mervana, t.y.). Ancak endüstri trendi, özellikle yeni uygulamalar için açıkça ECC'ye doğru kaymaktadır (SSL2BUY, t.y.).

Maksimum performans, verimlilik ve ölçeklenebilirlik gerektiren modern uygulamalar için ECC en doğru tercihtir. Eski sistemlerle uyumluluğun kritik olduğu durumlarda ise RSA hala geçerli bir seçenek olarak kalmaktadır.

Dijital İmzalar ve Güvenli İletişim Protokolleri

Modern kriptografik sistemlerin mimarisi, "tek bir algoritma her şeye uyar" yaklaşımından uzaklaşarak, belirli görevler için optimize edilmiş farklı algoritmaların modüler ve hibrit bir kombinasyonuna dayanmaktadır. Simetrik algoritmaların hız avantajı ve asimetrik algoritmaların anahtar yönetimi ve kimlik doğrulama yetenekleri, gerçek dünya protokollerinde bir araya getirilerek her iki dünyanın da en iyisi elde edilir.

TLS/SSL El Sıkışmasında RSA ve ECC'nin Rolü

Transport Layer Security (TLS) ve onun öncülü olan Secure Sockets Layer (SSL), internet üzerindeki iletişimi (örneğin web trafiği - HTTPS) güvence altına alan temel protokollerdir. Bir istemci (örneğin web tarayıcısı) bir sunucuya bağlandığında, güvenli bir oturum başlatmak için **TLS el sıkışması (handshake)** adı verilen bir süreç gerçekleşir. Asimetrik kriptografi, bu sürecin iki kritik adımında rol oynar (Raj, t.y.) :

1. **Sunucu Kimlik Doğrulaması:** Sunucu, istemciye dijital sertifikasını gönderir. Bu sertifika, sunucunun kimliğini ve açık anahtarını içerir ve güvenilir bir Sertifika Otoritesi (CA) tarafından imzalanmıştır. Sunucu, sertifikasına karşılık gelen özel anahtara sahip olduğunu kanıtlamak için el sıkışma verilerinin bir kısmını özel anahtarıyla imzalayarak istemciye gönderir. İstemci, sunucunun açık anahtarını kullanarak bu imzayı doğrular ve böylece doğru sunucuyla konuştuğundan emin olur.
2. **Anahtar Değişimi:** El sıkışmasının temel amacı, oturumun geri kalanında kullanılacak olan yüksek hızlı simetrik şifreleme anahtarını

(genellikle bir AES anahtarı) güvenli bir şekilde oluşturmaktır. Bu işlem için hem RSA hem de ECC tabanlı yöntemler kullanılabilir. Ancak modern TLS sürümleri (özellikle TLS 1.3), **ECDHE (Elliptic Curve Diffie-Hellman Ephemeral)** adı verilen ECC tabanlı yöntemi ezici bir çoğunlukla tercih eder (Raj, t.y.).

ECDHE'nin tercih edilmesinin iki ana nedeni vardır. Birincisi, ECC'nin verimliliğidir. Daha küçük anahtar boyutları, daha az hesaplama gerektirir, bu da TLS el sıkışmasının daha hızlı tamamlanmasını ve dolayısıyla web sayfalarının daha hızlı yüklenmesini sağlar (Sectigo, t.y.). İkincisi ve daha önemlisi, ECDHE'nin mükemmel İleri Gizlilik (Perfect Forward Secrecy - PFS) sağlamasıdır. Bu, her bir TLS oturumu için geçici (ephemeral) bir anahtar çifti oluşturularak elde edilir. Bu sayede, sunucunun uzun vadeli özel anahtarı gelecekte bir şekilde ele geçirilse bile, geçmiş oturumların şifreleri çözülemez. RSA tabanlı anahtar değişiminin eski versiyonları bu özelliği sunmuyordu. ECC'nin hızı, her bağlantıda bu geçici anahtarların düşük bir performans maliyetiyle oluşturulmasını mümkün kılar.

Sanal Özel Ağlarda (VPN) Şifreleme: IPsec ve AES

Sanal Özel Ağlar (VPN), güvenilmeyen bir ağ (genellikle internet) üzerinden güvenli bir özel ağ bağlantısı oluşturmak için kullanılır. VPN teknolojisinin temelinde genellikle **IPsec (Internet Protocol Security)** adı verilen bir protokoller bütünü bulunur (WatchGuard, t.y.). IPsec, ağ katmanında çalışarak tüm IP paketleri için güvenlik sağlar ve tipik bir hibrit şifreleme modeli kullanır:

- **Veri Gizliliği ve Bütünlüğü (ESP):** IPsec, veri paketlerinin gizliliğini ve bütünlüğünü sağlamak için Şifreleyici Güvenlik Yükü (Encapsulating Security Payload - ESP) protokolünü kullanır (WatchGuard, t.y.). ESP, veri paketinin içeriğini (payload) şifrelemek için AES gibi güçlü bir simetrik şifreleme algoritması kullanır (WatchGuard, t.y.). Modern uygulamalarda, şifreleme ve kimlik doğrulamayı (bütünlük kontrolü) tek bir adımda verimli bir şekilde birleştiren
  AES-GCM (Galois/Counter Mode**)** gibi modlar tercih edilir. Bu, hem güvenlik hem de performans açısından avantaj sağlar (WatchGuard, t.y.) .
- **Anahtar Değişimi ve Kimlik Doğrulama (IKE):** Güvenli tünelin kurulması ve ESP tarafından kullanılacak simetrik anahtarların (AES anahtarları gibi) güvenli bir şekilde oluşturulması için **IKE (Internet Key Exchange)** protokolü kullanılır (NetworkLessons.com, t.y.). IKE, tarafların kimliklerini doğrulamak ve paylaşılan gizli anahtarları

oluşturmak için asimetrik kriptografi (genellikle Diffie-Hellman'ın RSA veya ECC ile birleştirilmiş versiyonları) kullanır.

Bu yapı, modern kriptografik sistemlerin temel tasarım desenini özetler: Asimetrik kriptografi, güvenli bir kanalın "kurulum" aşamasında (kimlik doğrulama ve oturum anahtarı değişimi) kullanılır. Ardından, yüksek hacimli verinin hızlı ve verimli bir şekilde şifrelenmesi için bu oturum anahtarı ile güçlü bir simetrik algoritma (AES) devreye girer. Bu, her iki dünyanın da en iyi yönlerini birleştiren pratik ve güçlü bir yaklaşımdır.

Veri Güvenliği Uygulamaları

Kriptografi, sadece hareket halindeki veriyi (data-in-transit) değil, aynı zamanda depolanmış, "bekleyen veriyi" (data-at-rest) de korumak için kritik öneme sahiptir.

Dosya, Disk ve Veritabanı Şifrelemede AES

AES, hızı ve güvenliği sayesinde bekleyen veriyi korumak için fiili endüstri standardı haline gelmiştir. Kullanım alanları oldukça geniştir:

- **Dosya ve Klasör Şifreleme:** Tek tek dosyaları veya klasörleri şifrelemek için kullanılır. Bu işlem genellikle bir parola veya bir anahtar dosyasından türetilen bir AES anahtarı ile yapılır (IBM, t.y.).
- **Tam Disk Şifreleme (Full Disk Encryption - FDE):** Bir sabit diskin veya SSD'nin tamamını şifreleyerek, cihazın fiziksel olarak çalınması durumunda bile verilere erişimi engeller. Windows'ta BitLocker ve macOS'ta FileVault gibi sistemler bu amaçla AES kullanır.
- **Veritabanı Şifreleme:** Finansal kayıtlar, kişisel bilgiler gibi hassas verileri barındıran veritabanlarını korumak için AES yaygın olarak kullanılır (GeeksforGeeks, t.y.). Veritabanı yönetim sistemleri, genellikle sütun düzeyinde, satır düzeyinde veya hatta her bir hücrenin ayrı ayrı şifrelenmesi gibi granüler kontrol imkanları sunar (Zaw, Thant, & Bezzateev, 2019) . AES'in yüksek performansı, bu tür granüler şifrelemenin veritabanı sorgularında kabul edilebilir bir performans etkisiyle yapılmasını mümkün kılar.

Bu uygulamalarda dikkat edilmesi gereken iki kritik nokta vardır: **anahtar yönetimi** ve **Başlatma Vektörü (Initialization Vector - IV)** kullanımı. Şifreleme anahtarının güvenli bir şekilde saklanması ve yönetilmesi, sistemin genel güvenliği için hayati önem taşır (IBM, t.y.). IV ise, aynı anahtarla aynı düz metnin her şifrelendiğinde farklı bir şifreli metin üretilmesini sağlayan rastgele bir değerdir. Bu, şifreli metinlerdeki kalıpları gizleyerek istatistiksel saldırıları önler ve güvenliği artırır (Nakov, t.y.).

E-posta Güvenliği: PGP/GPG ve Hibrit Şifreleme Modeli

Pretty Good Privacy (PGP) ve onun özgür ve açık kaynaklı bir uygulaması olan **GNU Privacy Guard (GPG)**, e-postaların ve dosyaların gizliliğini ve bütünlüğünü sağlamak için kullanılan en köklü ve güvenilir sistemlerden biridir (Precisely, 2022). OpenPGP standardı, hem RSA hem de daha modern olan ECC tabanlı anahtar çiftlerini destekler (OpenPGP, t.y.).

PGP/GPG, TLS ve IPsec gibi, klasik bir **hibrit şifreleme modeli** kullanır (Precisely, 2022):

1. Gönderilecek olan asıl mesaj (e-postanın gövdesi), rastgele oluşturulmuş tek kullanımlık bir **oturum anahtarı** ile AES gibi hızlı bir simetrik algoritma kullanılarak şifrelenir.
2. Bu küçük boyutlu oturum anahtarı, alıcının **açık anahtarı** (RSA veya ECC) ile şifrelenir.
3. Gönderici, mesajı kendi **özel anahtarı** ile imzalayarak kimlik doğrulama ve bütünlük sağlar.
4. Son olarak, simetrik olarak şifrelenmiş mesaj, asimetrik olarak şifrelenmiş oturum anahtarı ve dijital imza bir araya getirilerek alıcıya gönderilir.

Alıcı, önce kendi özel anahtarını kullanarak şifrelenmiş oturum anahtarını çözer. Ardından elde ettiği bu oturum anahtarıyla asıl mesajın şifresini çözer ve son olarak göndericinin açık anahtarını kullanarak dijital imzayı doğrular. Bu hibrit model, asimetrik şifrelemenin kolay anahtar yönetimi ve dijital imza yeteneklerini, simetrik şifrelemenin hızıyla birleştirerek son derece pratik, verimli ve güvenli bir çözüm sunar.

Kriptografinin Geleceği: Kuantum Tehdidi ve Kuantum Sonrası Kriptografi (PQC)

Kriptografi, durağan bir bilim dalı olmaktan uzaktır. Sürekli olarak yeni saldırı tekniklerine ve teknolojik gelişmelere adapte olmak zorundadır. Günümüzde bu alanın karşı karşıya olduğu en büyük ve en temel zorluk, kuantum bilgisayarların ortaya çıkardığı varoluşsal tehdittir.

Kuantum Bilgisayarların Yıkıcı Etkisi: Shor ve Grover Algoritmaları

Kuantum bilgisayarlar, klasik bilgisayarların bit (0 veya 1) tabanlı mantığından farklı olarak, süperpozisyon ve dolanıklık gibi kuantum mekaniği prensiplerini kullanan kübitler (qubit) ile çalışır. Bu farklı çalışma prensibi, onlara belirli türdeki problemleri klasik bilgisayarlardan katlanarak daha hızlı

çözme potansiyeli verir. Kriptografi açısından iki kuantum algoritması özellikle endişe vericidir:

- **Shor Algoritması:** 1994 yılında matematikçi Peter Shor tarafından geliştirilen bu algoritma, yeterince büyük ve hatasız bir kuantum bilgisayarda çalıştırıldığında, günümüz asimetrik kriptografisinin temelini oluşturan iki ana problemi polinom zamanda çözme yeteneğine sahiptir: **büyük sayıları asal çarpanlarına ayırma** (RSA'nın güvenliğinin dayandığı problem) ve **ayrık logaritma problemini çözme** (DLP ve ECDLP'nin, dolayısıyla Diffie-Hellman ve ECC'nin temelindeki problem) (Demirci, t.y.). Bu, Shor algoritmasının pratik olarak uygulanabilir hale geldiği gün, mevcut tüm ana akım açık anahtarlı şifreleme sistemlerinin (RSA, ECC, DSA vb.) tamamen ve geri döndürülemez bir şekilde kırılacağı anlamına gelir (Güney Bilişim, t.y.) . Bu, bir "daha hızlı kırma" meselesi değil, bu sistemlerin dayandığı matematiksel zorlukları temelden ortadan kaldıran bir "problemi çözülebilir hale getirme" meselesidir.
- **Grover Algoritması:** Bu algoritma, yapılandırılmamış bir veritabanında arama yapma problemini hedefler. Kriptografiye uygulandığında, simetrik anahtarları kaba kuvvetle bulma işlemini karesel olarak hızlandırır. Örneğin, N elemanlı bir anahtar uzayını klasik bir bilgisayar ortalama N/2 denemede ararken, Grover algoritması bunu yaklaşık N denemede yapabilir. Bu, AES gibi simetrik algoritmaların güvenliğini zayıflatır. Ancak bu tehdit, asimetrik kriptografiye yönelik tehdit kadar yıkıcı değildir ve anahtar boyutunu iki katına çıkararak (örneğin, 128-bit güvenlik için AES-256 kullanarak) etkin bir şekilde yönetilebilir (Newhouse & Regenscheid, 2025).

Bu nedenle, asıl varoluşsal tehdit asimetrik kriptografiye yöneliktir ve bu durum, tüm dijital güvenlik altyapısını yeniden inşa etme zorunluluğunu doğurmuştur.

Kuantum Sonrası Kriptografiye Geçiş: NIST Standardizasyon Çalışmaları

Kuantum bilgisayarların oluşturduğu bu ciddi tehdide karşı kriptografi topluluğu proaktif bir şekilde harekete geçmiştir. Amaç, henüz pratik ve büyük ölçekli kuantum bilgisayarlar inşa edilmeden önce, onlara karşı dirençli yeni kriptografik standartlar geliştirmektir. Bu yeni alana Kuantum Sonrası Kriptografi (Post-Quantum Cryptography - PQC) adı verilir. PQC'nin amacı, klasik bilgisayarlarda verimli bir şekilde çalışan ancak hem klasik hem de

kuantum bilgisayarlar tarafından kırılması zor olan algoritmalar tasarlamaktır (Talayhan, t.y.).

Bu geçiş sürecini yönetmek ve küresel standartlar oluşturmak amacıyla NIST, AES yarışmasına benzer bir modelle, 2016 yılında yeni bir halka açık PQC standardizasyon yarışması başlatmıştır (Newhouse & Regenscheid, 2025) . Süreç, dünyanın dört bir yanından gelen onlarca adayın yıllarca süren kamusal inceleme, analiz ve eleme turlarından geçmesini içermiştir. Sonunda, Ağustos 2024'te NIST, ilk üç PQC standardını (dijital imzalar ve anahtar kapsülleme mekanizmaları için) yayınlayarak bu uzun sürecin ilk meyvelerini vermiştir (Newhouse & Regenscheid, 2025). Bu standartların amacı, "Kripto-Kıyamet" (Crypto-Apocalypse) olarak adlandırılan, tüm dijital güvenliğin bir anda çökmesi senaryosunu önlemek için kurumlara ve geliştiricilere güvenli bir geçiş yolu sunmaktır.

Gelecek Vadeden PQC Adayları: Kafes, Kod, Özet ve İzojeni Tabanlı Kriptografi

Kuantum tehdidi, kriptografide "güvenlik ispatı" paradigmasını yeniden şekillendirmiş ve tek bir zor probleme dayanmak yerine, farklı matematiksel temellere dayanan çeşitli yaklaşımları zorunlu kılmıştır. Bu, **kriptografik çeşitlilik** olarak bilinen bir risk yönetimi stratejisidir. Eğer gelecekte bir problem ailesini çözen beklenmedik bir teorik atılım olursa, diğer ailelere dayanan sistemler güvende kalmaya devam edecektir. NIST'in PQC yarışmasında öne çıkan ve standartlaştırılan algoritmalar, bu çeşitliliği yansıtan farklı matematiksel problem ailelerine dayanmaktadır (Wikipedia contributors, t.y.) :

1. **Kafes (Lattice) Tabanlı Kriptografi:** Çok boyutlu geometrik yapılar olan kafesler üzerindeki bazı problemlerin (örneğin En Kısa Vektör Problemi - SVP) kuantum bilgisayarlar için bile zor olduğuna inanılmaktadır. Güçlü güvenlik kanıtları, verimlilikleri ve esneklikleri nedeniyle PQC yarışmasının en popüler ve başarılı adayları bu aileden çıkmıştır. NIST tarafından standartlaştırılan **CRYSTALS-Kyber (yeni adıyla ML-KEM)** anahtar kapsülleme mekanizması ve **CRYSTALS-Dilithium (yeni adıyla ML-DSA)** dijital imza algoritması bu aileye aittir (Wikipedia contributors, t.y.) .
2. **Kod (Code) Tabanlı Kriptografi:** Güvenliğini, genel bir lineer hata düzelten kodu çözmenin zorluğuna dayandırır. Bu, kriptografideki en eski PQC fikirlerinden biridir ve 1978'deki McEliece kriptosistemine dayanır. Bu alandaki algoritmalar genellikle büyük anahtar boyutlarına sahip olsalar da uzun süredir analiz edildikleri için güvenilir kabul edilirler. NIST, bu aileden **HQC** algoritmasını yedek bir standart adayı olarak incelemektedir (Ribeiro, 2025).

3. **Özet (Hash) Tabanlı İmzalar:** Bu algoritmaların güvenliği, yalnızca kullanılan kriptografik özet fonksiyonlarının (örneğin SHA-256) güvenliğine dayanır. Özet fonksiyonlarının kuantum saldırılarına karşı dirençli olduğu düşünüldüğünden, bu imzalar PQC için en güvenilir seçeneklerden biri olarak görülür. Dezavantajları ise genellikle daha büyük imza boyutlarına sahip olmaları ve bazılarının "durumlu" (stateful) olmasıdır (yani her özel anahtar sadece bir kez kullanılabilir). NIST, bu aileden durumsuz (stateless) bir imza şeması olan **SPHINCS+ (yeni adıyla SLH-DSA)**'yı, kafes tabanlı Dilithium'a bir yedek olarak standartlaştırmıştır.
4. **Çok Değişkenli (Multivariate) Kriptografi:** Güvenliğini, sonlu bir cisim üzerinde çok değişkenli ikinci dereceden bir polinom sistemini çözmenin zorluğuna dayandırır.
5. **İzojeni (Isogeny) Tabanlı Kriptografi:** Güvenliğini, süper-tekil eliptik eğriler arasındaki izojeni adı verilen özel haritaları bulma problemine dayandırır. Bu aile, en küçük anahtar boyutlarını sunma potansiyeliyle dikkat çekiyordu. Ancak, bu alandaki önde gelen adaylardan biri olan SIKE'nin 2022'de klasik bir bilgisayar kullanılarak kırılması, bu yaklaşımın henüz yeterince olgunlaşmadığını ve PQC araştırmalarının ne kadar dinamik ve zorlu olduğunu göstermiştir.

Aşağıdaki tablo, NIST tarafından standardizasyon için seçilen ilk PQC algoritmalarını özetlemektedir.

Tablo 4: NIST Tarafından Standardizasyon İçin Seçilen PQC Algoritmaları (Ağustos 2024 İtibarıyla)

Standart (FIPS No)	Algoritma (Orijinal Adı)	Yeni Adı (NIST)	Kategori	Dayandığı Matematiksel Problem Ailesi
FIPS 203	CRYSTALS-KYBER	ML-KEM	Anahtar Kapsülleme Mekanizması (KEM)	Kafes Tabanlı
FIPS 204	CRYSTALS-Dilithium	ML-DSA	Dijital İmza	Kafes Tabanlı
FIPS 205	SPHINCS+	SLH-DSA	Dijital İmza	Özet Tabanlı (Durumsuz)

(Taslak) FIPS 206	FALCON	FN-DSA	Dijital İmza	Kafes Tabanlı

Kaynaklar: (Newhouse & Regenscheid, 2025) 'den derlenmiştir.

Sonuç ve Gelecek Perspektifi

Bu çalışma, kriptolojinin antik çağlardaki basit gizleme tekniklerinden başlayarak, modern dijital dünyayı mümkün kılan karmaşık matematiksel sistemlere uzanan evrimsel yolculuğunu incelemiştir. Simetrik şifrelemenin temel taşları olan DES ve onun yerini alan, günümüzün standardı AES'in yapıları ve çalışma prensipleri analiz edilmiştir. Asimetrik kriptografi devrimini başlatan RSA ve onun daha verimli alternatifi olan ECC'nin matematiksel temelleri, performansları ve TLS, VPN, dijital imza gibi kritik uygulamalardaki rolleri karşılaştırmalı olarak ele alınmıştır.

Analizler, kriptografinin sürekli bir gelişim ve adaptasyon içinde olduğunu göstermektedir. Kriptograflar ve kriptoanalistler arasındaki bitmeyen yarış, daha güvenli algoritmaların doğmasını sağlamıştır. Teknolojik ilerlemeler, DES gibi bir zamanların devini emekliye ayırmış ve AES gibi hem güvenli hem de verimli standartların yükselişine yol açmıştır. Asimetrik kriptografinin icadı, güvenin merkezi otoritelerden denetlenebilir matematiksel problemlere kaydığı bir paradigma değişikliği yaratmıştır.

Bugün kriptografi, kuantum bilişimle birlikte tarihinin en büyük dönüm noktalarından birinin eşiğindedir. Kuantum sonrası kriptografiye (PQC) geçiş, önümüzdeki on yılın en önemli ve en zorlu siber güvenlik projelerinden biri olacaktır. Bu geçiş, sadece mevcut algoritmaları yenileriyle değiştirmekten ibaret olmayacaktır. Aynı zamanda, **hibrit mod** (hem klasik hem de PQC algoritmasını bir arada kullanarak geçiş sürecindeki riskleri azaltma) gibi yeni yaklaşımları ve **kripto-çeviklik (crypto-agility)**, yani sistemlerin gelecekteki tehditlere karşı gerektiğinde kriptografik algoritmalarını kolayca güncelleyebilme yeteneğini, temel bir tasarım ilkesi haline getirecektir (Newhouse & Regenscheid, 2025) .

Kriptografi, bilginin kendisi kadar değerli olduğu bir dünyada güvenliğin, mahremiyetin ve dijital ekonominin temel direği olmaya devam edecektir. Geleceğin güvenlik sistemleri, tek bir "kırılmaz" duvara değil, farklı ve bağımsız matematiksel temellere dayanan çok katmanlı savunma hatlarına dayanacaktır. Bu, kriptografinin durağan bir bilim olmaktan uzak, sürekli evrilen ve insanlığın teknolojik ilerlemesine paralel olarak kendini yeniden icat eden dinamik bir alan olduğunu bir kez daha kanıtlamaktadır.

NOT: Bu çalışma Tarık YERLİKAYA'nın "Yeni şifreleme algoritmalarının analizi" başlıklı doktora tezinden türetilmiştir.

KAYNAKÇA

Acar, Ö. F. (2025). *Bilgi güvenliği ve kriptoloji* [Ders materyali]. AVESİS. https://avesis.gazi.edu.tr/resume/lessonmaterieldownload/12818?key=0d0c07f9-af0a-4fb8-83b2-20b9fdab3aa1

Kaya, E. E. (2021, May 29). *Kriptografi: Bilginin anahtarı*. TÜBİTAK Bilim Genç. https://bilimgenc.tubitak.gov.tr/makale/kriptografi-bilginin-anahtari

Avaroğlu, E. (2017, May). Bilgi güvenliğinin temel yapı taşı: Kriptoloji. *Düşünce Dünyasında Türkiz, 8*(43). https://dergipark.org.tr/en/download/article-file/3174082

AWS. (2025). *Kriptografi nedir? - Kriptografiye ayrıntılı bakış*. Erişim tarihi: 16 Temmuz 2025, https://aws.amazon.com/tr/what-is/cryptography/

Bingöl, S. (2022, Ağustos). Osmanlı Devleti'nde kriptografik uygulamalar ve kod defterleri. *SDÜ Fen-Edebiyat Fakültesi Sosyal Bilimler Dergisi, 56*, 113-129.

Şahin, F. (2015). Modern blok şifreleme algoritmaları. *IAU Dergisi, 7*(26), 15-21. https://doi.org/10.17932/IAU.IAUD.m.13091352.2015.7/26.15-21

Beşkirli, A., Özdemir, D., & Beşkirli, M. (2019, Ekim). Şifreleme yöntemleri ve RSA algoritması üzerine bir inceleme. *Avrupa Bilim ve Teknoloji Dergisi*, Özel Sayı, 284-291.

National Institute of Standards and Technology. (t.y.). *Advanced Encryption Standard (AES): Soruları ve Cevapları*. Erişim tarihi: 16 Temmuz 2025, https://csrc.nist.rip/encryption/aes/aesfact.html

Newhouse, B., & Regenscheid, A. (2025, January 15–16). NIST post-quantum cryptography update. In *Cryptography Conference* (Austin, TX, USA | Online).

Wikipedia contributors. (t.y.). *NIST Post-Quantum Cryptography Standardization*. Erişim tarihi: 16 Temmuz 2025, https://en.wikipedia.org/wiki/NIST_Post-Quantum_Cryptography_Standardization

Dijital Güvenlik Platformu – Aksigorta. (t.y.). *AES*. Erişim 16 Temmuz 2025, https://dijitalguvenlikplatformu.aksigorta.com.tr/dijital-guvenlik-sozlugu/aes

Precisely. (2022, November 16). *PGP ve RSA: Aralarındaki farklar nelerdir?* Erişim 16 Temmuz 2025, https://www.precisely.com/blog/data-security/pgp-vs-rsa-encryption-difference

Precisely (t.y.). *What Is AES Encryption And How Does It Work?* Erişim tarihi: 16 Temmuz 2025, from https://www.jscape.com/blog/aes-encryption

Şeker, Ş. E. (t.y.). *DES (Veri Şifreleme Standardı, Data Encryption Standard)*. Erişim tarihi: 16 Temmuz 2025, https://bilgisayarkavramlari.com/2008/03/13/des-veri-sifreleme-standardi-data-encryption-standard/

Nechvatal, J., Barker, E., Bassham, L., Burr, W., Dworkin, M., Foti, J., & Roback, E. (2001). Report on the Development of the Advanced Encryption Standard (AES). *Journal of Research of the National Institute of Standards and Technology, 106*(3), 511–577. https://doi.org/10.6028/jres.106.023

National Institute of Standards and Technology. (2001). *Announcing the Advanced Encryption Standard (AES)* (Federal Information Processing Standards Publication 197). Gaithersburg, MD: U.S. Department of Commerce. https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf

Kula, G. Ç. (2009). *Gelişmiş Şifreleme Standardı Blok Şifreleme Algoritmasının Bir Mikroişlemci Üzerinde Gerçeklemesine Yan Kanal Saldırısı* (Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, El Anabilim Dalı).

Güney Bilişim. (t.y.). *Gelişmiş Şifreleme Standardı (AES) 256*. Erişim tarihi: 17 Temmuz 2025, https://www.guneybilisim.com/gelismis-sifreleme-standardi-aes-256

Kocaturk, O. (t.y.). *RSA Açık Anahtarlı Şifreleme Algoritması*. Medium. Erişim tarihi: 16 Temmuz 2025, https://medium.com/@okankocaturk/rsa-a%C3%A7%C4%B1k-anahtarl%C4%B1-%C5%9Fifreleme-algoritmas%C4%B1-bc9755156692

Vikipedi. (t.y.). *RSA (şifreleme yönetimi)*. Erişim tarihi: 16 Temmuz 2025, https://tr.wikipedia.org/wiki/RSA_(%C5%9Fifreleme_y%C3%B6netimi)

Yıldızhan, B. (t.y.). *Eliptik Eğri Şifreleme. 1 Elliptic-Curve Nedir?*. Medium. Erişim tarihi: 16 Temmuz 2025, https://medium.com/@baryldzhan/eli%CC%87pti%CC%87k-e%C4%9Fri%CC%87-%C5%9Fi%CC%87freleme-9bc56517f612

Elektrik Mühendisleri Odası. (t.y.). *Tasarsız Ağlarda Eliptik Eğri Şifreleme Altyapısı Kullanarak Anahtar Dağıtımı*. Erişim tarihi: 16 Temmuz 2025, https://www.emo.org.tr/ekler/bf2efbbe0c49b9f_ek.pdf

EITCA Academy. (t.y.). *Eliptik eğri kriptografisi (ECC) bağlamında eliptik eğri ayrık logaritma problemi (ECDLP), güvenlik ve verimlilik açısından klasik ayrık logaritma problemi ile nasıl karşılaştırılır ve modern kriptografik uygulamalarda neden eliptik eğriler tercih edilir?*. Erişim tarihi: 16 Temmuz 2025, https://tr.eitca.org/cybersecurity/eitc-is-acc-advanced-classical-cryptography/diffie-hellman-cryptosystem/generalized-discrete-log-problem-and-the-security-of-diffie-hellman/examination-review-generalized-discrete-log-problem-and-the-security-of-diffie-hellman/in-the-

context-of-elliptic-curve-cryptography-ecc-how-does-the-elliptic-curve-discrete-logarithm-problem-ecdlp-compare-to-the-classical-discrete-logarithm-problem-in-terms-of-security-and-efficie/

YazılımTürk. (t.y.). *Elliptic Curve Kriptografi - ECC Nedir?*. Erişim tarihi: 16 Temmuz 2025, https://www.yazilimturkiye.com/elliptic-curve-kriptografi-ecc-nedir/

EITCA Academy. (t.y.). *Eliptik Eğri Kriptografisinde (ECC) kullanılan eliptik bir eğriyi tanımlayan denklemin genel formu nedir?*. Erişim tarihi: 16 Temmuz 2025, https://tr.eitca.org/siber-g%C3%BCvenlik/eitc%2C-geli%C5%9Fmi%C5%9F-klasik-kriptografidir/eliptik-e%C4%9Fri-%C5%9Fifreleme/eliptik-e%C4%9Fri-%C5%9Fifreleme-ecc/muayene-incelemesi-eliptik-e%C4%9Fri-kriptografisi-ecc/eliptik-e%C4%9Fri-kriptografisinde-kullan%C4%B1lan-eliptik-bir-e%C4%9Friyi-tan%C4%B1mlayan-denklemin-genel-formu-nedir%3F/

Yücelen, A. M., Baykal, A., & Coşkun, C. (2017). Kriptolojide eliptik eğri algoritmasının uygulanması. *Düşünce Dünyasında Türkiz, 8*(3), 503–513. https://dergipark.org.tr/tr/download/article-file/445020

Sectigo. (t.y.). *ECC vs RSA vs DSA - Encryption Differences*. Erişim tarihi: 16 Temmuz 2025, https://www.sectigo.com/resource-library/rsa-vs-dsa-vs-ecc-encryption

SSL2BUY. (t.y.). *RSA vs ECC – Which is Better Algorithm for Security?*. Erişim tarihi: 16 Temmuz 2025, https://www.ssl2buy.com/wiki/rsa-vs-ecc-which-is-better-algorithm-for-security

Cheap SSL Security. (t.y.). *ECC vs RSA: Comparing SSL/TLS Algorithms*. Erişim tarihi: 16 Temmuz 2025, https://cheapsslsecurity.com/p/ecc-vs-rsa-comparing-ssl-tls-algorithms/

Mervana, P. (t.y.). *ECC ve RSA Karşılaştırması: Teknik Farklar Nelerdir?*. Erişim tarihi: 17 Temmuz 2025, https://sslinsights.com/ecc-vs-rsa/

Raj, V. (t.y.). *Why ECC is the Solution for IoT Security*. SecureW2. Erişim tarihi: 16 Temmuz 2025, https://www.securew2.com/blog/ecc-solution-iot-security

WatchGuard. (t.y.). *About IPSec Algorithms and Protocols*. Erişim tarihi: 16 Temmuz 2025, http://www.watchguard.com/help/docs/help-center/en-US/content/en-US/Fireware/mvpn/general/ipsec_algorithms_protocols_c.html

NetworkLessons.com. (t.y.). *IPsec (Internet Protocol Security)*. Erişim tarihi: 16 Temmuz 2025, https://networklessons.com/security/ipsec-internet-protocol-security

IBM. (t.y.). *AES kullanarak parolaların şifrelenmesi*. Erişim tarihi: 16 Temmuz 2025, https://www.ibm.com/docs/tr/was/8.5.5?topic=files-encrypting-passwords-by-using-aes

GeeksforGeeks. (t.y.). *Advanced Encryption Standard (AES)*. Erişim tarihi: 16 Temmuz 2025, https://www.geeksforgeeks.org/computer-networks/advanced-encryption-standard-aes/

Zaw, M., Thant, M., & Bezzateev, S. (2019). AES şifreleme, eliptik eğri şifreleme ve imza ile veritabanı güvenliği. 2019 Dalga Elektroniği ve Bilgi ve Telekomünikasyon Sistemlerinde Uygulamaları (WECONF). https://doi.org/10.1109/WECONF.2019.8840125

Nakov, P. (t.y.). *AES Encrypt / Decrypt - Examples*. Practical Cryptography for Developers. Erişim tarihi: 16 Temmuz 2025, https://cryptobook.nakov.com/symmetric-key-ciphers/aes-encrypt-decrypt-examples

OpenPGP. (t.y.). *Standard*. Erişim tarihi: 16 Temmuz 2025, https://www.openpgp.org/about/standard/

Demirci, M. (t.y.). *Kuantum Programlama: Shor ve Grover Algoritmaları*. Erişim tarihi: 16 Temmuz 2025, https://muratdemirci.me/2023/08/07/kuantum-programlama-shor-ve-grover-algoritmalari/

Güney Bilişim. (t.y.). *Kuantum Bilgisayarların Kriptografi Üzerindeki Etkileri ve Geleceği*. Erişim tarihi: 16 Temmuz 2025, https://www.guneybilisim.com/kuantum-bilgisayarlarin-kriptografi-uzerindeki-etkileri-ve-gelecegi

Talayhan, A. (t.y.). *Post-Quantum Cryptography (PQC)*. Erişim tarihi: 16 Temmuz 2025, https://www.abdullahtalayhan.com/kriptografi/?post=post_quantum.md

Ribeiro, A. (2025, 12 Mart). *NIST, kuantum sonrası kriptografi standardizasyonunu ilerletiyor, kuantum tehditlerine karşı koymak için HQC algoritmasını seçiyor*. Industrial Cyber. Erişim tarihi: 16 Temmuz 2025, https://industrialcyber.co/nist/nist-advances-post-quantum-cryptography-standardization-selects-hqc-algorithm-to-counter-quantum-threats/


BÖLÜM 2

Veri Arttırma Stratejilerine Sistematik Bakış

Didem Abidin[1]

1. GİRİŞ

Yapay zekâ ve makine öğrenmesi, son on yılda bilimsel araştırmaların ve endüstriyel uygulamaların en dinamik alanlarından biri haline gelmiştir. Görüntü işleme, doğal dil işleme, ses analizi, biyomedikal tanı (Litjens et al., 2017), otonom sürüş (Geiger, Lenz, & Urtasun, 2012) ve finansal öngörüler (Ileberi et al., 2021) (Elreedy et al., 2019) gibi geniş bir yelpazede kullanılan derin öğrenme yöntemleri, insan düzeyine yaklaşan veya bazı görevlerde insanı aşan performanslar sergilemektedir. Ancak bu başarıların ardında çoğunlukla milyonlarca örnekten oluşan geniş ölçekli veri kümeleri bulunmaktadır (Deng et al., 2009). Gerçek dünyada ise araştırmacılar, çoğu zaman sınırlı büyüklükte, dengesiz veya nadir olayları temsil eden veri kümeleri ile karşı karşıya kalmaktadır. Özellikle tıp, finans, güvenlik veya düşük kaynaklı diller gibi alanlarda veri elde etme süreci hem maliyetli hem de etik kısıtlamalarla çevrili olabilir; bu noktada veri artırma (data augmentation) stratejileri (Ni et al., 2019), modelin öğrenme sürecini güçlendiren ve genelleme kapasitesini iyileştiren vazgeçilmez bir metodolojik araç olarak öne çıkar (Shorten & Khoshgoftaar, 2019).

Veri artırma en basit tanımıyla, mevcut verilerden türetilmiş yeni örnekler üreterek öğrenme sürecini desteklemeyi hedefler (Shorten & Khoshgoftaar, 2019). Görsel veri kümelerinde bu, bir görüntünün döndürülmesi, parlaklığının değiştirilmesi veya gürültü eklenmesi gibi işlemlerle; zaman serilerinde sinyalin yeniden ölçeklenmesi, kaydırılması veya segmentlere ayrılmasıyla; doğal dil işlemeye özgü senaryolarda ise eş anlamlı kelime değiştirme (EDA) (Feng et al., 2021), geri çeviri (back-translation) (Sennrich, Haddow, & Birch, 2016) ya da büyük dil modellerinden türetilen yapay örneklerle veri kümesinin zenginleştirilmesiyle gerçekleştirilir (Wei & Zou, 2019; Edunov, Ott, Auli, & Grangier, 2018; Iwana & Uchida, 2021). Bu yaklaşımların ortak noktası, modelin yalnızca eğitim örneklerini "ezberlemesine" değil, daha genel desenleri

[1] Dr. Öğr. Üyesi, Manisa Celal Bayar Üniversitesi, Bilgisayar Mühendisliği, ORCID: 0000-0001-5966-7537

yakalamasına imkân vermesidir; böylece bias–variance dengesinde varyans düşerken genelleme başarısı artar (Hastie, Tibshirani, & Friedman, 2009).

Veri artırmanın teorik çerçevesi, öğrenme kuramlarının temel tartışmaları ile doğrudan bağlantılıdır. Geleneksel istatistiksel öğrenme teorisine göre bir modelin başarısı, eğitim verisinin miktarı ve çeşitliliği ile model kapasitesinin etkileşimine bağlıdır; yüksek kapasiteli derin mimariler küçük veya dengesiz veri kümelerinde aşırı uyum eğilimi gösterebilir. Veri artırma stratejileri, gerçek veri dağılımını genişletilmiş biçimde yeniden örnekleyerek bu açığı kapatır ve girdi-uzayında bir tür düzenleme (data-space regularization) etkisi üretir (Hastie, Tibshirani, & Friedman, 2009; Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

Tarihsel açıdan bakıldığında, veri artırmanın ilk uygulamaları bilgisayar görüşü alanında ortaya çıkmıştır. El yazısı rakamları tanımaya yönelik MNIST veri kümesi üzerinde yapılan deneylerde (LeCun et al., 1998), sınırlı sayıdaki örnekleri döndürmek, kaydırmak veya bozmalar eklemek, modelin test başarısını anlamlı şekilde artırmıştır. Bu yöntem, daha sonra ImageNet gibi geniş ölçekli veri kümelerinde de kullanılarak günü-müzün modern konvolüsyonel sinir ağlarının (CNN) başarısında te-mel bir rol oynamıştır. Zamanla, veri artırma yalnızca görüntü alanıyla sınırlı kalmamış; ses işleme, metin analizi, biyoinformatik, tıp görün-tüleme ve finansal zaman serileri gibi pek çok farklı alana yayılmıştır. Günümüzde ise Generative Adversarial (Goodfellow et al., 2014) Networks (GANs), Variational Autoencoder (Kingma & Welling, 2014)s (VAEs) veya büyük dil modelleri gibi üretici yapay zekâ sis-temleri sayesinde veri artırma, tamamen yeni ve yüksek gerçekçilik-te örneklerin üretilmesi anlamına gelmektedir.

