

Görme Engelliler için Geliştirilmiş Destekleyici Mobil Artırılmış Gerçeklik Uygulamalarının Karşılaştırmalı Değerlendirmesi

Alper Tunga AKIN, Çetin CÖMERT, Ziya USTA

(1057, 31)-Aralıklı Dizilerden Üretilen Boole Fonksiyonlar

Selçuk KAVUT

Türkçe Konuşmadan Metne Dönüştürme için Ön Eğitimli Modellerin Performans Karşılaştırması: Whisper-Small ve Wav2Vec2-XLS-R-300M

Öykü Berfin MERCAN, Sercan ÇEPNİ, Davut Emre TAŞAR, Şükrü OZAN

Birleştirilmiş Derin Öznitelikleri Kullanarak BT ve X-Ray Görüntülerinden COVID-19 Tespiti

Asuman GÜNAY YILMAZ, Emine AYAZ, Muammer TÜRKÖĞLU

Görüntülerden veya Çizimlerden Otomatik Kot Oluşturma Teknikleri: Bir Derleme Çalışması

Musa Selman KUNDURACI, Turgay Tuğay BİLGİN

nDPI Derin Paket İnceleme Aracı Üzerinde Bir Çalışma

Zehra Nur ÖZBAY, Mehmet Emin DALKILIÇ

Yazılım Hata Tahmininde Farklı Alt Örneklemeye ve Üst Örneklemeye Yöntemlerinin Karşılaştırılması

Özge ŞEN KAYA, Sinem OZKURT KESE

İstatistiksel ve Derin Öğrenme Modellerini Kullanarak Hisse Senedi Fiyat Tahmini

Emre ALBAYRAK, A. Nurdan SARAN

Bilimsel Yazılım Geliştirme Süreçleri için bir Yazılım Mühendisliği Yetkinlik Envanteri

Bilge SAY

## EDİTÖR KURULU

### Eş-Baş Editörler

Prof. Dr. Eşref ADALI - İstanbul Teknik Üniversitesi

Prof. Dr. İbrahim SOĞUKPINAR - Gebze Teknik Üniversitesi

### Alan Editörleri

Prof. Dr. Banu DİRİ - Yıldız Teknik Üniversitesi

Prof. Dr. Sezer Gören UĞURDAĞ - Yeditepe Üniversitesi

Prof. Dr. Tunga GÜNGÖR - Boğaziçi Üniversitesi

Prof. Dr. Nevcihan DURU - Kocaeli Sağlık Üniversitesi

Doç. Dr. Mehmet KARAKÖSE - Fırat Üniversitesi

Doç.Dr. Hidayet TAKÇI- Cumhuriyet Üniversitesi

Dr. Burcu YILMAZ - Gebze Teknik Üniversitesi

### Adres:

Bilgisayar Bilimleri ve Mühendisliği Dergisi  
Darüşşafa Cad. 43L D:3 İstinye, Sarıyer-İstanbul  
E-posta: AkademikBilisimVakfi@gmail.com

## YAYIN KURULU

- Prof. Dr. İlyas Çiçekli - Hacettepe Üniversitesi  
Prof. Dr. Efendi Nasibođlu - Dokuz Eylül Üniversitesi  
Prof. Dr. Şule Gündüz Öğüdücü - İstanbul Teknik Üniversitesi  
Prof. Dr. Nizamettin Aydın Yıldız - Teknik Üniversitesi  
Prof. Dr. M. Halit S. Oğuztüzün - Ortadođu Teknik Üniversitesi  
Prof. Dr. Kemal Bıçakçı - İstanbul Teknik Üniversitesi  
Prof. Dr. Şeref Sađırođlu - Gazi Üniversitesi  
Prof. Dr. Erdal Kılıç - 19 Mayıs Üniversitesi  
Prof. Dr. N. Yasemin Topalođlu - Ege Üniversitesi  
Prof. Dr. Tolga Sakallı - Trakya Üniversitesi  
Doç. Dr. Serhan Yarkan - İstanbul Ticaret Üniversitesi  
Doç. Dr. Murat Yılmaz - Gazi Üniversitesi  
Doç. Dr. A. Hasan Koltuksuz - Yaşar Üniversitesi