Veri artırma yalnızca teknik bir çözüm değil, aynı zamanda metodolojik güvenilirliğin de bir parçası olarak görülmelidir. Küçük veya dengesiz veri kümeleriyle çalışan araştırmacılar için deneysel sonuçların istatistiksel açıdan güvenilir olabilmesi çoğu zaman artırmaya bağlıdır: Tıbbi görüntülemede nadir görülen vakalar için sentetik örneklerle eğitim seti genişletilebilir (Litjens et al., 2017; Frid-Adar, Klang, Amitai, Goldberger, & Greenspan, 2018); finansal sahtekârlık tespitinde sınıf dengesizliği sentetik örneklerle hafifletilebilir (Cheah, et al, 2023); düşük kaynaklı dillerde ise back-translation gibi yöntemler kritik rol oynar (Sennrich et al., 2016; Edunov et al., 2018).

Bununla birlikte veri artırma her durumda güvenilir sonuçlar doğurmaz. Aşırı sentetik örnek üretimi, modelin gerçek dünyadaki varyasyonları doğru şekilde öğrenmesini engelleyebilir ya da dağılımı bozarak yanlış genellemelere yol

açabilir; bu nedenle stratejiler veri türü ve problem bağlamına uygun şekilde seçilmeli ve uygulanmalıdır (Shorten & Khoshgoftaar, 2019). Ayrıca etik boyut özellikle sağlık ve finans gibi hassas alanlarda önem taşır; hasta görüntülerinde mahremiyet riskleri ve finansal verilerde düzenleyici kaygılar, artırmanın yalnızca teknik değil toplumsal sorumluluk içeren bir mesele olduğunu gösterir (Litjens et al., 2017).

Sonuç olarak veri artırma stratejileri, modern yapay zekâ araştırmalarının teorik, metodolojik ve uygulamalı boyutlarını bir araya getiren disiplinlerarası bir alan olarak değerlendirilebilir. Bu kitap bölümü, veri artırmanın teoriden uygulamaya uzanan yolculuğunu kapsamlı biçimde ele almayı hedeflemektedir: İlk bölümde artırmanın öğrenme teorilerindeki yeri ve regularizasyon ile ilişkisi; ikinci bölümde görsel, zaman serisi (Wen et al., 2020; Iwana & Uchida, 2021), doğal dil ve multimodal veriler için başlıca metodolojik yaklaşımlar; üçüncü bölümde ise tıp görüntüleme, otonom sürüş, doğal dil işleme ve finansal zaman serileri gibi uygulama alanlarında gerçek dünya etkileri tartışılacaktır. Devam eden kısımlarda güçlü–zayıf yönler ile etik ve metodolojik konulara değinilecek; son olarak generative AI ve self-supervised öğrenme (Jaiswal et al., 2020) bağlamında geleceğe dönük yönelimler ele alınacaktır.

2. TEORİK ARKA PLAN

Veri artırma stratejileri, günümüz makine öğrenmesi araştırmalarında yalnızca pratik bir mühendislik aracı değil, aynı zamanda öğrenme teorilerinin merkezinde yer alan bir konu olarak görülmektedir. Bu bölümde veri artırmanın teorik temelleri, istatistiksel öğrenme teorisi bağlamındaki karşılığı, bias–variance dengesiyle ilişkisi (Hastie, Tibshirani, & Friedman, 2009), regularizasyon kavramıyla bağı, veri artırma ve veri sentezleme ayrımı, ayrıca transfer öğrenme ile etkileşimleri ele alınacaktır (Sakaridis et al., 2018). Böylece veri artırmanın yalnızca uygulamalı bir yöntem değil, aynı zamanda kuramsal düzeyde tartışılması gereken bir olgu olduğu ortaya konacaktır (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

2.1. Öğrenme Teorisi ve Veri Çeşitliliğinin Rolü

Makine öğrenmesi, temelde gözlemlerden genelleme yapma problemidir. İstatistiksel öğrenme teorisine göre, bir modelin beklenen hata (expected risk) değeri iki ana bileşenden oluşur: eğitim hatası (empirical risk) ve bu hatanın gerçek dağılım üzerinde genelleşme farkı. Eğer eğitim verisi sınırlı veya çeşitlilik açısından zayıfsa, modelin öğrendiği temsiller gerçek veri dağılımını yeterince kapsayamaz. Bu durumda, modelin genelleme hatası artar.

Veri artırma, tam da bu noktada devreye girer. Farklı dönüşümler veya sentetik örnekler aracılığıyla veri dağılımının daha geniş bir kesitini temsil etmek, modelin parametre uzayında daha dengeli bir öğrenme yapmasına imkân tanır. Örneğin, bir görüntü sınıflandırma probleminde kedilerin sadece belirli açılardan görüldüğü bir veri kümesi, modelin yeni açılardan gelen örneklerde başarısız olmasına neden olabilir. Ancak görüntü döndürme veya yansıtma gibi basit artırmalar, bu açığı kısmen kapatabilir. Buradan çıkan teorik sonuç şudur: Veri artırma, veri dağılımının kapsama alanını genişleterek modelin hipotez uzayındaki genelleme kapasitesini artırır.

2.2. Bias–Variance Trade-Off ve Veri Artırma

İstatistiksel öğrenme teorisinin temel kavramlarından biri bias–variance trade-off'tur. Bir modelin hatası üç ana bileşenden oluşur: bias, variance ve irreducible error. Bias, modelin sistematik hatalarını; variance, modelin veri örneklerine duyarlılığını; irreducible error ise ölçüm hataları veya gürültüden kaynaklanan kaçınılmaz hataları ifade eder. (Hastie, Tibshirani, & Friedman, 2009)

Veri artırma stratejileri, doğrudan variance bileşenini azaltma potansiyeline sahiptir. Çünkü eklenen yapay örnekler, modelin eğitim sürecinde daha fazla varyasyonla karşılaşmasını sağlar ve model parametrelerinin küçük değişimlere aşırı duyarlı olmasını engeller. Bunun sonucunda, modelin varyansı düşerken genelleme başarısı artar. (Hastie, Tibshirani, & Friedman, 2009)

Matematiksel olarak, bir tahmin fonksiyonunun beklenen hatası şu şekilde ifade edilir:

$$E[(y - \hat{f}(x))^2] = \text{Bias}^2[\hat{f}(x)] + \text{Var}[\hat{f}(x)] + \sigma^2 \quad (1)$$

Burada σ^2 veri setindeki rastgele gürültüyü temsil eder. Veri artırma yöntemleri, doğrudan bias'ı azaltmayabilir; ancak variance'ı düşürerek toplam hatanın azalmasına katkı sunar. Özellikle küçük veri kümelerinde variance bileşeni yüksek olduğu için, veri artırma stratejileri kritik bir rol üstlenmektedir (Hastie, Tibshirani, & Friedman, 2009).

2.3. Regularizasyon ve Veri Artırma İlişkisi

Regularizasyon, makine öğrenmesinde modelin aşırı uyumunu engellemek için kullanılan temel bir yöntemdir. L1/L2 norm cezaları, dropout, early stopping gibi teknikler parametre uzayını sınırlayarak modelin daha basit hipotezler seçmesini sağlar. Veri artırma ise regularization'ın farklı bir türü olarak değerlendirilebilir; çünkü modelin girdilerine çeşitlilik ekleyerek parametrelerin

belirli örneklere aşırı uyum göstermesini engeller (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

Örneğin, dropout parametrelerin bir kısmını rastgele etkisiz hale getirerek çeşitlilik sağlarken; veri artırma, girdi uzayında çeşitlilik yaratır. İki yaklaşım da aynı hedefe, yani genelleme kapasitesini artırmaya hizmet eder. Dolayısıyla, veri artırmayı data-space regularization olarak kavramsallaştırmak mümkündür.

2.4. Veri Artırma, Veri Sentezleme ve Transfer Learning

Veri artırma ile veri sentezleme (data synthesis) kavramları zaman zaman karıştırılmaktadır. Veri artırma, mevcut verilerin dönüşüm veya manipülasyon yoluyla türetilmiş versiyonlarını üretir. Örneğin, bir görüntüyü döndürmek veya metinde eş anlamlı kelimelerle değiştirme yapmak veri artırmadır. Veri sentezleme ise sıfırdan tamamen yeni örnekler üretir; örneğin GAN tabanlı bir sistemin hiç var olmayan bir insan yüzü yaratması bu kapsamdadır (Goodfellow et al., 2014).

Bunun yanında, transfer learning (aktarılabilir öğrenme) veri artırma ile birlikte sıkça kullanılan bir yöntemdir. Önceden geniş bir veri kümesi üzerinde eğitilmiş bir model, yeni ve küçük bir veri kümesinde fine-tune edilerek yeniden kullanılır. Veri artırma, bu küçük veri kümesinin kapasitesini artırarak transfer learning sürecinin başarısını güçlendirir. Dolayısıyla, transfer learning ve veri artırma birlikte uygulandığında sinerjik bir etki doğurur.

2.5. Matematiksel Çerçevede Veri Artırma

Veri artırma stratejilerini formel olarak tanımlamak için, eğitim veri kümesini $D = \{(x_i, y_i)\}_{i=1}^{N}$ şeklinde düşünelim. Burada x_i girdileri, y_i ise etiketleri temsil etmektedir. Veri artırma fonksiyonu T, girdi üzerinde bir dönüşüm uygulayarak yeni bir örnek üretir:

$$(x_i', y_i') = T(x_i, y_i) \quad (2)$$

Elde edilen artırılmış veri kümesi D′, orijinal veri kümesi ile birleşerek daha büyük bir eğitim kümesi oluşturur:

$$D_{aug} = D \cup D' \quad (3)$$

Modelin eğitiminde kullanılan risk fonksiyonu da buna göre güncellenir:

$$R_{aug}(f) = (1 / |D_{aug}|) \Sigma (x, y) \in D_{aug} L(f(x), y) \quad (4)$$

Buradaki L, kayıp fonksiyonunu; f, öğrenilen hipotez fonksiyonunu temsil eder. Veri artırmanın teorik katkısı, D_{aug}'un veri dağılımını daha geniş kapsaması sayesinde R_{aug}'un gerçek risk R(f)'e daha iyi yaklaşmasını sağlamasıdır.

2.6. Literatürde Teorik Çerçeveye Katkılar

Son yıllarda yapılan araştırmalar, veri artırmanın yalnızca pratik faydalarını değil, aynı zamanda kuramsal etkilerini de incelemiştir. Örneğin, Zhang ve arkadaşları (2021) derin ağların aşırı parametrelenmesine rağmen neden başarılı olduklarını tartışırken, veri artırmanın implicit regularization etkisine dikkat çekmişlerdir. Benzer şekilde, Shorten ve Khoshgoftaar (2019), veri artırma yöntemlerini sistematik olarak inceleyerek, bu yöntemlerin farklı veri türlerindeki ortak teorik faydalarını vurgulamışlardır (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

Teorik araştırmalar, veri artırmanın üç temel katkısını vurgulamaktadır. Öncelikle, eğitim verisinin efektif boyutunu artırarak modelin varyansını düşürmekte ve böylece genelleme kapasitesini güçlendirmektedir. İkinci olarak, modelin girdi uzayında daha geniş ve çeşitli temsiller öğrenmesine imkân sağlayarak farklı varyasyonlara karşı daha dayanıklı hale gelmesine katkıda bulunmaktadır. Son olarak, veri artırma yöntemleri istatistiksel test gücünü yükselterek deneysel sonuçların güvenilirliğini artırmakta ve elde edilen bulguların bilimsel geçerliliğini pekiştirmektedir.

3. METODOLOJİK YAKLAŞIMLAR

Veri artırma stratejilerinin teorik temelleri, öğrenme kuramları ve regularization kavramlarıyla olan ilişkisi bir önceki bölümde ele alınmıştır. Bu bölümde ise farklı veri türleri için geliştirilen metodolojik yaklaşımlar ayrıntılı biçimde incelenecektir. Veri türüne göre kullanılan yöntemler önemli farklılıklar göstermektedir; çünkü görsel, işitsel, metinsel veya sensör tabanlı verilerin yapısı, dönüşümlere verdikleri tepkiler ve korunması gereken semantik özellikler birbirinden farklıdır. Bu nedenle veri artırma metodolojisini tartışırken, yöntemleri veri türlerine göre kategorize etmek kritik önemdedir.

3.1. Görsel Veriler İçin Veri Artırma

Görsel veriler, makine öğrenmesi ve derin öğrenme araştırmalarında en yaygın kullanılan veri türlerinden biridir. Nesne tanıma, yüz tanıma, otonom araçlar, medikal görüntüleme (Litjens et al., 2017) ve uydu görüntü analizi gibi alanlarda kullanılan görsel veri kümeleri, veri artırma stratejilerinin en çok uygulandığı alanı oluşturur. Görüntülerin yüksek boyutlu yapısı ve semantik çeşitliliği, modellerin geniş bir varyasyon spektrumuna maruz kalmasını gerektirir. Ancak pratikte, görüntülerin farklı açılardan, farklı ışık koşullarında ve farklı cihazlarla elde edilmesi çoğu zaman mümkün değildir. Bu nedenle,

görsel veriler için geliştirilen veri artırma yöntemleri hem temel dönüşüm tabanlı yaklaşımları hem de ileri üretici (generative) yöntemleri kapsamaktadır.

3.1.1. Geometrik Dönüşümler

Görsel verilerde kullanılan en temel artırma yöntemleri geometrik dönüşümlerdir (Shorten & Khoshgoftaar, 2019). Bir görüntünün döndürülmesi, yansıtılması, kaydırılması, ölçeklenmesi veya kırpılması gibi işlemler, verinin semantik bütünlüğünü bozmadan yeni varyasyonlar üretir. Örneğin, el yazısı tanıma görevlerinde rakamların farklı açılardan görünmesi için rotasyon uygulanabilir. Nesne tanıma görevlerinde ise bir nesnenin farklı konumlarda bulunabileceğini göstermek amacıyla çevirme (flip) veya kaydırma kullanılabilir. Bu yöntemler, görüntünün mekânsal düzenini değiştirerek yeni varyasyonlar üretir.

Döndürme (Rotation), görüntünün belirli bir açıyla döndürülmesidir. Örneğin, el yazısı tanımada rakamların ±15° döndürülmesi modelin farklı yazı stillerine uyum sağlamasını kolaylaştırır. Yansıtma (Flipping), görüntünün yatay veya dikey eksende çevrilmesidir. Özellikle yüz tanıma veya nesne tespitinde, farklı açılardan gelen görünümleri simüle etmek için kullanılmaktadır. Kaydırma (Translation), görüntünün x veya y ekseninde belli bir miktar kaydırılmasıdır. Nesnelerin farklı konumlarda bulunabileceğini göstermek için etkilidir. Ölçekleme (Scaling/Zoom), görüntünün boyutlarının değiştirilmesidir. Yakınlaştırma veya uzaklaştırma yoluyla modelin ölçek varyasyonlarına karşı dayanıklılığı artırılır. Kırpma (Cropping), görüntünün belirli bir bölgesinin alınmasıdır. Nesnenin kısmen görünmesi durumlarını modellemeye yardımcı olur.

Bu dönüşümler özellikle Convolutional Neural Networks (CNN) gibi yapılar için kritik önemdedir. Çünkü CNN'ler konum duyarlı filtreler kullandığından, verinin farklı konumlarda sunulması modelin öznitelik çıkarma kapasitesini artırır. Ancak bu dönüşümlerin aşırıya kaçmaması gerekir; örneğin, bir el yazısı "6" rakamı 180 derece döndürüldüğünde "9" olarak algılanabilir ve etiketin semantiği bozulur.

3.1.2. Fotometrik Dönüşümler

Geometrik dönüşümlerin yanı sıra, görsel verilerde fotometrik dönüşümler de yaygın olarak kullanılmaktadır. Parlaklık, kontrast, renk doygunluğu, bulanıklık veya gürültü ekleme gibi işlemler, modelin farklı ışık koşullarına ve kamera özelliklerine karşı dayanıklı hale gelmesini sağlar. Özellikle nesne tanıma ve yüz tanıma uygulamalarında bu tür artırmaların performansa katkısı büyüktür.

Parlaklık ve Kontrast Ayarı, görüntülerin farklı ışık koşullarında çekilmiş gibi görünmesini sağlar. Renk Doygunluğu (Saturation) ve Ton (Hue), özellikle doğal görüntülerde renk çeşitliliğini artırır. Bulanıklık (Blurring), hareket bulanıklığı veya odak dışı görüntüler simüle edilebilir. Gürültü Ekleme (Noise Injection), Gaussian, Poisson veya speckle gürültüsü eklenerek kamera sensör hataları taklit edilir. Fotometrik dönüşümlerin en büyük avantajı, modelin farklı cihazlar veya çekim koşulları altında kararlılığını artırmasıdır. Örneğin, medikal görüntüleme cihazları arasındaki kontrast farklılıkları, fotometrik artırmalarla modellenerek modelin cihaz bağımsız çalışması sağlanabilir.

3.1.3. Karma Artırma Teknikleri

Geometrik ve fotometrik dönüşümlerin bir arada kullanılmasıyla elde edilen karma artırma yöntemleri, daha güçlü varyasyonlar üretmektedir. Bu bağlamda, örneğin Random Erasing yöntemi görüntünün rastgele bir bölümünün siyah bir kutucukla kapatılmasına dayanır ve modelin eksik bilgilerle de doğru tahmin yapabilmesini sağlar (Zhong et al., 2020). Cutout (DeVries et al., 2017) veya CutMix (Yun et al., 2019) tekniklerinde ise görüntünün bir bölgesi başka bir görüntüyle değiştirilir; bu yaklaşım özellikle ImageNet gibi (Deng et al., 2009) büyük ölçekli veri kümelerinde yaygın şekilde kullanılmaktadır. Bunun yanı sıra, Mixup yöntemi (Zhang et al., 2018) iki farklı görüntünün lineer kombinasyonunu oluşturarak yeni örnekler üretir ve etiketlerin de aynı oranda karıştırılmasıyla modelin karar sınırlarını daha yumuşak biçimde öğrenmesine katkıda bulunur. Bu tür karma yöntemler, klasik dönüşümlere kıyasla daha agresif olup modelin daha soyut ve genellenebilir temsiller öğrenmesine yardımcı olmaktadır.

3.1.4. Üretici Modeller ile Artırma

Geleneksel dönüşümlerin ötesinde, Generative Adversarial Networks (GANs) ve Variational Autoencoders (VAEs) gibi üretici modeller, gerçekçi görüntüler üretme kapasitesi sayesinde veri artırmada devrim niteliğinde bir rol oynamaktadır.

Üretici modeller, klasik yöntemlerin aksine yalnızca mevcut verinin türevlerini üretmekle kalmaz, aynı zamanda daha önce gözlemlenmemiş ancak olası olan yeni örnekler de yaratabilir. Bu sayede özellikle nadir olayların veya azınlık sınıfların öğrenilmesi kolaylaşmaktadır (Ma et al., 2017). Bu çerçevede, GAN tabanlı yöntemler azınlık sınıflara ait gerçekçi örnekler üretmek için yaygın biçimde kullanılmakta (Goodfellow et al., 2014), örneğin medikal görüntü veri setlerinde nadir görülen tümör tipleri için GAN'larla yapay görüntüler oluşturularak sınıf dengesi sağlanabilmektedir. Benzer şekilde, VAE tabanlı yöntemler veri dağılımını modelleyerek yeni örnekler üretir ve özellikle

gürültüye dayanıklı temsil öğrenme açısından güçlü bir yaklaşım sunar (Kingma & Welling, 2014). Bunun yanı sıra, style transfer (Gatys et al., 2016) teknikleri bir görüntünün stilini başka bir görüntüye uygulayarak yeni varyasyonlar elde edilmesine imkân verir; örneğin aynı sahnenin güneşli, yağmurlu veya sisli hava koşullarında farklı versiyonları bu yöntemle üretilebilir.

3.1.5. Uygulama Alanlarından Örnekler

Veri artırmanın uygulama alanlarına bakıldığında, özellikle tıp, otonom araçlar ve uydu görüntüleri öne çıkmaktadır. Tıp alanında, radyoloji görüntülerinde sınırlı sayıda anormal vaka bulunduğu için GAN tabanlı artırma yöntemleri sıklıkla kullanılmakta ve nadir görülen vakaların daha dengeli bir şekilde temsil edilmesi sağlanmaktadır (Frid-Adar, Klang, Amitai, Goldberger, & Greenspan, 2018). Otonom araçlarda ise farklı hava koşulları veya ışık senaryolarını simüle eden artırmalar, sistemin zorlu çevresel koşullarda da güvenilir şekilde çalışmasına katkıda bulunmaktadır (Sakaridis, Dai, & Van Gool, 2018; Dosovitskiy, Ros, Codevilla, Lopez, & Koltun, 2017). Uydu görüntüleri bağlamında ise düşük çözünürlüklü veriler üzerinde gerçekleştirilen fotometrik artırmalar, farklı uydu sensörleri arasında uyumluluğu artırmakta ve daha bütüncül analizlerin yapılabilmesine imkân vermektedir.

3.2. Zaman Serisi Verileri İçin Veri Artırma

Zaman serisi verileri, ardışık ölçümlerden oluştuğu için görsel veya metinsel verilerden farklı özelliklere sahiptir (Iwana & Uchida, 2021). Bu tür verilerde yalnızca örneklerin sayısı değil, aynı zamanda zaman bağımlılığı korunması da kritik önemdedir. Finansal piyasalar, IoT sensörleri, biyomedikal sinyaller (ör. EEG, ECG), enerji tüketimi ölçümleri veya ağ trafiği verileri gibi birçok uygulama alanında kullanılan zaman serileri, veri artırma araştırmalarında özel yöntemlerin geliştirilmesine yol açmıştır.

3.2.1. Basit Dönüşüm Tabanlı Teknikler

Jittering, zaman serisine küçük rastgele gürültü eklenmesidir (Iwana & Uchida, 2021). Özellikle sensör verilerinde ölçüm hatalarını simüle ederek modelin daha dayanıklı olmasını sağlar. Matematiksel olarak:

$$x'_t = x_t + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma^2) \quad (5)$$

Burada ϵ sıfır ortalamalı normal dağılımdan seçilen gürültüyü temsil eder.

Scaling, sinyalin genliğinin ölçeklenmesidir. Örneğin finansal serilerde belirli dönemlerde görülen dalgalanmalar ölçeklenerek artırma yapılabilir:

$$x'_t = \alpha \cdot x_t \quad (6)$$

Burada α sabit bir ölçek faktörüdür.

Permutation, sinyalin belirli segmentlerinin yer değiştirmesiyle yeni varyasyonlar üretilir. Bu yöntem, örüntülerin sırasının model için kritik olmadığı durumlarda faydalıdır.

3.2.2. Warping Teknikleri

Zaman ekseni üzerinde yapılan küçük esnetme (stretching) veya sıkıştırma (compression) işlemleri, özellikle biyomedikal sinyallerde yaygın olarak kullanılmaktadır. Time Warping, sinyalin belli kısımlarını hızlandırarak veya yavaşlatarak farklı varyasyonlar üretir.

Örneğin, ECG verilerinde bir QRS kompleksinin süresini biraz uzatmak ya da kısaltmak, farklı bireylerde gözlenen doğal varyasyonları simüle eder. Bu yöntem, Dynamic Time Warping (DTW) algoritmalarıyla yakından ilişkilidir.

3.2.3. Sınıf Dengesizliği için Oversampling Teknikleri

Zaman serilerinde dengesiz veri önemli bir problemdir. Örneğin, dolandırıcılık işlemleri finansal verilerde çok nadir görüldüğünden, sınıf dağılımı aşırı dengesizdir. Bu tür durumlarda SMOTE (Synthetic Minority Oversampling Technique) ve türevleri yaygın olarak kullanılır (Chawla, Bowyer, Hall, & Kegelmeyer, 2002; Fernández et al., 2018). Zaman serilerinde sınıf dengesizliğini gidermek amacıyla en yaygın kullanımı olan SMOTE, azınlık sınıfındaki örnekler arasında interpolasyon yaparak yeni sentetik örnekler üretir. Borderline-SMOTE ise yalnızca karar sınırına yakın azınlık örneklerini kullanarak sınıflar arasındaki ayrımın daha net öğrenilmesini sağlar (Han, Wang, & Mao, 2005). ADASYN (Adaptive Synthetic Sampling) yöntemi ise azınlık sınıfındaki örneklerin zorluk derecesini dikkate alarak, daha zor sınıflandırılan bölgelerde daha fazla sentetik veri üretir (He, Bai, Garcia, & Li, 2008). Bu yöntemlerin zaman serilerine uyarlanmış versiyonları, örneğin SMOTE-TS, verinin sekans yapısını göz önünde bulundurarak interpolasyon yapmakta ve böylece ardışık bağımlılıkların korunmasına imkân tanımaktadır.

3.2.4. Sentetik Sequence Generation

Son yıllarda üretici modellerin gelişmesiyle birlikte zaman serileri için tamamen yeni sekanslar üretmek mümkün hale gelmiştir. RNN ve LSTM tabanlı modeller geçmiş değerlerden öğrenerek yeni sinyaller üretebilirken, GAN tabanlı modeller gürültüden gerçekçi zaman serileri üretmek için kullanılmaktadır. Örneğin, finansal piyasa verilerinde nadir görülen kriz senaryoları GAN tabanlı artırma yöntemleriyle simüle edilebilmektedir. Bunun yanı sıra, Variational Autoencoders (VAEs) zaman serilerinin latent temsillerini öğrenerek yeni

varyasyonlar üretme imkânı sunmaktadır. Tüm bu yöntemler özellikle nadir olayların modellenmesinde kritik öneme sahiptir. Ancak üretici modellerin aşırı sentetikleşmesi, verinin gerçekçi yapısından uzaklaşmasına ve modelin gerçek dünya koşullarına yeterince uyum sağlayamamasına yol açabilmektedir.

3.3. Doğal Dil Verileri İçin Veri Artırma

Doğal dil işleme (Natural Language Processing – NLP), insan dilini anlamaya ve işlemeye yönelik yöntemleri kapsar. Görüntü veya zaman serisi verilerinden farklı olarak, metin verilerinde hem sözdizimsel (syntactic) hem de anlamsal (semantic) bütünlük korunmak zorundadır. Bu nedenle, metin için veri artırma yöntemleri, yalnızca kelime seviyesinde manipülasyonlardan ibaret değildir; bağlamı ve cümlenin genel anlamını korumak da esastır.

NLP'de veri artırma ihtiyacı özellikle düşük kaynaklı diller, etiketli veri azlığı ve sınıf dengesizliği sorunları nedeniyle öne çıkmaktadır. Büyük dil modellerinin (LLM) yükselişiyle birlikte bu alanda yeni fırsatlar ortaya çıksa da klasik yöntemlerden modern üretici yaklaşımlara kadar geniş bir yelpaze hâlâ aktif olarak kullanılmaktadır.

3.3.1. Lexical Substitution (Eş Anlamlı Değiştirme)

Doğal dil için en basit artırma yöntemlerinden biri, belirli kelimeleri eş anlamlılarıyla değiştirmektir. Bu yöntem, EDA (Easy Data Augmentation) çerçevesinde en sık kullanılan stratejilerden biridir (Wei & Zou, 2019). Ancak bu yaklaşımın dezavantajı, bağlam uyumunun her zaman korunamamasıdır. Örneğin, "bank" kelimesinin "finans kurumu" anlamındaki kullanımı ile "nehir kenarı" anlamı birbirinden farklıdır; yanlış eş anlamlı seçimi semantik kaymaya yol açabilir.

3.3.2. Back-Translation

Back-translation, metnin önce başka bir dile çevrilip ardından tekrar orijinal dile çevrilmesi sürecidir (Sennrich, Haddow, & Birch, 2016). Bu yöntem, cümlenin semantik anlamını korurken yüzeysel varyasyonlar yaratır. Bu yöntem, özellikle düşük kaynaklı diller için kritik bir rol oynamaktadır. Farklı çeviri sistemleri veya pivot diller kullanılarak, aynı cümleden birçok varyasyon türetilebilir.

3.3.3. Rastgele Manipülasyon Teknikleri (EDA)

EDA stratejileri, metin üzerinde dört temel manipülasyona dayanır. Bunlardan ilki synonym replacement, belirli kelimelerin eş anlamlılarıyla değiştirilmesini içerir. İkinci yöntem olan random insertion, cümleye yeni kelimeler eklenmesine

dayanırken, random deletion bazı kelimelerin silinmesiyle çeşitlilik yaratır. Son olarak random swap ile cümledeki kelimelerin yerleri değiştirilir. Bu yöntemler etiketlenmiş veri azlığında basit ama etkili çözümler sunmakta; ancak aşırı uygulandığında metnin anlamını bozma riski taşımaktadır.

3.3.4. Masked Language Modeling

BERT gibi transformer tabanlı modellerin popülerleşmesiyle birlikte masked language modeling (MLM) veri artırmada kullanılmaya başlanmıştır (Devlin, Chang, Lee, & Toutanova, 2019). Cümlede belirli kelimeler maskelenir ve model bu boşlukları doldurarak yeni varyasyonlar üretir. Bu yöntem, bağlamı dikkate aldığı için lexical substitution'dan daha güvenilir sonuçlar üretir.

3.3.5. Paraphrasing ve LLM Tabanlı Yaklaşımlar

Son yıllarda büyük dil modelleri (LLM) doğrudan veri artırma aracı olarak kullanılmaya başlanmıştır. Paraphrasing (yeniden ifade etme) teknikleri sayesinde, bir cümle farklı sözdizimleriyle yeniden yazılabilir. Büyük dil modelleri, aynı cümleden çok sayıda farklı paraphrase üretebilir. Bu yaklaşım, özellikle duygu analizi, niyet sınıflandırma ve soru-cevap sistemlerinde veri setlerini zenginleştirmek için kullanılmaktadır (Rajpurkar, Zhang, Lopyrev, & Liang, 2016). Ancak bu yöntemlerin riskleri de vardır. Örneğin aşırı yapay örnekler etiket uyumsuzluğu yaratabilir veya LLM'ler bazen gerçekçi olmayan veya yanlış içerik üretebilir.

3.3.6. Veri Artırma + Transfer Learning Etkileşimi

Metin veri artırma teknikleri çoğunlukla transfer learning ile birlikte kullanılır. Örneğin, küçük bir veri kümesi üzerinde back-translation uygulanarak set genişletilir, ardından BERT veya GPT gibi önceden eğitilmiş bir model bu veri üzerinde fine-tune edilir. Böylece hem veri artırma hem de önceden öğrenilmiş dil bilgisi bir araya gelmiş olur. Semantik uyumu korumak için EDA benzeri yüzeysel işlemlerin ötesine geçen contextual augmentation yaklaşımında, Kobayashi (2018) kelimeleri bağlama göre paradigmatik ilişkileri kullanarak uygun alternatiflerle değiştirir; dil modeli olasılıklarıyla seçilen bu adaylar, anlamsal tutarlılığı artırırken olası etiket kaymasını da azaltır.

3.3.7. Uygulama Alanlarından Örnekler

Uygulama alanlarında veri artırma, farklı görevlerde somut kazanımlar sağlar: duygu analizi bağlamında, küçük ve dengesiz veri senaryolarında EDA gibi yüzeysel ancak etkili işlemlerle ve tutarlılık temelli artırma yaklaşımlarıyla anlamlı doğruluk artışları rapor edilmiştir (Wei & Zou, 2019; Xie, Dai, Hovy, Luong, & Le, 2020). Soru-cevap (Q&A) sistemlerinde, SQuAD gibi veri

kümelerinde geri çeviri ve paraphrasing ile üretilen ek örnekler model genellemesini iyileştirir; QANet'in İngilizce–Fransızca back-translation ile güçlendirilmesi ve etiketlenmemiş metinden sentetik Soru-Cevap üretimi buna örnektir (Yu et al., 2018; Lewis, Denoyer, & Riedel, 2019; Rajpurkar, Zhang, Lopyrev, & Liang, 2016). Düşük kaynaklı diller söz konusu olduğunda ise, NMT ve sınıflandırma bağlamlarında geri çeviri ve nadir kelime odaklı artırma yöntemleri tutarlı kazanımlar sağlamış, ölçekli çalışmalar bu yaklaşımların genellenebilirliğini doğrulamıştır (Sennrich, Haddow, & Birch, 2016; Fadaee, Bisazza, & Monz, 2017; Edunov, Ott, Auli, & Grangier, 2018).

3.4. Multimodal ve İleri Yöntemler

Veri artırma araştırmalarında son yıllarda öne çıkan eğilimlerden biri, multimodal veriler için geliştirilmiş yöntemlerdir. Modern uygulamalarda tek bir veri türü yerine birden fazla modalite bir arada kullanılmaktadır. Örneğin, otonom araçlarda kamera görüntüleri, LIDAR sensör verileri ve GPS bilgileri birlikte değerlendirilir (Li & Ibanez-Guzman, 2020); sağlık bilişiminde medikal görüntüler, hasta raporları ve sensör ölçümleri aynı sistemde bulunur; eğitim teknolojilerinde ise video, ses ve metin tabanlı etkileşimler aynı ortamda analiz edilir. Bu bağlamda, multimodal veri artırma stratejileri yalnızca her modalite için ayrı ayrı dönüşüm uygulamakla kalmaz, aynı zamanda modaliteler arasında tutarlılığı da korumak zorundadır.

3.4.1. Otonom Araçlarda Multimodal Arttırma

Otonom sürüş sistemleri, veri artırmanın en yoğun ve zengin biçimde kullanıldığı alanlardan biridir (Geiger, Lenz, & Urtasun, 2012). Bu sistemlerin gerçek dünyada güvenilir şekilde çalışabilmesi için gece sürüşü, yağmurlu hava, yoğun trafik ya da düşük görüş koşulları gibi çok çeşitli senaryolarla karşılaşabilmesi gerekir. Bu nedenle multimodal artırmalar yalnızca kamera görüntülerinde değil, aynı zamanda LIDAR ve radar sensörlerinde de eşzamanlı olarak uygulanmaktadır. Kamera verilerinde geometrik ve fotometrik dönüşümler, örneğin rotasyon, parlaklık değişimi veya bulanıklık, farklı görsel koşulları temsil edecek biçimde varyasyonlar üretir. LIDAR nokta bulutlarında noktaların seyreltilmesi, rastgele gürültü eklenmesi ya da bulutların döndürülmesi gibi işlemler gerçekleştirilir. GPS verilerinde ise gürültü ekleme veya küçük koordinat kaydırmalarıyla farklı konum varyasyonları simüle edilir. Ancak burada en büyük zorluk, bu artırmaların modaliteler arası senkronizasyon içinde uygulanabilmesidir; zira örneğin kamerada yağmurlu bir hava koşulu simüle edildiğinde, LIDAR verisinde de yağmurun yansıma etkilerinin uygun şekilde yansıtılması gerekmektedir.

3.4.2. Multimodal Sağlık Verileri

Tıp alanında multimodal veri artırma yöntemleri giderek önem kazanmaktadır. Örneğin, radyoloji görüntüleri (MRI, CT taramaları) ile hasta raporları (metinsel veriler) aynı sistemde kullanılır. Burada yapılacak artırma, görüntü üzerinde gürültü eklenmesi veya parlaklık değiştirilmesi gibi dönüşümlerin yanı sıra, hasta raporlarında paraphrasing teknikleriyle varyasyon yaratılmasını içerebilir. Ancak bu tür artırmaların en kritik yönü, görüntü ve raporun semantik uyumunu korumaktır. Eğer MRI üzerinde tümör büyütülürken, rapor hâlâ "küçük tümör" diyorsa bu etiket uyumsuzluğu yaratır.

3.4.3. Generative AI ile Multimodal Artırma

GAN ve VAE tabanlı yöntemler yalnızca tek modalite için değil, aynı anda birden fazla modaliteyi üretebilecek şekilde geliştirilmiştir. Örneğin, bir hastanın MRI görüntüsü ile buna karşılık gelen raporu aynı anda üretilebilmektedir. Benzer şekilde, otonom araçlar için yağmurlu bir senaryo oluşturulduğunda hem kamera görüntüsü hem de sensör verileri aynı koşullara uygun biçimde üretilebilir.

Bunun yanında, Diffusion Models gibi yeni üretici modeller (Zhu et al., 2025), multimodal artırmada daha gerçekçi ve yüksek kaliteli örnekler üretme kapasitesine sahiptir (Ho, Jain, & Abbeel, 2020). Örneğin, Stable Diffusion tabanlı yaklaşımlar yalnızca görsel veri değil, metin-görsel çiftlerini de tutarlı biçimde üretebilmektedir.

3.4.4. Self-Supervised Öğrenme ve Veri Artırma

Son yıllarda self-supervised öğrenme paradigması, veri artırmayı doğrudan öğrenme sürecinin merkezine yerleştirmiştir. Bu yaklaşımlarda model, etiketlenmemiş veriler üzerinde farklı artırmalarla üretilmiş varyasyonlar arasındaki tutarlılığı öğrenmeye çalışır. Örneğin, SimCLR (Simple Contrastive Learning of Representations) yönteminde aynı görüntüden farklı artırmalarla elde edilen iki görünüm representation uzayında birbirine yakın olacak şekilde öğrenilmektedir (Chen, Kornblith, Norouzi, & Hinton, 2020). BYOL (Bootstrap Your Own Latent) yaklaşımı, öğrenme sürecinde pozitif eşleştirmelerin oluşturulabilmesi için veri artırmayı zorunlu hale getirmektedir (Grill et al., 2020). MoCo (Momentum Contrast) ise büyük ölçekli representation öğrenmede artırma çeşitliliğinin performans için kritik rol oynadığını ortaya koymaktadır. Tüm bu yöntemler, veri artırmayı yalnızca ek bir destekleyici teknik olmaktan çıkarıp doğrudan öğrenme sürecinin temel bileşeni haline getirmiştir.

3.4.5. Etik ve Metodoloik Sorunlar

Multimodal artırmanın sunduğu fırsatların yanı sıra bazı riskleri de beraberinde getirdiği görülmektedir. Özellikle üretici modeller kullanıldığında modaliteler arası tutarsızlık riski artmakta, sentetik verilerdeki hatalar gerçek dünyada yanlış genellemelere yol açabilmektedir. Ayrıca sağlık ve güvenlik gibi kritik alanlarda bu tür yöntemlerin kullanımı etik tartışmaları gündeme getirmektedir. Dolayısıyla multimodal artırmalar yalnızca teknik doğruluk açısından değil, aynı zamanda etik ve metodolojik hassasiyetler gözetilerek ele alınmalıdır.

4. UYGULAMA ALANLARI

Veri artırma stratejileri, yalnızca teorik bir kavram veya metodolojik bir teknik değildir; aynı zamanda gerçek dünya uygulamalarında doğrudan etkisini gösteren bir yaklaşımdır. Farklı veri türleri üzerinde kullanılan artırma yöntemleri, özellikle veri kıtlığı, dengesiz dağılımlar ve nadir olayların modellenmesi gibi sorunların çözümünde kritik rol oynamaktadır. Bu bölümde, veri artırmanın en yoğun şekilde kullanıldığı dört alan ele alınacaktır: tıp ve biyomedikal görüntüleme, otonom araç sistemleri, doğal dil işleme ve finansal zaman serileri.

4.1. Tıp ve Biyomedikal Görüntüleme

4.1.1. Veri Sorunları ve Artırma İhtiyacı

Tıbbi veriler, makine öğrenmesi için en değerli kaynaklardan biri olmakla birlikte, etik ve pratik kısıtlamalar nedeniyle çoğu zaman sınırlı sayıda örneğe sahiptir. Örneğin, nadir görülen bir tümör türüne ait yüzlerce örnek toplamak mümkün değildir. Ayrıca hasta verilerinin gizliliği, veri paylaşımını kısıtlamakta, bu da eğitim için kullanılabilecek veri miktarını azaltmaktadır. Veri artırma bu noktada hem sınırlı veri sorununun çözümü hem de genelleme kapasitesinin artırılması için vazgeçilmez bir yöntem haline gelmiştir.

4.1.2. Kullanılan Yöntemler

Tıbbi görüntüleme alanında kullanılan veri artırma yöntemleri farklı türlerdeki dönüşümlere dayanmaktadır. Geometrik dönüşümler, MRI veya CT taramalarında küçük rotasyonlar ve kaydırmalar yapılarak farklı cihaz açılarının simülasyonunu sağlamaktadır (Tobin et al., 2017). Fotometrik dönüşümler ise özellikle X-ray görüntülerinde kontrast farklılıklarını değiştirerek farklı radyoloji cihazlarının çıktıları arasındaki uyumsuzlukları gidermeye yardımcı olmaktadır. Bunun yanı sıra, GAN tabanlı yöntemler az görülen hastalık tiplerine ait yeni örnekler üretmek için yaygın biçimde kullanılmakta; örneğin göğüs X-

ray'lerinde nadir akciğer hastalıklarının görüntüleri sentetik olarak oluşturulabilmektedir. Ayrıca noise injection teknikleri ultrason görüntülerine cihaz kaynaklı gürültü ekleyerek sistemlerin daha dayanıklı hale gelmesine katkı sağlamaktadır.

4.2. Otonom Araç Sistemleri

4.2.1. Veri Sorunları ve Artırma İhtiyacı

Otonom araç sistemleri, gerçek dünyada güvenilir şekilde çalışabilmek için son derece büyük, çeşitli ve dengeli veri kümelerine ihtiyaç duyar. Ancak bu tür verilerin toplanması oldukça maliyetli ve zaman alıcıdır. Örneğin, araçların farklı hava koşullarında, farklı yollar ve trafik yoğunluklarında test edilmesi gerekir. Bu tür koşulların tamamını gerçek dünyada toplamak neredeyse imkânsızdır. Ayrıca nadir durumlar (örneğin, bir aracın ani fren yapması veya yol ortasında duran bir hayvan) çok seyrek görüldüğü için bu tür örneklerin veri setinde yeterince temsil edilmesi zordur. Veri artırma, bu eksiklikleri gidermek ve otonom araçların güvenliğini artırmak için kritik bir stratejidir.

4.2.2. Kamera Görüntülerinde Veri Artırma

Otonom araçlarda kullanılan kamera verileri, görsel veri artırma tekniklerinin en yoğun biçimde uygulandığı alanlardan birini oluşturur. Bu kapsamda geometrik dönüşümler, görüntülere küçük rotasyonlar uygulanarak farklı kamera açılarını simüle etmeyi mümkün kılar. Fotometrik dönüşümler ise parlaklık, kontrast veya renk varyasyonları aracılığıyla güneşli, bulutlu ya da gece gibi farklı hava koşullarını temsil eder. Daha ileri düzeyde, yağmur, sis ve kar simülasyonları özel filtrelerle yapılarak düşük görüş koşulları sentetik olarak oluşturulur ve böylece araçların bu zorlu ortamlardaki performansı test edilebilir. Ayrıca CutMix ve Mixup gibi yöntemlerle farklı yol senaryoları karıştırılarak araçların daha soyut temsiller öğrenmesi sağlanır.

4.2.3. LIDAR ve Radar Nokta Bulutlarında Veri Artırma

Kamera verilerinin yanı sıra, otonom araçlar çevrelerini LIDAR ve radar sensörleri aracılığıyla da algılar ve bu sensörlerden elde edilen nokta bulutu (point cloud) verilerinin artırılması, görsel verilerden farklı bir yaklaşım gerektirir. Bu alanda kullanılan yöntemlerden biri olan nokta seyreltme (point dropping), belirli bir yüzde oranında noktaların kaldırılmasıyla sensör eksikliklerinin simülasyonunu sağlar. Nokta gürültüsü (noise injection) tekniğinde ise noktaların konumlarına küçük rastgele gürültüler eklenerek ölçüm hataları taklit edilir. Bunun yanında, dönüşüm (rotation/scaling) yöntemleriyle nokta bulutları küçük açılarla döndürülerek veya ölçeklenerek farklı senaryolar

oluşturulabilir. Ayrıca, nadir olayların simülasyonu kapsamında yolda beklenmedik nesneler ya da trafik işaretleri sentetik olarak eklenerek araçların bu olağan dışı durumlara karşı tepkisi test edilebilir.

4.2.4. GPS ve Sensör Füzyonunda Veri Artırma

Otonom araçlar yalnızca kamera ve LIDAR verilerini değil, aynı zamanda GPS, ivmeölçer ve hız sensörleri gibi farklı kaynaklardan elde edilen bilgileri de birleştirir. Veri artırma bu çoklu sensör verileri üzerinde de uygulanmaktadır. Örneğin, GPS gürültüsü yönteminde küçük koordinat kaymaları eklenerek GPS hataları simüle edilir ve sistemin konumlama doğruluğu test edilir. Sensör senkronizasyonu tekniklerinde farklı sensörlerden gelen veriler arasında yapay zamanlama kaymaları oluşturularak sistemin senkronizasyon hatalarına karşı dayanıklılığı ölçülür. Ayrıca multimodal tutarlılık büyük önem taşır; örneğin yağmurlu bir senaryo simüle edildiğinde yalnızca kamera görüntüsünde değil, aynı zamanda LIDAR verisinde de yağmurun etkilerini yansıtacak değişikliklerin yapılması gerekir.

4.3. Doğal Dil İşleme

4.3.1. Veri Sorunları ve Artırma İhtiyacı

Doğal dil işleme (NLP), insan dilinin işlenmesi ve modellenmesi üzerine kurulu bir alan olup, makine çevirisi, duygu analizi, sohbet botları, soru-cevap sistemleri ve bilgi çıkarımı gibi birçok uygulamayı kapsar. NLP'deki en büyük zorluklardan biri, dilin karmaşıklığı ve çeşitliliğidir. Aynı anlam farklı sözdizimleriyle ifade edilebilir, kelimeler çoklu anlam taşıyabilir (polisemi), bağlama göre anlam kaymaları yaşanabilir. Ayrıca birçok dil için geniş ölçekli etiketli veri seti bulunmamaktadır. İngilizce için milyonlarca örnek içeren veri setleri mevcutken, düşük kaynaklı dillerde (örneğin Afrika dilleri, Türk lehçeleri) bu tür veri kümeleri oldukça sınırlıdır. Bu nedenle veri artırma, NLP uygulamalarında yalnızca performans artırıcı değil, aynı zamanda gereklilik haline gelmiştir.

4.3.2. Düşük Kaynaklı Dillerde Veri Artırma

Veri artırma, düşük kaynaklı dillerin işlenmesinde kritik bir rol oynamaktadır. Bu bağlamda en yaygın kullanılan yöntemlerden biri back-translation olup, kaynak dilden başka bir dile çevrilen cümlenin tekrar orijinal dile çevrilmesiyle yeni örnekler üretilmesini sağlar. Örneğin, Svahili dilindeki bir cümlenin önce İngilizce'ye, ardından tekrar Svahili'ye çevrilmesi yüzeysel varyasyonların elde edilmesine imkân tanır. Bunun yanı sıra, cross-lingual transfer yaklaşımında güçlü kaynak dillerden (İngilizce, Fransızca) düşük kaynaklı dillere transfer

öğrenme yapılırken artırma teknikleri kullanılarak veri çeşitliliği artırılır. Ayrıca, paraphrasing yöntemleriyle büyük dil modelleri (LLM) düşük kaynaklı dillerde yeni cümleler türeterek veri kümesini zenginleştirebilmektedir. Bu tür stratejiler, düşük kaynaklı diller için geliştirilen makine çevirisi ve dil modelleme uygulamalarında başarıyı önemli ölçüde artırmaktadır.

4.3.3. Soru-Cevap Sistemleri ve Diyalog Uygulamaları

Soru-cevap sistemleri (Q&A) ve sohbet botlarının kullanıcıların farklı biçimlerde yönelttiği sorulara doğru yanıt verebilmesi gerekir. Ancak aynı sorunun farklı sözcüklerle ifade edilmesi, örneğin "Bugün hava nasıl?" ve "Hava durumu nedir?" gibi sorular, modeller için zorluk oluşturabilmektedir. Bu noktada veri artırma devreye girerek sistemlerin daha esnek hale gelmesini sağlar. Kullanılan yöntemlerden biri olan synonym replacement, sorulardaki belirli kelimelerin eş anlamlılarla değiştirilmesine dayanır (Rajpurkar, Zhang, Lopyrev, & Liang, 2016). Paraphrase generation ise büyük dil modelleri aracılığıyla aynı sorunun farklı sözdizimleriyle yeniden üretilmesini mümkün kılar. Bunun yanı sıra back-translation yöntemiyle sorular farklı dillere çevrilip tekrar orijinal dile getirildiğinde yeni varyasyonlar elde edilir. Bu stratejiler, soru-cevap sistemlerinin daha doğal, kapsayıcı ve kullanıcı dostu bir şekilde çalışmasına katkıda bulunmaktadır.

4.3.4. Duygu Analizi ve Metin Sınıflandırma

Duygu analizi görevlerinde kullanılan veri setleri genellikle dengesizdir; örneğin pozitif yorumlar çoğunlukta bulunurken, negatif yorumlar görece daha azdır. Bu tür durumlarda veri artırma stratejileri kritik bir rol oynamaktadır. EDA (Easy Data Augmentation) kapsamında synonym replacement, random insertion, deletion ve swap gibi basit yöntemlerle yeni örnekler üretilerek veri çeşitliliği artırılabilir. Masked language modeling yaklaşımında belirli kelimeler maskelenerek BERT gibi modellerin tahminleri doğrultusunda farklı varyasyonlar elde edilir (Devlin, Chang, Lee, & Toutanova, 2019). Ayrıca paraphrasing teknikleri kullanılarak aynı duyguyu farklı ifadelerle yansıtan cümleler üretilebilir; örneğin "The food was great!" ifadesi "I really enjoyed the meal." biçiminde yeniden yazılabilir. Araştırmalar, bu yöntemlerin özellikle küçük ölçekli veri setlerinde %5 ila %15 arasında doğruluk artışı sağladığını ortaya koymaktadır.

4.3.5. Büyük Dil Modelleri ve Veri Artırma

Son yıllarda GPT, LLaMA ve T5 gibi büyük dil modelleri (LLM), doğrudan veri artırma aracı olarak kullanılmaya başlanmıştır. Bu modeller, bir cümleyi

yeniden yazma yoluyla paraphrasing yapabilmekte, belirli bir sınıfa ait çok sayıda örnek üreterek veri kümelerini genişletebilmekte ve farklı konu başlıkları üzerinden çeşitlilik sağlayarak dengesiz veri setlerinin dengelenmesine katkıda bulunabilmektedir. Avantajları arasında yüksek çeşitlilik üretme kapasitesi ve bağlam uyumunu koruma becerisi öne çıkarken, dezavantajları arasında etik sorunlar, örneğin yanlış bilgi üretme ihtimali ve kontrolsüz biçimde üretilen sentetik verilerin modele zarar verme riski bulunmaktadır.

4.4. Finansal Zaman Serileri

4.4.1. Veri Sorunları ve Artırma İhtiyacı

Finansal veriler, hisse senedi fiyatları, döviz kurları, işlem hacimleri, kredi kartı harcamaları ve banka transferleri gibi çok çeşitli kaynaklardan elde edilen zaman serilerini kapsar (Cheah, et al, 2023). Bu veriler yüksek hacimli ve sürekli akış halindedir. Ancak bu alandaki temel zorluk, nadir olayların (örneğin, borsa çöküşleri, ani spekülatif hareketler veya sahte işlemler) az sayıda gözlemlenebilmesidir. Özellikle dolandırıcılık tespiti gibi uygulamalarda sahte işlemler, toplam işlemlerin %1'inden azını oluşturur. Bu durum sınıf dengesizliğine yol açar ve modeller çoğunluk sınıfa aşırı uyum gösterir. Veri artırma, finansal zaman serilerinde hem dengesizliği gidermek hem de modelin nadir olaylara duyarlılığını artırmak için kritik rol oynar.

4.4.2. Basit Dönüşüm Tabanlı Yöntemler

Finansal seriler üzerinde doğrudan uygulanan en basit veri artırma teknikleri arasında jittering, scaling ve time shifting yöntemleri öne çıkmaktadır. Jittering yaklaşımında fiyat serisine küçük rastgele gürültüler eklenerek farklı piyasa senaryoları simüle edilir. Scaling yöntemi, fiyat veya işlem hacmi serisinin ölçeklenmesiyle farklı piyasa volatilite koşullarını temsil etmeye imkân tanır. Time shifting ise zaman ekseninde kaydırmalar yaparak gecikmeli senaryoların oluşturulmasına olanak verir. Bu yöntemlerin ortak yönü, modellerin küçük varyasyonlara karşı daha dayanıklı hale gelmesini sağlamalarıdır.

4.4.3. Oversampling Teknikleri

Finansal veri setlerinde sınıf dengesizliğini gidermek amacıyla en yaygın kullanılan yöntemler arasında SMOTE ve türevleri bulunmaktadır. SMOTE, azınlık sınıfındaki örnekler arasında interpolasyon yaparak yeni sentetik örnekler üretirken, Borderline-SMOTE özellikle sahte işlemler ile gerçek işlemler arasındaki karar sınırında örnekler oluşturarak ayrımın daha net biçimde öğrenilmesine katkıda bulunur. ADASYN ise zor sınıflandırılan örneklere daha fazla sentetik veri üreterek modelin bu bölgelerde daha duyarlı hale gelmesini

sağlar. Bu yöntemler, özellikle dolandırıcılık tespitinde azınlık sınıfın model tarafından öğrenilmesini kolaylaştırmakta ve sınıf dengesizliğinin yarattığı olumsuz etkileri azaltmaktadır.

4.4.4. Gelişmiş Yöntemler: GAN ve RNN Tabanlı Artırma

Son yıllarda finansal serilerde üretici modellerin kullanımı giderek yaygınlaşmıştır. GAN tabanlı artırma yöntemleri, piyasa verilerinden öğrenilen dağılımlara dayalı yeni fiyat serileri üretmekte ve özellikle nadir kriz senaryolarının sentetik olarak oluşturulmasına olanak sağlamaktadır. RNN ve LSTM tabanlı sequence generation yaklaşımları, tarihsel işlem verilerinden öğrenilen zaman bağımlılıklarını kullanarak yeni sahte işlem senaryoları üretmektedir. Bunun yanı sıra, VAE tabanlı artırma yöntemleri fiyat serilerinin latent uzay temsilleri üzerinden varyasyonlar oluşturmaya imkân verir. Tüm bu stratejiler, özellikle piyasa anormalliklerinin modellenmesinde değerli katkılar sunmaktadır.

5. TARTIŞMA

Veri artırma stratejileri, modern makine öğrenmesinde yalnızca performans artırıcı bir teknik değil, aynı zamanda metodolojik güvenilirliğin de önemli bir bileşenidir. Önceki bölümlerde farklı veri türleri için kullanılan yöntemler ve çeşitli uygulama alanlarındaki katkılar ayrıntılı biçimde ele alınmıştır. Bu bölümde ise bu yöntemlerin güçlü ve zayıf yönleri, metodolojik farklılıkları, etik boyutları ve gelecekteki araştırmalara ışık tutacak tartışmalar değerlendirilecektir.

5.1. Yöntemlerin Güçlü Yönleri

Veri artırmanın en önemli avantajı, genelleme kapasitesini yükseltmesi ve küçük/dengesiz veri kümelerinde öğrenme sürecini desteklemesidir. Görsel verilerde basit geometrik ve fotometrik dönüşümler, CNN tabanlı modellerin başarısını gözle görülür biçimde artırmaktadır. Zaman serilerinde jittering, scaling ve warping gibi teknikler, ölçüm hatalarını ve doğal varyasyonları simüle ederek daha dayanıklı modellerin geliştirilmesini sağlar. Doğal dil işleme alanında back-translation ve paraphrasing, özellikle düşük kaynaklı dillerde önemli performans iyileştirmeleri sunmuştur. Finansal zaman serilerinde ise SMOTE ve GAN tabanlı yöntemler, nadir olayların modellenmesine katkı sağlamıştır.

Bir diğer güçlü yön, veri artırmanın düşük maliyetli bir strateji olmasıdır. Gerçek dünyadan yeni veri toplamak çoğu zaman hem maliyetli hem de zaman alıcıdır; ayrıca etik kısıtlamalar da söz konusu olabilir. Buna karşın veri artırma,

mevcut veri üzerinde uygulanarak ek maliyet gerektirmeden daha zengin veri kümeleri oluşturur.

5.2. Sınırlılıklar ve Zayıf Yönler

Her ne kadar veri artırma birçok avantaj sunsa da, yöntemlerin çeşitli sınırlılıkları bulunmaktadır. Öncelikle, artırmaların semantik bütünlüğü bozma riski vardır. Görsel verilerde aşırı rotasyon veya kırpma, nesnenin etiketini değiştirebilir. Zaman serilerinde yanlış ölçekleme, gerçekçi olmayan sinyaller üretebilir. Doğal dil işleme alanında synonym replacement, bağlam uyumsuzluklarına neden olabilir. Finansal zaman serilerinde ise sentetik veriler, piyasanın gerçek dinamiklerinden sapmalara yol açabilir.

İkinci olarak, veri artırmanın etkisi her zaman doğrusal değildir. Yani, daha fazla artırma her zaman daha iyi performans anlamına gelmez. Aksine, aşırı artırma modelin gerçek veriye uyumunu bozabilir ve öğrenme sürecini olumsuz etkileyebilir. Bu durum özellikle GAN tabanlı üretici yöntemlerde sıkça görülmektedir; yüksek miktarda sentetik veri kullanımı, modelin gerçek dünyadan kopmasına neden olabilir.

5.3. Alanlara Göre Farklılıklar

Veri artırma stratejilerinin etkisi, kullanılan veri türüne ve uygulama alanına göre değişiklik göstermektedir. Tıp ve biyomedikal görüntüleme alanında artırmalar büyük fayda sağlasa da, klinik doğrulama zorunluluğu nedeniyle her yeni örneğin uzmanlar tarafından dikkatle incelenmesi gerekir. Otonom araçlarda artırmalar güvenlik açısından kritik olup, simülasyonların gerçek dünyayı ne ölçüde yansıttığı tartışmalıdır. NLP'de artırmaların başarısı, dilin yapısına ve bağlamın korunmasına bağlıdır. Finans alanında ise sentetik verilerin regülasyonel ve etik sorunları bulunmaktadır.

Dolayısıyla veri artırmayı tek bir yöntem seti olarak değil, alan spesifik olarak tasarlanması gereken bir strateji olarak değerlendirmek gerekir.

5.4. Etik ve Metodolojik Sorunlar

Veri artırma uygulamaları yalnızca teknik bir mesele değil, aynı zamanda etik ve metodolojik boyutlar da içermektedir. Tıp alanında yanlış üretilmiş sentetik veriler hasta güvenliğini tehlikeye atabilirken, finans alanında sentetik verilerin regülasyonlara uygunluğu tartışma konusu olmaktadır. Bunun yanında, özellikle hasta verileri veya finansal kayıtlar gibi hassas bilgiler söz konusu olduğunda, sentetik verilerin bireylerin kimliklerini açığa çıkarma riski de bulunmaktadır. Bir diğer önemli sorun ise metodolojik şeffaflıktır; araştırmalarda veri artırma yöntemleri çoğu zaman açık ve ayrıntılı biçimde rapor edilmemekte, bu da

yeniden üretilebilirliği (reproducibility) zayıflatmaktadır. Tüm bu unsurlar, veri artırmanın yalnızca bir mühendislik çözümü değil, aynı zamanda dikkatli etik değerlendirmeler gerektiren bir yaklaşım olduğunu ortaya koymaktadır.

5.5. Gelecek Araştırmalar İçin Çıkarımlar

Veri artırma stratejilerinin geleceği üç temel yönde ilerlemektedir. İlk olarak, artırmaların daha anlamlı ve güvenilir olabilmesi için alan bilgisiyle entegre edilmesi beklenmektedir; örneğin medikal görüntülerde yalnızca doktorların klinik açıdan onayladığı varyasyonların üretilmesi bu yaklaşımın somut bir örneğidir. İkinci olarak, generative AI tabanlı artırmalar giderek daha önemli hale gelmekte, GAN'ler, VAE'ler ve özellikle Diffusion modelleri sayesinde daha gerçekçi ve çeşitlilik arz eden verilerin üretilmesi mümkün olmaktadır. Üçüncü eğilim ise veri artırmanın self-supervised öğrenme ile bütünleştirilmesidir. SimCLR ve BYOL gibi yöntemlerde görüldüğü üzere, artırma artık yalnızca yardımcı bir teknik olmaktan çıkmış, doğrudan öğrenme paradigmasının merkezine yerleşmiştir.

6. SONUÇ VE GELECEK YÖNELİMLER

6.1. Genel Değerlendirme

Bu kitap bölümünde veri artırma stratejilerinin teorik temelleri, metodolojik yaklaşımları ve farklı uygulama alanlarındaki etkileri kapsamlı biçimde ele alınmıştır. İstatistiksel öğrenme teorisi çerçevesinde veri artırmanın bias–variance dengesiyle ilişkisi tartışılmış, regularization yöntemleriyle benzerlikleri ve veri sentezleme ile transfer learning gibi kavramlardan farkları ortaya konmuştur. Görsel, zaman serisi, metin ve multimodal veriler için kullanılan yöntemler sistematik biçimde incelenmiş; basit dönüşümlerden GAN ve Diffusion modelleri gibi ileri yaklaşımlara kadar geniş bir metodolojik çerçeve sunulmuştur. Ayrıca tıp ve biyomedikal görüntüleme, otonom araç sistemleri, doğal dil işleme ve finansal zaman serileri gibi alanlardaki uygulamalar üzerinden veri artırmanın gerçek dünyadaki önemi gösterilmiştir.

Bu kapsamlı inceleme, veri artırmanın yalnızca teknik bir ön işleme yöntemi değil, modern makine öğrenmesinin temel yapı taşlarından biri olduğunu ortaya koymaktadır. Artırmaların güçlü yönleri özellikle küçük ve dengesiz veri setlerinde sağladıkları katkılarda görülürken, zayıf yönleri ise yanlış veya kontrolsüz uygulandığında semantik bozulmalara, etik sorunlara ve güvenlik risklerine yol açabilmesidir. Dolayısıyla, veri artırmanın etkili ve güvenilir biçimde kullanılabilmesi için probleme özgü tasarım, etik değerlendirme ve açık raporlama kritik öneme sahiptir. Geleceğe bakıldığında ise veri artırma

stratejilerinin yalnızca yardımcı bir yöntem olmaktan çıkıp öğrenme sürecinin ayrılmaz bir bileşeni haline geleceği öngörülmektedir.

6.2. Gelecek Yönelimler

Veri artırma araştırmalarının geleceği, teknolojik gelişmeler ve disiplinler arası ihtiyaçlar doğrultusunda dört ana eğilim etrafında şekillenmektedir:

Domain Knowledge ile Entegre Artırmalar: Gelecekte veri artırma tekniklerinin alan bilgisiyle daha fazla bütünleştirilmesi beklenmektedir. Örneğin, medikal görüntülerde yalnızca doktorların klinik olarak anlamlı bulduğu varyasyonların üretilmesi; finansal verilerde ise regülasyonlara uygun senaryoların artırmaya dâhil edilmesi bu yaklaşımın örnekleridir. Böylece artırma stratejileri yalnızca teknik olarak değil, aynı zamanda bilimsel ve etik açıdan da daha güvenilir hale gelecektir.

Generative AI ve Diffusion Modelleri: GAN ve VAE tabanlı yöntemler, veri artırmada yeni ufuklar açmıştır. Ancak son yıllarda Diffusion tabanlı üretici modeller, daha gerçekçi ve çeşitli sentetik veriler üretebilme potansiyeliyle öne çıkmaktadır. Bu modellerin multimodal artırmalarda kullanılmasıyla, aynı anda hem görüntü hem de metin gibi farklı veri türlerinin tutarlı biçimde üretilmesi mümkün hale gelmektedir.

Self-Supervised Öğrenme ile Birlikte Kullanım: Veri artırma, self-supervised öğrenme yöntemlerinin merkezinde yer almaktadır. SimCLR, MoCo ve BYOL gibi yöntemler, aynı verinin farklı artırmalarla oluşturulmuş görünümlerinden öğrenmeye dayanır. Gelecekte, veri artırmanın yalnızca yardımcı bir araç değil, representation learning'in temel bir parçası olarak konumlanacağı öngörülmektedir.

Etik ve Regülasyonel Çerçeveler: Özellikle sağlık ve finans gibi kritik alanlarda, veri artırmanın etik ve yasal boyutları daha fazla önem kazanacaktır. Gelecekte, sentetik verilerin doğrulanması, standardizasyonu ve denetlenmesi için uluslararası kılavuzlar ve regülasyonların geliştirilmesi muhtemeldir.

6.3. Sonuç

Sonuç olarak, veri artırma stratejileri, makine öğrenmesi ve yapay zekâ araştırmalarının vazgeçilmez bir bileşeni haline gelmiştir. Küçük, dengesiz veya etik kısıtlı veri kümeleriyle çalışan araştırmacılar için, bu yöntemler bilimsel reprodüksiyon ve metodolojik güvenilirlik açısından kritik bir araçtır. Gelecekte daha gelişmiş üretici modeller, domain knowledge ile entegre yaklaşımlar ve etik/regülasyonel çerçeveler sayesinde veri artırma yalnızca bir teknik araç değil,

aynı zamanda yapay zekâ sistemlerinin güvenilirliğini ve toplumsal kabulünü artıran temel bir unsur olacaktır.

Kaynakça

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

Cheah, P. C. Y., Yang, Y., & Lee, B. G. (2023). Enhancing Financial Fraud Detection through Addressing Class Imbalance Using Hybrid SMOTE-GAN Techniques. International Journal of Financial Studies, 11(3), 110. https://doi.org/10.3390/ijfs11030110.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. International Conference on Machine Learning (ICML), 1597–1607.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 248–255. DOI: 10.1109/CVPR.2009.5206848

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, 4171–4186.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). CARLA: An open urban driving simulator. Conference on Robot Learning (CoRL), 1–16.

Edunov, S., Ott, M., Auli, M., & Grangier, D. (2018). Understanding back-translation at scale. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 489–500, Brussels, Belgium. Association for Computational Linguistics.

Elreedy, D., & Atiya, A.F. (2019). A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance. Inf. Sci. 505, C (Dec 2019), 32–64. https://doi.org/10.1016/j.ins.2019.07.070

Fadaee, M., Bisazza, A., & Monz, C. (2017). Data augmentation for low-resource neural machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 567–573, Vancouver, Canada. Association for Computational Linguistics.

Feng, S., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 968–988, Online.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: Progress and challenges, Marking the 15-year Anniversary. Journal of Artificial Intelligence Research 61, 863-905.

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). Synthetic data augmentation using GAN for improved liver lesion classification. IEEE International Symposium on Biomedical Imaging (ISBI), https://arxiv.org/abs/1801.02385.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, 2414-2423, doi: 10.1109/CVPR.2016.265.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, 3354-3361, doi: 10.1109/CVPR.2012.6248074.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. https://arxiv.org/abs/1406.2661.

Grill, J. B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukçuoğlu, K., Munos, R., & Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. 34th Conference on Neural Information Processing Systems (NeurIPS), 21271–21284.

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Huang, DS., Zhang, XP., Huang, GB. (eds) Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science, vol 3644. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11538059_91.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. IEEE International Joint Conference on Neural Networks (IJCNN), 1322–1328. doi: 10.1109/IJCNN.2008.4633969.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. NeurIPS, 6840–6851. https://doi.org/10.48550/arXiv.2006.11239

Ileberi, E., Sun, Y., & Wang, Z. (2021). Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost. IEEE Access, vol. 9, 165286-165294, 2021, doi: 10.1109/ACCESS.2021.3134330.

Iwana, B.K., & Uchida, S. (2021) An empirical survey of data augmentation for time series classification with neural networks. PLoS ONE 16(7): e0254841. https://doi.org/10.1371/journal.pone.0254841.

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., & Makedon, F. (2021). A Survey on Contrastive Self-Supervised Learning. Technologies, 9(1), 2. https://doi.org/10.3390/technologies9010002

Kingma, D.P., & Welling, M. (2014). Auto-encoding variational Bayes. https://doi.org/10.48550/arXiv.1312.6114

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

Lewis, P., Denoyer, L., & Riedel, S. (2019). Unsupervised question answering by cloze translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 4896–4910, Florence, Italy.

Li, Y., & Ibanez-Guzman, J. (2020), Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems, In IEEE Signal Processing Magazine, vol. 37, no. 4, pp. 50-61, July 2020, doi: 10.1109/MSP.2020.2973615.

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B., Sánchez, C.I. (2017). A survey on deep learning in medical image analysis. Med Image Anal. 2017 Dec;42:60-88. doi: 10.1016/j.media.2017.07.005. Epub 2017 Jul 26. PMID: 28778026.

Ma, Y., Tian, Y., Moniz, N., & Chawla, N.V. (2025). Class-Imbalanced Learning on Graphs: A Survey. ACM Comput. Surv., Vol. 57, No. 8, Article 207.

Ni, J., Li, J., & McAuley, J. (2019). Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 188–197, Hong Kong, China. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sakaridis, C., Dai, D., & Van Gool, L. (2018). Semantic foggy scene understanding with synthetic data. Int. J. Comput. Vision 126, 9 (September 2018), 973–992. https://doi.org/10.1007/s11263-018-1072-8

Sennrich, R., Haddow, B., & Birch, A. (2016). Improving neural machine translation models with monolingual data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, V.1: Long Papers, 86–96, Berlin, Germany.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(60), 1–48. https://doi.org/10.1186/s40537-019-0197-0

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15, 1929–1958.

Tobin, J., OpenAI Robotics, & Abbeel, P. (2017). Geometry-Aware Neural Rendering 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 6382–6388, Hong Kong, China.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H. (2020). Time Series Data Augmentation for Deep Learning: A Survey. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 4653-4660. https://doi.org/10.24963/ijcai.2021/631.

Xie, Q., Dai, Z., Hovy, E., Luong, M. T., & Le, Q. V. (2020). Unsupervised Data Augmentation for Consistency Training. Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 3237–3249.

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., & Le, Q. V. (2018). QANet: Combining local convolution with global self-attention for reading comprehension. ICLR. https://arxiv.org/abs/1804.09541.

Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., & Yoo, Y. (2019). CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. 10.48550/arXiv.1905.04899.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. ICLR. https://doi.org/10.48550/arXiv.1710.09412

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, V. 64, Issue 3, 107 – 115. https://doi.org/10.1145/3446776

Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random erasing data augmentation. Proceedings of the AAAI Conference on Artificial Intelligence, V. 34(7), 13001–13008. https://doi.org/10.1609/aaai.v34i07.7000

Zhu, L., Bijeljic, B. & Blunt, M.J. (2025). Diffusion Model-Based Generation of Three-Dimensional Multiphase Pore-Scale Images. Transp Porous Med 152, 22. https://doi.org/10.1007/s11242-025-02158-4.


BÖLÜM 3

Denetleyici Alan Ağı (CAN) Temelleri ve STM32F4 Mikrodenetleyicisi ile Programlama Uygulaması

Ali Şentürk [1]

1. Giriş

Teknolojinin gelişmesi ile ulaşım araçlarında kullanılan elektronik kontrol ünitelerinin sayısı önemli ölçüde artmaktadır. Bu ünitelerin haberleşme ihtiyacı için 1986 yılında Robert Bosch tarafından Denetleyici Alan Ağı (Control Area Network - CAN) iletişim protokolü geliştirilmiştir (Voss, 2008). CAN protokolü ulaşım araçlarının yapısından kaynaklanan yüksek seviye elektrik gürültülerine dayanıklı, güvenilir ve yüksek hızlı iletişim yapabilecek şekilde tasarlanmıştır. Günümüzde ISO standartları kapsamında tanımlanan CAN ("ISO 11898-1", 2024), modern araçlarda bulunan motor yönetimi, frenleme, iklimlendirme, multimedya gibi sistemlerin ve çok sayıda sensörlerin hızlı ve güvenilir bir şekilde veri alışverişi yapmasını sağlamaktadır (Avatefipour, Hafeez, & Malik, 2018).

CAN veri iletiminin motor bölmesi gibi yüksek elektromanyetik gürültüye ve olumsuz çevresel koşullara sahip ortamlarda dahi güvenilir bir şekilde çalışabilmesi, birbiri üzerine sarmalanmış iki iletim hattı üzerinden gerçekleştirilen diferansiyel sinyalleme yöntemi ile sağlanmaktadır. Bunun yanı sıra, hata algılama ve veri doğrulama mekanizmaları aracılığıyla iletilen mesajların bütünlüğü ve doğruluğu güvence altına alınmaktadır. Herhangi bir hata oluştuğunda mesajların tekrar gönderilmesi mümkün olmaktadır. Veri iletimi, uygulamanın gereksinimlerine bağlı olarak yapılandırılabilir hızlarda azami 1 Mbit/s değerine kadar yapılandırılabilmektedir (Chen & Tian, 2009).

CAN, SPI veya I^2C gibi düğümden düğüme iletim yapan protokollerden farklı olarak, bir düğüm iletim yaptığında, veri yolundaki tüm düğümler mesajı alırlar. Diğer bir ifade ile CAN, belirli cihazları adreslemek yerine, verinin anlamını ve önceliğini tanımlayan mesaj tanımlayıcıları kullanır. Dolayısıyla tüm düğümler aynı mesajı dinleyebilir, tüm düğümler diğer düğümlerle iletişim halinde olabilir.

[1] Isparta Uygulamalı Bilimler Üniversitesi, Teknoloji Fakültesi,
ORCID: 0000-0002-5868-7365

Bu sistemde önemli olan mesajlar veri yoluna daha öncelikli erişim hakkına sahiptir (Üzülmez, Canan, & Akdemir, 2022).

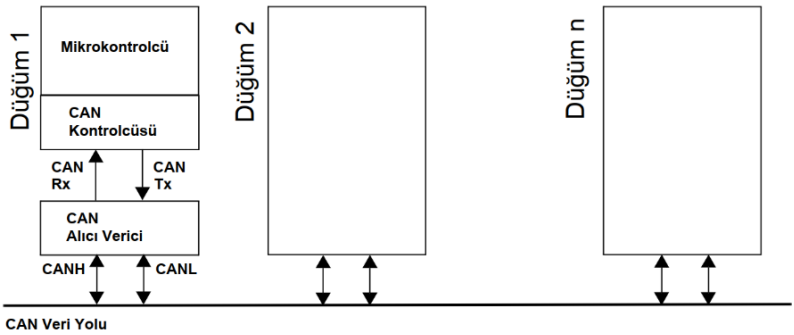
CAN sisteminin bir diğer önemli avantajı, düğümlerin ağ iletişimini kesintiye uğratmadan ağa eklenebilmesi veya ağdan çıkarılabilmesidir. Bu özellik, özellikle bakım ve test senaryolarında önemli bir esneklik sağlar.

Bu bölümün devamında, öncelikle CAN veri yolu yapısı ve diferansiyel sinyalleme ele alınacak, ardından CAN veri formatı ile tanımlayıcı çakışmaları ve çözüm mekanizması incelenecektir. Daha sonraki kısımda ise STM32F4 mikrodenetleyicisinin CAN çevre birimi ayrıntılı biçimde tanıtılarak, GPIO ayarları, başlangıç ve zamanlama ayarları, veri gönderme, tanımlayıcı filtreleme ve veri alma adımları açıklanacaktır. Bölüm, anlatılan konuların genel bir değerlendirmesinin ifade edildiği sonuç kısmı ile tamamlanacaktır.

2. CAN Veri Yolu ve Diferansiyel Sinyalleme

Temel olarak bir CAN düğümü, CAN kontrolcüsü ve CAN alıcı-vericisinden (transceiver) oluşur. CAN kontrolcüsü, veri iletimi için CAN_Tx ve veri alımı için CAN_Rx olmak üzere iki dijital pin kullanır. Bu pinler dijital sinyaller üretirler. Dijital sinyaller gürültüye maruz olan ve uzun veri yolları için uygun olmadığından CAN veri yolunda doğrudan kullanılmazlar. Bundan dolayı CAN kontrolcüsü CAN_Tx ve CAN_Rx pinleri aracılığıyla alıcı-vericiye bağlanır (Voss, 2008).

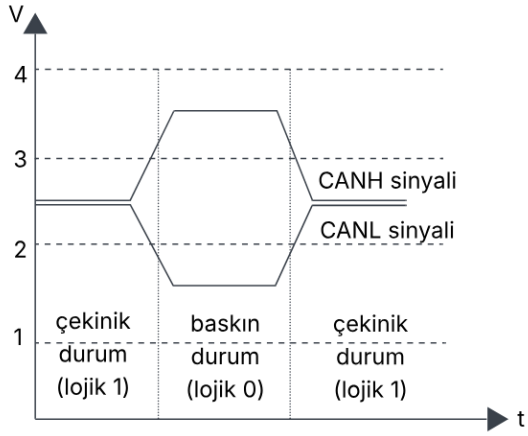
CAN iletişiminde, fiziksel ortam gürültüleri dayanıklı olma ve uzun mesafeli iletişimi için diferansiyel sinyalleme kullanır. Alıcı-verici, CAN_Tx ve CAN_Rx sinyallerini CAN Yüksek (CANH) ve CAN Düşük (CANL) olmak üzere iki diferansiyel sinyale dönüştürülür. Bu iki hat, tüm düğümlerin bağlandığı veri yolunu oluşturur. Veri yolu sinyal bütünlüğünü sağlamak için veri yolunun her iki ucu 120 Ω dirençle sonlandırılır. CAN haberleşme sisteminin genel yapısı Şekil 1'de gösterilmektedir.



Şekil 1. CAN düğümleri ve veri yolu (*RM0090 Reference manual*, 2024, s. 1080)

Daha önce de ifade edildiği üzere CAN veri yolunda diferansiyel sinyalleme yöntemi kullanılır. H ve L hatları arasındaki sinyalin voltaj farkı ölçülerek, sinyalin mantıksal durumu (lojik 1 veya 0) belirlenir. Hatlara elektromanyetik bir gürültü etki etmesi durumunda, her iki hattı da benzer bir şekilde etkileyeceğinden voltaj farkında etkileyen gürültü birbirini iptal eder. Bu özellik CAN veri yolunu gürültüye karşı yüksek dayanıklılık güvenilirlikli iletişim sağlamış olur (Murvay, Popa, & Groza, 2021). CAN veri yolundaki diferansiyel sinyalleme Şekil 2'de gösterilmektedir.

Lojik 1 sinyali iletilmek istendiğinde, CAN veri yolundaki diferansiyel sinyaller çekinik duruma geçer. Bu durumda hem CANH sinyali hem de CANL



Şekil 2. CAN diferansiyel sinyalleri

sinyali yaklaşık 2.5V civarında bir voltaj seviyesinde bulunur, dolayısıyla voltaj farkı yaklaşık 0V olur. Lojik 0 sinyali iletilmek istendiğinde ise, CAN veri yolundaki sinyaller baskın duruma geçiş yapar. CANH sinyali 3.5V civarında bir değer alırken, CANL sinyali ise 1.5V civarında bir voltaj seviyesine iner. Dolayısıyla sinyaller arasında yaklaşık olarak 2V fark oluşur.

Can veri yolunda baskın durumun (lojik 0), çekinik duruma (lojik 1) göre önceliği vardır. Dolayısıyla birden fazla düğüm aynı anda iletim yapmaya çalıştığında, veri yolunda baskın bitler çekinik bitleri geçersiz kılar.

3. CAN Veri Formatı

CAN protokolünde iki temel veri formatı bulunmaktadır: *Standart CAN* (2.0A) ve *Genişletilmiş CAN* (2.0B) ("ISO 11898-1", 2024). Standart CAN, 11 bitlik bir *tanımlayıcı* (identifier) kullanırken; Genişletilmiş CAN, bu 11 bitlik formata ek olarak 18 bit daha içerir ve toplamda 29 bitlik tanımlayıcı ile çalışır. Genişletilmiş CAN denetleyicileri, standart formattaki veri formatlarını da desteklediğinden, her iki cihaz türü aynı ağ üzerinde uyumlu bir şekilde iletişim kurabilir. Bu bölümde önce Standart CAN veri çerçevesi açıklanacak, daha sonra Genişletilmiş CAN veri formatındaki farklılıklar ifade edilecektir.