## İÇİNDEKİLER

## Sayfa

Görme Engelliler için Geliştirilmiş Destekleyici Mobil Artırılmış Gerçeklik Uygulamalarının Karşılaştırmalı Değerlendirmesi ( <i>Araştırma</i> ) Alper Tunga AKIN, Çetin CÖMERT, Ziya USTA	94-101
(1057, 31)-Aralıklı Dizilerden Üretilen Boole Fonksiyonlar ( <i>Araştırma</i> ) Selçuk KAVUT	102-108
Türkçe Konuşmadan Metne Dönüştürme için Ön Eğitimli Modellerin Performans Karşılaştırması: Whisper-Small ve Wav2Vec2-XLS-R-300M ( <i>Araştırma</i> ) Öykü Berfin MERCAN, Sercan ÇEPNİ, Davut Emre TAŞAR, Şükrü OZAN	109-116
Birleştirilmiş Derin Öznitelikleri Kullanarak BT ve X-Ray Görüntülerinden COVID-19 Tespiti ( <i>Araştırma</i> ) Asuman GÜNAY YILMAZ, Emine AYZ, Muammer TÜRKOĞLU	117-124
Görüntülerden veya Çizimlerden Otomatik Kot Oluşturma Teknikleri: Bir Derleme Çalışması (Derleme) Musa Selman KUNDURACI, Turgay Tugay BİLGİN	125-136
nDPI Derin Paket İnceleme Aracı Üzerinde Bir Çalışma ( <i>Araştırma</i> ) Zehra Nur ÖZBAY, Mehmet Emin DALKILIÇ	137-146
Yazılım Hata Tahmininde Farklı Alt Örnekleme ve ÜstÖrnekleme Yöntemlerinin Karşılaştırılması ( <i>Araştırma</i> ) Özge ŞEN KAYA, Sinem OZKURT KESE	147-160
İstatistiksel ve Derin Öğrenme Modellerini Kullanarak Hisse Senedi Fiyat Tahmini ( <i>Araştırma</i> ) Emre ALBAYRAK, A. Nurdan SARAN	161-169
Bilimsel Yazılım Geliştirme Süreçleri için bir Yazılım Mühendisliği Yetkinlik Envanteri ( <i>Araştırma</i> ) Bilge SAY	170-176



# Görme Engelliler için Geliştirilmiş Destekleyici Mobil Artırılmış Gerçeklik Uygulamalarının Karşılaştırmalı Değerlendirmesi

## Comparative Assessment of Mobile Augmented Reality Applications Developed for Visually Impaired Persons

Alper Tunga AKIN  
Karadeniz Teknik Üniversitesi  
Harita Mühendisliği Bölümü  
Trabzon, Türkiye  
alpertunga@ktu.edu.tr  
ORCID: 0000-0002-4535-9143

Çetin CÖMERT  
Karadeniz Teknik Üniversitesi  
Harita Mühendisliği Bölümü  
Trabzon, Türkiye  
ccomert@ktu.edu.tr  
ORCID: 0000-0002-2019-6990

Ziya USTA  
Artvin Çoruh Üniversitesi  
Harita Mühendisliği Bölümü  
Artvin, Türkiye  
ziyausta@artvin.edu.tr  
ORCID: 0000-0003-2232-2011

### Öz

Tablet, akıllı telefon gibi mobil cihazlar için artırılmış gerçekliğe (AR) yönelik sunulan geliştirme araçlarının ve mobil cihazların kullanımının yaygınlaşmasıyla, özellikle son on yılda, birçok yeni destekleyici AR uygulaması hayatımıza girmiştir. Bu uygulamalar cihazların kamera, kızılötesi, ToF, lidar gibi aygıtlarından gelen bilgiyle gerçekliğin artırımını sağlamaktadırlar. Böylesi uygulamaların yaygınlaşmasıyla, görme engelli odaklı uygulamaların geliştirilmesi de hem bir ihtiyaç hem de faal bir araştırma alanı olmuştur. Bu uygulamaların görme engelli bireylerin (GEB) kullanımına uygunluğunun değerlendirilmesinde hedef kitlenin ihtiyaçları, hassasiyetleri ve konforları göz önünde bulundurulmalıdır. Bu çalışmada kendine yer bulan gerek yazılım sektöründen gerekse akademik kaynaklardan seçilmiş çalışmalar, GEB'e sağladığı çıktılar, gerçek zamanlı kullanıma hız ve doğruluk ölçütleri yönünden uygunluk ve ulaşılabilirlik açısından değerlendirilmiştir. Yazılım sektöründen seçilen çalışmalar kendine pazarda yer edinip belirli bir kullanıcı kitlesine ulaşmış ürünler olup, akademik çalışmalar ise 2013'ten günümüze yayınlanmış güncel ve özgün çalışmalardır. Çalışmaların karşılaştırılmasının yanı sıra, söz edilen ölçütlere en uygun bir uygulamanın nasıl ortaya konulabileceği yönünde de irdelemelerde bulunulmuştur.