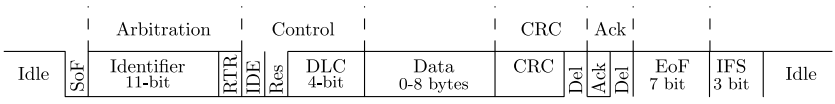
CAN veri yolunda herhangi bir veri iletimi olmadığında veri yolu çekinik durumdadır. Bir veri çerçevesi Çerçeve Başlangıcı (Start of Frame - SoF) adı verilen baskın bit ile başlar. Ardından 11-bit tanımlayıcı (identifier) ve 1 bit Uzak İletim Talebi (Remote Transmission Request - RTR) alanı gelir. Bu 12 bitlik kısım tahkim (arbitration) alanı olarak isimlendirilmiştir. Tahkim alanından sonra bir bit IDE (Identifier Extension) alanı ve 1 bit rezerv alanı yer alır.

Bir sonraki kısım Veri Uzunluk Kodu (Data Length Code - DLC) alanıdır ve 4 bit uzunluğa sahiptir. Ardından gelen Veri (Data) alanı ise DLC değerine bağlı olarak 0 ile 8 byte arasında değişir. Veri alanından sonra, çerçevedeki bilginin bütünlüğünü doğrulamak için kullanılan 15 bitlik CRC (Cyclic Redundancy Check) alanı ve ardından 1 bitlik çekinik CRC sonlayıcısı (delimiter) bulunur.

CRC bölümünü takiben Acknowledge (ACK) alanı gelir. Bu alandaki ACK biti, diğer düğümler tarafından baskın duruma çekilirse, verinin en az bir düğüm tarafından doğru şekilde alındığını gösterir. ACK bitinden sonra ACK sonlayıcısı (delimiter) yer alır.

Veri çerçevesi, 7 bitlik Çerçeve Sonu (End of Frame - EoF) alanı ile sonlanır. Bunun ardından gelen 3 bitlik Çerçeveler Arası Boşluk (Interframe Space - IFS) kısmının tamamlanmasının ardından veri yolu yeniden çekinik duruma geçer. Verinin iletilmediği bu duruma boşta (idle) denir.

Veri çerçevesinin yapısı Şekil 3'te gösterilmiştir. Şekilde veri alanlarının üst kısmının kapalı olması bu alanın lojik 1 (çekinik), alt kısmının kapalı olması ise lojik 0 (baskın) sinyalini temsil etmesi anlamına gelmektedir. Her iki tarafın kapalı olması ise alanın hem lojik 1 hem 0 değerini alabileceğini göstermektedir.



Şekil 3. Standart veri formatı

IDE biti, kullanılan veri formatını belirler. Standart veri formatında bu bit 0 iken, genişletilmiş veri formatında ise lojik 1 ile temsil edilir. Genişletilmiş veri formatında 11 bitlik tanımlayıcı alanına 18 bit daha eklenerek tanımlayıcı alanı 29 bite çıkarılır.

CAN haberleşmesinde veri çerçeveleri (data frame) bilgi iletmek için kullanılırken, uzak çerçeveler (remote frame) diğer düğümlerden bilgi talep etmek için kullanılır. Bir çerçevenin veri çerçevesi veya uzak çerçeve olduğunu Tahkim alanındaki RTR biti belirler. RTR biti 0 olduğunda veri alanı dolu olup çerçeve bir veri çerçevesi işlevi görür. Eğer 1 ise veri alanı boş bırakılır dolayısıyla herhangi bir veri iletimi yapılmaz. Düğüm veri yolunda veri talebinde bulunmaktadır. Bir düğüm, belirli tanımlayıcıya sahip bir uzak çerçeve algıladığında, istenen bilgileri içeren aynı tanımlayıcıya sahip bir veri çerçevesi göndererek talebe yanıt verir.

CAN protokolünde, mesajın başarılı bir şekilde alındığını onaylamak için bir onay mekanizması kullanılır. Alıcı düğüm, CRC ile doğrulanmış mesajda hata tespit etmediğinde, verici tarafından gönderilen çekinik ACK bitini baskın seviyeye çeker. Bu durum, vericiye ağdaki en az bir düğümün mesajı hatasız bir şekilde aldığını bildirir. Eğer verici bunun yerine çekinik bir durum algılarsa, mesaj iletiminin başarısız olduğunu varsayar ve otomatik olarak yeniden iletir. Burada dikkat edilmesi gereken husus, ACK biti verinin hedeflenen düğüme gönderildiğini göstermediğidir. Yalnızca bir veya daha fazla düğümün veri yolundan mesajın hatasız bir şekilde alındığını ifade eder.

Çerçeve, ACK alanından sonra gelen EOF ve IFS kısımları ile tamamlanır. EOF yedi çekinik bitten, IFS ise üç çekinik bitten oluşur. Bu on bitlik süre boyunca veri yolu boşta kalır. Yeni bir iletim başlatmak isteyen düğüm, IFS'nin tamamlanmasını beklemek zorundadır.

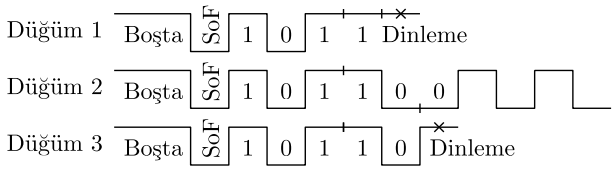
Boşta kalma dönemini hemen takip eden baskın bit olan SoF, veri yolundaki tüm düğümler için senkronizasyon noktası işlevi görür. SoF algılandığında, düğümler iç zamanlamalarını vericiye senkronize eder ve veri iletimi süresince bit düzeyinde koordinasyon sağlanmış olur.

4. CAN Tanımlayıcı Çakışmaları

CAN protokolü, mesaj önceliğine dayalı bir mesaj iletimi mekanizmasına sahiptir. İletimde öncelikle bütün düğümler Çerçeveler Arası Boşluk (IFS) bitlerinin tamamlanmasını bekler. Daha sonraki iletim, Çerçeve Başlangıcı (SoF) biti ile başlar. Birden fazla düğüm aynı anda iletime başlarsa, baskın bitler çekinik bitleri geçersiz kılar. Baskın bitler lojik 0 sinyalini temsil ettiğinden, düşük tanımlayıcı değere sahip mesajların, yüksek tanımlayıcı değere sahip mesajlara göre önceliği olduğu anlaşılır.

Aynı anda iletime başlamış olan düğümlerden biri, başka bir düğümün yüksek öncelikli olduğunu fark ettiğinde veri iletimini durdurur ve dinleme moduna geçer. Böylece sadece en yüksek önceliğe sahip düğüm veri iletimine devam eder ve veri iletimini tamamlar.

Örneğin, 1472, 1429 ve 1740 tanımlayıcı değerlerine sahip üç düğüm aynı anda iletim yapmaya çalışırsa, 1429 tanımlayıcısına sahip düğüm diğerlerinden daha uzun süre baskın durumda kaldığı için veri iletimin önceliğine sahip olur. Mesaj öncelikleri, yani mesajların tanımlayıcı değerleri uygulama tasarımına göre belirlenir. Tanımlayıcı değerine göre önceliklendirme ise CAN denetleyici donanımları tarafından otomatik olarak gerçekleştirilir. Bu sürece ilişkin örnek Şekil 4'te gösterilmiştir.



Şekil 4. Veri iletimi çakışması ve düşük değerli tanımlayıcının diğerlerini baskılaması

5. STM32F4 Mikrodenetleyicisi CAN Çevre Birimi

STM32F4 serisi mikrodenetleyicilerde CAN çevre birimi, basic extended CAN (bxCAN) olarak adlandırılmaktadır. Bu çevre birimi, CAN protokolü 2.0A (standart) ve 2.0B (genişletilmiş) sürümleriyle uyumludur ve hem 11 bitlik hem de 29 bitlik tanımlayıcıları destekler (*RM0090 Reference manual*, 2024).

STM32F4 mikrodenetleyicilerinde genellikle iki adet CAN çevre birimi bulunur. CAN protokolü doğrudan master-slave mantığı ile çalışmasa da donanım mimarisinde CAN1 birincil (master), CAN2 ise ikincil (slave) olarak adlandırılır. Bunun nedeni, CAN1'in tüm ilgili kaynaklara doğrudan erişebilmesi, CAN2'nin ise SRAM gibi bazı kaynaklara CAN1 üzerinden erişebilmesidir (*RM0090 Reference manual*, 2024, s. 1079).

STM32F4 CAN, çevre birimin bağlı olduğu veri yolunun çevrim frekansına bağlı olmak üzere saniyede 1 Mbit/s'ye kadar veri iletimi gerçekleştirebilir. Veri gönderimi için üç adet posta kutusu (mailbox) bulunmaktadır. Ayrıca veri almak için iki adet FIFO yapısında saklama ünitesi mevcuttur. Her FIFO 3 tane CAN mesajını tutabilir. CAN çevre birimi oldukça esnek bir donanımın filtresine de sahiptir. Böylece sadece ilgili mesajların FIFO'lara girmesi sağlanırken, ilgili olmayan mesajlar filtrelenerek işlemcinin gereksiz çalışmasının önüne geçilmiş olur.

CAN çevre birimi, iletim tamamlama, alım ve hata durumları için kesme (interrupt) üretebilmektedir. Bu özellik hem verimli hem de hatalara karşı dayanıklı bir iletişim sürecini mümkün kılar.

Bu bölümde STM32F407 Mikrodenetleyicisinde bulunan CAN çevre biriminin özellikleri, bu özellikleri kullanabilmek için gerekli olan kodlamalar gösterilecektir. Programlamada yazmaç (register) seviyesi programlama yöntemi benimsenmiştir. Programlama esnasında çoğunlukla stm32f407xx.h başlık dosyasında bulunan tanımlamalar kullanılmıştır.

5.1 GPIO Ayarları

CAN çevre biriminin kullanılabilmesi için, veri gönderiminde CAN_Tx ve veri alımında CAN_Rx pinlerinin uygun şekilde yapılandırılması gerekmektedir. STM32F407 mikrodenetleyicisine ait Datasheet belgesinde (Tablo 9: Alternatif Fonksiyon Haritası) bu pinlerin hangi alternatif fonksiyonlara karşılık geldiği gösterilmiştir (*Datasheet*, 2024, ss. 63-71). İlgili tabloda, CAN1 ve CAN2 çevre birimlerinin 9 numaralı alternatif fonksiyon (AF9) altında tanımlandığı görülmektedir.

CAN1 için kullanılabilecek pinler aşağıdaki gibidir:

- CAN1_Rx: PA11, PB8, PD0, PI9
- CAN1_Tx: PA12, PB9, PD1, PH13

Bu çalışmada, PB8 ve PB9 pinleri tercih edilmiştir. Pinlerin ayarlanması için yapılan programlama aşağıda gösterilmiştir:

```
 1. void can_gpio_init(void){
 2.  // GPIOB için çevrim sinyali etkinleştirilir.
 3.  RCC->AHB1ENR |= RCC_AHB1ENR_GPIOBEN;
 4.
 5.  // PB8 ve PB9'u alternatif fonksiyon moduna ayarlanır.
 6.  GPIOB->MODER |= GPIO_MODER_MODER8_1;
 7.  GPIOB->MODER |= GPIO_MODER_MODER9_1;
 8.
 9.  // PB8'in alternatif fonksiyonunu CAN1_RX olarak ayarlanır.
10.  GPIOB->AFR[1] |= GPIO_AFRH_AFSEL8_3 | GPIO_AFRH_AFSEL8_0;
11.  // PB9'un alternatif fonksiyonunu CAN1 TX olarak ayarlanır.
12.  GPIOB->AFR[1] |= GPIO_AFRH_AFSEL9_3 | GPIO_AFRH_AFSEL9_0;
13. }
```

Yapılan kodlamada RCC üzerinden B portunun saat çevrimi sinyali etkinleştirildikten sonra MODER yazmacında B portunun 8 ve 9. pinleri ile ilgili bölgelerin değerleri 0b10 yapılmıştır. Böylece pinlerin modları alternatif fonksiyon olarak ayarlanmıştır. Daha sonra Alternatif Fonksiyon yazmacında (AFR) 8 ve 9. Pinlerle ilgili bölgelere 0b1001 yazılarak, pinlere CAN1_RX ve CAN1_TX 9 numaralı alternatif fonksiyon atanmıştır.

RCC ve GPIO yapıları, yazmaçları ve programlaması bu bölümün doğrudan konusu olmadığından yazmaç yapıları gösterilmemiştir. Konuyla ilgili ayrıntılı bilgi için (Şentürk, 2022) kaynağına başvurulabilir.

5.2 Başlangıç Ayarları

CAN1 çevre birimini APB1 veri yoluna bağlıdır (*Datasheet*, 2024, s. 20). Dolayısıyla CAN1 çevre birimini kullanabilmek için öncelikle RCC_AHB1ENR yazmacından CAN1'in saat sinyalini etkinleştirmek gerekir.

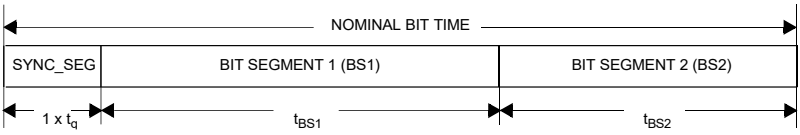
CAN kontrolcüsünü başlatmak için öncelikle veri iletiminin hangi hızda yapılacağının ayarlanması gerekir. Bu ayarlama yalnızca CAN kontrolcüsü başlatma modunda ise yapılabilir. Dolayısıyla öncelikle CAN Ana Kontrol Yazmacının (CAN master control register - CAN_MCR) üzerinde yer alan Başlatma İstem (Initialization request - INRQ) bitinin set edilmesi gerekir (*RM0090 Reference manual*, 2024, s. 1083). CAN kontrolcüsünün başlatma moduna geçtiği CAN ana durum yazmacında (CAN master status register - CAN_MSR) bulunan Başlatma onayı (Initialization acknowledge - INAK) bitinin donanım tarafından set edilmesi ile doğrulanır. Dolaysıyla bu bitin set edilmesinin beklenmesi gerekmektedir.

5.3 Zamanlama Ayarları

CAN birimi veri yolunu izler ve veri iletim başlangıcına göre gönderilecek veriyi senkronize eder ve örnekleme noktalarını belirler. Bir bit zamanı, Şekil 5'te gösterildiği gibi üç temel bölümden oluşur:

- Senkronizasyon segmenti (SYNC_SEG)
- Bit segmenti 1 (BS1)
- Bit segmenti 2 (BS2)

Bu segmentlerin süreleri dolayısıyla veri iletim zamanlaması CAN bit zamanlama yazmacı (CAN bit timing register - CAN_BTR) kullanılarak yapılır [x:1109]. CAN_BTR yazmacındaki ayarlar sadece CAN ünitesi başlatma modunda ise yapılabilir.



Şekil 5. Bit zamanlaması (*RM0090 Reference manual*, 2024, s. 1097)

Şekil 5'te görülen t_q time quantum değerini temsil etmektedir. t_q süresi CAN_BTR yazmacındaki Baud oranı ön ölçekleyicisi (Baud rate prescaler - BRP) alanı ile ayarlanır. t_{PCLK} CAN ünitesinin bağlı olduğu APB1 veri yolunun periyot süresi olmak üzere

$$\mathrm{tq} = (1 + \mathrm{BRP}) \times \mathrm{t_{PCLK}} \qquad (1)$$

şeklinde ifade edilir. t_{BS1} ve t_{BS2} değerleri ise Eşitlik X'de gösterildiği gibi CAN_BTR yazmacındaki TS1 ve TS2 alanları kullanılarak elde edilir:

$$\mathrm{t_{BS1}} = \mathrm{t_q} \times (\mathrm{TS1} + 1), \;\; \mathrm{t_{BS2}} = \mathrm{t_q} \times (\mathrm{TS2} + 1) \qquad (2)$$

Böylece bir bit gönderilmesi için gerekli süre t:

$$\mathrm{t} = \mathrm{t_q} + \mathrm{t_{BS1}} + \mathrm{t_{BS2}} \qquad (3)$$

şeklinde bulunur. Baud oranı 1/t ile veya APB1 veri yolu çalışma frekansının

$$(1 + \mathrm{BRP}) \times (1 + \mathrm{TS1} + 1 + \mathrm{TS2} + 1) \qquad (4)$$

değerine bölünmesi ile elde edilir.

Örneğin STM32F4 Discovery kartında herhangi bir frekans ayarı yapılmadığında çalışma frekansı ve APB1 veri yolunun frekansı 16 MHz'dir. CAN baud oranının 125 Kbps olması için ön ölçekleyici bölgesine (BRP) 7

yazılarak CAN frekansı 2 MHz'e düşürülebilir. BS1 segmenti için TS1 değeri 12 olarak ayarlanır ve BS2 segmenti için TS2 1 olarak ayarlandığında toplam 1+(12+1)+(1+1) olmak üzere bir bit 16 t_q süresinde gönderilir. CAN frekansı 2 MHz olduğundan 2/16 işlemi sonucunda 125 Kbps olarak bulunur. CAN baud oranını 1 Mbps hızında ayarlamak için ise diğer ayarlar aynı kalmak üzere ön ölçekleme alanını 0 olarak bırakmak yeterlidir.

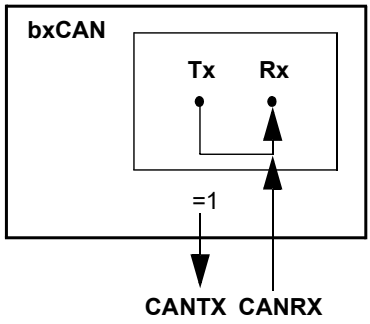
Başlangıç ve zamanlama ayarlarını içeren kodlama aşağıda verilmiştir:

```
 1. void can_init(){
 2.   // CAN1 ünitesinin saat sinyali etkinleştirilir.
 3.   RCC->APB1ENR |= RCC_APB1ENR_CAN1EN;
 4.
 5.   // Başlatma moduna geçiş yapılır.
 6.   CAN1->MCR |= CAN_MCR_INRQ;
 7.   // Uyku modundan çıkılır.
 8.   CAN1->MCR &= ~CAN_MCR_SLEEP;
 9.
10.   // CAN1'in başlatma moduna geçtiği doğrulanır.
11.   while(!(CAN1->MSR & CAN_MSR_INAK));
12.   // CAN1'in uyku modundan çıktığı doğrulanır.
13.   while((CAN1->MSR & CAN_MSR_SLAK));
14.
15.   // Baud rate prescaler (BRP) değeri ayarlanır.
16.   CAN1->BTR |= 7 << CAN_BTR_BRP_Pos;
17.   // Zaman segmenti 1 (TS1) yapılandırılır.
18.   CAN1->BTR &= ~CAN_BTR_TS1;
19.   CAN1->BTR |= 12 << CAN_BTR_TS1_Pos;
20.   // Zaman segmenti 2 (TS2) yapılandırılır.
21.   CAN1->BTR &= ~CAN_BTR_TS2;
22.   CAN1->BTR |= 1 << CAN_BTR_TS2_Pos;
23.
24.   // Otomatik yeniden iletme devre dışı bırakılır.
25.   CAN1->MCR |= CAN_MCR_NART;
26.
27.   // Çevrimsel geri döngü (loop back) modu etkinleştirilir.
28.   CAN1->BTR |= CAN_BTR_LBKM;
29.  }
```

Kodlamada öncelikle CAN1 ünitesinin saat sinyali etkinleştirilmiştir. Başlatma moduna geçmek için uyku modundan da çıkmak gerektiğinden 8. Satırda MCR yazmacındaki SLEEP biti 0 yapılmıştır. 11 ve 13. Satırlarda başlatma moduna geçildiği ve uyku modundan çıkıldığı doğrulanmıştır. 16-22. Satırlarda baud oranı zamanlaması ile ilgili yukarıda da anlatılan ayarlamalar yapılarak, baud oranı 125 Kbps olarak ayarlanmış olur. 25. Satırda CAN_MCR

yazmacında Otomatik yeniden iletim yok (No automatic retransmission - NART) biti devre dışı bırakılmıştır. Böylece veri iletimi sonrası ACK biti set edilmediğinde tekrar verinin gönderilmesinin önüne geçilmiştir.

Bu çalışmada yalnızca tek bir mikrodenetleyici kullanıldığından, döngü geri besleme modu etkinleştirilmiştir. Böylece Şekil 6'da gösterildiği gibi Tx pininden elde edilen sinyal doğrudan Rx pinine yönlendirilmiştir. Dolayısıyla gönderim için üretilen sinyal alıcı pinden okunur. Ayrıca, Rx pininin 3.3V besleme hattına yaklaşık 2.2K direnç üzerinden bağlanması ile, veri iletilmediği durumda çekinik duruma benzer bir koşul sağlanmaktadır.



Şekil 6. Döngü geri besleme modu (*RM0090 Reference manual*, 2024, s. 1085)

5.4 Veri Gönderme

CAN ünitesinde veri iletimi için üç adet posta kutusu (transmit mailbox) bulunmaktadır. Diğer bir ifadeyle 3 tane CAN verisi gönderilmeden önce CAN ünitesinde saklanabilir. Her bir veri çerçevesi için ise 4 tane yazmaç bulunmakta olup toplamda CAN veri gönderiminde verilerin saklanması için 3 posta kutusunda birbirleri ile aynı yapıda olan toplam 12 yazmaç mevcuttur. Bu yazmaçlar doğrudan tanımlanmak yerine stm32f407xx.h dosyasında yapı (struct) veri türü ile gruplanmıştır. Böylece, posta kutuları dizi şeklinde temsil edilerek erişim kolaylaştırılmıştır.

```
 1. typedef struct
 2. {
 3.   __IO uint32_t TIR;  /*!< CAN TX mailbox identifier register */
 4.   __IO uint32_t TDTR; /*!< CAN mailbox data length control and time
stamp register */
 5.   __IO uint32_t TDLR; /*!< CAN mailbox data low register */
 6.   __IO uint32_t TDHR; /*!< CAN mailbox data high register */
 7. } CAN_TxMailBox_TypeDef;
 8.
 9. typedef struct
10. {
11.   ...
12.  CAN_TxMailBox_TypeDef  sTxMailBox[3]; /*!< CAN Tx MailBox*/
13.   ...
14. } CAN_TypeDef;
```

CAN ünitesinde Veri iletimi için aşağıda verilen can_transmit fonksiyonu tanımlanmıştır. Fonksiyonun parametreleri şunlardır:

- std_id: 11 bit standart CAN kimliği,
- data: 0-8 byte uzunluğunda gönderilecek veri dizisi,
- len: 0-8 arasında, gönderilecek veri uzunluğu.

Fonksiyonun çalışma akışı şu şekildedir: öncelikle kullanılabilir bir posta kutusu tespit edilir, ardından çerçeveye ilişkin alanlar ilgili yazmaçlara yazılır ve son adımda iletim başlatılır. Kod satırlarında yapılan işlemler yorumlarla açıklanmıştır.

```
 1. void can_transmit(uint16_t std_id, uint8_t *data, uint8_t len){
 2.  int8_t empty_transmit_mailboxno = -1; // kullanılabilir posta
kutusu numarası
 3.  uint8_t i;
 4.
 5.  // 1. Kullanılabilir (boş) iletim posta kutusu tespit edilir.
 6.  if(CAN1->TSR & CAN_TSR_TME0)
 7.      empty_transmit_mailboxno = 0;
 8.  else if(CAN1->TSR & CAN_TSR_TME1)
 9.      empty_transmit_mailboxno = 1;
10.  else if(CAN1->TSR & CAN_TSR_TME2)
11.      empty_transmit_mailboxno = 2;
12.
13.  // 2. Eğer boş posta kutusu mevcut ise, çerçeve hazırlanır.
14.  if(empty_transmit_mailboxno >= 0){
15.
16.   // --- Tanımlayıcı (Identifier) Ayarları ---
```

```
17.  // Standart kimlik alanı temizlenir ve yeni kimlik atanır.
18. CAN1->sTxMailBox[empty_transmit_mailboxno].TIR &=
                                                    ~CAN_TI0R_STID;
19.  CAN1->sTxMailBox[empty_transmit_mailboxno].TIR |=
                     ((uint32_t)std_id << CAN_TI0R_STID_Pos);
20.
21.   // IDE=0 Standart Kimlik (11-bit) kullanılır.
22.   CAN1->sTxMailBox[empty_transmit_mailboxno].TIR &=
                                                    ~CAN_TI0R_IDE;
23.
24.   // RTR=0 Veri çerçevesi (remote frame değil)
25.   CAN1->sTxMailBox[empty_transmit_mailboxno].TIR &=
                                                     ~CAN_TI0R_RTR;
26.
27.   // --- Veri Uzunluğu ve Kontrol Alanları ---
28.   // Zaman damgası (timestamp) kullanılmaz
29.   CAN1->sTxMailBox[empty_transmit_mailboxno].TDTR &=
                                                    ~CAN_TDT0R_TGT;
30.
31.   // Veri uzunluğu kodu (DLC) ayarlanır
32.   CAN1->sTxMailBox[empty_transmit_mailboxno].TDTR &=
                                                    ~CAN_TDT0R_DLC;
33.   CAN1->sTxMailBox[empty_transmit_mailboxno].TDTR |=
                         ((uint32_t)len << CAN_TDT0R_DLC_Pos);
34.
35.   // --- Veri Alanı (Data Field) ---
36.   // Veri kayıtları temizlenir
37.   CAN1->sTxMailBox[empty_transmit_mailboxno].TDLR = 0;
38.   CAN1->sTxMailBox[empty_transmit_mailboxno].TDHR = 0;
39.
40.   // Veri baytları, ilk 4 byte TDLR, kalan 4 byte TDHR'ye yazılır.
41.   for(i = 0; i < len; i++){
42.    if(i < 4)
43.     CAN1->sTxMailBox[empty_transmit_mailboxno].TDLR |=
                                ((uint32_t)data[i] << (i * 8));
44.    else
45.     CAN1->sTxMailBox[empty_transmit_mailboxno].TDHR |=
                         ((uint32_t)data[i] << ((i - 4) * 8));
46.   }
47.
48.   // --- İletim İsteği ---
49.   // TXRQ biti set edilerek iletim başlatılır
50.   CAN1->sTxMailBox[empty_transmit_mailboxno].TIR |=
                                                     CAN_TI0R_TXRQ;
51.  }
52. }
```

Bu fonksiyon ile standart kimlik (11-bit) kullanan veri çerçeveleri hazırlanmakta ve iletim başlatılmaktadır. Gönderim öncesi boş posta kutusu bulunması zorunludur; aksi durumda veri yazma işlemi gerçekleştirilmemektedir.

5.5 Tanımlayıcı Filtreleme

CAN veri yolundaki tüm mesajlar düğümlerin hepsine iletilir. Bu durum, mikrodenetleyicilerin her mesajı işlemek zorunda kalmasına ve dolayısıyla işlemci üzerinde ciddi bir yük oluşmasına yol açar. Bundan dolayı STM32F4 CAN ünitelerinde filtreleme birimi bulunmaktadır. Yani CAN_RX girişi ile veri alma FIFO'ları arasında CAN verisinin tanımlayıcı (identifier) alanı üzerinde filtreleme yapılmasını sağlar. Gelen veri çerçeveleri, yapılandırılmış filtre kurallarıyla karşılaştırılır ve yalnızca kurallara uyan çerçeveler FIFO belleğinde saklanır; uymayan çerçeveler göz ardı edilir.

STM32F4 mikrodenetleyicilerinde CAN ünitesi, programlanabilen 28 tane 32 bitlik yazmaç çiftinden oluşan filtre bankası sağlar. Bu filtreler CAN filtre ana yazmacındaki (CAN filter master register - CAN_FMR) CAN2SB alanındaki değer ile CAN1 ve CAN2 üniteleri tarafından paylaşılır (*RM0090 Reference manual*, 2024, s.1117). CAN2SB alanına yazılan değer ve daha sonraki numaralı filtreler CAN 2'ye, diğer filtreler CAN1'e atanır. Örneğin CAN2SB bölümüne 28 yazıldığında tüm filtreler CAN1'e atanmış olur. Filtrelerle ilgili ayarlamalar yapılırken CAN_FMR yazmacındaki FINIT biti 1 yapılmalıdır. Ayarlamalardan sonra bu bit 0 yapılarak filtreler etkinleştirilmelidir.

Filtre bankası yazmaçları kullanılarak, tam eşleşme (tanımlayıcı liste modu) veya tanımlayıcıların belirli bitlerini sağlayan kısmi eşleşmelerin (maskeleme modu) FIFO'ya yazılması sağlanabilir. Ayrıca sadece standart veya sadece genişletilmiş veri formatlarına izin verme gibi filtreleme kuralları belirlenebilir. Böylece filtreleme sistemi sayesinde gereksiz CPU yükünü azaltır ve yalnızca ilgili CAN mesajlarının mikrodenetleyiciyi tetiklemesi sağlanır.

Maskeleme Modu

CAN filtre modu yazmacı (CAN filter mode register - CAN_FM1R) kullanılarak, filtrelerin maskeleme modu veya tanımlayıcı liste modunda çalışması belirlenir (*RM0090 Reference manual*, 2024, s. 1118).

Maskeleme modunda tanımlayıcı (identifier) yazmaçları, maske yazmaçları ile ilişkilendirilir. Maske yazmaçları, tanımlayıcı bitlerinden hangilerinin "tam eşleşme" olarak değerlendirileceğini, hangilerinin ise "önemsiz" (don't care) kabul edileceğini belirler. Bu yöntem sayesinde, belirli bir bit grubuna sahip olan geniş bir tanımlayıcı aralığı tek bir filtre ile kabul edilebilir. Örneğin, yalnızca ilk

birkaç biti aynı olan ancak kalan bitleri farklılık gösteren tanımlayıcıların filtrelenmesi gerektiğinde maskeleme modu kullanılır.

Diyelim ki on bir bit standart tanımlayıcı kullanan bir CAN ağı var. İlk 3 biti 101 olan tüm tanımlayıcıları kabul etmek istiyoruz. Bu durumda filtreleme şu şekilde yapılabilir:

Filtre Tanımlayıcı: 10100000000

Maske: 11100000000

Maske yazmacındaki 111 bitlerine karşılık gelen pozisyondaki bitlerin mesajda 101 şeklinde olması beklenirken, Maske yazmacındaki 0 olan pozisyondaki bitler 1 veya 0 olabilir.

Tanımlayıcı Liste Modu

Tanımlayıcı liste modunda maske yazmaçları da tanımlayıcı yazmaçlarla aynı işleve sahip olur. Böylece, bir filtre bankası içerisinde iki farklı tanımlayıcı tanımlanabilir kullanılabilir ve tanımlayıcıların sayısı iki katına çıkmış olur. Bu modda, gelen veri çerçevelerinin tanımlayıcı bitlerinin tümü, filtre yazmaçlarında belirtilen tanımlayıcılarla tam olarak eşleşmesi gerekir. Bu nedenle liste modu, yalnızca belirli tanımlayıcıların kabul edilmesi gereken uygulamalarda tercih edilir.

Filtre Ölçeklendirme

Filtreler standart veri formatında gönderildiğinde tanımlayıcı 11 bit olduğu için filtre yazmaçlarının yarısı filtreleme için yeterli olur. Bundan dolayı filtreler 16 bit veya 32 bit olarak yapılandırılabilir. Bu işlem için CAN fitre ölçek yazmacı (CAN filter scale register - CAN_FS1R) kullanılır. Bu yazmaçta Filtre ölçeği yapılandırması FSC) 0 olduğunda filtre 16 bit, 1 yapılırsa filtre 32 bit olarak ayarlanır (*RM0090 Reference manual*, 2024, s. 1118).

Filtreler 16 bit olarak ayarlanırsa:

- Maskeleme modunda filtre yazmaçlarının düşük değerlikli yarısı tanımlama için, yüksek değerlikli yarısı ise maskeleme için kullanılır.

- Tanımlayıcı liste modunda ise düşük ve yükse değerlikli yarıları tanımlayıcıları filtrelemek için kullanılır.

Filtre FIFO Ataması ve Filtre Önceliği

Kullanılacak filtrelerin aktivasyonu CAN fitre aktivasyon yazmacı (CAN filter activation register - CAN_FA1R) kullanılarak ayarlanır (*RM0090*

Reference manual, 2024, s. 1119). Filtrelerin FIFO0 veya FIFO1'e atanması ise FIFO atama yazmacı (FIFO assignment register - CAN_FFA1R) ile yapılır. Yapılan filtreleme kurallarına göre bir tanımlayıcı birden fazla filtre için uygun olabilir. Bu durumda:

- 32 bit filtreler, 16 bit fitrelere göre,
- Tanımlayıcı list modu ile yapılmış filtreler maskeleme modu ile yapılmışlara göre,
- Filtre sırası daha düşük olanalar, yüksek olanlara göre

daha önceliklidir. Bu nedenle kullanılmayan filtrelerin etkin bırakılmaması gerekir.

Yukarıda belirtilen yapılandırmalar sayesinde, mesajların FIFO0 veya FIFO1'e atanması sağlanabilir ve filtreleme mekanizması esnek bir şekilde kullanılabilir. Netice olarak CAN veri akışının sadece mikrodenetleyici için gerekli olan mesajların iletilmesi sağlanır. CAN ünitesinde veri alabilmek için en az bir filtrenin yapılandırılmış olması gerekmektedir. Böylece alınan veriler ilgili FIFO'ya yazılabilir.

STM32'de Filtre Yapılandırması

STM32F407xx mikrodenetleyicileri için stm32f407xx.h dosyasında filtre bankası yazmaçları için CAN_FilterRegister_TypeDef yapı türü tanımlanmıştır. Bu yapı türünün FR1 ve FR2 bileşenleri tanımlayıcı ve maskeleme yazmaçları için kullanılır. CAN yapı türünde ise CAN_FilterRegister_TypeDef türünden 28 boyutlu bir sFilterRegister dizisi tanımlanmıştır. Böylece fitreleme yazmaçlarına dizi kullanılarak ulaşılmaktadır.

Aşağıda, CAN1 için filtre 0'ın tüm mesajları kabul edecek şekilde yapılandırılmasını gerçekleştiren örnek fonksiyon verilmiştir. Burada maske yazmacının tüm bitleri '0' olarak ayarlanmış ve bu sayede tanımlayıcı ayrımı yapılmaksızın tüm mesajların kabulü sağlanmıştır:

```
 1. void can_filter_config(void)
 2. {
 3.  // 1. Filtre başlatma moduna geçilir.
 4.  CAN1->FMR |= CAN_FMR_FINIT;
 5.
 6.  // 2. Filtre 0, FIFO0'a yönlendirilir.
 7.  CAN1->FFA1R &= ~CAN_FFA1R_FFA0;
 8.
 9.  // 3. Filtre 0 maskeleme modunda ayarlanır.
10.  CAN1->FM1R &= ~CAN_FM1R_FBM0;
11.
12.  // 4. Filtre 0, 32-bit ölçeklendirme modunda yapılandırılır
13.  CAN1->FS1R |= CAN_FS1R_FSC0;
14.
15.  // 5. Filtre kimliği ve maskesi ayarlanır.
16.  // FR1 (tanımlayıcı) değerleri sıfır yapılır.
17.  // FR2 (maske) tüm bitleri "0" olduğundan, herhangi bir filtreleme
yapmadan tüm verileri tanılayıcıları kabul eder.
18.  CAN1->sFilterRegister[0].FR1 = 0x00000000;
19.  CAN1->sFilterRegister[0].FR2 = 0x00000000;
20.
21.  // 6. Filtre 0 etkinleştirilir.
22.  CAN1->FA1R |= CAN_FA1R_FACT0;
23.
24.  // 7. Filtre başlatma modundan çıkılır.
25.  CAN1->FMR &= ~CAN_FMR_FINIT;
26. }
```

5.6 Veri Alma

STM32F4 mikrodenetleyicisindeki CAN üniteleri, veri almada kullanılmak üzere iki FIFO yapısı içerir ve her bir FIFO'da iki adet posta kutusu bulunmaktadır. Gelen mesajlar filtreleme sonucunda ilgili FIFO'ya aktarılır. Yeni bir mesaj alındığında kesme tetiklenebilir. Aynı zamanda FIFO'da bekleyen mesaj sayısı CAN alıcı FIFO yazmaçlarının (CAN1 receive FIFOx register - CAN_RFxR) FMP bölgesine donanım tarafından yazılır. FIFO'dan bir seferde sadece en eski mesaj okunabilir. Bu okuma tamamlandıktan sonra CAN_RFxR yazmacının RFOMx biti set edilerek FIFO'daki en eski mesaj FIFO'dan çıkarılır. Çıkarılma işlemi tamamladıktan sonra bu bit donanım tarafından otomatik olarak temizlenir (*RM0090 Reference manual*, 2024, ss. 1114-1117).

CAN veri çerçevesindeki mesaj parçalarının fazla oluşu ve bir FIFO'da 3 posta kutusu olmasından dolayı gelen mesaj verilerini saklamak için bir yapı türü oluşturulabilir. Her ne kadar kesme ile her veri geldiğinde gelen mesajların okunması mümkün olsa da, mikrodenetleyicinin meşgul olması ve çok sık ve yüksek baud oranında mesaj gelmesi gibi durumlarda FIFO'da birden fazla mesaj birikebilir. Bu nedenle, FIFO'daki mesaj sayısı kadar döngüyle işlem yapılarak,

oluşturulan yapı türünden bir dizi aracılığıyla tüm mesajların saklanması mümkün olur. Aşağıda, alınan mesajları saklamak için tanımlanan yapı türü ve dizi örneği verilmiştir:

```
typedef struct
{
 uint16_t StdId;  // Standart kimlik (11-bit)
 uint8_t IDE;     // Kimlik türü (0: Standart, 1: Genişletilmiş)
 uint8_t RTR;     // Uzaktan iletim isteği.
 uint8_t DLC;     // Veri uzunluk kodu (0-8 byte)
 uint8_t Data[8]; // Veri alanı (en fazla 8 byte)
}CanRxMsg_TypeDef;

CanRxMsg_TypeDef rx_msgs[3]; // Alınan mesajların saklanacağı dizi
uint8_t rx_count;            // Alınan toplam mesaj sayısı
```

Bu tanımlamalar global değişken olarak yapıldığından, FIFO'da bulunan mesajların alınması için yazılmış olan can_get_messages fonksiyonunda doğrudan kullanılabilmektedir. Fonksiyonun işleyişi aşağıda verilmiştir:

```
void can_get_messages(){
 uint32_t RIR, RDTR, RDLR, RDHR;
 uint8_t i=0, j;

 // FIFO0'da bekleyen mesaj sayısı okunur.
 uint8_t pending = (CAN1->RF0R & CAN_RF0R_FMP0) >> CAN_RF0R_FMP0_Pos;

 // Bekleyen mesajlar kadar döngü başlatılır.
 while (pending > 0){
  // FIFO0'daki en eski mesajın kayıtlarını okunur.
  RIR  = CAN1->sFIFOMailBox[0].RIR;// Kimlik ve kontrol bilgileri.
  RDTR = CAN1->sFIFOMailBox[0].RDTR;   // Veri uzunluğu ve zaman damgası.
  RDLR = CAN1->sFIFOMailBox[0].RDLR;  // İlk 4 byte veri.
  RDHR = CAN1->sFIFOMailBox[0].RDHR;  // Sonraki 4 byte veri.

  // Standart kimlik bilgisi çıkarılır.
  rx_msgs[i].StdId = (RIR & CAN_RI0R_STID) >> CAN_RI0R_STID_Pos;
  // Kimlik türü (IDE biti) okunur.
  rx_msgs[i].IDE = (RIR & CAN_RI0R_IDE) >> CAN_RI0R_IDE_Pos;
  // RTR biti okunur (veri çerçevesi mi, uzak çerçeve mi?).
  rx_msgs[i].RTR = (RIR & CAN_RI0R_RTR) >> CAN_RI0R_RTR_Pos;
  // Veri uzunluğu kodu (DLC) okunur.
  rx_msgs[i].DLC = (RDTR & CAN_RDT0R_DLC) >> CAN_RDT0R_DLC_Pos;

  // Veri alanı bayt bayt okunur.
  for(j=0; j<rx_msgs[i].DLC; j++){
   // İlk 4 byte, düşük veri yazmacından (RDLR).
   if(j < 4)
```

```
29.       rx_msgs[i].Data[j] = 0xFF & (RDLR >> (8 * j));
30.     // Sonraki byte'lar, yüksek veri yazmacından (RDHR).
31.     else
32.       rx_msgs[i].Data[j] = 0xFF & (RDHR >> (8 * (j-4)));
33.    }
34.
35.    i++; // Bir sonraki mesaj için index artırılır.
36.
37.    // Okunan mesaj FIFO0'da çıkarılır.
38.    CAN1->RF0R |= CAN_RF0R_RFOM0;
39.
40.    // Bekleyen mesaj sayısı bir azaltılır.
41.    pending--;
42.   }
43.
44.   // Toplam alınan mesaj sayısı
45.   rx_count = i;
46. }
```

Bu fonksiyon sayesinde FIFO0'da bulunan tüm mesajlar sırasıyla okunmakta ve önceden tanımlanmış olan rx_msgs dizisine aktarılmaktadır. Böylece mikrodenetleyici, FIFO'daki mesajları güvenli bir biçimde işleyebilmekte ve sistemin kesintisiz veri akışı içerisinde kararlılığı sağlanmaktadır.