**Anahtar sözcükler:** Artırılmış Gerçeklik, Mobil Uygulamalar, Destekleyici Teknolojiler, Görme Engelliler

Gönderme, düzeltme ve kabul tarihi: 03.02.2023 - 01.06.2023 – 20.06.2023

Makale türü: Araştırma

### Abstract

Thanks to the widespread use of augmented reality (AR) development tools and mobile devices such as tablets and smartphones, many new assistive AR applications have found a place in our lives, especially in the last decade. These applications provide reality augmentation with the information that comes from the inertial sensors of the devices such as cameras, infrared, ToF, and lidar. With the spread of such applications, the development of visually impaired-focused applications has become both a need and an active research field. The needs, sensitivities and comforts of the target audience should be taken into account in evaluating the suitability of these applications for the use of visually impaired people (VIPs). In this study, selected studies from both the software industry and the academic literature are evaluated in terms of their suitability, accessibility and outputs provided to VIP. Studies selected from the software industry are products that have gained a place in the market and reached an evident user base, and academic studies are actual and original studies published from 2013 to the present. In addition to the comparison of the studies, it has also been examined how an application that is most suitable for the mentioned criteria can be put forward.

**Keywords:** Augmented Reality, Mobile Applications, Assistive Technologies, Visually Impaired Persons

### 1. Giriş

Görme duyusunun sağladığı görüntü bilgisi bireylerin dünyayı

derecede önemlidir. Görme duyusunun eksikliği ya da tamamıyla yoksunluğu bireylerin yaşam kalitesini doğrudan olarak olumsuz yönde etkileyen bir durumdur. Günlük yaşamda görme duyusunda bir sorun yaşamayan insanların özel dikkat ve emek sarf etmeksizin gerçekleştirdiği yolda engellerden kaçarak yürümek, giyinmek, karşıdan karşıya geçmek gibi eylemler görme engelli birey (GEB) için bir yardım ve destek olmadan gerçekleştirilemeyecek eylemlere dönüşebilmektedir. Böylesi bir yardım ve destek için bir yardımcının gerekliliği bireyin yaşam bağımsızlığını olumsuz etkileyen bir durumdur. Bu nedenle günümüzde, GEB'lere yönelik çözümlerin giyilebilir teknolojiler ya da mobil platformlar kullanılarak geliştirilmesi önemli bir çalışma alanıdır [1].

Belirtilen bu çözümler GEB'in günlük yaşam etkinliklerinin yönetmesini ve yönlendirilmesini (navigasyonunu) sağlama gibi işlevleri yerine getirirler ve destekleyici teknolojiler olarak adlandırılırlar [2]. Destekleyici teknolojilerde eğilim, son on yılda mobil cihazların kullanımının dünya genelinde yaygınlaşması ve mobil platformlara yönelik geliştirme araçlarının sunulmasıyla birlikte mobil uygulamalar yönündedir [3][4]. Bu uygulamalar genelde mobil cihazların duyularını kullanarak kullanıcıya titreşim ya da seslendirilmiş metin olarak yanıt döndürmektedirler. Bu bakımdan, kişinin içinde bulunduğu gerçeklikte, gerçekliğin boyutlarını artırıcı bir görev üstlendikleri için bu uygulamalar artırılmış gerçeklik (AR) uygulamaları olarak nitelendirilebilir [1]. Görme engeli bulunmayan bir bireyin gerçekliği eğer, beş duyu oranı bulunmasından ötürü, beş boyutlu olarak nitelendirilirse, GEB'in gerçekliği bu AR uygulamalarıyla dört boyuttan beşinci boyuta yükseltilmeye çalışılmaktadır. Bu gerçekleştirilirken görme duyusu ile sağlanan, kişinin karşısındaki nesnelere türü, karşılaşılabileceği engellerin mesafesi ve yaya yönlendirmesi gibi çıktılar GEB'e ses ya da titreşimle sunulmaktadır [5].

Söz konusu AR uygulamalarının değerlendirilmesinde, GEB'lerin bu yöndeki ihtiyaçları, kaygıları ve talepleri, uygulamanın ne denli kullanıcı dostu bir arayüze sahip olduğu, ürettiği çıktılarının çeşitliliği ve doğruluğu belirleyici ölçüt olmaktadır. Bunun yanı sıra, özellikle ülkemiz Türkiye gibi gelişmekte olan ülkelerde, uygulamanın kullanımı için gerekli maliyet ve ulaşılabilirlik de değerlendirmede belirleyici kriterlerdir [6].

Metnin izleyen bölümlerindeki kompozisyonu nesne türü, engel mesafesi ve yönlendirme için gerekli rota bilgilerinin üretim tekniklerine göre şekillendirilmiştir. Her bilgi üretim tekniğine ilişkin özgün uygulama örnekleri verilmiş ve bu örneklerin yukarıda sözü edilen ölçütlere göre uygunlukları değerlendirilmiştir. Değerlendirme yalnızca bilimsel çalışmalarla sınırlandırılmamış, yazılım sektöründen de öne çıkan örneklerle yer verilmiştir.