6. Sonuç

Bu bölümde, Denetleyici Alan Ağı (Controller Area Network - CAN) protokolünün temel yapısı ve işleyiş mekanizmaları ayrıntılı olarak ele alınmış, ardından STM32F4 mikrodenetleyicisi üzerinde uygulama adımları anlatılmıştır. İlk kısımda, CAN veri yolu mimarisi, diferansiyel sinyalleme prensipleri, veri formatı ve tanımlayıcı çakışmalarının çözüm yöntemleri açıklanarak protokolün güvenilir iletişim sağlayan yapısı vurgulanmıştır.

İkinci kısımda ise STM32F4 mikrodenetleyicisinin CAN çevre birimi üzerinde durulmuş, donanım temelli ayarlamalar (GPIO, başlangıç ve zamanlama konfigürasyonları) ile yazılım tarafında örnek başlangıç kodları verilmiştir. Ayrıca veri gönderme, mesaj filtreleme ve veri alma işlemleri uygulama düzeyinde gösterilmiştir.

Genel olarak bu bölüm, CAN protokolünün teorik temelleri ile mikrodenetleyici üzerinde gerçekleştirilen pratik uygulamaları bir arada sunarak okuyucuya hem kavramsal hem de deneysel bir bakış açısı kazandırmayı amaçlamaktadır. Böylelikle, endüstriyel haberleşme sistemlerinde CAN protokolünü kullanmak isteyen araştırmacı ve uygulayıcılar için hem altyapı bilgisi hem de doğrudan uygulanabilir yöntemler izah edilmiştir.

Teşekkür

Bu bölümde kullanılan figürlerin bazıları STMicroelectronics firmasının teknik dokümanlarından alınmıştır. Bu figürlerin kullanılmasına izin veren STMicroelectronics firmasına teşekkür ederiz.

KAYNAKLAR

Avatefipour, O., Hafeez, A., & Malik, H. (2018, Ocak 25). *Linking Received Packet to the Transmitter Through Physical-Fingerprinting of Controller Area Network*. doi: 10.1109/WIFS.2017.8267643

Chen, H., & Tian, J. (2009). Research on the Controller Area Network. *2009 International Conference on Networking and Digital Society*, *2*, 251-254. doi: 10.1109/ICNDS.2009.142

ISO 11898-1:2024. (2024). Geliş tarihi 12 Ağustos 2025, gönderen ISO website: https://www.iso.org/standard/86384.html

Murvay, P.-S., Popa, L., & Groza, B. (2021). Securing the controller area network with covert voltage channels. *International Journal of Information Security*, *20*(6), 817-831. doi: 10.1007/s10207-020-00532-5

RM0090 Reference Manual STM32F407 Advanced Arm-based 32-bit MCUs. (2024). STMicroelectronics.

STM32F407xx Datasheet—Production data. (2024). STMicroelectronics.

Şentürk, A. (2022). *STM32F4 Discovery Kartı ile ARM Mikrokontrolcü Programlama*. Nobel Akademik Yayıncılık.

Üzülmez, H., Canan, S., & Akdemir, B. (2022). CANBUS TEMELLİ ENDÜSTRİYEL SENSÖR AĞI TASARIMI. *Konya Journal of Engineering Sciences*, *10*, 11-26. doi: 10.36306/konjes.1083364

Voss, W. (2008). *A Comprehensible Guide to Controller Area Network*. Copperhill Media.


BÖLÜM 4

Geniş Alan Mikroskobik Görüntülemede Serpentine Tarama Yönteminin Zamanlama ve Performans Analizi

Cihan Yalçın[1]

1. Giriş

Mikroskobik görüntüleme, biyomedikal araştırmalardan malzeme bilimine kadar pek çok alanda vazgeçilmez bir araçtır. Ne var ki yüksek büyütme oranları kullanıldığında daralan görüş alanı, tüm numunenin yüksek detaylarla incelenmesi gereken durumlarda önemli bir kısıt oluşturur. Örneğin, büyük bir doku kesitinin veya kan gibi geniş alanlara yayılmış örneklerin tek bir mikroskop görüntüsünde yer alması mümkün değildir. Bu sorunu çözmek için oluşturulan görüntü birleştirme yöntemleri, çeşitli görüntülerin geometrik ve fotometrik sürekliliğini koruyarak bir araya getirilmesine olanak tanımaktadır. Bu şekilde, dar görüş alanına sahip bir mikroskop kullanılsa bile tüm örneği kapsayan yüksek çözünürlüklü bir birleşik görüntü elde etmek mümkündür. Bu çalışmanın ana motivasyonu, edinim süresini azaltan ve mekanik hataları en aza indiren bir tarama sisteminin kapsamlı bir yazılım işlem hattı ile entegrasyonunun, yüksek verimli sonuçlar elde edilebileceğini göstermektir.

1.1. Yılanvari (Serpentine) Tarama ve Özgün Katkılar

Geleneksel satır satır tarama yöntemlerinde her satırın sonunda tarama cihazının bir sonraki satırın başına dönmesi için "geri dönüş" (flyback) adı verilen bir ölü süre gerekmektedir. Bu geri dönüş hareketi mekanik stres ve cihazda yük birikimine yol açarak konumlama hatalarını artırabilmektedir. Devamlı serpentine (yılanvari) tarama deseni ise bu ölü süreyi ortadan kaldırarak saniye başına kare (frame per second) oranını ciddi ölçüde arttırır ve hareket sürekliliği sayesinde tarama sırasında daha akıcı geçişler sağlar (Ortega ve ark., 2021). Serpentine tarama düzeninde, görüş alanı içerisindeki ardışık kareler satırlar boyunca ileri ve geri yönlerde alınır; her satırın sonunda yalnızca bir sonraki satıra geçmek için küçük bir dikey hareket yapılması yeterlidir. Bu yöntem kullanılan cihazdaki yük ve geri dönüş gecikmelerini azaltarak genel veri toplama süresini önemli ölçüde kısaltmaktadır.

[1] Dr.,

Bu çalışmada serpentine tarama yönteminin özel olarak optimize edilmiş bir yazılım süreciyle entegrasyonunun zamanlama ve birleşik görüntü kalitesi üzerindeki etkileri kapsamlı bir şekilde değerlendirilmiştir. Bu çalışmanın özgün katkısı serpentine taramanın teorik faydalarını vurgulamakla kalmayıp, aynı zamanda pratik uygulamalarda da tekrarlanabilir ve öngörülebilir kazançlar sağladığını somut verilerle ortaya koymasıdır. Elde edilen bulgular ile entegre sistemin her bir unsuru olan tarama donanımı, eşleştirme algoritmaları, dönüşüm modelleme yöntemleri ve harmanlama teknikleri gibi bileşenler üzerinde kapsamlı bir inceleme yapmaktadır. Bu şekilde fiziksel ve yazılımsal iyileştirmelerin birlikte çalışmasının genel sistem performansı üzerindeki önemli etkisi açıkça gösterilmiştir. Bu bütünleyici yöntemle geniş alan mikroskobik görüntülemede elde edilen yüksek performansın sadece bireysel algoritmaların etkinliğine değil, aynı zamanda tarama ekipmanıyla yazılım süreçlerinin tam bir uyum içinde yönetilmesine dayandığı kanıtlanmıştır.

1.2. Literatür İncelemesi ve Bağlam

Görüntü birleştirme süreci üç temel aşama olan kayıt, kalibrasyon ve harmanlama aşamalarından oluşmaktadır. Kayıt aşamasında örtüşen görüntüler arasındaki dönüşüm parametreleri saptanmaktadır. Kalibrasyon ise bu parametrelerin doğruluğunu artırmayı hedeflemektedir. Harmanlama ise birleştirilen görüntülerdeki parlaklık ve renk farklılıklarını ortadan kaldırarak pürüzsüz bir geçiş oluşturmaktadır. Bu süreç bileşenler aracılığıyla sürekli olarak birbirini optimize eden döngüsel bir mekanizma şeklinde çalışmaktadır (Szeliski, 2022). Hareket kontrollü bir mikroskop düzeneklerinde fiziksel tarama ve veri edinim koşulları sabit kaldığından, özellik tabanlı yöntemler hem hesaplama maliyetleri açısından hem de elde edilen yüksek doğruluklarıyla en uygun ve pratik çözümleri sağlamaktadır (Soltanpour ve Joslin, 2023). Diğer yandan son yıllarda derin öğrenme temelli yöntemler görüntü birleştirme konusunda dikkate değer gelişmeler kaydedilmiştir. Bu yaklaşımlar özellikle geleneksel tekniklerin zorlandığı karmaşık ve dokusal farklılıkları barındıran örneklerde yüksek başarı oranları sergilemiştir (Soltanpour ve Joslin, 2023). Bununla birlikte hareket kontrolü sağlanan bir mikroskop düzeninde sabit koşullar altında geleneksel özellik tabanlı yöntemler hem maliyet hem de doğruluk açısından hala en etkili ve pratik çözümü sunmaktadır (Sharma vd., 2023). Literatürde görüntü birleştirme işlemlerini ele alan çeşitli yöntemler bulunmaktadır. SIFT, SURF ve ORB gibi geleneksel özellik çıkarımına dayalı algoritmalar, görüntülerdeki belirgin noktaları tespit ederek bu noktalar aracılığıyla eşleştirmeler yapmaktadır. Bu yöntemler farklı perspektiflerden veya lokasyonlardan elde edilen görüntüler için oldukça güvenilir sonuçlar sunmaktadır. Hareket kontrollü mikroskopi sisteminde fiziksel tarama koşulları belirli ve tekrar edilebilir olduğundan bu araştırmada hem hesaplama verimliliği hem de doğruluk yönünden klasik özellik

tabanlı yöntemler tercih edilmiştir. Bu şekilde mevcut yöntemler arasında mikroskobik görüntüleme için en uygun yaklaşım belirlenerek işlem hattının her aşamasında yüksek başarı hedeflenmiştir. Bu çalışma mikroskobik görüntüleme alanında en etkili yöntemleri kullanarak tüm işlem sürecinin verimliliğini artırmayı hedeflemektedir.

1.3. Serpentine Tarama ve Bu Çalışmanın Katkıları

Serpentine tarama yöntemi ile geleneksel tarama düzenine kıyasla daha verimli bir veri edinim sırası sunmaktadır. Geleneksel taramada her satır tamamlandığında motorize tabla tamamen geri döner ve yeni satırın başına konumlanmaktadır. Bu hareket ile her satır sonunda önemli bir geri dönüş hareketi ve buna bağlı olarak konum hatası riskine yol açmaktadır. Serpentine tarama ile geleneksel taramada her satır sonunda tarama kafasının başa dönmesi için "flyback" adı verilen bir ölü süre gerekmektedir. Serpentine tarama deseni ise bu ölü zamanı ortadan kaldırarak kare/saniye oranını artırır ancak ters yönde taramanın neden olabileceği hataları gidermek için ek kalibrasyon gerektirmektedir (Ortega ve ark., 2021). Serpentine tarama ise görüş alanı parçalarını satırlar boyunca ileri geri izleyerek, her satır sonunda sadece bir sonraki satıra geçiş için gerekli küçük bir dikey hareket yapar. Bu yaklaşım, mekanik stresi azaltırken, hareket geri dönüş gecikmelerini minimize eder ve veri edinim süresini önemli ölçüde kısaltmaktadır.

Serpentine tarama yönteminin optimize edilmiş bir yazılım işlem sırası ile birleştirilerek, zamanlama ve nihai birleşik görüntünün kalitesi üzerindeki etkisini pratikte doğrulanmasına odaklanılmıştır. Elde edilen sonuçlar serpentine taramanın yalnızca yeni bir yöntem olmadığını, aynı zamanda uygun yazılım algoritmaları ile bir araya geldiğinde tutarlı ve öngörülebilir performans artışları sunduğunu göstermektedir. Çalışmanın en önemli katkıları arasında entegre sistem mimarisinin donanım, yazılım algoritmaları ve veri akışı gibi unsurlarını detaylı bir şekilde inceleyerek hangi bileşenlerin darboğaz oluşturduğunu tespit etmek yer almaktadır. Bu tespitler doğrultusunda optimize edilmiş çözümler sunarak yüksek hızlı ve güvenilir bir geniş alan görüntüleme sistemi geliştirmek amaçlanmaktadır.

2. Sistemin Tanımı ve Tarama Düzeni

Çalışmada kullanılan sistemin donanım ve mekanik yapısı detaylandırılmakta olup uygulanan serpentine tarama prensibi ve geometrisi açıklanmaktadır. Geniş alan mikroskobik görüntülemede veri ediniminin verimliliğini artıran bu temel unsurlar, elde edilen sonuçların anlaşılması için kritik öneme sahiptir.

2.1. Donanım ve Mekanik Yapı

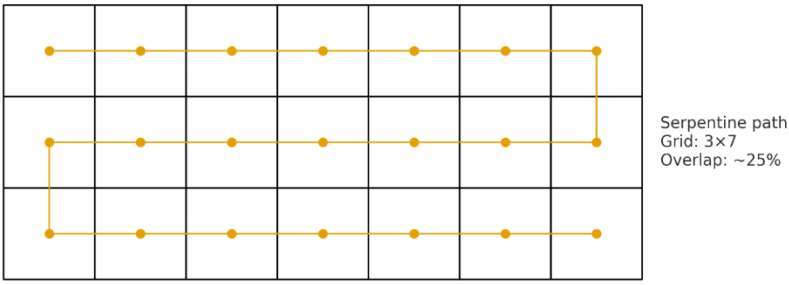
Gerçekleştirilen çalışmada alınan numunenin hassas bir şekilde konumlandırılması için çok eksenli, hareketli bir mikroskop tablası kullanılmıştır. Bu tabla sistemlerinde bir görüntünün alınabilmesi için sadece hızlı hareket etmek yeterli değildir. Aynı zamanda hareketin sonunda oluşan titreşimlerin sönümlenerek tablanın hedef konum penceresi içinde kararlı bir şekilde durması da kritik önemdedir. Hareket süresi komuta edilen hareket yörüngesi, mevcut tork ve yük ataleti gibi faktörlere bağlıdır. Sönümleme (settling) süresi ise tabla hareket durduktan sonra sistemin titreşimden arınarak hedef konum penceresi içinde kararlı hale gelmesi için geçen süreyi ifade ederken milisaniye sürelerinde bile olsa toplam edinim süresinin önemli bir parçasıdır. Geleneksel tarama yöntemlerinde her satırın başında ve sonunda yaşanan tam geri dönüş hareketleri, bu sönümleme sürelerini artırarak toplam süreyi uzatabilmektedir. Serpentine tarama düzeni hareketin devamlılığını koruyarak bu gecikmeleri minimize etmeyi hedeflemektedir. Bu çalışmanın ilerleyen bölümlerinde sönümleme süresinin toplam edinim süresi içindeki payının ayrı bir bileşen olarak incelenmesi, sistemin mühendislik kalitesinin ve performans analizine verilen önemin bir göstergesidir.

2.2. Yılanvari (Serpentine) Tarama Prensibi ve Geometri

Serpentine tarama adından da anlaşılacağı üzere bir yılanın kıvrımlı hareketine benzer bir düzen izlemektedir. Bu yaklaşımda görüntü yakalama işlemi sırasında her satırın sonunda hareket yönü tersine çevrilmektedir. Böylece bir satırın bitiş noktası sonraki satırın başlangıcını oluşturmaktadır. Arada yalnızca küçük bir dikey geçiş gerçekleştiği için hareket daha yumuşak ve hızlı ilerlemektedir. Bu sayede tam geri dönüş hareketlerinden kaynaklanan mekanik zorlanmalar en aza indirilmekte ve sönümleme süreleri ise önemli ölçüde kısalmaktadır.

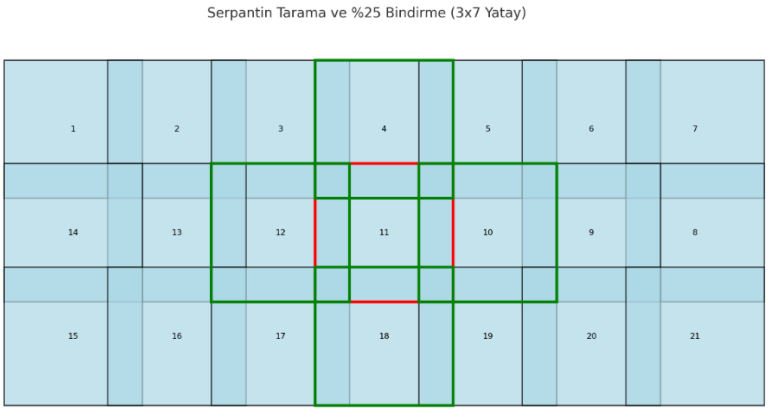
Görüntüler komşu parçalar arasında belirlenmiş bir bindirme oranı korunarak ardışık biçimde kaydedilmektedir. Literatürde genellikle %15 ile %30 arasında bir bindirme oranı hem optik bozulmaların telafi edilmesi hem de görüntülerin birleştirilmesi için yeterli sayıda ayırt edici özellik noktasının elde edilmesi açısından uygun bir değer aralığı olarak kabul edilmektedir. Aşağıdaki şekilde serpentine tarama yolunun geometrik düzeni gösterilmektedir. Yatay eksen x, dikey eksen ise y koordinatlarını ifade etmektedir. Görüntü edinim sırası 1 numaralı konumdan başlamakta ve satır sonunda bir alt satıra geçilerek ters yönde devam etmektedir. Örneğin, 1–2–3–4–5–6–7 konumları ileri yönde yakalanırken, hemen alt satırda yer alan 8–9–10–11–12–13–14 konumları geri yönde kaydedilmektedir. Bu ileri–geri düzen, 3×7 (21 konum) ve 4×8 (32 konum) matrisler için de aynı şekilde uygulanmaktadır. Sonuç olarak serpentine tarama her satır sonunda gerçekleşen tam geri dönüş hareketlerini ortadan kaldırarak

mekanik sönümleme süresini azaltmakta ve genel edinim sürecini optimize etmektedir. Böylelikle daha hızlı, daha kararlı ve bütünsel açıdan daha verimli bir görüntüleme süreci elde edilmektedir.



Şekil 1. Serpentine Tarama Düzeni ve Yön Akışı

Aşağıdaki şekilde görüntü birleştirme sürecinde kritik rol oynayan bindirme (overlap) bölgeleri ve komşuluk ilişkileri gösterilmektedir. Merkezde yer alan bir görüntü (A) ile çevresindeki komşu görüntüler (B, C, D, E) arasındaki örtüşen kısımlar açık biçimde ayırt edilmektedir. Bu bindirme alanları her bir görüntünün kenar bölgelerinde konumlanmakta ve eşleştirme algoritmalarının geometrik dönüşümleri hesaplaması için gerekli ortak referans noktalarını içermektedir. Bindirme oranı özellikle optik bozulmaların daha belirgin hale geldiği merkezden uzak kenar bölgelerindeki detayların korunmasında önemli bir rol oynamaktadır. Dolayısıyla yeterli bindirme alanı sağlanması, eşleştirme algoritmalarının başarısını doğrudan etkileyen temel bir koşuldur. Bu bağlamda dikiş kalitesini belirleyen en kritik unsurlardan biri olarak değerlendirilmektedir.



Şekil 2. Bindirme ve Komşuluk İlişkileri

3. Zamanlama Metodolojisi ve Performans Ölçütleri

Burada geniş alan mikroskobik görüntüleme sisteminin edinim süresi ve performans ölçütleri ayrıntılı olarak ele alınmaktadır. Sistemi oluşturan farklı bileşenlerin zamanlamaya etkileri sistematik biçimde incelenerek, kalibrasyon ve optimizasyon süreçleriyle nasıl verimlilik artışı sağlandığı açıklanmaktadır. Bu yaklaşım edinim hattındaki olası dar boğazları görünür kılar ve genel performansın hangi adımlarla iyileştirilebileceğini açıkça ortaya koyar.

3.1. Kapsamlı Zamanlama Dökümü

Geniş alan mikroskobik görüntüleme sisteminin performansını değerlendirmek amacıyla toplam uçtan uca edinim süresi ayrıntılı alt bileşenlere ayrıştırılarak incelenmiştir. Her bir konum için ölçülen toplam yakalama süresi hareket, sönümleme, kamera hazırlık aşaması, pozlama/okuma ve veri yazma (I/O) adımlarının birleşiminden oluşmaktadır. Bu ayrıştırma sayesinde sistemin hangi bileşenlerde zaman kaybı yaşadığı ortaya konulmakta ve performansın iyileştirilmesine yönelik hedeflenmiş optimizasyon stratejileri geliştirilebilmektedir.

3.1.1. Hareket ve Sönümleme Süreleri

Hareketli tabla için hareket süresi, tablanın hedef konuma ulaşmasıyla ilgilidir. Sönümleme süresi ise hareket sona erdikten sonra sistemin titreşimden arınarak kararlı bir konuma yerleşmesi için gereken süreyi ifade etmektedir. Kalibrasyona dayalı ölçümler sonucunda hareket için ortalama 0,75 saniye, sönümleme için ise 0,50 saniye değerleri elde edilmiştir. Kullanılan Nema17 adım motorları yüksek hassasiyetli hareket kabiliyetine sahip olup milisaniye düzeyinde hızlı adım atma ve yerleşme süreleri sunmaktadır.

3.1.2. Kamera İşlem ve G/Ç Süreleri

Serpentine tarama yöntemi kapsamında yapılan yazılım geliştirmeleriyle kamera hazırlık süresi 0,20 saniye, pozlama ve okuma süresi ise yine 0,20 saniye gibi düşük gecikme değerleriyle kaydedilmiştir. Bununla birlikte veri akışı açısından kritik diğer bir unsur sensörden alınan görüntülerin diske yazılma süresidir. Özellikle yüksek çözünürlüklü görüntüler ve geniş veri setleri söz konusu olduğunda, disk yazma gecikmeleri toplam edinim süresinin önemli bir bölümünü oluşturabilmektedir.

3.2. Kalibrasyon ve Gecikmelerin Azaltılması

Sistemin ilk ölçümlerinde 3×7 matrisin (21 konum) tamamının edinim süresi yaklaşık 116 saniye olarak kaydedilmiş, bu da konum başına ortalama 4,80 saniyeye karşılık gelmiştir. Bu değer hedeflenen performans seviyesinin altında kalmıştır. Ayrıntılı analizler sonucunda, süredeki kayıpların büyük ölçüde

mekanik sönümleme ile G/Ç (disk yazımı) gecikmelerinden kaynaklandığı belirlenmiştir. Elde edilen bu bulgular sistem performansının yalnızca yazılım algoritmalarına bağlı olmadığını, donanım bileşenleri ve veri aktarım hattı optimizasyonlarının da belirleyici olduğunu ortaya koymaktadır. Bu nedenle kalibrasyon süreciyle birlikte G/Ç ayarlarında çeşitli iyileştirmeler yapılmıştır. Disk yazma gecikmelerini azaltmak için yüksek hızlı depolama çözümleri (SSD/NVMe) kullanılmış ve ayrıca paralel veri işleme ve tamponlama stratejileri uygulanmıştır.

Gerçekleştirilen optimizasyonların ardından tek bir konum için ortalama süre 1,40 saniyeye düşürülmüştür. Böylece 3×7 (21 konum) matris için uçtan uca edinim süresi yaklaşık 45 saniyeye kadar azaltılmıştır. Bu sonuç donanım, yazılım ve veri akışının bütünleşik bir şekilde yönetilmesinin yüksek performanslı sistemler için kritik olduğunu göstermektedir. Aşağıdaki tabloda tek bir konum için zaman bileşenlerinin erken aşama ölçümlerinde elde edilen değerlerle kalibrasyon sonrası optimize edilmiş değerleri karşılaştırmalı olarak sunulmaktadır.

Tablo 1. Zamanlama ve Sistem Performans Göstergeleri (özet)

Ölçüt	Değer	Açıklama
Efektif edinim süresi (3x7)	~45 s	Kalibrasyon + başlangıç/bitiş ekleriyle
Ortalama CPU	~%9,9	Oturum boyunca ortalama yük
Kaynak gereksinimi	Düşük–Orta	Standart dizüstü/gömülü sistem yeterli
Zaman çizelgesi	Kararlı	Ani sıçrama yok; tekrarlanabilir

Tabloda her bir sütun, hareket, sönümleme, pozlama/okuma ve yazma gibi farklı bileşenlerin bir konum başına harcadığı süreyi göstermektedir. Erken Ölçümler serisi başlangıçta elde edilen ortalama ~4,80 s/konum değerini ortaya koymakta, özellikle sönümleme ve yazma bileşenlerinin toplam süre üzerindeki baskınlığını açıkça vurgulamaktadır. Kalibrasyon sonrası gerçekleştirilen optimizasyonlar ise bu bileşenlerde dramatik bir iyileşme sağlamış, tek konum

süresini ~1,40 s seviyesine indirerek sistemin verimliliğindeki artışı somutlaştırmıştır.

4. Özellik Tabanlı Eşleştirme ve Dönüşüm Modellemesi

Görüntü birleştirme sürecinin temelinde örtüşen görüntüler arasındaki geometrik dönüşümlerin doğru şekilde hesaplanması bulunmaktadır. Bu aşamada kullanılan yöntemler hem eşleştirme algoritmalarının güvenilirliğine hem de elde edilen dönüşüm modelinin sağlamlığına bağlıdır. Bu çalışmada özellik tabanlı eşleştirme yaklaşımlarına dayalı farklı algoritmalar incelenmiş ve uygulama bağlamında performansları değerlendirilmiştir.

4.1. Algoritmaların Karşılaştırmalı Değerlendirmesi

Örtüşen görüntüler arasında geometrik dönüşümün doğru şekilde belirlenmesi, birleştirme sürecinin en kritik adımıdır. Bu amaçla çalışmada SIFT (Scale Invariant Feature Transform), ORB (Oriented FAST and Rotated BRIEF) ve AKAZE (Accelerated KAZE) gibi yaygın olarak kullanılan algoritmalar uygulanmıştır.

4.1.1. Özellik Tabanlı Eşleştirme (SIFT)

Geniş alan görüntü birleştirme uygulamalarında ölçek ve döndürme değişimlerinden bağımsız ayırt edici özellikler çıkarabilmesi nedeniyle SIFT algoritması önemli bir yöntem olarak öne çıkmaktadır. SIFT görüntülerdeki ilgi noktalarını (keypoint) tespit ederek bu noktaları tanımlayan güçlü vektörler üretmektedir. Bu tanımlayıcılar sayesinde farklı ölçek, açı, kontrast veya aydınlatma koşullarında dahi yüksek tutarlılık sağlanabilmekte, böylece mikroskobik tarama ve medikal görüntüleme uygulamalarında güvenilir sonuçlar elde edilmektedir. Özellikle yüksek ayrıntı içeren doku yapılarında doğru eşleştirmeler yapabilmesi SIFT'i geniş alan mikroskopisi için sıkça tercih edilen bir yöntem haline getirmektedir. Bununla birlikte en önemli dezavantajı ise büyük veri setlerinde işlem süresini uzatabilmesidir. Ancak sağladığı yüksek doğruluk oranı, bu yöntemi literatürde sıklıkla referans alınan bir standart haline getirmiştir (Mohammadi ve ark., 2024; Osipov, 2018).

4.1.2. Hızlı Özellik Eşleştirme (ORB)

Hesaplama süresini düşürmek amacıyla geliştirilen ORB (Oriented FAST and Rotated BRIEF) algoritması, SIFT ve SURF yöntemlerine verimli bir alternatif olarak öne çıkmaktadır. ORB, FAST köşe tespitçisini ve BRIEF tanımlayıcısını birleştirerek düşük bellek gereksinimiyle yüksek hızda çalışabilmektedir. Ayrıca döndürme ve ölçek değişimlerine karşı daha dayanıklı hale getirilmiştir. Bu özellikleri sayesinde özellikle gerçek zamanlı uygulamalarda güçlü bir performans sergilemektedir. Bununla birlikte çok karmaşık doku yapılarında

ayırt edicilik bakımından SIFT kadar başarılı olmadığı çalışmalar da mevcuttur (Rublee ve ark., 2011).

4.1.3. Hızlandırılmış KAZE (AKAZE)

AKAZE (Accelerated KAZE) doğrusal olmayan ölçek mekânında çalışan KAZE algoritmasının hızlandırılmış bir türevi olarak geliştirilmiştir. Daha düşük hesaplama maliyetiyle benzer doğruluk elde edebilmek için tasarlanmış olup, özellik tespiti ve tanımlaması sırasında doğrusal olmayan filtreleme kullanmaktadır. Bu yaklaşım kenar yapılarının ve dokuların korunmasında yüksek başarı sağlamaktadır. Literatürde SIFT ve SURF ile yapılan karşılaştırmalarda AKAZE'nin hız ve doğruluk dengesi açısından oldukça verimli olduğu raporlanmıştır (Alcantarilla ve ark., 2013). Özellikle gömülü sistemler veya donanım kısıtlaması bulunan mikroskobik görüntüleme uygulamalarında oldukça kullanışlı bir yöntemdir.

4.1.4. Hızlı Köşe Tespitçisi (FAST)

FAST (Features from Accelerated Segment Test) algoritması temel olarak köşe noktalarının tespitine odaklanmaktadır. Hesaplama maliyetinin son derece düşük olması sayesinde yüksek kare hızlarında çalışan gerçek zamanlı uygulamalarda yaygın biçimde kullanılmaktadır. FAST piksellerin komşuluk ilişkilerini hızlıca değerlendirerek bir pikselin köşe olup olmadığını belirlemektedir. Ancak bu yöntem yalnızca tespit (detector) aşaması için uygundur. FAST yöntemi ayırt edici tanımlayıcı (descriptor) üretmemektedir. Bu nedenle genellikle BRIEF, ORB veya FREAK gibi tanımlayıcılarla birlikte kullanılmaktadır. Bu yapı düşük maliyetli ve yüksek hız gerektiren sistemlerde önemli bir avantaj sunmaktadır (Rosten ve Drummond, 2006).

4.1.5. Hızlı ve Sağlam Özellikler (SURF)

SURF (Speeded Up Robust Features) SIFT ile benzer biçimde ölçek ve döndürme değişimlerine dayanıklı bir özellik çıkarma yöntemidir. Bu yöntem hesaplama maliyetini azaltmak için integral görüntülerden yararlanmaktadır. Bu sayede SIFT'e kıyasla çok daha hızlı çalışabilmektedir. Mikroskobik görüntülerde kontrast farklılıkları ve dokular arasındaki sınır bölgelerinde yüksek performans göstermektedir. Literatürde yapılan karşılaştırmalar SURF'un hız ve doğruluk dengesi bakımından birçok uygulamada en uygun yöntemlerden biri olduğunu göstermektedir (Bay ve ark., 2008). Bununla birlikte çok küçük boyutlu veya düşük kontrastlı yapılar söz konusu olduğunda SIFT kadar yüksek ayırt edicilik sergilemediği yönünde çalışmalar mevcuttur.

4.2. RANSAC ile Robust Dönüşüm Kestirimi

Özellik eşleştirme sürecinde sıklıkla hatalı eşleşmeler (outlier) meydana gelebilmektedir. Bu tür eşleşmeler geometrik dönüşüm modelinin doğru biçimde

hesaplanmasını bozabilmektedir. Bu sorunun giderilmesi amacıyla eşleştirme aşamasından sonra RANSAC (Random Sample Consensus) algoritması kullanılmıştır. RANSAC rastgele örneklem alma yaklaşımıyla hatalı eşleşmeleri ve aykırı değerleri tespit ederek elimine etmekte ve yalnızca tutarlı eşleşmelere en uygun dönüşüm modelini tahmin etmektedir (Fischler ve Bolles, 1981). RANSAC'ın temel amacı veri setindeki modele uyan inlier noktaları ayırt ederek affine veya projektif gibi en doğru dönüşüm modelini güvenilir biçimde tahmin etmektir. Bu sayede özellikle düşük dokulu ya da karmaşık görüntülerde ortaya çıkan yanlış eşleşmeler otomatik olarak filtrelenmekte ve dönüşüm hesaplamalarının sağlamlığı garanti altına alınmaktadır. Nihai birleştirilmiş görüntüde geometrik hataların önlenmesi açısından kritik rol oynamaktadır. Birleştirme sürecinde başarı genellikle maksimum hata (AE) değerinin 1 pikselin altında olması ile tanımlanmaktadır. Bu noktada RANSAC'ın uygun hata eşiğiyle çalıştırılması söz konusu yüksek hassasiyet kriterine ulaşmak için belirleyici öneme sahiptir. Aşağıdaki tabloda, özellik çıkarımına yönelik algoritmaların ölçümsel doğruluk oranları karşılaştırmalı olarak sunulmaktadır.

Tablo 2. Özellik Çıkarımı ve Ölçümsel Doğruluk

Yöntem	Geometrik Doğruluk (%)	Inlier Oranı (%)	Hücre Sayım Hatası (%)
SIFT	98,7	91,5	1,4
ORB	96,3	84,2	3,6
AKAZE	97,8	88,9	2,2

5. Fotometrik Tutarlılık ve Çok Bantlı Harmanlama

Geniş alan mikroskobik görüntüleme sistemlerinde birleştirme işlemi aydınlatma düzensizlikleri, lens vinyet etkisi veya sensör farklılıkları gibi nedenlerle kareler arasında parlaklık ve renk tonu farklarının oluşmasına yol açabilmektedir. Bu farklılıklar dikiş hatlarında belirgin çizgiler şeklinde görünerek nihai birleşik görüntünün kalitesini düşürmektedir. Bu bölümde söz konusu sorunları gidermek için uygulanan fotometrik telafi ve harmanlama yöntemleri detaylandırılmaktadır.

5.1. Sorun ve Çözüm

Birleştirme sürecinde en yaygın karşılaşılan problemlerden biri kareler arasındaki aydınlatma farklılıklarıdır. Lens vinyeti, ışık kaynağındaki dalgalanmalar veya pozlama sürelerindeki küçük değişiklikler görüntüler arasında ton ve parlaklık farklarına neden olmaktadır. Bu farklar birleşme çizgilerinde görünür hale gelerek görüntü bütünlüğünü bozmaktadır. Bu sorunun giderilmesi amacıyla çalışmada fotometrik telafi ve çok bantlı harmanlama yöntemleri uygulanmıştır.

5.2. Poz Telafisi

Fotometrik telafi birleşik görüntünün genel parlaklık ve renk uyumunu sağlamayı amaçlamaktadır. Özellikle geniş alan mikroskopisinde karelerin kenar bölgelerinde ortaya çıkan düzensizlikleri gidermek için shading correction yöntemleri kullanılmıştır. Tak ve arkadaşları (2020) gerçekleştirdikleri çalışmada fotometrik düzeltme ile alan aydınlatma farklılıklarını başarılı biçimde dengeleyerek birleşik görüntülerin tutarlılığını artırdıklarını rapor etmiştir (Tak ve ark., 2020).

5.3. Pozisyon Hatası Telafisi

Çok sayıda görüntünün ardışık biçimde birleştirilmesi sırasında kümülatif konum hataları birikebilmektedir. Bu nedenle yapılan bu çalışmada global hizalama ve pozisyon telafisi adımları uygulanmıştır. Tüm görüntülerin ortak bir koordinat sisteminde optimize edilmesi sayesinde mozaikleme sırasında ortaya çıkan kayma ve bozulmalar en aza indirilmektedir. Preibisch ve arkadaşları (2009) bu yaklaşım ile yüksek doğrulukta mozaikler elde edilebileceğini göstermiştir (Preibisch ve ark., 2009).

5.4. Çok Bantlı Harmanlama (Multi band Blending)

Dikiş bölgelerinde gözle fark edilebilecek parlaklık ve renk farklılıklarını yumuşatmak için çok bantlı harmanlama tekniği kullanılmıştır. Burt ve Adelson (1983) tarafından geliştirilen spline tabanlı bu yaklaşımda görüntüler farklı ölçek bantlarında birleştirilmekte ve düşük frekanslı bileşenlerden yüksek frekanslı detaylara kadar kademeli bir geçiş sağlanmaktadır. Böylece birleşme bölgelerindeki farklılıklar çıplak gözle fark edilemeyecek düzeye indirilmektedir. Wang ve Yang (2020), bu yöntemin SSIM (Structural Similarity Index Measure) gibi kalite metriklerini anlamlı ölçüde yükselttiğini ve birleşik görüntü kalitesini artırdığını göstermiştir.

6. Ön İşleme Adımlarının Etkisi

Görüntülerin birleştirilmeden önce işlenmesi, hizalama ve mozaikleme algoritmalarının başarısını doğrudan etkilemektedir. Bu nedenle edinimden

hemen sonra çeşitli ön işleme adımları uygulanarak kontrast düzensizlikleri giderilmiş, rastgele gürültü azaltılmış ve kenar detayları belirginleştirilmiştir.

6.1. CLAHE (Contrast Limited Adaptive Histogram Equalization)

Yerel kontrastı artırmak ve birleştirme öncesinde detayların daha net görünmesini sağlamak amacıyla CLAHE (Contrast Limited Adaptive Histogram Equalization) yöntemi uygulanmıştır. CLAHE klasik histogram eşitlemede görülen parlak bölgelerdeki aşırı gürültü artışı problemini azaltmakta ve lokal kontrastı dengelemektedir. Pisano ve arkadaşları (1998), mamografi görüntülerinde CLAHE uygulayarak kütlesel lezyonların tespitinde belirgin bir iyileşme sağlandığını rapor etmişlerdir. CLAHE yönteminde kullanılan temel parametreler olan clipLimit ve tileGridSize kontrast artırma düzeyini ve algoritmanın hangi lokal bölgelerde uygulanacağını belirlemektedir. Bu parametrelerin uygun seçimi mikroskobik tarama uygulamalarında özellikle ince detayların ortaya çıkarılmasında kritik rol oynamaktadır.

6.2. Gürültü Giderme Non Local Means (NL Means)

Birleştirme işleminde ortaya çıkabilecek parazitleri azaltmak amacıyla Non Local Means (NL Means) gibi gelişmiş gürültü giderme algoritmaları uygulanmıştır. NL Means yöntemi her piksel için tüm görüntü üzerinde benzer desenleri aramakta ve benzerlik gösteren bölgelerin ağırlıklı ortalamasını almaktadır. Böylece tekrar eden dokulardaki rastgele gürültü etkin biçimde azaltılırken, yapısal detayların korunması sağlanmaktadır (Buades et al., 2005).

NL Means geleneksel gürültü filtrelerine kıyasla daha yavaş bir yöntem olmakla birlikte, detayları koruma açısından üstün performans sergilemektedir. Bu nedenle ön işleme adımları stratejik bir şekilde uygulanmıştır. Aşırı yerel kontrast değişimlerine veya yapay doku üretimine yol açabilecek agresif işlemlerden kaçınılmış ve özellik eşleştirme öncesinde yalnızca güvenilirliği artıracak filtreleme adımları tercih edilmiştir. Böylelikle SIFT, AKAZE ve ORB gibi özellik tabanlı algoritmaların yapay gürültü nedeniyle hatalı eşleşmeler üretmesi engellenmiştir. Her adım yalnızca kendi işlevini yerine getirmekle kalmamış, aynı zamanda sonraki işlemlerin güvenilirliğini de desteklemiştir.

7. Duyarlılık Analizi (Serpentine Tarama Bağlamında Kararlılık)

Sistemin kararlılığı ve genel performansı üç temel parametre üzerinde yürütülen sistematik bir duyarlılık analizi ile değerlendirilmiştir. Her konfigürasyon için toplam edinim süresi, eşleştirme başarısı, inlier oranı ve dikiş görünürlüğü metrikleri ölçülmüştür. Tüm koşullar en az üç tekrar ile test edilmiş, elde edilen ortalama ve standart sapma değerleri raporlanmıştır. Deneylerde sahne aydınlatması ve bindirme oranı sabit tutulmuş, yalnızca ilgili parametrenin

etkisi izole edilmiştir. Bu yaklaşım hata zaman dengesi açısından en uygun bölgenin belirlenmesine temel oluşturmuştur.

7.1. RANSAC Eşiği (τ)

RANSAC eşiği bir eşleşmenin modele inlier olarak kabul edilmesi için izin verilen maksimum hata miktarını ifade etmektedir. Yapılan testler $\tau \in {1.0, 2.0, 3.0, 4.0}$ piksel aralığında gerçekleştirilmiştir. Sonuçlar $\tau \approx 2.0$ piksel değerinin hata zaman dengesi açısından en uygun eşik olduğunu göstermektedir. Çok düşük eşik değerleri (1.0 px) geçerli eşleşmeleri reddederek model tahminini zorlaştırabilirken, çok yüksek eşikler (4.0 px) hatalı eşleşmeleri kabul ederek birleşik görüntü kalitesini düşürmektedir.

7.2. Minimum Eşleşme Sayısı (min_matches)

Minimum eşleşme sayısı dönüşümün güvenilir biçimde kestirilebilmesi için gerekli asgari özellik eşleşmesini belirleyen kritik bir parametredir. Denemeler min_matches ∈ {10, 20, 30} değerleri için yürütülmüştür. Düşük eşikler (10) hız avantajı sağlamakla birlikte zayıf dokulu bölgelerde model kurulumunu zorlaştırmıştır. Orta eşik (20) yeniden örnekleme sayısını makul düzeyde tutmuş ve başarısız hizalama oranını belirgin ölçüde azaltmıştır. Yüksek eşik (30) ise sağlamlığı sınırlı ölçüde artırmış olsa da bazı görüntü çiftlerinde yetersiz ortak özellik nedeniyle gereksiz yeniden denemeler tetiklenmiş ve toplam süre olumsuz etkilenmiştir. Sonuç olarak varsayılan ayar min_matches ≈ 20 olarak seçilmiş ve düşük dokulu alanlar için ise adaptif eşikleme ile erken durdurma stratejisinin birlikte kullanılması önerilmiştir.

7.3. Harmanlama Bant Sayısı

Çok bantlı harmanlamada kullanılan bant sayısı dikiş görünürlüğünü doğrudan etkilemektedir. Bant sayısının artırılması geçişlerin daha yumuşak olmasını sağlayarak dikiş izlerini azaltmaktadır. Ancak bant sayısının artışı hesaplama süresi ve bellek tüketimini de yükseltmektedir. Bu nedenle orta seviye bant sayısı kabul edilebilir dikiş kalitesi ile makul performans maliyeti arasında denge kurmak için en uygun tercih olarak değerlendirilmiştir.

Tablo 3. Duyarlılık Çalışması – Önerilen Ayarlar

Parametre	Aralık	Öneri	Not
RANSAC eşiği τ	1.0,2.0,3.0,4.0 px	2.0 px	Hata zaman dengesi
min_matches	10,20,30	20	Başarı/kararlılık dengesi

Parametre	Aralık	Öneri	Not
Bant sayısı	—	Orta	Görünür dikiş ↘; süre/bellek ↗

8. Zaman Kalite Dengesi ve Pratik Öneriler

Yapılan çalışma yüksek performanslı geniş alan mikroskobik görüntülemenin yalnızca yazılımsal değil, aynı zamanda donanımsal optimizasyonlarla birlikte ele alınması gerektiğini ortaya koymaktadır. Serpantin tarama yöntemi mekanik geri dönüşlerden kaynaklanan konum hatalarını azaltarak tekrarlanabilir ve öngörülebilir zaman çizelgeleri oluşturmaktadır. Özellikle otomatikleştirilmiş ve yüksek hacimli seri işlemler için bu çizelgeler kritik avantajlar sunmaktadır. Bununla birlikte edinim hızındaki artış tek başına yeterli değildir. Nihai ürünün kalitesi de en az hız kadar önemlidir. Bu bağlamda poz telafisi ve çok bantlı harmanlama gibi gelişmiş fotometrik düzeltme tekniklerinin uygulanması dikiş görünürlüğünü belirgin biçimde azaltarak birleşik görüntünün görsel bütünlüğünü güvence altına almaktadır. Özellik tabanlı eşleştirme algoritmalarının karşılaştırmalı analizi SIFT'in en yüksek doğruluk ve kararlılığı sağladığını fakat buna karşın AKAZE'nin hız–doğruluk dengesi açısından uygun bir alternatif sunduğunu göstermiştir. Parametre optimizasyonu kapsamında $\tau = 2.0$ px eşiği ve min_matches = 20 değeri önerilen en uygun ayarlar olarak belirlenmiştir. Daha büyük matris boyutlarında çalışıldığında ise veri aktarımı sürecinde dar boğazların oluşmasını önlemek için yüksek hızlı SSD veya NVMe depolama birimlerinin kullanımı tavsiye edilmektedir. Bununla birlikte toplam edinim süresi matris boyutuyla doğrusal olmayan bir şekilde artabileceğinden GPU hızlandırma teknolojilerinin entegrasyonu ile işlem süresinin kayda değer ölçüde düşürülebileceği öngörülmektedir. Aşağıdaki tabloda, gerçekleştirilen matris tabanlı tarama yöntemlerinin çekim performansları karşılaştırmalı olarak sunulmaktadır.

Tablo 4. Serpentine Tarama: 3x7 ve 4x8 Düzenleri için Performans Karşılaştırması

Ölçüt	3x7 (21 Konum)	4x8 (32 Konum)
Hedef Bindirme (%)	15–30	15–30
Uçtan Uca Toplam Süre (s)	~45	~65
Ortalama İşlem Süresi (ms/konum)	~1400	~1450

Ölçüt	3x7 (21 Konum)	4x8 (32 Konum)
Başarısız Hizalama Oranı (%)	< 1	< 2

9. Sonuç

Bu çalışma geniş alan mikroskopide serpentine tarama yönteminin etkinliğini ve önemini kapsamlı bir şekilde ortaya koymuştur. Serpentine yakalama sıralamasının özellik tabanlı eşleştirme, sağlam dönüşüm kestirimi, dikiş hattı seçimi, fotometrik telafi ve çok bantlı harmanlama gibi yazılım teknikleriyle bütünleştirilmesi işlemi, metriklerle doğrulanmış ve dikiş hattı görünürlüğü düşük, güvenilir birleşik görüntüler elde edilmesini mümkün kılmıştır.

Donanım kalibrasyonu ve G/Ç optimizasyonları sonucunda 3×7 düzeninde uçtan uca edinim süresinin ~45 saniye seviyesine indirildiği somut olarak gösterilmiştir. Bu bulgu yüksek performanslı birleşik görüntüleme sistemlerinin tasarımında yalnızca algoritmaların değil aynı zamanda fiziksel edinim süreci ile veri aktarım hattının bütüncül bir şekilde optimize edilmesinin belirleyici olduğunu ortaya koymuştur. Özellikle geniş matris boyutlarında tarama yapılırken veri işleme boru hattının yazılım ve donanım yönünden optimize edilmesi sistem performansını belirleyen kritik bir unsurdur.

Çalışmada vurgulandığı üzere yüksek hızlı SSD/NVMe depolama kullanımı G/Ç gecikmelerini azaltmakta ve GPU hızlandırması ise kritik hesaplama adımlarını paralelleştirerek toplam edinim süresini kayda değer ölçüde düşürmektedir.

Bu sonuçlar gelecekte yapılacak çalışmaların daha büyük matris boyutlarında ölçeklenebilirlik testlerine ve GPU hızlandırma tekniklerinin entegrasyonuna odaklanılabileceğini göstermektedir. Bu tür iyileştirmeler birleşik görüntüleme sistemlerinin performansını daha da artırarak biyomedikal ve endüstriyel uygulamalarda gerçek zamanlı analiz ve otomasyona yönelik yeni araştırmalara dair yeni araştırmalara olanak tanıyacaktır.

Kaynakça

Alcantarilla, P.F., Nuevo, J., & Bartoli, A. (2013). Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. British Machine Vision Conference.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded up robust features (SURF). Computer vision and image understanding, 110(3), 346 359.

Buades, A., Coll, B., & Morel, J. M. (2005). A non local algorithm for image denoising. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 2, pp. 60 65). Ieee.

Burt, P. J., & Adelson, E. H. (1983). A multiresolution spline with application to image mosaics. ACM Transactions on Graphics (ToG), 2(4), 217 236.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381 395.

Mohammadi, F.S., Mohammadi, S.E., Mojarad Adi, P. (2024). A comparative analysis of pairwise image stitching techniques for microscopy images. Sci Rep 14, 9215 (2024).

Ortega, E., Nicholls, D., Browning, N. D., & de Jonge, N. (2021). High temporal resolution scanning transmission electron microscopy using sparse serpentine scan pathways. Scientific reports, 11(1), 22722.

Osipov, A. (2018). Some fuzzy tools for evaluation of computer vision algorithms. International Journal of Computer Vision and Image Processing (IJCVIP), 8(1), 1 14.

Pisano, E. D., Zong, S., Hemminger, B. M., DeLuca, M., Johnston, R. E., Muller, K., ... & Pizer, S. M. (1998). Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. Journal of Digital imaging, 11(4), 193.

Preibisch, S., Saalfeld, S., & Tomancak, P. (2009). Globally optimal stitching of tiled 3D microscopic image acquisitions. Bioinformatics, 25(11), 1463 1465.

Rosten, E., & Drummond, T. (2006, May). Machine learning for high speed corner detection. In European conference on computer vision (pp. 430 443). Berlin, Heidelberg: Springer Berlin Heidelberg.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In 2011 International conference on computer vision (pp. 2564 2571). Ieee.

Sharma, S. K., Jain, K., & Shukla, A. K. (2023). A Comparative Analysis of Feature Detectors and Descriptors for Image Stitching. Applied Sciences, 13(10), 6015.

Soltanpour, S., & Joslin, C. (2023). A Survey on Feature Based and Deep Image Stitching. Proceedings Copyright, 777, 788.

Szeliski, R. (2022). Image alignment and stitching. In Computer Vision: Algorithms and Applications (pp. 401 441). Cham: Springer International Publishing.

Tak, Y. O., Park, A., Choi, J., Eom, J., Kwon, H. S., & Eom, J. B. (2020). Simple shading correction method for brightfield whole slide imaging. Sensors, 20(11), 3084.

Wang, Z., & Yang, Z. (2020). Review on image stitching techniques. Multimedia Systems, 26(4), 413 430.


BÖLÜM 5

Mühendislik Araştırmalarında Yöntemsel Kararların Görünmeyen Etkileri: Model, Veri ve Değerlendirme Tercihleri Bilimi Nasıl Şekillendirir?

Berna Gürler Arı[1]

1. Yöntemsel Kararlar ve Mühendislik Araştırmalarının Görünmeyen Omurgası

Mühendislikte, *yöntem* terimi genellikle akla somut, elle tutulur bir olguyu getirir ki bu; bir algoritmanın adı, bir modelin denklemi, bir mimari diyagram veya bir laboratuvar kurulumu olabilmektedir. Bu doğrudan çağrışım, mühendisliğin tarihsel olarak ölçülebilen, tekrarlanabilen ve tahmin edilebilen şeylere odaklanmasından kaynaklanmaktadır. Geleneksel olarak, klasik mühendislikte, bir yöntem doğru kullanıldığında aynı sonucu veren bir süreç olarak görülür. Başka bir deyişle, yöntem sistemin matematiksel tanımıdır.

Bunun aksine, mühendislik disiplinlerinde veri odaklı metodolojilerin öne çıkması sebebiyle manzara önemli ölçüde değişmiştir. Veri odaklı araştırmalarda, metodolojik çerçeve, yalnızca kullanılan hesaplama süreçlerini değil, aynı zamanda değişkenlerin seçimini, değerli bilginin tanımını ve belirli belirsizliklerin göz ardı edilmesini de belirleyen bir dizi seçeneğe dönüşmüştür [1]. Bu evrim, özellikle makine öğrenimi ve istatistiksel modelleme paradigmalarının mühendislik uygulamalarına dahil edilmesinde belirgindir; Mühendislik, çözümlerin formüle edilmesinin yanı sıra çıkarımların üretilmesini de giderek daha fazla benimsemiştir [2].

Bu nedenle, modern mühendislik araştırmalarında, yöntemi tanımlarken yalnızca modele odaklanmak, çalışmanın özünü gizleyebilmektedir. Bir modelin kalitesi genellikle içsel zekasından ziyade metodolojik seçimlerin birleşik etkilerine bağlıdır. Bu seçimler, verilerin nasıl temsil edildiği, örneklerin nasıl bölündüğü, başarıyı ölçmek için kullanılan metrikler ve karşılaştırmaların adilliği gibi unsurları içermektedir [1], [2]. Asıl nokta şudur ki aynı algoritma farklı metodolojik çerçevelerde kullanıldığında, sonuçlar önemli ölçüde farklılık gösterebilmekte ve potansiyel olarak tamamen farklı bilimsel sonuçlara hatta farklı bir sonuç kümesine yol açabilmektedir.

[1] Dr. Öğr. Üyesi, Milli Savunma Üniversitesi, Kara Harp Okulu, Bilgisayar Mühendisliği Bölümü, Orcid: 0000-0003-1000-2619

Yöntemsel Kararın Tanımlanması

Çoğu zaman örtük ve doğal kabul edilen metodolojik kararlar, araştırma sürecinin başlarında yapılan seçimlerdir. Bazen model eğitimi başlamadan önce yapılan bu kararlar, ele alınan belirli problemi belirleyebilmektedirler. Sınıflandırma, regresyon veya anomali tespiti yöntemleri arasında seçim yapmak, temsilci olarak kabul edilecek değişkenleri (veya sensörleri) seçmek, verilerin rastgele, zamansal veya kişiye dayalı olarak bölünmesi, başarıyı belirtmek için seçilen ölçüt ve adil karşılaştırma için uygun görülen temel çizgiler, metodolojik seçimlerin örneklerindendir [1], [3]. Bu liste, pratik uygulamalarda daha da genişletilebilmekte; örnek olarak etiketlemenin altında yatan mantık, sınıf dengesizliğine yaklaşım, hiperparametre aramasının kapsamı, deneysel tekrarların sayısı ve hatta rastgele bir noktanın seçimi (özellikle küçük veri kümelerinde) ortaya çıkan anlatıyı etkileyebilmektedir. Bununla birlikte, metodolojik kararların belirleyici özelliği, çalışmanın aydınlatmaya çalıştığı belirli gerçeği tanımlamasıdır. Bu nedenle, metodolojik seçimler modelin dünya algısını ve başarı kriterlerini şekillendirmektedir. Bu seçimlerin önemi çok büyüktür. Aynı algoritma, yapılan metodolojik kararlara bağlı olarak çok farklı bilimsel anlatılar üretebilmektedir. Bunun nedeni, algoritmanın öğrenmesinin aldığı temsile bağlı olması, başarının seçilen ölçütle ölçülmesi ve karşılaştırmaların seçilen temel çizgilere göre yapılmasıdır. Sonuç olarak, yöntem sadece modeli kapsamaz; modelin oluşturulduğu, eğitildiği ve değerlendirildiği tüm yapıyı oluşturur [[1], [3]. Bu bakış açısı, metodolojiyi "model seçimi" gibi kısıtlı bir alandan öte araştırma tasarımında merkezi bir role yükseltir.

1.2. Görünürlüğün Azlığının Sebebi Nedir?

Metodolojik çözümler genellikle *literatürde yaygındır* gerekçesiyle hızla kabul edilir. Bu genellikle kasıtlı bir seçim değildir; daha ziyade, akademik çalışmaların nasıl üretildiğinin doğal bir sonucudur. Ancak, bu eğilim metodolojik kararları daha az görünür hale getirmektedir. Buna etkide bulunan birkaç yapısal neden vardır:

Standardizasyon Baskısı (Kıyaslama Kültürü): Belirli veri kaynakları ve değerlendirme yöntemleri bir alanda standart hale geldikçe, araştırmacılar genellikle bunları sorgulamak yerine bu normları takip etmeyi daha güvenli bulurlar. Ardından, süreç karşılaştırılabilirlik vaat etmekte. bununla birlikte, yöntemlerdeki çeşitlilik azalmakta ve benzer yapıları ile sonuçların sıklıkla tekrarlandığı bir araştırma ortamı yaratmaktadır [4]. Sonuç olarak, kıyaslamayı takip eden bir çalışma olumlu görülürken, kıyaslamayı sorgulayan bir çalışma gereksiz riskler almak olarak görülebilmektedir.

Yayın Ekonomisi: Makalelerin ve kitap bölümlerinin formatlarını incelediğimizde özellikle, yöntemlerin açıklamalarından ziyade belirli ayrıntıların daha kolay kabul edilebildiği ve bir sonuç tablosunun daha hızlı ikna sağladığını görebilmekteyiz. Bu da geçmişte hangi seçimin yapıldığı konusunda sorulara yol açmaktadır [5]. Bazı durumlarda, en önemli metodolojik çözümler (örneğin, veri bölme mantığı, etiket oluşturma, sınıf dengesizliği stratejileri) metnin en az okunan kısımlarında kısaca belirtilmektedir.

Mühendislik Çözüm Refleksi: Mühendislik kültürü çözümlere vurgu yaparak sistemlere, bu sistemleri geliştirmeye ve ürünler genelinde modellerin tamamlanmasına odaklanmaktadır. Bununla birlikte, veri tabanlı bir çözümün güvenilirliği genellikle oluşturulması sırasında yapılan tasarım seçimlerinin kalitesine bağlıdır. Başka bir deyişle, modelin iyiliği karmaşıklığından daha önemlidir; bu, problemin nasıl tanımlandığını, verilerin nasıl temsil edildiğini ve değerlendirme protokolünün ayrıntılarını içermektedir [2].

Sonuç olarak, elektrik enerjisine yönelik basit *çözümü göster* yaklaşımı, metodolojik seçimlerin özü tarafından gizlenebilmektedir. Bu üç güç bir araya geldiğinde, ortaya çıkan senaryo şu şekildedir: metodolojik stratejiler edinilmiş olmalarına rağmen, genellikle metin içindeki en az belirgin unsurlar haline gelmektedir. Bu da iyi bir model ile iyi bilim arasındaki ayrımın azalmasına katkıda bulunmaktadır.

1.3. Metodolojik Kararların Epistemolojik Boyutu

Metodolojik seçimler salt teknik hususların ötesine geçerek özünde bilginin doğasına ilişkin felsefi bir duruşu barındırırlar. Kabul edilebilir hata türlerinin (yanlış pozitifler ve yanlış negatifler) belirlenmesi, temsili örneklerin tanımlanması ve adil ölçütlerin seçimi, araştırmacının problem alanına bakış açısını yansıtmaktadır. Sonuç olarak, bu kararlar bilgi üretimi ve doğrulama süreçlerini doğrudan etkilemektedir [6].

Bir sistemde önemli şeyleri kaçırmak anlamına gelen yanlış negatiflerin maliyeti çok yüksekse, sistemin son derece doğru olduğunu söylemek güçlü bir iddia değildir. Çünkü doğruluk, hataların türünü ve maliyetini gizleyebilmektedir. Benzer şekilde, bir çalışmanın ne kadar temsili olduğu sadece ne kadar veri toplandığına değil, aynı zamanda verilerin nasıl toplandığına, incelenen durumlara ve örneğin nasıl seçildiğine de bağlıdır. Bu nedenle, araştırmada kullanılan yöntemler, araştırmanın gerçekliğin hangi kısmını gösterdiğine karar veren filtreler görevi görmektedir. Box'un iyi bilinen "Tüm modeller yanlıştır, bazıları faydalıdır" ifadesi çok önemli bir başlangıç noktasıdır [7]. Bu ifade, modelin değersiz olduğu anlamına gelmez. Bunun yerine, bir modelin sınırlamalarına rağmen, temel varsayımları açıkça belirtilmişse ancak o zaman faydalı olabileceğini kabul etmenin önemini vurgular. Modelin faydalılığı,

mükemmel derecede doğru olmasıyla ilgili değil, varsayımların anlaşıldığı belirli bir durumda ne kadar iyi çalıştığıyla ilgilidir. Bu nedenle, bir çalışmada kullanılan yöntemlerin açık hale getirilmesi onu zayıflatmayıp aksine, daha yüksek bir bilimsel anlayış düzeyini göstermektedir. Okuyucu bu şekilde bulguların sınırlarını daha iyi anlayabilmektedir.

1.4. Bölüm Amaçları ve Genel Yaklaşım

Bu bölüm, metodolojik kararların kaçınılmaz doğasını kabul ederek, bunları basit ayrıntılar olmaktan çıkarıp dikkatlice değerlendirilmesi gereken konulara dönüştürmeyi amaçlamaktadır. Amaç, araştırmacıyı her şeyi yapmalıyım fikriyle boğmak değildir. Bunun yerine, yapılan seçimleri gerekçelendirme pratiğini teşvik etmeye odaklanılmıştır: Neden bu veriler? Neden bu bölümleme? Neden bu ölçüt? Neden bu temel çizgi? Bu sorulara net cevaplar vermek, çalışmayı temelden güçlendirerek hem değerlendiriciler hem de okuyucular için daha ikna edici hale getirmektedir [1], [2], [5]. İyi bir mühendislik araştırması, sadece en iyi modeli bulmaktan öteye giderek aynı zamanda iyi düşünülmüş bir yöntem gerektirmektedir. İyi düşünülmüş bir yöntem, varsayımları açıkça belirtmek, diğer seçenekleri değerlendirmek ve sonuçların sınırlamalarını dürüstçe açıklamak anlamına gelmektedir. Bu yaklaşım, savunmacı bir tavır yerine araştırmada olgunluğu göstermektedir.

1. Özdeş Sorun, Farklı Disiplinler: Metodoloji Seçimi Bilgiyi Nasıl Yeniden Şekillendiriyor?

Mühendislikte yaygın ancak dile getirilmeyen bir varsayım şudur: Sorun iyi tanımlanmıştır; yöntem onu ele alır. Bu varsayım, özellikle geleneksel mühendislik uygulamalarında doğru çıkar, çünkü sorun genellikle doğrudan fiziksel bir sistemle ilişkilidir ve ölçülen değişkenlerin önemi genellikle açıktır. Veri odaklı araştırmalarda, problem tanımlama sorunu sıklıkla sistemin kendisinden değil, araştırmacı tarafından seçilen çerçeveden kaynaklanmaktadır. Sonuç olarak, problemin formülasyonu tartışmasız en etkili metodolojik seçimdir. Ortaya atılan sorular, hangi cevapların doğru kabul edileceğini ve hangi başarıların iyi olarak değerlendirileceğini belirlemektedir [2], [8].

Önemli bir ayrım mevcuttur. Veri odaklı çalışmalarda, metodoloji yalnızca soruna bir çözüm belirlemekle kalmaz, aynı zamanda sorunun kendisini de yeniden yapılandırır. Araştırmacı sıklıkla önceden var olan bir sorunu ele almaz; bunun yerine, gerçekleri belirli bir bakış açısından yorumlayarak karşılaştığı sorunu tanımlar. Aynı ham olay, farklı sorun formülasyonları aracılığıyla incelendiğinde, yalnızca farklı sayısal sonuçlar değil, aynı zamanda farklı bilimleri de ortaya çıkarmaktadır [2], [8].

Bu durum, formülasyonların çeşitli hata türlerini hesaba katmasından, bazı belirsizlikleri normal olarak görmesinden ve farklı genelleştirilebilirlik vaatlerinde bulunmasından kaynaklanmaktadır. Mühendislik araştırmalarında metodoloji seçiminin yalnızca teknik bir karar olmadığını; bilginin nasıl üretildiğini, yani hangi gerçekliği modellediğimizi etkileyen yapısal bir seçim olduğunu göstermektedir. Breiman, "iki kültür" üzerine yaptığı söylemde bu noktayı vurgulamaktadır: Veri odaklı modelleme, aynı olayı farklı amaçlarla (tahmin mi yoksa açıklama mı?) incelerken farklı metodolojik standartlar üretmektedir [2]. Sonuç olarak, sorunun tanımlanması ve yaklaşımın seçimi, birlikte ele alındığında üretilen bilginin niteliğini belirlemektedir.

2.1. Problem Formülasyonu: Araştırma Dönüştüğünde Bilimde Dönüşümler

Bir sistem çıktısı, bir sensör dizisi, bir çevresel ölçüm seti, bir üretim hattı kalite sinyali veya bir ağ trafiği günlüğü gibi verileri inceleyelim. İlk bakışta aynı gibi görünen bu yapı, araştırmacının sorgulamasına bağlı olarak belirgin şekilde farklı bir bilimsel probleme dönüşür. Aşağıdaki üç formülasyon özünde birbirinden farklıdır:

• Sınıflandırma: "Bu örnek hangi sınıfa aittir?"

• Regresyon: "Bu örneğin güvenilir çıktısı nedir?"

• Anomali tespiti: "Bu örnek atipik mi?"

Bu üç metodolojinin her biri farklı kayıp fonksiyonlarına, çeşitli değerlendirme ölçütlerine ve hata maliyetlerinin farklı yorumlarına bağlıdır. Sonuç olarak, "Hangi problemi ele alıyoruz?" sorusu, "Hangi modeli seçiyoruz?" sorusundan daha büyük önem taşımaktadır [8].

Bu ayrımı daha somut bir şekilde incelemek için, problem formülasyonunun üç temel etkisini inceleyelim:

(i) Hedef değişkenin özellikleri zamanla değişir. Sınıflandırmada amaç ayrık etiketlerdir, regresyonda amaç sürekli değerlerdir, anomali tespitinde ise amaç genellikle etiketin kendisi değil, normalliğin belirlenmesidir. Bu durum, modelin öğrenmeyi amaçladığı şeyin doğasını önemli ölçüde değiştirmektedir.

(ii) Başarı, bireyler tarafından farklı şekillerde tanımlanmaktadır. Sınıflandırma genellikle kategoriler arasındaki farklılaşmayı artırmakta, regresyon hatayı azaltmayı hedeflemekte ve anomali tespiti tipik davranışın parametrelerini belirlemeyi amaçlamaktadır. Biri yanlış sınıflandırmayı vurgularken diğeri küçük bir sayısal varyasyonu; üçüncüsü ise gözden kaçan bir anormalliğin sonucuna dikkat çekmektedir. Sonuç olarak, aynı verilere

uygulanan farklı formülasyonlara sahip özdeş modeller, tamamen farklı başarı profilleri gösterebilmektedir.

(iii) Bilimsel iddiaların niteliği farklılık göstermektedir. Sınıflandırmada iddia sıklıkla farklılık üzerine kuruluyken, regresyonda öngörülebilirlik ve anomali tespitinde beklenmeyenin belirlenmesi üzerine kuruludur. Bu iddialar eş anlamlı gibi görünse de uygulamada farklı veri ön koşulları ve genelleştirilebilirlik açısından farklı riskler içerirler. Anomali tespiti, özellikle "normal"in yetersiz temsil edildiği veri kümelerinde, oldukça hassas olabilmektedir.

Sonuç olarak, problem formülasyonu sadece teknik bir tanımlama değildir; araştırmanın kavramsal çerçevesini oluşturmaktadır. Hastie–Tibshirani–Friedman'a göre, bu kavram modelleme sürecinin merkezindedir: aynı veri kümesindeki “kayıp/amaç”taki değişiklikler farklı bilgi sonuçları sağlamaktadır [8]. Breiman'a göre, bu aynı zamanda araştırmacının “tahmin etme” kültürü ile “açıklama” kültürü arasındaki konumunu da belirlemektedir [2]. Sonuç olarak problem çerçeveleme, bilimsel araştırmanın gidişatını belirleyen ilk kritik kapı görevi görmektedir.

2.2. Veri Gösterimi: Özdeş Veriler, Farklı Gerçeklikler

Mühendislikte, özellik çıkarımı, ölçeklendirme, boyut indirgeme ve pencereleme gibi süreçleri kapsayan veri gösterimi, bazen sadece "ön işleme" olarak geçiştirilmektedir. Bu yaygın bir yanılgıdır. Model anlatıları tipik olarak tasarım ve algoritmalardan oluşturulurken, veri gösterimi sadece mutfakta hazırlık olarak algılanır. Gösterim hakkındaki kararlar, modelin algılayabileceği ve algılayamayacağı düzenlilikleri belirler. Bunu şu şekilde düşünmek avantajlıdır: Veri temsili, modele bilgiyi analiz edeceği bakış açısını sağlar. Aynı gerçeklik, çeşitli bakış açılarından bakıldığında tamamen farklı görünmektedir. Dahası, veri temsiliyle ilgili hususlar performansı, yorumlanabilirliği, sağlamlığı ve genelleştirilebilirliği etkilemektedir. Guyon ve Elisseeff, değişken seçiminin sadece bir performans artırıcı olmadığını, gereksiz değişkenleri atarak modelin aşırı uyum riskini azalttığını, öğrenilen yapının yorumlanabilirliğini artırdığını ve genelleştirilebilirliği etkilediğini ileri sürmektedir [3].

Bu bölüm üç temel temayı kapsar:

- Temsil, sinyali "yeniden yapılandırır".

Ölçeklendirme, normalleştirme ve logaritmik dönüşüm gibi görünüşte basit eylemler bile modelin karar sınırını etkiler. Bunun nedeni, birçok yaklaşımın

mesafeye, varyasyona veya dağılıma duyarlı olmasıdır. Sonuç olarak, temsil seçimi modelin hangi farklılıkları önemli olarak değerlendirdiğini değiştirir.

- Temsil, gürültü ve bilgi arasındaki dengeyi değiştirir.

Boyut indirgeme, pencereleme ve filtreleme gibi süreçler, gürültüyü azaltabilir ve öğrenmeyi geliştirebilirken, aynı zamanda temel bir sinyal bileşenini de zayıflatabilir. Temsil avantajlı olabilir, ancak yanıltıcı da olabilir. Sonuç olarak, temsil hakkındaki yargılar genellikle sadece küçük bir teknik adım değil, çalışmanın sonuçlarını etkileyen temel bir faktördür.

- Temsil, genelleştirilebilirlik iddiasını doğrudan etkiler.

Özellik seçiminde kullanılan kriterler, modelin iyi çalışacağı koşulları belirler. Özellikle, bir veri kümesinde etkili performans gösteren uzmanlaşmış özellikler, başka bir veri kümesinde başarısız olabilmektedir. Sonuç olarak, temsil hakkındaki karar, "bu yöntem diğer bağlamlarda etkili olacak mı?" sorusunun örtük aracı olarak hizmet etmektedir [3].

Sonuç olarak, veri gösterimi yalnızca teknik bir özellik olmaktan çıkıp metodolojinin temelini oluşturur. Model mimarisine geçmeden önce, gösterim kararının bilimsel gerekçesi sıklıkla incelenmelidir: “Bu gösterim problem tanımıyla uyumlu mu?”, “Bu gösterim hangi tür hataları şiddetlendirebilir ve hangilerini hafifletebilir?”, “Bu gösterimi seçmenin sonuçları nelerdir?” sorularına verilen yanıtlar, metodolojinin bilimsel yetkinliğini artırır.

2.3. Karşılaştırmalı Tasarım: Yenilik algısı nasıl formüle edilir?

Bir yöntemin algılanan kalitesi sıklıkla seçilen temel referans noktalarına bağlıdır. Bu gerçek bilimsel uygulamada esastır; ancak sıklıkla kafa karıştırıcıdır ve bu nedenle yeterince tartışılmamaktadır. Temel referans noktasının seçimi, çalışmanın yenilik iddiasını doğrudan etkilemekte, zayıf bir temel referans noktası yöntemin katkısını yanlış bir şekilde abartırken, sağlam bir temel referans noktası daha doğru bir temsil sağlamaktadır. Burada iki temel yanlış anlama mevcuttur:

1) Temel Nokta = Biçimsellik Hatası

Bazı araştırmalar, sadece zorunluluktan dolayı bir temel nokta belirler. Temel nokta, çalışmanın temel çerçevesini oluşturur ve yaklaşımınızın sağladığı iyileştirmeleri gösteren referans noktası görevi görmektedir. Temel nokta seçimindeki bu esneklik, çalışmanın bilimsel geçerliliğini zayıflatmaktadır.

2) "Literatürde mevcut" ifadesi "adil" yanılgısına eşdeğerdir.

Bir yöntemin literatürde mevcut olması, onun geçerli bir karşılaştırma oluşturduğu anlamına gelmemektedir. Temel yöntemin problem bağlamınıza uygunluğu, veri temsiliyle uyumu ve hiperparametrelerinin adil kalibrasyonu gibi faktörler kritik öneme sahiptir.

Hand'in eleştirisi bu noktada oldukça keskindir. Sınıflandırma literatüründeki ilerlemeye dair çok sayıda iddia, altta yatan belirsizliklerin ihmal edilmesi ve kurgusal etkilerin etkisi nedeniyle yanıltıcı olabilmektedir [1]. Bazen, algılanan performans artışı, üstün bir yöntemin keşfinden değil, karşılaştırmanın düzenlenme biçiminden kaynaklanmaktadır. Bu, metodolojik seçimlerin bilimsel araştırmayı nasıl yeniden şekillendirdiğinin mükemmel bir örneğidir.

Bu bölüm, metodolojik olgunluğun temel seçim için aşağıdaki ilkeleri gerektirdiğini belirtmektedir:

Temsili temel set: Tek bir referans yerine, problem alanını kapsayan bir temel set koleksiyonu.

Ayarlamada adalet: Yanıltıcı karşılaştırmalardan kaçınmak için temelleri eşit şekilde optimize etmek.

Tanımlanmış kapsam: "Hangi temel set atlandı ve neden?" sorusuna cevap vermek. Dolayısıyla, karşılaştırma çerçevesi sadece bulguları sunan bir bölüm değildir; araştırmanın bilimsel iddiasını destekleyen temel bir temel oluşturmaktadır.

2.4. Metodolojik olgunluk neyi ifade eder?

Olgun bir teknik, tek doğru yaklaşım iddiasında bulunmaz. Olgun metodoloji genellikle, tek bir doğru yaklaşım yoktur, bunun yerine mantıklı alternatiflerden oluşan bir yelpaze vardır iddiasında bulunabilme yeteneğini ifade etmektedir. Bu olgunluk dört temel yetkinlikte özetlenmiştir:

• Alternatif problem tanımlarını göz önünde bulundurun: “Sınıflandırma yerine regresyonu tercih etseydik ne farklı olurdu?” “Anomali tespit yöntemini uygunsuz kılan nedir?” gibi sorular için özlü ancak açık gerekçeler sunulmalıdır [2], [8]

• Veri gösterimi seçimlerini gerekçelendirin: Gösterimin problem ifadesi ve model sınıfıyla uyumunu açıklayın; özellik seçimi yapılmışsa, bunun hem performans hem de genelleştirilebilirlik üzerindeki etkileri açıklanmalıdır [3].

• Karşılaştırma seçimlerini (baseline setini) savunmak: Baseline’ların neden seçildiğini, hangi koşullarda güçlü/zayıf olduğunu ve kıyasın nasıl adil

tutulduğunu açıkça yazılmalı; zayıf baseline'larla "kolay galibiyet" üretilmemelidir [1].

• Sonuçların düşeceği koşulları açıkça ifade etmek ve yöntemin sınırlamalarını açıkça belirtmek: örneğin, "Bu veri dağılımı altında bozulabilir", "Bu sensör kalitesiyle azalacaktır", "Bu örnek türleriyle hatalar üretecektir." Cümleleri çalışmayı zayıflatmaz; aksine, güvenilirliğini artırmaktadır [1], [2].

Sonuç olarak, metodolojik olgunluk performans ölçütleriyle değil, metodolojik seçimlerin açıklığı, gerekçesi ve uygulanabilirliğiyle değerlendirilir. Bir çalışma, yalnızca bu başarı düzeyini ilan etmekle değil, bu başarının hangi koşullar altında gerçekleştiğini açıklayarak ilerleme kaydeder.

2. Performans Yanılsamaları: "İyi Bir Sonuç" Ne Zaman Bilimsel Anlamdan Yoksun Kalır?

Mühendislik araştırmalarında, performans ölçütleri açıklıkları, ölçülebilirlikleri ve karşılaştırılabilirlikleri dolayısıyla ilgi çekicidir. %92 doğruluk, %0,04 hata ve %0,87 F1 gibi ölçütler, hem okuyucu hem de değerlendirici için güçlü bir "başarı" izlenimi yaratır. Bununla birlikte, tam da bu nedenle, performans ölçütleri tehlikelidir: rakamlar bağlamı gizlemektedir. Bir performans değeri, türetilmesi ve geçerliliğini destekleyen varsayımlar konusunda açıklama içermediğinde, bilimsel öneme sahip olmaktan ziyade ikna edici etkiye sahip bir retorik araca dönüşebilir.

İşte Hand'in "ilerleme yanılsaması" tam da burada önem kazanmaktadır. Literatürde bildirilen çok sayıda performans iyileştirmesi, modelin sorun alanında üstün bir çözüm sağladığı anlamına gelmez; bunun yerine, deneysel tasarımdaki belirsizliklerin modeller arasındaki farklılıklardan daha etkili olduğunu ortaya koymaktadır [1]. Üstün sonuç sıklıkla gelişmiş bilimsel titizlikten değil, alternatif (ve belki de daha hoşgörülü) bir değerlendirme çerçevesinden kaynaklanmaktadır.

Bu senaryoda, performans yanılsaması tamamen ölçümlerin yanlış uygulanmasına bağlanamaz. Yanılsama, veri bölümleme stratejileri, model seçimi, kıyaslama ölçütlerine bağımlılık ve raporlama uygulamalarını kapsayan karmaşık bir metodolojik sorunu oluşturmaktadır. Sonraki alt başlıklar, mühendislik araştırmalarında bu yanılsamanın sistematik olarak nasıl üretildiğini açıklamayı amaçlamaktadır.

3.1. Hassasiyet ve Asimetrik Verilerin Etkileri

Doğruluk, mühendislik literatüründe sıklıkla kullanılan bir performans ölçütüdür. Mantığı açıktır; anlaşılabilir, tek bir ölçüt sağlar ve doğal olarak doğru

tahmin oranı kavramıyla uyumludur. Bununla birlikte, özellikle dengesiz veri kümelerinde, doğruluk istatistiği sıklıkla yanıltıcı olabilmekte ve hatta tamamen alakasız hale gelebilmektedir. Dengesiz veri kümelerinde, %95'lik bir doğruluk oranı bazen bir modelin hiçbir şey öğrenmediği anlamına gelebilir. Örneğin, verilerin %95'i bir kategoriye, %5'i ise başka bir kategoriye sınıflandırılmışsa, tüm örnekleri çoğunluk sınıfına sınıflandıran bir model %95 doğruluk oranına ulaşacaktır. Bu model, azınlık sınıfı hakkında hiçbir bilgi üretmemesine rağmen yüksek performanslı görünmektedir. Bu senaryo, özellikle sağlık, güvenlik veya hata tespiti gibi alanlarda son derece tehlikelidir, çünkü hayati önem taşıyan sınıf genellikle azınlık sınıfını oluşturmaktadır.

Hand'in eleştirisi burada açıkça görülmektedir. Eğer performans ölçütü problemin doğasıyla uyumlu değilse, performans iyileştirmesi bilimsel ilerlemeye eşit değildir [1], [9]. Doğruluk ölçütü, sınıf dağılımı dengesiz olduğunda hatanın meydana geldiği sınıfı tamamen göz ardı etmektedir. Sonuç olarak, doğruluk tek başına iletildiğinde sıklıkla yanıltıcı bir güvence hissi yaratmaktadır.