## 2. Değerlendirme

Değerlendirme bölümüne geçmeden önce araştırmada kullanılan sorgulamaya ilişkin ayrıntıların belirtilmesinde fayda görülmüştür. Konuya ilişkin akademik yayınların taranmasında Web of Science (WoS) ve IEEE Xplore veri tabanlarında anahtar sözcük kombinasyonundan oluşan

“(visually impaired OR blind) AND (augmented reality OR navigation OR assistive)” sorgusu kullanılmıştır. Sorguda kullanılan anahtar sözcük kombinasyonu, ilgili çalışmaların gerek başlık ve özetlerinde gerekse anahtar sözcüklerinde sıkça rastlanan sözcüklerdir. Sunulan bu çalışma, yalnızca bu sorgu ile elde edilen çalışmalardan ibaret değildir. “Connected Papers” adlı web uygulaması [URL-1] kullanılarak, sorgu sonucu elde edilen çalışma kümesinin bir atıf ağı elde edilmiş ve bu ağ yapısı üzerinden, küme içerisindeki çalışmalarca sıklıkla başvurulan kaynaklar da bu çalışmaya dahil edilmiştir. Böylelikle, anahtar sözcük sırasında gözden kaçması olası çalışmalar ve küme içerisindeki çalışmaların referans aldığı kaynaklar da çalışmaya dahil edilmiştir. Ayrıca söz edilen atıf ağı yapısı, ağ içerisindeki diğer çalışmalarla düşük etkileşimde bulunan çalışmaların da kapsamdan çıkarılmak üzere yeniden gözden geçirilmesini sağlamıştır. Elde edilen sonuçların teknolojilerin günümüzdeki seviyesi bağlamında anlamlı olması için sorgu 2013-2023 yılları aralığıyla sınırlandırılmıştır. Ayrıca şu da göz ardı edilmemelidir ki, bu değerlendirme yalnızca mobil cihazlar ile kullanılan, kullanıcıya nesne tanıma, engel uyarısı ve yönlendirme ile rehberlik sağlayan uygulamaları konu edinmektedir. Yazılım sektöründen sunulan örnekler de basında, internet aramalarında ve mobil cihazların uygulama mağazalarında konuya ilişkin aramalarda başlıca karşılaşılan örneklerdir. Yayınlarda, ilgili kapsamda geliştirilen uygulama sayısı burada irdelenenle sınırlı olmayıp, irdeleme kullanılan teknikler bakımından diğerleri arasından sıyrılan özgün uygulamalar üzerinden gerçekleştirilmiştir. Bu çalışmadan önce ilgili alanda gerçekleştirilmiş kaynak araştırması çalışmaları da bu çalışmanın ortaya çıkmasında ilham verici olmuşlardır [2, 3, 5]. Bu çalışmayı, verilen çalışmalardan ayıran en belirgin özellik, destekleyici uygulamaların sınıflandırılmasında ve kıyaslanmasında kullanılan tekniklerin yanı sıra uygulamanın erişilebilirliğinin de gözetilmesidir. Ayrıca bu çalışmanın, konuyu güncel uygulamalarla ele alan ve Türk dilinde kaleme alınmış bir çalışma olması da özgünlüğünü perçinleyen bir etken olarak görülmektedir.

GEB'lere yönelik geliştirilen AR uygulamaları, kullandıkları yöntemler bakımından, iki kümeye ayrılabilir. Bunlardan birincisi QR (quick response) kod, RFID, BLE (bluetooth low energy), UWB (ultra-wideband), NFC, zemin işaretçisi ya da tabela gibi konum ve anlam karşılığı ön tanımlı işaretlerle donatılmış ve haritaları uygulama veri tabanında bulunan mekanlarda kullanılabilen “bağlamsal” uygulamalardır. Bağlamsal uygulamalar genelde GPS sinyalinin zayıf olduğu iç mekanlarda kullanılmaktadır. Diğerleri ise çeşitli duyulardan gelen veriyi anlık olarak işleyip ön tanımlı olmayan mekanlarda da bilgi üretip sunabilen “algısal” uygulamalardır. Bu duyular başta kamera, ToF (Time of flight) olmak üzere sesötesi duyurga, lidar ya da kızılötesi duyurga olabilmektedir.

Yayınlarda geçmiş inceleme çalışmalarında daha farklı sınıflandırmalarla karşılaşmak mümkündür. Örneğin [5] çalışmasında, ele alınan uygulamalar duyurga ağına dayalı uygulamalar ve video kamera kullanılan uygulamalar olmak üzere iki ana sınıfa ayrılıp alt sınıflarla incelemeye devam edilmiştir. [2] çalışmasında ise uygulamalar, uygulamanın kullandığı duyurmanın aktif ya da pasif olma durumuna,

hareketli nesnelere ayırt edebilme yeteneklerine göre, derinlik tespitinin çalışma menziline, arka plan algoritmalarının çevrimiçi veya çevrimdışı çalışma durumlarına kadar inen derin bir hiyerarşide ele alınmıştır. Sunduğumuz bu çalışmada ise uygulamalar kullandıkları tekniklerin neticesinde, koşturulacakları alanda bir ön çalışmaya bağımlı (bağlamsal) ya da bilinmeyen ortamların anlık olarak algılanmasına dayalı (algısal) uygulamalar olmak üzere iki ana kümeye ayrılmıştır. Böylelikle sadece profesyonellerin değil, kullanmak üzere bir uygulama arayışında bulunan GEB'lerin ya da konuya teknoloji okur yazarlığı düzeyinde merak duyan bireylerin de fikir edinebileceği nitelikte üst bakışa sahip bir sınıflandırma ve değerlendirme elde edilmiştir.

## 2.1 Bağlamsal Uygulamalar

Bağlamsal yaklaşım kümesine ilk örnek olarak ARIANNA (pAth Recognition for Indoor Assisted Navigation with Augmented perception) verilebilir [7]. ARIANNA, iç mekanlarda GEB'lerin yönlendirmesini sağlamak üzere geliştirilmiştir. Mekanın kat planı üzerindeki konum karşılığı bilinen zemine işlenmiş QR kodlar ve bu kodlar arasına çekilmiş şeritler kullanılmıştır. QR kodlu akıllı telefon kamerasıyla taranıp uzak sunucuya gönderilmekte, sunucuda kodun kat planı üzerindeki konumsal tanımı çözümlenip GEB'in hedefe ulaşması için hesaplanan rota istemciye gönderilmektedir. GEB'in rota üzerinde hareket ettirilmesi, zemindeki şeritlerin kameranın görüş alanından çıktığı anda gerçekleşen titreşim kesintileriyle kullanıcının uyarılması ile sağlanmaktadır. Anlatılan şekliyle uygulamanın kullanılabilmesi için öncelikle GEB'in ziyaret edebileceği ya da zaman geçirebileceği tüm mekanlar için ön çalışma yürütülmesi gerekmektedir. Bu mekanların zeminleri QR kod ve şeritlerle işaretlenmeli, bu işaretçilerin kat planı üzerinde konum karşılıkları bilinmeli, sayısal kat planı üzerinden rota hesaplanabilmesi için kaydedilmelidir. ARIANNA'daki bu yükümlülükleri ortadan kaldıran, algısal yaklaşım grubuna dahil, bir güncelleme çalışması 2021 yılında ARIANNA+ adıyla sunulmuştur [8]. ARIANNA+, ilgili başlık altında irdelenecektir.

NavCog uygulaması ile GEB'lerin iç mekanlarda bluetooth bağlantılı BLE cihazlarla konumlandırılması ve yönlendirilmesi sağlanmıştır [9]. 6-12 metre aralıklarla yerleştirilmiş BLE cihazlar periyodik olarak kullanıcıların cihazlarına bluetooth aygıtları üzerinden kendi konumlarını yayınlamaktadırlar. Kullanıcı aygıtlarda bu yayınların toplamı ile bir BLE ağı oluşmakta ve konumlandırma bu ağ üzerinden enterpolasyon ile gerçekleşmektedir. Yönlendirme işlemi ise kullanıcı cihazının jiroskop aygıtı ile sağlanmakta ve sesli yönlendirme yapılmaktadır. BLE aygıtları iç mekanın kat planı üzerindeki dağılımlarının geometrisi, üretilen konum bilgisinin doğruluğunu direkt olarak etkilemektedir. Gerekli dağılımı sağlayabilmek de, özellikle geniş iç hacimli kapalı mekanlarda temin edilecek BLE sayısı fazla olabileceğinden, yüksek maliyetli olabilmektedir. Ayrıca yalnızca iOS işletim sistemi için geliştirilmiş olması, kullanıcı için bir donanım bağımlılığı da getirmektedir.

NavCog'da olduğu gibi konumsal etiketlerden enterpolasyon amaçlı ağ oluşturulmasında kullanılan bir diğer yapı da RFID'lerdir. RFID ile kullanılan çözümlerin çalışma ilkesi

yukarıda açıklanan BLE ile sunulan çözümlere benzerdir. Pasif RFID duyargaların etki alanı 0,5-10 metre aralığındadır. Bu nedenle çok sık yerleştirilmeleri gerekmektedir. Aksi takdirde, yönlendirme esnasında, kullanıcı cihazının jiroskop aygıtında hata birikimi çözülememekte ve sağlıklı bir şekilde yönlendirme sağlanamamaktadır. Aktif RFID kullanımında duyargaların 1 Km'ye kadar uzanabilen etki mesafeleri sayesinde BLE cihazlara kıyasla daha seyrek yerleştirilebilmeleri mümkündür [10]. Ancak bu duyargalar yüksek enerji tüketimi ve buna bağlı olarak fazladan bakım gerektirmektedirler [11]. BLE kullanan çözümlerdeki duyargaların dağılım geometrisinin doğruluğa etkisi burada da söz konusudur. Ayrıca bu çözümlerde GEB üzerinde akıllı telefona ilaveten mobil RF okuyucu bulunması gerekmektedir, bu da ulaşılabilirliği kısıtlayabilecek bir etkidir. Günümüzde tercih edilebilirlikleri BLE ve UWB cihazlara göre daha düşüktür.