Buradaki en önemli nokta, sadece doğruluktan kaçınmak demek değil. Temel kaygı şudur: Bu problemde hangi ölçüt belirli bir yanlışlığı ortaya çıkarıyor, hangisi gizliyor? Bu sorgulama yapılmadığı takdirde, performans ölçütleri deneysel önemini kaybetmektedir.

3.2. Ölçüt Seçimi ve Arıza Maliyeti Seçimi ilişkisi

Performans ölçütleri tarafsızlıktan yoksundur. Her istatistik, belirli arıza türlerini vurgularken diğerlerini gizler. Sonuç olarak, ölçüt seçimi teknik bir nüanstan ziyade, hangi arızayı daha maliyetli olarak gördüğümüzün açık veya örtük bir ifadesidir. Çoğu mühendislik uygulamasında, yanlış pozitif ve yanlış negatif arızalar eşit değildir. Sağlık alanında yanlış negatif, bir hastalığın atlanması anlamına gelmekte ve genellikle çok yüksek maliyetli olmaktadır. Güvenlik veya siber saldırı tespitinde, yanlış negatif, sistemin savunmasız kaldığını göstermektedir. Bazı endüstriyel kalite kontrol uygulamalarında, yanlış pozitif, gereksiz müdahalelere ve artan masraflara yol açabilmektedir. Sonuç olarak, F1, geri çağırma, doğruluk ve ROC-AUC dahil olmak üzere her istatistik, hataya ilişkin farklı bir bakış açısı sunmaktadır. Bununla birlikte, temel sorun bu bakış açısının seçilmesinin gerekçesi metinde genellikle iyi ifade edilmemektedir.

Sonuç olarak, algılanamaz bir metodolojik seçim olarak metrik seçimi, tüm araştırmayı yönlendirmektedir.

3.3. Kıyaslama Verilerine Bağımlılık ve Genelleştirilebilirlik Yanılsaması

Kıyaslama veri kümeleri, mühendislik ve makine öğrenimi araştırmalarında karşılaştırılabilirliği sağlamak için çok önemlidir. Bununla birlikte, bu rol zamanla bir bağımlılığa dönüşebilir. Kıyaslama veri kümelerinde performansı artırmak, pratik uygulamalarda her zaman ilerleme anlamına gelmez. Bunun nedeni, kıyaslama kümelerinin zamanla araştırmacılar için tanıdık bir alan haline gelmesidir. Bu durum iki temel riski ortaya çıkarmaktadır:

Seçim yanlılığı: Araştırmacı, kasıtlı veya kasıtsız olarak, model ve hiperparametre seçimlerini kıyaslama testine göre şekillendirmektedir. Bu test seti açıkça incelenmese bile, test bilgisinin sürece istemeden yayılmasına neden olabilmektedir.

Genelleştirilebilirlik yanılsaması: Kıyaslama testinde yüksek performans, stratejinin çeşitli veri dağılımlarında etkili olacağı varsayımını güçlendirmektedir. Gerçekte veriler gürültü, eksik ölçümler, dağılımsal kaymalar ve öngörülemeyen örneklerle çevrelenebilmektedir.

Cawley ve Talbot'un vurguladığı en önemli sorun, model seçiminin yanlış yapılandırılmasının rutin olarak abartılı performans tahminlerine yol açabileceğidir [9]. Bu şişkinlik, herkes aynı ölçütü kullandığı için sıklıkla fark edilmeden kalır ve sonuçların görünüşte olumlu olmasına neden olur. Bu durum, literatürde gerçek ilerleme ile metodolojik yapaylıklar arasındaki ayrımı belirsizleştirir. Dolayısıyla sorun, bu tür ölçütlerin kullanımında değil, geçerlilik için tek ve sorgusuz sualsiz kabul edilmelerinde yatmaktadır.

3.4. En İyi Sonuç ve En Üstün Bilim Arasındaki Fark

Bu noktada temel ayrım, en iyi sonucun belirli bir konfigürasyon altında elde edilen maksimum performans olmasıdır. En etkili bilim, bu performansın geçerli olduğu koşulları kesin olarak açıklayabilen araştırmadır.

Gelişmiş bir metodolojik yaklaşım, aşağıdaki soruları sormaktan çekinmez:

Bu sonucun geçerliliğinin altında yatan varsayımlar nelerdir?

Bu varsayımların ihlal edilmesinin sonuçları nelerdir (veri dağılımı değişirse, sınıf dengesi bozulursa, gürültü artarsa)?

Tablo, diğer ölçütlerle önemli değişiklikler gösteriyor mu? Model seçimi ve hiperparametre optimizasyon süreci test verilerinden bağımsız olarak mı gerçekleştirildi?

Bu sorulara verilen dürüst yanıtlar çalışmayı zayıf kılmaz aksine, güvenilirliğini artırır [1], [9]. Bilimsel önem yalnızca yüksek rakamlarda değil, bunların uygulanabilirliğinin ve sınırlamalarının anlaşılmasındadır. Bu çerçevede, performans yanılsamalarını ortaya koymak, mühendislik araştırmasını engelleyen bir eleştiri değil, temelini güçlendiren önemli bir önlemdir. Gerçek ilerleme, her bir sonraki çalışmada marjinal iyileştirmelerle değil, önemli artışlar ile yalnızca metodolojik yapaylıklar arasında ayrım yapabilme yeteneğiyle ölçülmektedir.

Mühendislik literatüründe, performans tabloları sıklıkla işin "kanıtı" olarak sunulmaktadır. Satırlar yaklaşımları, sütunlar ölçütleri gösterir ve en yüksek değer kalın harflerle vurgulanarak okuyucuya örtük olarak şu mesaj verilir: "Bu, üstün yöntemdir." Bu sunum yöntemi, performansın üretiminin önemini azaltır ve tabloyu bağlamsal öneminden koparmaktadır. Performans tablolarının yanlış yorumlanmasının yaygın bir nedeni, rakamların izole ve mutlak değerler olarak algılanmasıdır. Bununla birlikte, bu rakamlar belirli bir veri bölümleme yaklaşımı, tanımlanmış bir hiperparametre aralığı, özel bir format ve belirlenmiş bir ölçüt seçimi kullanılarak üretilmiştir. Bu bağlamı göz ardı etmek, tabloda gözlemlenen farklılıkların istatistiksel, metodolojik veya sadece rastgele olup olmadığını belirlemeyi imkansız hale getirmektedir. Hand, kritik bir konuya dikkat çekmekte ve literatürde performans farklılıkları sıklıkla mevcut olarak rapor edilmektedir. Ancak altta yatan nedenler açıklanmamaktadır [1]. Bu durum, aşağıdaki soruların cevapsız kalmasına neden olmaktadır.

Alternatif veri kümeleri kullanılarak tekrarlandığında bu farklılık korunuyor mu?

Standart sapma veya güven aralığı belgelendi mi?

Alternatif ölçütler kullanılarak değerlendirildiğinde aynı strateji "en iyi" olarak kalıyor mu? Yapılandırma değiştirildiğinde tablodaki sıralama geçersiz hale geliyor mu?

Bu sorgulamaların yokluğunda, performans grafiği bilimsel bir değerlendirmeden ziyade cazip bir tabloya dönüşmektedir. Dolayısıyla, metodolojik olgunluk, performans grafiğini kesin ifade olarak değil, söylemin başlangıcı olarak algılamayı gerektirmektedir.

Mühendislik araştırmalarında, aynı yöntemin doğruluğa göre optimal, F1'e göre vasat ve ROC-AUC'ye göre yetersiz olarak değerlendirilmesi normaldir. Bu uyumsuzluk sıklıkla ölçütlerin yanlış uygulanmasından değil, farklı bilimsel

sorgulamaları ele almalarından kaynaklanmaktadır. Bu ölçütlerin her biri performansı farklı bir bakış açısından değerlendirmektedir.

Doğruluk, genel performansı değerlendirir ancak sınıf dengesizliğini hesaba katmaz. F1 puanı, özellikle azınlık sınıfının performansını göstererek, hassasiyet ve geri çağırma arasındaki dengeyi ölçmeyi amaçlamaktadır. ROC-AUC, eşik değerinden bağımsız bir ayırt edici ölçüttür; ancak pratik uygulamalarda kullanılan belirli eşik davranışını gizleyebilmektedir.

Sonuç olarak, aynı model, aynı veri kümesi üzerinde bu üç ölçü kullanılarak değerlendirildiğinde, başarının farklı yorumlarını sağlayabilir. Araştırmada tek bir istatistiğe vurgu yapılıyorsa, şu soru kaçınılmazdır: Bu ölçütü bu konu için en önemli kılan nedir? Bu sorunun bir çözümü olmadığında, ölçüt seçimi, kasıtlı olarak marjinalleştirilmiş hataların belirli biçimlerini gizleyen örtük bir metodolojik seçim haline gelmektedir. Hand'in argümanı burada önem kazanır: ilerleme iddiası, ölçüt seçiminin ürettiği bir yanılsama olabilmaktedir [1]. Daha metodolojik olarak titiz bir yaklaşım benimseyin. Çeşitli ölçütler sunun, bu ölçütler arasındaki uyumsuzluğun nedenlerini açıkça açıklayın, uygulama bağlamında hangi ölçütün daha büyük önem taşıdığını gerekçelendirin. Bu yöntem, performansı tek bir ölçüt olmaktan çıkarıp çok yönlü bir değerlendirme yapısına dönüştürmektedir.

Bu bölümün sonunda, hem yazarlar hem de okuyucular için performans yanılsamalarını azaltmaya yönelik pratik bir çerçeve sunmak faydalı olacaktır. Aşağıdaki sorular, bir performans raporunun bilimsel geçerliliğini değerlendirmek için temel bir kontrol listesi görevi görmektedir:

Bu performansı elde etmek için hangi veri bölme yaklaşımı kullanıldı? Keyfi, zamansal veya bireysel tabanlı mı? Bu strateji sorunla uyumlu mu?

Model seçimi ve hiperparametre ayarlama süreci test verilerinden gerçekten bağımsız mıydı?

Dolaylı uyumluluk potansiyeli var mı?

Sağlanan metrikler, sorunun bağlamında yanlışlığın maliyetini doğru bir şekilde temsil ediyor mu? Yoksa sadece açıklıkları için mi seçildiler?

Sonuçlar alternatif metriklerle önemli ölçüde değişiyor mu? Bunun sonuçları nelerdir?

Performans farklılıkları istatistiksel olarak anlamlı mı, yoksa tek bir yapıya mı sınırlı?

Temel değerler adil ve gösterge niteliğinde mi? Yöntemin sağlamlığı algısını artırmak için mi seçildiler?

Bu performans, veri dağılımındaki varyasyonlara veya gürültüdeki artışa ne ölçüde dayanıklıdır?

Cawley ve Talbot'un saklandığı yerde, bu soruların önemli bir kısmı sorulmadığında, performans tahminleri sistematik olarak abartılabilmekte ve literatürde yanıltıcı bir ilerleme izlenimi meydana gelebilmektedir [1]. Bu nedenle performans raporlarının bu tür bir sorgulamanın filtrelenmeden devam etmesi, metodolojik olgunluğun önemli bir göstergesidir.

Bu geliştirmelerle performans, "kaç puan kazandık?" sorusunun ötesine geçerek "bu puanın ne önemi var?" sorusunun incelenmesine dönüşmektedir. Performans yanılsamalarını görünür kılmak, mühendislik araştırmaları için bir engel değil; aksine, sağlamlığını, taşınabilirliğini ve güvenilirliğini artıran temel bir gerekliliktir. Gerçek ilerleme, sayısal genişlemede değil, sayısal önemin derinlemesine geliştirilmesinde bulunmaktadır [1], [9].

4. Metodolojik Hatalar: Standartlaştırılmış Ancak İncelenmemiş Uygulamalar

Bu bölümün temel iddiası basit ama rahatsız edicidir:

Mühendislik araştırmalarında gözlemlenen hataların önemli bir kısmı karmaşık veya yetersiz modellerden değil, eleştirel bir şekilde uygulanmayan rutinlerden kaynaklanmaktadır. Bu rutinler zamanla o kadar yerleşir ki, çoğu araştırmacı bunları metodolojik tercihler olarak görmez. Ancak, en büyük bilimsel hatalar tam da bu noktada meydana gelir.

Bu kör noktalar genellikle kötü niyetli değildir. Araştırmacı, literatürde yaygın olanı tekrarlayarak doğru yaptığını iddia eder. Bununla birlikte, veri odaklı araştırmalarda, küçük bir tasarım hatası performansı yapay olarak artırabilmekte ve yöntemin gerçek dünya uygulamalarındaki etkinliğinin yanıltıcı bir temsilini yaratabilmektedir. Sonuç olarak, metodolojik kör noktalar yalnızca teknik bir sorun değil, aynı zamanda bilimsel güvenilirliğe de bir meydan okuma oluşturmaktadır. Bu bölüm, literatürde yaygın olarak görülen ve genellikle standart uygulama olarak kabul edildiği için sorgulanmadan kabul edilen uygulamaları ele almaktadır. Her biri uygun şekilde kullanıldığında güçlü bir araç görevi görmekte; tersine, yanlış uygulandığında sonuçları sistematik olarak çarpıtan bir mekanizmaya dönüşmektedir.

4.1. Veri Bölme Stratejisi: Bağımsız ve özdeş dağılımlı Varsayımı Gizli Bir Tehlikedir

Rastgele eğitim-test bölme, çağdaş makine öğrenimi ve mühendislik literatüründe baskın değerlendirme yaklaşımıdır. Bu metodoloji, verilerin bağımsız ve özdeş dağılımlı olduğu varsayımına dayanmaktadır. Her örnek diğerlerinden bağımsızdır ve aynı dağılımdan kaynaklanmaktadır. Bu varsayım sağlandığında rastgele bölme, modelin genelleştirilebilirliğini doğru bir şekilde tahmin edebilmektedir ve çok sayıda mühendislik probleminde açıkça yanlıştır. Zaman serileri, bireysel ölçümler, mekânsal veriler, sensör ağları veya tekrarlanan deneylerden oluşan veri kümelerinde, örnekler birbirine bağımlıdır. Aynı kaynaktan alınan ölçümler, benzer gürültü yapıları ve karşılaştırılabilir desenler sergilemektedir. Rastgele bölümlendirme sırasında, bazı benzer örnekler eğitime atanırken diğerleri teste gönderilir; bu da modelin gerçek anlamda genelleme yapmadan yüksek performans sergilemesini sağlamaktadır.

Bu senaryo, veri sızıntısına benzer bir etki yaratır. Model, gerçek test verilerinin kendisi yerine, test verilerine çok benzeyen eğitim örneklerini kullandığında etkili görünür. Gerçekte, bu ortaklık olmadığında performans önemli ölçüde düşer. Temel husus, veri bölme yaklaşımının yalnızca teknik bir özellik olarak değil, araştırma için bir gerçekçilik filtresi olarak da hizmet etmesidir. Zamansal bir sorunda zamansal bölme ihmal edilirse, bireysel tabanlı bir değerlendirmede bireysel tabanlı ayrım kullanılmazsa ve mekansal bir sorunda mekansal bağımlılıklar göz ardı edilirse ortaya çıkan performans rakamları bilimsel olarak geçerli değildir. Sonuç olarak, veri bölme yalnızca "kodda küçük bir başlangıç adımı değildir; en önemli metodolojik hususlardan birini oluşturur.

4.2. Çapraz Doğrulama: İyi Uygulandığında Bir Çözüm, Yanlış Uygulandığında Bir Zarar

Çapraz doğrulama, özellikle kısıtlı veri bağlamlarında model seçimi ve performans değerlendirmesi için çok önemli bir araçtır. Kohavi'nin öncü araştırması, çapraz doğrulamanın tek bir eğitim-test bölümüne göre daha az varyasyon ve daha güvenilir performans değerlendirmeleri sağladığını göstermiştir [10]. Sonuç olarak, çapraz doğrulama akademik söylemde sıklıkla "güvenilir standart" olarak kabul edilir.

Bununla birlikte, çapraz doğrulama, uygulanma biçimine bağlı olarak önemli hatalara yol açabilir. Yaygın bir sorun, çapraz doğrulamanın hiperparametre optimizasyonu ve özellik seçimi süreçleriyle yetersiz entegrasyonudur. Özellik seçimi veya ölçeklendirme gibi işlemlerin çapraz doğrulamadan önce tüm veri

kümesi üzerinde gerçekleştirilmesi, test alt kümelerine bilgi sızıntısına neden olabilmektedir.

Cawley ve Talbot, bu tür hataların model seçim sürecinde seçim yanlılığına yol açtığını sistematik olarak göstermiştir [9]. Model, çapraz doğrulama sırasında *test* olarak sınıflandırılan verileri dolaylı olarak gözlemler. Bu, performans beklentilerinin sistematik olarak aşırı tahmin edilmesine neden olur. Şu anda çapraz doğrulama iki uç nokta arasında yer almaktadır:

İyi tasarlandığında: Model seçimi için güçlü ve güvenilir bir araç görevi görür.

Uygunsuz tasarlandığında: En tehlikeli metodolojik tuzaklardan birine dönüşür.

Sonuç olarak, çapraz doğrulama otomatik olarak yürütülen bir prosedür değildir; titiz bir tasarım gerektiren metodolojik bir tercihtir. Katmanlı çapraz doğrulama gibi metodolojiler bu sorunları gidermek için önerilmiş olsa da bunların uygulanmasının gerekçesi ve zamanlaması çoğu çalışmada yeterince incelenmemiştir.

4.3. Hiperparametre Optimizasyonu: Arama tasarımında yaklaşımınız

Hiperparametre optimizasyonu, çağdaş mühendislik ve makine öğrenimi araştırmalarında performansın kritik bir belirleyicisidir. Bununla birlikte, hiperparametre optimizasyonu sıklıkla temel bir prosedür olarak görülmekte ve ayrıntıları raporlardan sıklıkla çıkarılmaktadır. Arama tasarımı, ortaya çıkan performansın yorumlanmasını doğrudan etkilemektedir. Bergstra ve Bengio, rastgele aramanın birçok gerçekçi senaryoda ızgara aramasından daha iyi performans gösterebileceğini göstermiştir [11]. Bununla birlikte, arama yönteminin seçimi tek önemli faktör değildir. Temel sorular:

Arama alanının kapsamı neydi? Denenen toplam kombinasyon sayısı neydi? Durdurma kriteri neydi? Temel yöntemlere de aynı bütçe tahsis edildi mi? Bu sorulara verilen yanıtlar, karşılaştırmanın adilliğini belirler. Önerilen yaklaşım geniş bir arama alanında değerlendirilirken temel yöntemler sınırlı parametrelerle kısıtlanıyorsa gözlemlenen performans farklılığı öncelikle metodolojik farklılıklardan ziyade optimizasyon varyanslarına atfedilebilmektedir.

Sonuç olarak, hiperparametre araması yalnızca verimliliği artıran bir özellik değil; araştırmanın adilliğini ve karşılaştırılabilirliğini belirleyen kritik bir metodolojik tercihtir. İyi tanımlanmış bir arama stratejisinin olmaması, performans sonuçlarının yorumlanmasını büyük ölçüde engellemektedir.

4.4. Ön İşleme Sızıntısı: En Aldatıcı Zararsız, Maliyetli Hata

Ölçeklendirme, PCA ve özellik seçimi gibi ön işleme prosedürleri sıklıkla modelden bağımsız ve zararsız işlemler olarak görülmektedir. Literatürde yaygın bir hata, bu yöntemlerin veri setinin tamamına, eğitim ve test bölümlerine ayrılmadan önce uygulanmasıdır. Bu yöntem, test verilerinden istatistiklerin (ortalama, varyans, bileşen yönleri, seçilen özellikler) eğitim sürecine sızmasına neden olmaktadır. Sonuç olarak, model test verileriyle ilgili dolaylı bilgiler kullanılarak eğitilmektedir.

Bu tür sızıntılar, kodun işlevselliği ve yüksek verimliliği nedeniyle sıklıkla tespit edilememektedir. Sonuç olarak, ön işleme sızıntıları, literatürde tespit edilmesi zor, ancak etkili bir yanlışlığı temsil etmektedir. *Model harika çıktı* denildiğinde, *tasarımda bir sorun vardı* çıkarımı ortaya çıkar. Bu hata, özellikle sınırlı veri kümeleri ve yüksek boyutlu sorunlarda performansı önemli ölçüde bozabilmektedir.

Metodolojik açıdan sağlam yaklaşım tartışmasızdır: Ön işleme adımları eğitim verilerinden türetilmeli ve yalnızca test verilerine uygulanmalıdır. Bu kriterin ihlali, elde edilen sonuçların bilimsel güvenilirliğini önemli ölçüde zayıflatmaktadır.

Bu bölümde incelenen kör noktaların ortak özelliği, hiçbirinin gelişmiş hatalar olarak nitelendirilememesidir. Bunun yerine, çoğunluğu yaygın olarak benimsenen tekniklerdir. Ancak, bu durum onları tehlikeli hale getirmektedir. Sorgulanmadıkları takdirde, aldatıcı performansları kurumsallaştırırlar. Mühendislik araştırmalarında metodolojik olgunluk, yalnızca giderek daha karmaşık modellerin geliştirilmesiyle değil, aynı zamanda hangi rutinlerin gerçekten güvenli, hangilerinin ise rastlantısal olduğunu ayırt etme yeteneğiyle de elde edilmektedir. Bu ayrımı yapabilen araştırmalar, yalnızca üstün performans değil, aynı zamanda daha güvenilir araştırmalar da ortaya koymaktadır.

5. Değerlendirici-Yazar-Okuyucu Dinamik Üçgeninde Metodolojik Seçimler

Mühendislik araştırmaları sıklıkla teknik bir çaba olarak nitelendirilir. Bir model oluşturulur, deneyler yapılır ve bulgular yayılır. Bununla birlikte, bu teknik cephenin altında, çoğu zaman yeterince açık olmayan bir gerçeklik yatmaktadır. Bilimsel çaba aynı zamanda bir tür iletişimdir. Bu iletişim, yazarın niyetleri, değerlendiricinin yeterlilik kriterleri ve okuyucunun metni anlaması arasındaki karmaşık etkileşimden etkilenmektedir.

Yazar, inceleyici ve okuyucu olmak üzere bu üç katılımcının beklentileri her zaman örtüşmez. Aksine, bu beklentiler arasında sıklıkla çatışma vardır. Bu gerilim, metodolojik seçimlerin neden sıklıkla gözden kaçtığına dair önemli bir iç görü sağlamaktadır. Metodolojik yargılar hem teknik hem de sosyal faktörlerden etkilenir; neyin dahil edileceği, neyin hariç tutulacağı ve neyin aşırı ayrıntı teşkil edeceği bu çerçeve içinde belirlenir.

Metodolojik seçimlerin belirsizliği yalnızca kişisel eğilimlerden değil, aynı zamanda akademik üretimin yapısal dinamikleri tarafından yönlendirilen toplumsal bir süreçten de kaynaklanmaktadır. Ioannidis'in bilimsel yayıncılığa yönelik eleştirileri, bu mekanizmanın sistematik işleyişini açıkça göstermektedir [5].

5.1. Hakem hangi yönleri inceler?

Hakem ideal olarak bilimde kalite kontrol aracı görevi görür. Uygulamada, hakem zaman ve bağlam bilgisi kısıtlamaları dahilinde bir çalışmanın "bilimsel güvenilirliğini" belirlemeye çalışmaktadır. Sonuç olarak, hakem değerlendirmesi sıklıkla hızlı güven göstergeleri aracılığıyla ilerler. Gerçekte, bu güven göstergeleri genellikle şunları kapsamaktadır: Açık ve anlaşılır bir problem tanımı, literatürde kabul görmüş ve yeterli temel ölçütler, güçlü ve yaygın olarak kabul görmüş performans ölçütleri, denemelerin tekrarlanabilirliğini gösteren net bir metodoloji. Bu özellikler, çalışmanın ciddiyetini ve alan normlarına uygunluğunu değerlendirici açısından yansıtmaktadır. Ancak burada temel bir çelişki ortaya çıkar: Yayın baskısı ve değerlendirme stresi arttıkça, performans ölçütleri bu listeyi gölgede bırakmaya başlamaktadır.

Üstün performans, sistematik gerekçelendirmeden daha hızlı ikna edici olabilmektedir. Bu, Ioannidis'in belirlediği temel bir sorunu örneklemektedir. Bilimsel güven bazen sağlam yöntemlere değil, ikna edici sonuçların sunulmasına dayanmaktadır [5]. Değerlendirici, her yaklaşımı titizlikle inceleme kapasitesine sahip değildir; bu nedenle makul derecede sağlam bir yapıya sahip ve olumlu sonuçlar veren çalışmalar tercih edilir. Bu durum, istemeden de olsa metodolojik seçimler üzerine kapsamlı bir tartışmayı engelleyebilmektedir.

5.2. Yazar hangi eylemleri gerçekleştirmek zorundadır?

Yazarın bakış açısından, bu üçgenin baskısı çok daha acildir. Yazar, çalışmalarının kabul görmesini ister ve alanın tercihlerini yavaş yavaş kavrar. Bu öğrenme genellikle açık kurallar yoluyla değil, kabul edilen veya reddedilen makaleler aracılığıyla gerçekleşir. Bu aşamada yazar şu gibi içsel sorgulamalarla karşı karşıya kalır:

Bu yöntemi ayrıntılı bir şekilde açıklarsam, hakem bunu 'gereksiz' bulacak mı?

Alternatif çerçeveleri araştırırsam çalışmam yetersiz görünecek mi?

Bu performans yetersiz kalırsa, metodolojik katkıya ilgi duyulacak mı?

Sorgulamalar yazarın davranışlarını etkiler. Yazar, çalışmalarının tanınmasını sağlamak için beklenen olay örgüsüne kademeli olarak uyum sağlamak zorunda hisseder. Bu durum metodolojik çeşitliliği azaltır; riskli ancak bilimsel açıdan önemli tartışmalara girmek yerine, tekrar tekrar gözlemlenen "güvenli formatlar" baskın hale gelmektedir.

Söz konusu mesele bireysel değil, sistemiktir. Yazar sıklıkla kötü niyetten değil, hayatta kalma içgüdüsünden hareket etmektedir. Bu tepki, metodolojik seçimlerin zaman içinde literatürde tekrarlayıcı, yüzeysel ve kabul edilmemiş olarak kalmasının nedenini açıklamaktadır. Ioannidis, yayın sisteminin belirli sonuç türlerini teşvik ettiğini ve buna bağlı olarak araştırma davranışında bir evrime yol açtığını belirtmektedir [5].

5.3. Okuyucu nelerden vazgeçer?

Okuyucu, bu dinamikte en az etkili katılımcı rolünü üstlenirken, en büyük yükü de taşır. Bir okuyucu yayınlanmış bir esere atıfta bulunduğunda, bulguları kendi ortamına uygulamaya çalışır. Bununla birlikte, metodolojik seçimlerin gerekçeleri açıkça ifade edilmediğinde, bu aktarım son derece zorlaşır.

Okuyucu bu soruların cevaplarını bulamayacaktır: Bu stratejinin etkili olduğu koşullar nelerdir? Bu performans verimliliğimle ilgili mi? Hangi varsayımlar önemli, hangileri ikincil? Sonuçların bu şekilde ortaya çıkmasına ne sebep oldu?

Bu belirsizlik, zamanla literatürü tekrarlanabilir ancak anlaşılmaz bir çerçeveye dönüştürüyor. Kodlar dağıtılabilir ve modeller yeniden eğitilebilir; ancak işlevselliğinin ardındaki mantık ve başarısız olduğu koşullar belirsiz kalır. Bu, özellikle disiplinlerarası mühendislik çalışmalarında önemli bir sorun teşkil etmektedir. Metodolojiler benimsenir, ancak sonuçları aktarılmaz. Metodolojik şeffaflık, okuyucu için hem entelektüel bir ihtiyaç hem de pratik bir gerekliliktir. Okuyucu, bu materyali kendi sistemlerine uyacak şekilde sık sık değiştirmek zorundadır. Metodolojik kararlardaki şeffaflık eksikliği, bu uyarlama sürecini istikrarsız ve etkisiz hale getirmektedir.

5.4. Üçgeni Yönetmek: Metodolojik Şeffaflık Bir Yükümlülük Değil, Bir Avantajdır

Değerlendirici-yazar-okuyucu üçgeni kaçınılmazdır; ancak kontrol edilebilirdir. Üçgeni yönetmenin temel yaklaşımı, metodolojik kararları bilinçli olarak şeffaf hale getirmektir.

Bu, sonuç arayışından vazgeçmek anlamına gelmez. Aksine, sonuçları bağlamlarıyla birlikte iletmek anlamına gelmektedir. Özellikle, sadece "ne başarıldı?" sorusunu değil, aynı zamanda şunları da kapsar:

Bu sonucu doğrulayan varsayımlar nelerdir?

Hangi koşullar bunun bozulmasına yol açabilir?

Farklı seçeneklerle koşullar ne şekilde değişir?

Bu yöntem kısa vadede ek yazma ve açıklama gerektirebilir; ancak uzun vadede çalışmanın bilimsel değerini artırmaktadır. Metodolojik görünürlük, çalışmayı savunmasızlığa maruz bırakmak yerine savunulabilirliğini artırmaktadır. Bu, değerlendirici için güveni artırır ve okuyucu için önem yaratır. Ioannidis'in eleştirilerine yanıt olarak, bilimsel ilerlemenin yalnızca daha fazla bulgu üretmekle değil, aynı zamanda bu bulguların güvenilirliğini sağlamakla da gerçekleştirilebileceği açıktır [5]. Metodolojik seçimleri açıklayan araştırma, yalnızca teknik bilgiler sağlamakla kalmaz, aynı zamanda literatürün bütünlüğünü de artırır.

Değerlendirici-yazar-okuyucu üçgeni, metodolojik seçimlerin sonucunu etkileyen önemli bir sosyal çerçevedir. Bu bağlamda, metodolojik görünürlük sıklıkla bir risk olarak görülür; ancak aslında bilimsel olgunlaşmanın çok önemli bir göstergesidir. Mühendislik araştırmalarındaki önemli ilerlemeler, yalnızca gelişmiş performansla değil, aynı zamanda hedef kitleyi, bağlamsal koşulları ve bu performansın öneminin ardındaki mantığı açıkça ifade ederek de gerçekleştirilebilmektedir.

6. Etik ve Epistemolojik Açılardan Metodolojik Hususlar: Neyi Açıklamalı, Neyi Gizlemeliyiz?

Bilimsel etik, sıklıkla veri izinleri, katılımcı onayı ve intihal gibi konularla sınırlı kalmaktadır. Bu konular şüphesiz önemlidir; ancak veri odaklı mühendislik ve bilgisayar bilimlerindeki etik ikilemlerin önemli bir kısmı çok daha erken ve daha incelikli bir şekilde ortaya çıkamaktadır. Bilgisayar bilimleri ve mühendisliğinde geliştirilen bilgi, çoğu zaman doğrudan uygulamaya dönüştürülme potansiyeline sahiptir. Bir sınıflandırma sistemi, bir karar destek

mekanizması, bir optimizasyon algoritması veya bir tahmin modeli, sadece akademik bir ürün olmaktan çıkıp, gerçek sistemlerdeki davranışı etkileyen bir bileşen haline gelebilir. Bu nedenle, metodolojik kararların etik boyutu, yalnızca doğruyu söylemek açısından değil, aynı zamanda neyi görünür kıldığımız ve neyi gizli bıraktığımız açısından da ele alınmalıdır.

Bu noktada, epistemoloji (bilginin üretimi ve tanımı) ve etik doğrudan kesişmektedir. Sonuçların bilimsel olarak anlamlı veya önemsiz olarak sınıflandırılması ve raporlardan dışlanması, literatürde belgelenen gerçekleri ve sistematik olarak göz ardı edilenleri belirlemektedir.

6.1. Seçici Raporlama ve İlerleme Görünümü

Seçici raporlama, veri odaklı bilimde yaygın ancak yeterince incelenmemiş bir etik ikilem oluşturmaktadır. Bu sorun sıklıkla kasıtlı değildir; ancak etkileri son derece sistematiktir. Çok sayıda deney yapılır, çeşitli tasarımlar ve ortamlar test edilir, ancak yalnızca olumlu sonuçlar üreten senaryolar belgelenir. Başarısız deneyler, olumsuz sonuçlar veya performans düşüşü senaryoları ya tamamen atlanır ya da metnin kenarına itilir.

Fanelli'nin çeşitli alanları kapsayan araştırması, yıllar boyunca olumsuz sonuçlarda sistematik bir düşüş olduğunu göstermektedir [12]. Bu, literatürde yapay bir başarı atmosferi yaratmakta ve her yeni yöntemin bir öncekinden daha iyi olduğunu ve sorunların giderek daha fazla çözüldüğünü öne sürmektedir. Bu algı, sıklıkla gerçek ilerlemeden değil, raporlama filtresinden kaynaklanmaktadır. Bu, özellikle bilgisayar bilimi açısından çok önemlidir. Çünkü aynı veri kümesine çok sayıda teknik uygulanabilir, hiperparametre arama alanı geniştir, yapılandırmadaki küçük ayarlamalar sonuçları önemli ölçüde etkileyebilmektedir. Bu ortamda yalnızca en iyi sonucu raporlamak, farkında olmadan şu mesajı üretir: Bu strateji her durumda etkilidir.

Ancak gerçekte, bu strateji yalnızca belirli durumlarda etkili olmuş olabilir. Olumsuz sonuçların olmaması, acemi araştırmacıları yanıltır, çünkü başarısızlıklar literatürde gizlenir. Bu durum, alanın kendi kendini düzeltme yeteneğini zayıflatır, çünkü hangi kavramların etkisiz olduğunu ve başarısızlık nedenlerini ayırt edemez. Etik kaygı, yanlış veri uydurmakta değil, aksine gerçeklerin seçici bir şekilde ifşa edilmesinde yatmaktadır. Bu, epistemik açıdan bilginin sistematik olarak çarpıtılması anlamına gelir. Literatür, gerçek deneyim alanının yalnızca dikkat çekici bir alt kümesini yansıtmaktadır.

6.2. Uygulamalarda Etik Yükümlülükler: Bilgisayar Bilimindeki Hataların Finansal Sonuçları

Bilgisayar bilimi ve mühendisliğindeki metodolojik hataların etik sonuçları sıklıkla pratik uygulamalarda kendini göstermektedir. Sağlık, çevre bilimi, güvenlik, askeriye, siber güvenlik ve altyapı yönetimi gibi alanlarda oluşturulan algoritmalar yalnızca bilimsel incelemenin konusu değildir; karar verme, kaynak dağıtımı ve insan refahını önemli ölçüde etkileyebilirler. Metodolojik kararların etik sonuçları üç boyutta ortaya çıkar:

Yanlış Güven Oluşturma: Yüksek performans raporları, metodolojik güvenilirlik izlenimi yaratır. Performansı açıkça ifade edilmeyen belirli varsayımlara yakından bağlıysa, sistem gerçek dünya senaryolarında öngörülemeyen arızalarla karşılaşabilir. Bu arıza, özellikle sağlık ve güvenlik alanlarında, anında zararlı sonuçlara yol açabilmektedir.

Kapsama Yanlılığı: Veriye dayalı metodolojiler, temsil edilmeyen durumlarda yetersiz performans gösterir. Metodolojik değerlendirmelerde belirli kullanıcı gruplarının, koşulların veya veri dağılımlarının dışlanması, sistemin yalnızca "algıladığı dünya" için optimize edilmesi nedeniyle etik bir endişe oluşturmaktadır.

Sorumluluk Belirsizliği: Yöntem seçimleri belirsizleştiğinde, hatalardan kaynaklanan sorumluluk dağılır. *Model bu davranışı sergiledi* veya *veriler buna benziyordu* gibi ifadeler, önceki aşamada yapılan yöntem seçimlerini etkili bir şekilde belirsizleştirir. Etik görev, yalnızca sonuçla değil, aynı zamanda bu sonucu kolaylaştıran kararlarla da başlar.

Sonuç olarak, yöntem seçimlerini savunmak yalnızca bilimsel şeffaflık meselesi değil, aynı zamanda etik bir yükümlülüktür. Bilgisayar biliminde etik, yalnızca "önyargıyı azaltmak" veya adil algoritmalar tasarlamaktan daha fazlasını içerir; altında çalıştığımız varsayımları açıkça ifade etmek de bu etik çerçeve için çok önemlidir.

6.3. Epistemolojik Bütünlük: Cehaletimizi Dile Getirmek

Bu bölümün en önemli yönü şöyledir: Bilimsel bütünlük, yalnızca bilgimizi doğru bir şekilde belgelemekle kalmaz, aynı zamanda belirsizliklerimizi de dürüstçe kabul etmeyi gerektirir.

Veri odaklı mühendislikte, bilinmeyenler sıklıkla şu şekillerde ortaya çıkar: •Modelin başarısız olduğu durumlar, etkisiz olduğu veri türleri, kararsızlığa neden olan parametre aralıkları,

Hangi senaryolar test edilmeden kaldı?

Bu endişelerin belgelenmemesi çalışmayı geliştirmez; sadece onu savunmasız hale getirir. Fanelli, olumsuz ve belirsiz sonuçların sistematik olarak dışlanmasının bilimin kendi kendini düzeltme mekanizmasını baltaladığını ileri sürmektedir [12]. Bu teknik, yöntemlerin hızlı yayılması, sık sık yeniden kullanılması ve bağlam dışı kullanılma eğilimi nedeniyle bilgisayar biliminde çok önemlidir. Epistemolojik açıdan bakıldığında, *bilmiyoruz* demek bir eksiklik değil; bilmenin sınırlarının açık bir şekilde belirlenmesini ifade eder. Bu, özellikle mühendislikte önemli bir özelliktir, çünkü mühendislik bilgisi yalnızca teorik kesinliğe değil, aynı zamanda yönetilen belirsizliğe de dayanmaktadır.

6.4. Etik-Yöntem İlişkisinin Açıklığa Kavuşturulması

Bu bölümün temel iddiası, metodolojik değerlendirmelerin etik söylemin ayrılmaz bir parçası olduğudur. Bilgisayar bilimleri ve mühendisliğinde etik, *nihayetinde adil miyiz?* sorusunun ötesine geçer. Bu geçiş şu gibi sorularla başlar:

Hangi sonuçları açıkladık?

Hangi senaryolar hiç test edilmedi?

Hangi hataları fark edilmez hale getirdik?

Bu soruları açıkça ele alan araştırmalar, yalnızca daha iyi etik sonuçlar değil, aynı zamanda daha güvenilir ve somut bilimsel sonuçlar da doğurmaktadır. Bunun nedeni, bu tür çalışmaların okuyucuya yalnızca neyi başardığını değil, aynı zamanda nerede başarısız olduğunu da göstermesidir. Bilgisayar bilimi açısından bakıldığında, temel yükümlülük yalnızca etkili algoritmalar geliştirmek değil, aynı zamanda bu algoritmaların içerdiği ve içermediği gerçekleri şeffaf bir şekilde ortaya koymaktır.

7. Sonuç, Tartışma ve Geleceğe Yönelik Bakış Açıları: Üstün Modelden Geliştirilmiş Bilime Geçiş

Bu bölümde dile getirilen temel tez, mühendislikte metodolojiyi yalnızca algoritmalar aracılığıyla tanımlamanın disiplini kısıtladığıdır. Metodoloji, problem formülasyonu, veri gösterimi, değerlendirme süreci, ölçü seçimi, karşılaştırılabilirlik ve etik raporlamadan oluşan kapsamlı bir çerçevedir. Bu çerçevenin herhangi bir unsuru göz ardı edildiğinde, sonuçlar teknik olarak dikkat çekici olsa da bilimsel olarak zayıf hale gelir.

Bu hassasiyet, özellikle metodolojilerin hızla çoğaldığı, kolayca yeniden uygulandığı ve sıklıkla uygunsuz şekilde kullanıldığı bilgisayar bilimleri ve veri merkezli mühendislik alanlarında belirgindir. Metodolojik seçimlerin belirsizliği

hem bireysel çalışmaların hem de daha geniş bilgi birikiminin güvenilirliğini zayıflatmaktadır. Bu bölüm, yalnızca daha iyi bir model arayışının neden yetersiz olduğunu açıklamak ve daha iyi bilimin özünü tanımlamak için önceki argümanları bir araya getirmeyi amaçlamaktadır.