SUGAR isimli çalışma UWB duyargalar kullanarak GEB'lerin iç mekanlarda yönlendirmesini sağlamaktadır [12]. 50-60 metre aralıklarla yerleştirilen UWB'lerle ağ oluşturulmaktadır. UWB, yukarıda söz edilen BLE ve RFID'ye göre daha pahalı olsa dahi, menzilin geniş olması sayesinde daha az UWB cihazı ile verim sağlanabilmektedir. Bu avantaj sayesinde bu çalışmada UWB'ler tercih edilmişlerdir. Oluşturulan UWB ağı üzerinden GEB'lerin konumlandırılması sağlanmış ve jiroskoptan gelen veri kullanılarak A\* algoritmasıyla hesaplanan bir rota üzerinden ses ile yönlendirilmiştir. BLE ve RFID çözümlerindeki duyargaların iç mekanlara yerleştirilmesi-bakımı ve yayılım geometrisinin doğruluğa etkisi durumları, SUGAR ve benzeri UWB kullanan çözümler için de geçerlidir.

Bağlamsal uygulamalar sınıfına dahil edilebilecek bir diğer çalışma Gören Göz projesi kapsamında sunulan Navibaston uygulamasıdır [13]. Bu çalışma kapsamında, çalışmanın finansörü bakanlık tarafından alınan başvurular sonucunda uygun görülen GEB'lere uygulamanın dağıtımını gerçekleştirilmektedir. Uygulama, çevrimiçi harita altlıkları üzerinden dış mekan yönlendirmesini sesli geri bildirimle sağlamaktadır. Bu uygulama, söz edilen erişim kısıtlaması sebebiyle test edilebilirlik ve ulaşılabilirlik açısından kısıt taşımaktadır. Ayrıca, kullanıcı geri dönüşleri ve sağlanan çıktılarının doğruluk ölçünleri paylaşılmadığı için güvenilirlik değerlendirmesinde bulunabilmek güçtür. Ülkemizde GEB'ler tarafından yaygın kullanılan bir uygulama olması sebebiyle bu çalışmada kendine yer bulmuştur.

Belirtildiği üzere bağlamsal yaklaşım grubundaki çalışmalar genelde iç mekan çözümlerine odaklanmışlardır. Bunun gerekçesi, dış mekanlarda konumlandırmada yaygın kullanılan GPS duyargalarının sağladığı verinin doğruluk düzeyinin iç mekanlarda oldukça düşük olmasıdır. Öyle ki GPS sinyalleri kapalı bir alana girdiklerinde engellerden ötürü zayıflamaktadırlar. Bu nedenle, iç mekanlarda GPS'in yerini alacak bir konum verisi sağlayıcıya ihtiyaç duyulduğu için BLE, RFID, UWB gibi vericilere ya da QR kodlara başvurulmuştur. Bu nesnelere yerleştirilmesi, alana dağılım geometrilerinin optimizasyonu, sayısal kat planı üzerine işlenmeleri, uygulamanın kullanılması olası her kapalı alan için bir ön çalışma yapılmasını gerektirmektedir. Bu nedenle, yalnızca iç

mekanlarla sınırlı kalmayarak, algısal yaklaşım grubunda irdelenecek çalışmalar ortaya çıkmıştır.

## 2.2 Algısal Uygulamalar

WeWALK akıllı baston projesi, GEB'nin karşısındaki engellerin sesli geri bildirimini sağlamaktadır. Sistem bir sesötesi duyurğa barındıran baston ve akıllı telefon uygulamasından oluşmaktadır. Sesötesi duyurğa ile kullanıcının göğüs ve baş hizasındaki 0,8-2,5 metre aralığındaki engeller algılanıp bildirilmektedir. Bununla birlikte kullanıcının akıllı telefonuna bluetooth bağlantısıyla bağlanarak, telefonun konum servisleri aracılığıyla çevrimiçi haritalar üzerinden yönlendirme sağlamaktadır[14].

ARIANNA'daki zemin işaretçisi tesisi yükümlülüğünü gideren ARIANNA+, iOS işletim sistemli cihazlarda ARKit [15] AR kütüphanesi fonksiyonları sayesinde GEB'in yönlendirmesini sağlamaktadır. ARKit ve Android tabanlı ARKit karşıt seçeneği ARCore [16], kullanıcı kamerasıyla algılanan sahnenin yapısını modellemek için ToF ve lidar duyurğalarını kullanırlar ve ARIANNA+'da da bu sayede sahnedeki zemin ve zemin üstü nesnelere ayrılması sağlanmıştır. GEB'in yatay zeminde bir engele çarpmadan hareket edebilmesi sağlanmıştır. Ayrıca buna ek olarak uygulamaya, anlık görüntü üzerinden nesne tespiti yeteneği eklenmiş ve bunun için SSD [17] ve Inception [18] derin öğrenme (DL) modelleri kullanılmıştır. Uygulamanın ARKit tabanlı çalışıyor olması, Apple firmasının bu kütüphane için destek sağladığı belirli cihazlara bağımlılığı beraberinde getirmektedir [19].

[20] ile, GEB'e karşılaştığı nesnenin türünü ve nesne ile kullanıcı arasındaki mesafeyi sesli geri bildirimle sunan bir istemci sunucu mimarisi geliştirilmiştir. Kullanıcının akıllı telefonu ile alınan canlı görüntü kareleri uzak sunucuya gönderilip kareler üzerinden nesne türü ve mesafe bilgisi türetilmektedir. Bu bilgiler metin veri olarak istemciye gönderilip istemci tarafında metni sese dönüştürme teknolojisiyle sese dönüştürülerek kullanıcıya sunulmaktadır. Sunucu tarafındaki nesne türü çıkarımı YOLO [21] ve Fast R-CNN [22] DL modelleri ile sağlanmaktadır. Derinlik bilgisi çıkarımı için kamera odak ekseninin zemine paralel olduğu varsayımına dayanan bir üçgen benzerliği çözümü tercih edilmiştir. İstemci sunucu mimarisi ile işlem yükü uzak sunucuya aktarıldığı için, kullanıcı donanımından bağımsız olma niteliği taşımaktadır. Ancak derinlik çıkarımında tercih edilen yaklaşım, temel alınan varsayımın canlı kullanımda geçersiz kalma ihtimalinin yüksekliği sebebiyle riskli görülmüştür. Öyle ki, çalışmanın sinama sonuçlarında üretilen derinlik bilgisindeki hatanın 10 metre ve üzerindeki mesafelerde giderek arttığı görülmektedir.

"Watch Your Step" adlı çalışmada, GEB'in karşılaştığı engelleri kızılötesi duyurğa barındıran Google Tango destekli cihazlarla algılayarak titreşim ve ses ile geri bildirim veren bir çözüm geliştirilmiştir [23][24]. Kızılötesi duyurğadan gelen veri Google Tango fonksiyonları ile işlenerek cihazın algıladığı sahnenin fiziksel yapısı modellenilebilmektedir. Bu çalışmayla GEB'in karşılaştığı engellerle arasındaki mesafe bilgisi üretilip, titreşim ve akustik geri bildirim ile bu engellerden kaçınması sağlanmıştır. Bu çözüm yalnızca Google Tango destekli cihazlar ile çalışıyor olmasından ötürü platform bağımlı bir

çözümdür. Ayrıca, Google firmasının 2018 yılında Tango'yu sonlandırıp ARCore projesini başlatmasından dolayı sürdürülebilir değildir [25].

[26] çalışmasında, GEB'in karşılaştığı nesne türü ve mesafe bilgisini canlı görüntü kareleri üzerinden DL modelleri ile üretilen ses ve titreşimle geri bildirim sağlayan bir istemci sunucu mimarisi geliştirilmiştir. Akıllı telefona Wi-fi ile bağılı kablosuz bir kamerayla alınan anlık görüntü kareleri sunucuya gönderilmekte, sunucuda YOLO ile nesne türü, [27] modeli ile de mesafe bilgisi 0,5-20 m aralığında üretilip istemciye gönderilmektedir. Bu bilgiler istemciye sese ve titreşime dönüştürülüp kullanıcıya sunulmaktadır. Titreşimli geri bildirim kullanıcının koluna taktığı iki akıllı saat ile verilmektedir. Çalışma, istemci sunucu mimarisi kurulması sayesinde kullanıcı cihazının kurulumuna bağımlılık barındırmamaktadır. Ancak akıllı telefonun kendi kamerası yerine ek kamera kullanılması bir ek donanım yükümlülüğü getirmektedir. Bu kamera aynı zamanda şarj edilmesi gereken fazladan bir donanım anlamına gelmektedir.