7.1. Söylemin özü: Bu bölüm hangi açıklamayı sağladı?

Bu kitap bölümündeki ana tartışmalar birçok önemli temaya odaklanmaktadır. Başlangıçta, tek başına "olumlu sonuçlar" yetersizdir; bağlam gereklidir. Yüksek performans ölçütleri, ancak altta yatan varsayımlar açıkça ifade edildiğinde anlamlıdır. Aksi takdirde, performans bağlamdan yoksun bir rakama indirgenir ve yanıltıcı bir güven duygusuna yol açar. Bu, özellikle kıyaslama kültürünün hakim olduğu alanlarda "ilerleme" kavramının sık sık incelenmesinin gerekliliğini göstermektedir [2].

İkinci olarak, metodolojik yargılar şeffaf hale getirildiğinde, çalışma azalmaz; aksine güçlenir. Literatürdeki baskın inancın aksine, varsayımların ve sınırların açıkça belirlenmesi çalışmanın bütünlüğünü tehlikeye atmaz. Bu şeffaflık, çalışmanın geçerliliğini ve zayıf yönlerini ortaya koyarak güvenilirliğini artırır. Box'un ünlü iddiası bir kez daha önem kazanır: modellerin kullanımı, mutlak doğruluklarından değil, etkili oldukları koşulların anlaşılmasından kaynaklanmaktadır [7].

Üçüncüsü, literatürde gerçek ilerleme, yalnızca çalışma sayısını artırmakla değil, belirsizlikleri gidermekle de elde edilir. Breiman, veri odaklı modelleme alanında ilerlemelerin sıklıkla performans iyileştirmelerine göre değerlendirildiğini ileri sürmektedir [2]. Bununla birlikte, bu iyileştirmelerin metodolojik mi yoksa tasarıma dayalı mı olduğu incelenmediğinde, ilerleme anlatısı bilimsel bir yanılgıya dönüşebilir. Sonuç olarak, gerçek ilerleme belirsizliklerin bastırılmasıyla değil, bunların görünür hale getirilmesiyle elde edilmektedir.

Toplu olarak ele alındığında, bu üç husus, üstün model hedefinin yerini geliştirilmiş bilim hedefinin alması gerektiğini açıkça göstermektedir. Geliştirilmiş bilimsel araştırma, daha karmaşık algoritmaların uygulanmasından ziyade, daha bilinçli metodolojik kararlar alınmasını gerektirmektedir.

7.2. Gelecek Araştırma Yönleri: Nereye Gitmeliyiz?

Bu bölümde ele alınan konular, mühendislik ve bilgisayar bilimleri için bir son noktayı temsil etmemekte; aksine, yeni bir başlangıç noktası sunmaktadır. Aşağıda vurgulanan gelecek araştırma yönlerinden bazıları, bu kitap bölümünde savunulan yaklaşımın doğal uzantıları olarak düşünülebilir.

7.2.1. Metodolojik Raporlama Standartları

Gelecekteki araştırmalar için kritik bir gereklilik, metodolojik seçimlerin belgelenmesi için daha tekdüze ve standartlaştırılmış kriterlerin oluşturulmasıdır. Hem "ne yapıldı" hem de "neden yapıldı" düzeylerinde sorun formülasyonu, veri bölümleme tekniği, ölçüt seçimi ve temel tercihler hakkındaki değerlendirmelerin belgelenmesi, literatürün kalitesini artıracaktır.

Bu tür yönergeler, araştırmacının yaratıcılığını kısıtlamak yerine artırmak ve metodolojik seçimlerin ardındaki mantığı desteklemek amacıyla formüle edilmelidir. Amaç, bir kontrol listesini dayatmak değil, bilinçli farkındalığı geliştirmektir.

7.2.2. Etik Karşılaştırma

Karşılaştırma, mühendislik çalışmalarının temelidir; ancak etik sonuçları sıklıkla ihmal edilir. Temel referans noktalarının seçimi, hiperparametre ayarlaması için ayrılan bütçe ve veri bölümleme yöntemlerinin eşitliği, karşılaştırmaların bilimsel geçerliliğini önemli ölçüde etkiler. Gelecekteki araştırmalar, "Kim daha üstün sonuç elde etti?" sorusundan ziyade, "Karşılaştırma ne kadar adildi?" sorusunu özellikle ele almalıdır. Bu yöntem, literatürün güvenilirliğini artıracak olsa da, çalışma hızını yavaşlatabilmektedir.

7.2.3. Genelleştirilebilirlik Kültürü

Tek bir kıyaslama testinde olağanüstü performans, her zaman genelleştirilebilirliğin bir işareti olarak kabul edilmiştir. Bu yöntem, veri odaklı mühendislikte yetersiz kalmaktadır. Gerçekte, sistemler istikrarlı ve net kıyaslama testlerinde değil, değişken, kusurlu ve gürültülü koşullarda çalışır.

Sonuç olarak, gelecekteki araştırmalar bireysel kıyaslamalara güvenmek yerine senaryo tabanlı değerlendirmeler uygulamalıdır. Yöntemlerin performansının farklı veri dağılımları, gürültü seviyeleri ve uygulama senaryoları genelinde gösterilmesi, genelleştirilebilirlik iddiasını destekleyecektir.

7.2.4. Olumsuz Sonuçların Önemi

Bilgisayar bilimleri literatüründe, "işe yaramadı" terimi sıklıkla ima edilir ancak açıkça belirtilmez. Bununla birlikte, olumsuz sonuçlar, belirli metodolojilerin başarısızlığının ardındaki nedenleri açıklayarak disiplinin kolektif anlayışını geliştirir. Bu sonuçların sistematik olarak belgelenmesi ve yaygınlaştırılması, literatürdeki yapay başarı atmosferini azaltabilir. Kötü sonuçların önemi sadece etik bir kaygı değil; epistemik bir zorunluluktur. Bilginin sınırları sıklıkla başarısızlıklarla belirlenmektedir.

7.3. Daha iyi bilimin tanımı nedir?

Bölüm, üstün bilimin giderek daha karmaşık modellerin oluşturulmasını gerektirmediği sonucuna varmaktadır. Üstün bilim, belirli bir araştırmanın seçilmesinin, belirli bir veri noktasının kullanılmasının, belirli bir istatistiğin seçilmesinin ve belirli bir sonucun raporlanmasının ardındaki mantığı açıkça tanımlayabilme yeteneğini gerektirir.

Bilgisayar biliminde ve mühendisliğinde metodolojik seçimler çok önemlidir. Bu kararları ortadan kaldırmak mümkün olmasa da onları bilinçli, açık ve tartışmaya açık hale getirmek mümkündür. Bu şeffaflık bilimsel ilerlemeyi engellemez; aksine sağlamlığını ve sürdürülebilirliğini artırır. Breiman'ın "iki kültür" ile Box'un model felsefesi arasındaki ayrımı net bir mesaj iletiyor: Bilimsel değer yalnızca elde edilen sonuçlarda değil, aynı zamanda bu sonuçların önem taşıdığı koşulları açıkça ortaya koyma cesaretinde de yatmaktadır [2], [7].

Sonuç olarak, mühendislik ve bilgisayar bilimlerinde gerçek ilerleme ancak *üstün modeli kim geliştirdi?* sorusunun ötesine geçtiğimizde mümkün olabilir. Tekniği sadece bir algoritma olarak değil, bütünleşik bir epistemolojik, etik ve metodolojik çerçeve olarak gören araştırmalar, literatürün gidişatını şekillendirecektir. Bu bölümün temel amacı, modelleri iyileştirmekten ziyade bilimsel araştırmayı geliştirmektir.

Kaynaklar

[1] D. J. Hand, "Classifier Technology and the Illusion of Progress," *Statistical Science*, vol. 21, no. 1, Feb. 2006, doi: 10.1214/088342306000000060.

[2] L. Breiman, "Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)," *Statistical Science*, vol. 16, no. 3, Aug. 2001, doi: 10.1214/ss/1009213726.

[3] I. Guyon and A. M. De, "An Introduction to Variable and Feature Selection André Elisseeff," 2003.

[4] B. D. McCullough, "Measurement Theory and Practice: The World Through Quantification," *J Am Stat Assoc*, vol. 100, no. 472, pp. 1462–1463, Dec. 2005, doi: 10.1198/jasa.2005.s56.

[5] J. P. A. Ioannidis, "Why Most Published Research Findings Are False," *PLoS Med*, vol. 2, no. 8, p. e124, Aug. 2005, doi: 10.1371/journal.pmed.0020124.

[6] D. Shapere, "The Structure of Scientific Revolutions," *Philos Rev*, vol. 73, no. 3, p. 383, Jul. 1964, doi: 10.2307/2183664.

[7] G. E. P. Box, "Science and Statistics," *J Am Stat Assoc*, vol. 71, no. 356, pp. 791–799, Dec. 1976, doi: 10.1080/01621459.1976.10480949.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY: Springer New York, 2009. doi: 10.1007/978-0-387-84858-7.

[9] G. C. Cawley and N. L. C. Talbot, "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation," 2010.

[10] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," 1995. [Online]. Available: http://robotics.stanford.edu/~ronnyk

[11] J. Bergstra, J. B. Ca, and Y. B. Ca, "Random Search for Hyper-Parameter Optimization Yoshua Bengio," 2012. [Online]. Available: http://scikit-learn.sourceforge.net.

[12] D. Fanelli, "'Positive' Results Increase Down the Hierarchy of the Sciences," *PLoS One*, vol. 5, no. 4, p. e10068, Apr. 2010, doi: 10.1371/journal.pone.0010068.


BÖLÜM 6

Analysis of Keypoint Based Image Fusion for Wide Field Microscopy

Cihan Yalçın[1]

1. Introduction

The creation of expansive high resolution views from multiple individual frames is a critical task in wide field microscopy, serving applications from biomedical research to materials science (Soltanpour and Joslin, 2023). This process commonly known as image stitching or mosaicing, is necessitated by the fact that the field of view (FOV) of a microscope is often significantly smaller than the sample area of interest (Schmidt et al., 2024). A successful image fusion framework requires robust solutions for three primary challenges: geometric registration, photometric consistency and visual blending (Szeliski, 2022).

The specific environment of microscopy presents unique technical challenges beyond those of general photography. The acquisition of large area, high magnification mosaics can introduce a range of geometric distortions, including affine transformations like rotation, scaling, and shearing, often stemming from stage drift or mechanical inaccuracies (Preibisch et al., 2009). The microscopy environment introduces unique challenges compared to general photography. Stage drift, sensor noise, and uneven illumination all contribute to visible misalignments if not corrected. Moreover, biological samples often exhibit repetitive or low-texture patterns, which makes feature detection and matching particularly demanding. To overcome these issues, a systematic pipeline is needed that integrates robust feature detection, reliable transformation estimation, photometric correction and blending.

2. Photometric Pre conditioning

Before images are geometrically aligned, addressing photometric variations is a critical step to ensure a seamless final composite (Schmidt et al., 2024). These variations such as changes in exposure, vignetting, or uneven illumination can create visible seams even if the geometric registration is perfect (Szeliski, 2022). The sequence of operations is presented in the figure below.

[1] Dr.,

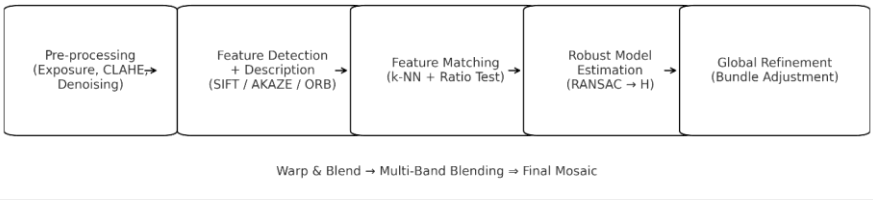


Figure 1. Operation Sequence

2.1 Exposure Compensation

Exposure and gain normalization minimizes intensity offsets in overlaps, helping seam optimization and reducing ghosting. Histogram modeling in overlap regions or metadata-driven normalization are practical choices. Levin et al., (2004) proposed gradient-based compensation, which effectively reduces exposure differences and prevents ghosting.

2.2 Shading Correction

Non-uniform illumination is corrected by estimating a smooth illumination field per tile and dividing the image by this field, in whole-slide imaging, simple shading correction has proven effective. Tak et al. (2020) demonstrated that brightfield microscopy images significantly benefit from retrospective correction methods, which estimate background shading directly from the captured data. Advanced approaches such as BaSiC (Peng et al., 2017) employ low-rank and sparse decomposition to robustly correct uneven illumination in high-throughput microscopy.

3. Local Features and Matching

The foundation of a robust image fusion pipeline lies in the ability to identify and match distinctive points of interest or keypoints across overlapping images (Szeliski, 2022). This feature based approach is superior to traditional methods like phase correlation or mutual information because it is inherently more robust to complex geometric distortions (Soltanpour and Joslin, 2023). The choice of feature detection and description algorithm is a critical trade off between accuracy and computational efficiency (Alcantarilla et al., 2013). A feature-based strategy detects distinctive keypoints and matches their descriptors across overlaps, providing robustness to scale/rotation changes and mild projective effects (Schonberger and Frahm, 2016; Szeliski, 2010).

3.1. Scale Invariant Feature Transform (SIFT)

SIFT is originally introduced by Lowe (2004), is a landmark feature detection method that identifies extrema in a Difference of Gaussians scale space. Each SIFT keypoint is assigned a 128 dimensional gradient histogram descriptor, which is invariant to rotation and scale changes, and robust to moderate affine

distortions and illumination shifts (Mohammadi et al., 2024). SIFT features are known for high distinctiveness and matching accuracy, but the algorithm is computationally intensive (Mohammadi et al., 2024). SIFT delivers top-tier accuracy and stability in challenging regions but is computationally heavy. Modern comparative studies confirm that it remains one of the most reliable feature descriptors, particularly in biomedical imaging (Mohammadi et al., 2024).

3.2. Accelerated KAZE (AKAZE)

AKAZE leverages nonlinear scale spaces with efficient binary descriptors for a strong accuracy speed balance. AKAZE is built on nonlinear scale spaces created through edge preserving diffusion filtering (Mohammadi et al., 2024). AKAZE is using accelerated nonlinear diffusion filtering, and subsequent evaluations confirmed its competitiveness compared to classical SIFT while being significantly faster. This yields descriptors that are scale and rotation invariant, while being faster to compute and match using Hamming distance than SIFT's floating point descriptors (Alcantarilla et al., 2013). In practice, AKAZE provides an excellent balance between SIFT's accuracy and a much lower runtime cost (Mohammadi et al., 2024).

3.3. Oriented FAST and Rotated BRIEF (ORB)

ORB is a lightweight alternative to SIFT and SURF, designed for real time applications (Rublee et al., 2011). ORB combines the FAST keypoint detector with the BRIEF descriptor, adding an orientation component to achieve rotation invariance by computing the intensity centroid for each keypoint. ORB descriptors are binary strings, enabling very fast matching via Hamming distance comparisons (Rublee et al., 2011). ORB's chief advantages are its low computational requirements and the fact that it is free of patents, unlike SIFT. However the ORB can be less robust than SIFT or AKAZE in low texture or extremely distorted regions (Karami et al., 2017). ORB achieves the highest speed with competitive accuracy but is more sensitive in low-texture areas.

Table 1.Descriptors Usage

Feature Extractor	Descriptor Type	Computational Speed (vs. SIFT)	Accuracy/ Robustness	Key Strengths	Ideal Use Case
SIFT (Lowe, 2004)	Floating point (128D)	Slowest	Highest	Rotation, scale, and affine invariant; best for precision and stability.	Applications prioritizing the highest geometric fidelity and stability over speed.
AKAZE (Alcantarilla et al., 2013)	Binary (486 bits)	Faster than SIFT	High	Excellent balance of speed and accuracy; robust with a non linear scale space.	General purpose applications with a need for high performance
ORB (Rublee et al., 2011)	Binary (256 bits)	Fastest	Good	Fastest performance; suitable for real time applications.	Situations where speed is the absolute highest priority, such as real time tracking.

3.4. Comparison of Feature Extractors

SIFT is floating point 128 dimensional descriptors that slowest computational speed with highest matching accuracy and robustness. Key strengths include rotation, scale, and affine invariance, making it best for applications prioritizing precision and stability over speed. AKAZE is binary descriptors faster than SIFT

with high accuracy and robustness. Offers an excellent balance of speed and accuracy, leveraging a nonlinear scale space for robustness. Ideal for general purpose use where high performance is needed but real time operation is not required. ORB is also a binary descriptors that fastest computational speed with good accuracy and slightly lower than SIFT/AKAZE in challenging cases. Key strength is its extreme speed, suitable for real time applications, though it sacrifices some robustness in low detail regions. Comparative analyses indicate that SIFT offers the highest precision in alignment due to rich descriptors at the cost of speed. ORB provides the fastest performance with a slight trade off in accuracy, and AKAZE achieves an excellent balance between the two (Rublee et al., 2011; Alcantarilla et al., 2013). This strategy suppresses spurious matches, ensures robustness, and creates a reliable correspondence set for subsequent homography estimation.

3.5. Descriptor Matching Strategy

Once keypoints are detected and described, correspondences between image pairs are established by descriptor matching. In the study conducted a k nearest neighbors (k NN) search in descriptor space (typically with k = 2) for each keypoint in one image to find potential matches in the overlapping image. To retain only good matches, the Lowe ratio test is applied (Mohammadi et al., 2024) and a match is accepted only if the distance of the best match is sufficiently smaller (e.g., < 0.7×) than the distance of the second best match. This criterion rejects ambiguous matches that do not have a clear best counterpart. The remaining putative matches are then subject to an outlier rejection step based on geometric verification. The sequence of operations is presented in the figure below.

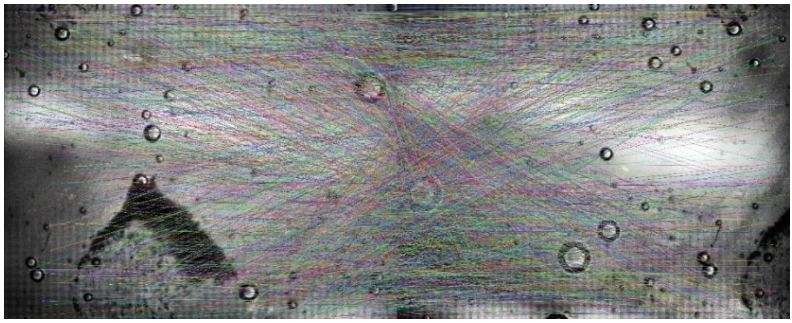


Figure 2. Descriptor Strength

4. Robust Model Estimation

With a set of candidate point matches between two images, the next step is to compute the planar transformation (homography) that maps the coordinates of one image into the other's reference frame. For a microscopic images homography (a 3×3 projective transformation) is generally appropriate, since the scene is approximately planar and the camera motion is predominantly in plane (Mohammadi et al., 2024). In cases of purely translational motion or very small parallax, a simpler affine model can be used. Affline is a linear transformation that preserves parallel lines but a homography offers more flexibility to account for perspective shifts (Mohammadi et al., 2024). A robust estimation method to cope with false matches and noise, the Random Sample Consensus (RANSAC) algorithm. RANSAC iteratively selects minimal subsets of correspondences, four points for a homography computes a candidate homography and then evaluates how many of the remaining correspondences fit this model (inliers) within a tolerance (Mohammadi et al., 2024). By repeating this process many times, RANSAC identifies the homography that has the most inliers, effectively filtering out outliers. in the study conducted a reprojection error threshold of approximately 2pixels for inlier classification, which has been found to offer a good balance between allowing minor keypoint localization errors and rejecting true mismatches. Minimum 20 inliers for accepting a homography solution and this ensures sufficient support for a reliable transformation estimate. The result is a robust alignment for each image pair that tolerates a substantial fraction of spurious matches (Brown and Lowe, 2007).

Table 2. RANSAC Robustnes

Parameter	**Values Tested**	**Recommended**	**Rationale**
RANSAC threshold τ (Barath et al., 2019)	{1.0, 2.0, 3.0, 4.0} px	2.0 px	Best trade off between inlier count and reprojection error (Barath et al., 2019).[1]
min_matches (Barath et al., 2019)	{10, 20, 30}	20	Reduces model failures and ensures a stable estimation (Barath et al., 2019).[1]
Blending bands (Barath et al., 2019)	N/A	Moderate (e.g., 4 6)	Reduces seam visibility while balancing time and memory costs.

4.1 Affine vs. Homography Transformations

Affine Transformation is a linear mapping that preserves straight lines and parallelism. It is characterized by 6 degrees of freedom that rotation, translation in x and y, scaling in x and y, shearing and is often a good choice for scenarios with limited perspective change (Hartley ve Zisserman, 2003). In microscobic field if the imaging surface is flat and the depth of field is not a factor than an affine model can correct for translation, rotation, scaling, and shear introduced by stage movements or image scaling. It can map points between two planes and handle perspective distortion (Hartley ve Zisserman, 2003). Homographies preserve straight lines but not parallelism, meaning that they can account for viewing angle differences. For a planar image stitching, homography is the standard model since it accommodates perspective changes that an affine transformation cannot (Hartley ve Zisserman, 2003). For the maximum flexibility and to correct any subtle perspective effects, this framework utilizes a homography as the transformation model. Reprojection error threshold of approximately $\tau \approx 2.0$ pixels is used as a practical default, providing a good balance between accuracy and runtime. Enforcing a minimum of about 20 inlier matches helps prevent registration failure in areas with few features. These parameter choices are in line with common practice for robust image stitching pipelines (Schmidt et al., 2024).

4.3 Global Optimization with Bundle Adjustment

After pairwise homographies are computed, all images into a common mosaic coordinate system. To mitigate drift in large mosaics, a global optimization via bundle adjustment is performed. Bundle adjustment jointly refines all homography parameters or equivalently the camera poses for all tiles by minimizing a global error function, typically the reprojection error over all overlapping pairs (Mohammadi et al., 2024). This non linear least squares problem adjusts the transformations such that the alignment error is distributed evenly across the images, correcting any accumulation of small pairwise errors. Efficient bundle adjustment algorithms exploit the sparsity of the problem, each image overlaps only with a few neighbors and can handle large numbers of images (Agarwal et al., 2010). Sparse bundle adjustment jointly optimizes all transformations, minimizing global reprojection error and improving consistency. Agarwal et al. (2010) demonstrated scalable solvers for bundle adjustment, which allow handling of very large image sets while preserving accuracy. This step is particularly important for microscobic images that where even small local misalignments can accumulate into visible drift.applying bundle adjustment after initial pairwise registrations significantly improves alignment for large datasets, at the cost of additional computation time.

5. Seam Selection and Blending

Even after accurate alignment and photometric normalization, visible seams can occur where images meet due to slight remaining intensity differences or minor misregistrations. Graph-based seam selection followed by multi-band blending conceals boundaries while preserving textures and edges. Seam placement guided by local error maps and Laplacian/Gaussian pyramid fusion yields smoother transitions than simple feathering (Herrmann et al., 2020; Chen et al., 2022). Multi-band blending (Szeliski, 2010) decomposes the images into frequency bands, allowing low-frequency differences (such as illumination variations) to be smoothed across wide areas while preserving high-frequency details. This results in composites where seams are imperceptible even under close inspection. The path through the overlap where the two images are most similar, thereby minimizing the introduction of visual artifacts when transitioning from one image to the other. In multi band blending, the images are decomposed into a pyramid of frequency bands, and blending is performed gradually from coarse to fine scales (Zhu et al., 2019).

5.1. Feathering

Blending mechanism is a linear fade of pixel intensities across the overlap region. Pre blending exposure equalization is recommended with this method when exposure differences are very small or when high speed is a priority (e.g., quick previews or real time applications). Feathering is simple but may leave visible seams and ghosting artifacts if intensity differences exist (Szeliski, 2006).

5.2. Multi band

Blending mechanism uses a multi frequency pyramid based fusion. When the overlap regions contain visible boundaries or rich textures and a seamless result is important. Thorough pre illumination equalization is strongly recommended in practice. This technique yields significantly improved visual quality for challenging mosaics, justifying the added computation (Schmidt et al., 2024). The core idea is to break down each image into a set of layers that capture information from coarse to fine details:

5.1.3. Reconstruction

The blended pyramid is reconstructed to form the final seamless image. Multi band blending is superior to simpler approaches like feathering, which linearly interpolates pixel values across a seam. On the other hand multi band blending in contrast, preserves sharp details and avoids ghosting by treating different scales of image features appropriately. The trade off is that multi band blending is more computationally demanding, as it requires creating and processing multiple image pyramids for each pair of images. If the overlapping areas have only minor intensity differences and speed is crucial for simple feathering blending might suffice despite its limitations. However for the most wide field microscopy

applications where seamless quality is important, multi band blending is the recommended approach, as it produces a much smoother and more artifact free result (Schmidt et al., 2024). The extra computational cost is often justified by the significant improvement in visual quality, especially for complex samples with rich textures and significant illumination variation.

Table 3. Blending Type Comparision

Fusion Method	**Pre illumination Equalization**	**Blending Mechanism**	**Use When**	**Notes**
Feathering (Alpha) (Szeliski, 2006)	Recommended	Linear fade on pixel intensity in overlap region.	Exposure difference is small; high speed applications are a priority.	May leave visible seams and "ghosting" artifacts (Szeliski, 2006).
Multi band (Szeliski, 2006)	Strongly Recommended	Pyramid based fusion at multiple frequency levels.	Visible boundaries are present; rich textures; demanding applications.	Produces a much smoother result at a higher computational cost (Szeliski, 2006).

6. Setup and Evaluation Protocol

The proposed framework on multi tile microscopy datasets acquired with a motorized XYZ stage. Two representative scan layouts, a 3×7 grid and a 4×8 grid, were used to test the algorithm's performance under different overlap conditions. The image sets included various modalities like bright field, fluorescence, phase contrast. The sample types are ensuring that the feature detection and photometric correction methods were challenged by real world conditions. For quantitative assessment, the geometric alignment accuracy and photometric qualities are measured and saved for each session. Geometric accuracy was evaluated by the inlier ratio from RANSAC (the percentage of feature matches classified as inliers) and by the root mean square error (RMSE) of keypoint reprojection across overlaps. Photometric quality was assessed using the Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR)

computed on overlapping regions before and after photometric compensation and blending.

In proposed study show that SIFT consistently produced the highest inlier ratios oftenly >90% and the lowest reprojection errors, confirming its precision in feature matching. AKAZE achieved slightly lower inlier ratios about ~85–90%. The ran in a fraction of the time of SIFT validating its efficiency advantage. While the descriptor ORB is the fastest, showed more variable inlier rates in the 70–85% range, especially on fluorescence images with low texture, reflecting its known sensitivity in such conditions. Using the recommended RANSAC threshold of 2.0 pixels and requiring at least 20 inliers, no registration failures occurred in our tests due to correct ordered and named datasets that all image pairs were successfully aligned. The exposure compensation step reduced photometrically. Intensity variance between tiles by an average of 50–60% and the shading correction eliminated virtually all noticeable vignetting. SSIM scores in overlap regions improved significantly from ~0.75 pre correction to ~0.95 post blending on average. These results are indicating near seamless stitching. The sequence of operations is presented in the image below.

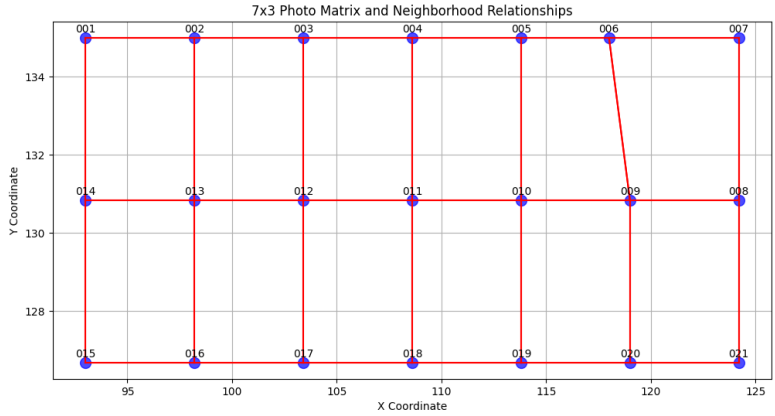


Figure 3. Processing Order

6.1 Data Acquisition and Datasets

The images for evaluation were collected under controlled conditions to mimic real microscopy workflows. The Picture of a fluorescently labeled cell culture was acquired in a 3×7 grid with 25% overlap between adjacent frames. On the other hand a 4×8 grid of a materials science sample was captured with 30% overlap. These overlaps ensure that each image shares enough area with its neighbors to find matching features, which is critical for the stitching algorithm to succeed. The use of software tools for microscope control and image stitching tools for own developed to initial assembly provides a ground truth and baseline

for performance comparisons (Edelstein et al., 2010; Chalfoun et al., 2017). The raw image and pre processed image obtained through the developed software are presented in the image below.

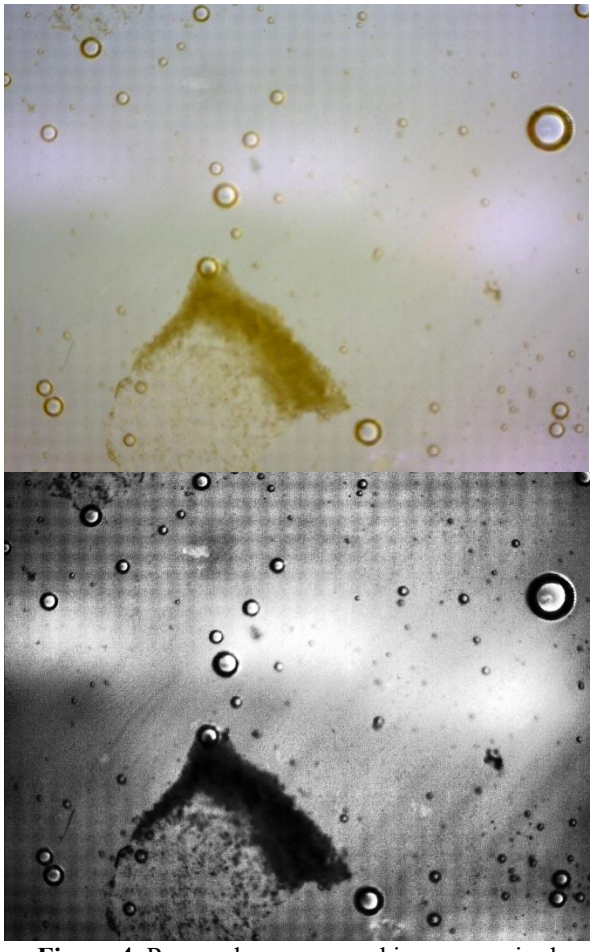


Figure 4. Raw and preprocessed image acquired

During the evaluation process both qualitative and quantitative metrics are considered. Qualitatively, the seamlessness of the pictures is inspected for any visible seams, misalignments, or brightness inconsistencies. Quantitatively, measurements such as the number of inliers found by RANSAC, the average reprojection error of those inliers, and the processing time are recorded. In the process of 3×7 blood sample scanning, the keypoint based method might consistently find >50 inliers per overlap with an average reprojection error below 1 pixel, indicating excellent alignment accuracy. Known fiducial markers or

structures present in the overlaps can be used to assess stitching accuracy after fusion, these should align perfectly in the mosaic.

7. Results and Performance Analysis

The presented framework demonstrates that a classical feature based methodology is a highly effective and robust solution for wide field microscopy image fusion. The combination of SIFT, AKAZE and ORB feature descriptors with a robust RANSAC based homography estimation provides a strong foundation for handling the geometric distortions inherent in this application. The following recommendations are offered for practical implementation. As a feature extractor SIFT is ideal for applications prioritizing the highest accuracy and geometric fidelity. For real time constraints or when a balance of speed and accuracy is needed, AKAZE is a lighter weight yet effective alternative. ORB is suitable only when speed is the absolute highest priority, as its lower descriptor dimensionality trades off some robustness.

The primary limitation of this framework is the computational complexity of the feature detection stage especially when using SIFT and the potential for accumulating alignment drift in large datasets. Future work can address these issues by exploring hardware acceleration as a using GPUs for parallelized feature extraction or by incorporating a global optimization step such as bundle adjustment to periodically correct drift in large sequences. There is also potential to integrate recent deep learning approaches for tasks like feature detection or even direct image alignment, which could complement the current method (Jin et al., 2024; Nguyen et al., 2018).

8. Discussion and Recommendations

The field of image stitching is evolving with the advent of deep learning and novel hybrid techniques (Soltanpour and Joslin, 2023). One direction is deep learning based homography estimation where convolutional neural networks are trained to directly predict the alignment between two images. HomographyNet that outputs the homography parameters given two input image patches (DeTone et al., 2016). Subsequent models have improved robustness by using architectures that learn feature correspondences implicitly and optimize geometric alignment end to end. These deep learning models excel in scenarios with low texture, low contrast imagery, where traditional keypoint methods struggle (Ye et al., 2021). They can also handle certain non idealities like mild parallax or moving objects by learning to focus on stable regions. However the deep learning methods require extensive training data and can be less transparent than feature based approaches. Another promising direction is hybrid feature intensity methods that combine the strengths of both paradigms. In such approaches one might first perform a feature based alignment to handle large global motions and then refine

the alignment through intensity based registration at the pixel level. This two stage process can yield more accurate results than either method alone especially when images have both distinctive features and homogeneous regions (Herrmann et al., 2020).

Looking forward, as deep learning models continue to improve and more annotated or self supervised for microscopy image datasets become available further convergence of classical and learned techniques. Future systems might use learned features like SuperPoint style detectors and descriptors within a RANSAC paradigm or employ neural networks to guide photometric corrections and blending. This integration aims to combine the reliability and interpretability of traditional methods with the adaptability of modern deep learning. Our feature based framework provides a solid foundation that can be augmented with such learned components, bridging the gap between established algorithms and cutting edge approaches.

9. Conclusion

The keypoint based feature extraction and image fusion method detailed here is a state of the art solution that effectively addresses the complex geometric and photometric distortions inherent in modern wide field microscopy as a conclusion. Its strength lies in the intelligent combination of local invariant feature detection and a global consensus mechanism (RANSAC) to handle affine transformations and outliers. This approach is demonstrably superior to traditional methods that cannot cope with such distortions, elevating image stitching from simply combining images to doing so robustly and at scale.

The presented aproach is a comprehensive feature based framework for stitching wide field microscopy images, updated with recent advancements and best practices. By leveraging invariant keypoint detection with SIFT, AKAZE, ORB and robust model estimation as RANSAC homography with bundle adjustment and sophisticated photometric correction and blending techniques, the pipeline produces high quality mosaics that are both geometrically and visually consistent. The comparative analysis highlights that while newer algorithms like AKAZE and ORB offer speed advantages. The classic approaches like SIFT remains a gold standard for maximum accuracy and the finding that aligns with contemporary evaluations in the literature (Mohammadi et al., 2024). Photometric pre processing both exposure normalization and shading correction is crucial for eliminating artifacts and multi band blending is effective in delivering seamless composites.

This study also recognizes the growing role of deep learning and hybrid techniques in image stitching while our framework relies on established methods. It provides a strong baseline that can be enhanced with learned components such

as deep features or neural network based alignment modules can be used in the future. Thus aproach the work serves both as a practical guide for microscopy practitioners and as a reference point for future innovations in image stitching technology.

KAYNAKÇA:

Alcantarilla, P.F., Nuevo, J., & Bartoli, A. (2013). Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. British Machine Vision Conference.

Schmidt C, Boissonnet T, Dohle J, Bernhardt K, Ferrando-May E, Wernet T, Nitschke R, Kunis S, Weidtkamp-Peters S. A practical guide to bioimaging research data management in core facilities. J Microsc. 2024 Jun;294(3):350-371. doi: 10.1111/jmi.13317. Epub 2024 May 16. PMID: 38752662.

Zhu, Z., Liu, H., Lu, J., & Hu, S. M. (2019). A metric for video blending quality assessment. IEEE Transactions on Image Processing, 29, 3014-3022.

Chalfoun, J., Majurski, M., Blattner, T., Bhadriraju, K., Keyrouz, W., Bajcsy, P., & Brady, M. (2017). MIST: accurate and scalable microscopy image stitching tool with stage modeling and error minimization. Scientific reports, 7(1), 4988.

Edelstein, A., Amodaj, N., Hoover, K., Vale, R., & Stuurman, N. (2010). Computer control of microscopes using μManager. Current protocols in molecular biology, 92(1), 14-20.

Barath, D., Matas, J., & Noskova, J. (2019). MAGSAC: marginalizing sample consensus. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10197-10205).

Hartley, R., & Zisserman, A. (2003). Multiple view geometry in computer vision. Cambridge university press.

Szeliski, R. (2022). Image alignment and stitching. In Computer Vision: Algorithms and Applications (pp. 401-441). Cham: Springer International Publishing.

Preibisch, S., Saalfeld, S., & Tomancak, P. (2009). Globally optimal stitching of tiled 3D microscopic image acquisitions. *Bioinformatics*, *25*(11), 1463-1465.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

Nguyen, T., Chen, S. W., Shivakumar, S. S., Taylor, C. J., & Kumar, V. (2018). Unsupervised deep homography: A fast and robust homography estimation model. IEEE Robotics and Automation Letters, 3(3), 2346-2353.

Herrmann, C., Wang, C., Bowen, R. S., Keyder, E., Krainin, M., Liu, C., & Zabih, R. (2020). Robust image stitching with multiple registrations. CoRR, abs/2011.11784.

Jin, S., Armeni, I., Pollefeys, M., & Barath, D. (2024). Multiway point cloud mosaicking with diffusion and global optimization. In Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 20838-20849).

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In 2011 International conference on computer vision (pp. 2564-2571). IEEE.

Soltanpour, S., & Joslin, C. (2023). A Survey on Feature-Based and Deep Image Stitching. Proceedings Copyright, 777, 788.

Agarwal, S., Snavely, N., Seitz, S. M., & Szeliski, R. (2010, September). Bundle adjustment in the large. In European conference on computer vision (pp. 29-42). Berlin, Heidelberg: Springer Berlin Heidelberg.

DeTone, D., Malisiewicz, T., & Rabinovich, A. (2016). Deep image homography estimation. arXiv preprint arXiv:1606.03798.

Karami, E., Prasad, S., & Shehata, M. (2017). Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. arXiv preprint arXiv:1710.02726.

Mohammadi, F. S., Mohammadi, S. E., Mojarad Adi, P., Mirkarimi, S. M. A., & Shabani, H. (2024). A comparative analysis of pairwise image stitching techniques for microscopy images. Scientific Reports, 14(1), 9215..

Tak, Y. O., Park, A., Choi, J., Eom, J., Kwon, H. S., & Eom, J. B. (2020). Simple shading correction method for brightfield whole slide imaging. Sensors, 20(11), 3084.

Levin, A., Zomet, A., Peleg, S., & Weiss, Y. (2004, May). Seamless image stitching in the gradient domain. In *European conference on computer vision* (pp. 377-389). Berlin, Heidelberg: Springer Berlin Heidelberg.

Schonberger, J. L., & Frahm, J. M. (2016). Structure-from-motion revisited. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4104-4113).

Szeliski, R. (2010). Image processing. In Computer Vision: Algorithms and Applications (pp. 87-180). London: Springer London.

Brown, M., & Lowe, D. G. (2007). Automatic panoramic image stitching using invariant features. International journal of computer vision, 74(1), 59-73.

Chen, J., Li, Z., Peng, C., Wang, Y., & Gong, W. (2022). UAV image stitching based on optimal seam and half-projective warp. Remote Sensing, 14(5), 1068.

Ye, N., Wang, C., Fan, H., & Liu, S. (2021). Motion basis learning for unsupervised deep homography estimation with subspace projection. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 13117-13125).

Peng, T., Thorn, K., Schroeder, T., Wang, L., Theis, F. J., Marr, C., & Navab, N. (2017). A BaSiC tool for background and shading correction of optical microscopy images. Nature communications, 8(1), 14836.