[1] çalışmasıyla, GEB'in karşılaştığı engelin türünü ve engel ile arasındaki mesafeyi, yalnızca akıllı telefon kamerasıyla alınan anlık görüntü üzerinden, sesli geri bildirimle sunan bir istemci sunucu mimarisi geliştirilmiştir. Bunun yanı sıra bu uygulama, kullanıcı akıllı telefonunun konum servislerini kullanarak GEB'in dış mekanlarda sesli yönlendirmesini sağlamaktadır. Sunucu tarafında, telefon kamerasından gelen görüntüye karşılık görüntü üzerindeki nesnelere türü, konumları ve nesnelere ile kamera arasında mesafe DL modelleri ile üretilmektedir. Nesne tespiti için YOLO, mesafe çıkarımı için ise BTS (from Big To Small) DL modelleri kullanılmıştır [28][29]. Bu bilgiler istemciye metin veri formatında gönderilmektedir. Metin formattaki bilgiler metinden ses üretme yöntemi ile sese dönüştürülerek GEB'e sunulmaktadır. İstemci tarafında aynı zamanda, Mapbox Web Servisleri kullanılarak yönlendirme sağlanmaktadır [30]. Bu çalışma ile herhangi bir donanım bağımlılığı bulunmayan, düşük yetenekli mobil cihazlarda da sorunsuz çalışabilecek, DL modellerinin problem çözme yeteneğini web üzerinden sunabilen bir uygulama üretilmiştir.

[31] çalışmasında, GEB'in karşılaştığı engelin bilgisi, anlık görüntü üzerinden sahne yorumlaması sağlayan bir DL modeliyle bir istemci sunucu mimarisi aracılığı ile sunulmuştur. Görüntü üzerindeki nesnelere tanınması YOLO modeli ile sağlanmış, görüntü üzerinden algılanan nesnelere arasındaki uzamsal ilişkinin konuşma diline dökümü Doğal Dil İşleme (DDİ) mimarileri ile sağlanmıştır. Konuşma dilinde üretilen metin formatındaki sahne yorumu, 70'i aşkın dil seçeneği ile sese dönüştürülerek kullanıcıya sunulmuştur. Bunun yanı sıra, eğer ki anlık görüntü bir yazı ya da tabela içeriyorsa OCR tekniği ile sayısallaştırılıp sese dönüştürülerek kullanıcıya sunulmaktadır.

[32] çalışmasında ise anlık görüntü üzerinden GEB'e kalabalıkta uyarı verme amacıyla yaya tanıma sağlayan bir DL modeli geliştirilip, taşınabilir atanmış dizgeler üzerinde koşum zamanı ve doğruluk sinamaları gerçekleştirilmiştir. YOLO modeli yalnızca yaya tespiti yapmak üzere uygulamaya uyarlanarak Nvidia Jetson Nano atanmış dizge üzerinde MİB



ve GiB (Grafik İşleme Birimi) ile yanıt süresi değerlendirilmiştir. Bu çalışmada kullanıcıya sağlanan geri bildirim türüne ilişkin bir bilgi verilmemiş olup, YOLO mimarisinin yaya tespiti bağlamındaki başarımı ele alınmıştır.

### 2.3 Uygulamaların Karşılaştırılması

Belirtildiği gibi gerek akademik yayınlardan gerekse yazılım sektöründen GEB'lere yönelik geliştirilmiş AR uygulaması örnekleri irdelenmiştir. İrdeme platformdan bağımsızlık, ek donanım yükümlülüğü, doğruluk ölçünleri, üretilen çıktıların çeşitliliği açılarından gerçekleştirilmiştir. Çizelge-1'de irdelenen uygulamaların özgünlükleri verilmiş olup, Çizelge-2'de de uygulamalar birbirleri ile karşılaştırılmış; metnin devamında da en uygun uygulamanın seçimine ve iyileştirmelere yönelik çıkarımlarda bulunulmuştur. Çizelgedeki “-” ifadesi ilgili özelliğin çalışmada bulunmadığını ya da açıklanmadığını, “+” ifadesi ise ilgili özelliğin bulunduğunu belirtmektedir.

Uygulamanın platform bağımlılığının bulunması ve beraberinde ek donanım yükümlülüğü getirmesi ulaşılabilirliğini olumsuz yönde etkilemektedir. Bu açıdan bakıldığında [9], [8] ve [23] çalışmaları belirli işletim sistemlerine yönelik geliştirme araçlarıyla geliştirilmiş oldukları için mevcuttaki halleriyle evrensel çözümler olmayıp, çapraz platform geliştirme araçlarıyla yeniden düzenlenerek tüm işletim sistemlerinin kullanıcılarına hitap eder duruma getirilebilirler. Öyle ki, Unity ya da Godot gibi çapraz platform geliştirme ortamlarıyla [33] ya da PWA (Progressive Web Apps) gibi web tabanlı çözümlerle bu bağımsızlık sağlanabilmektedir [34]. Bunun yanı sıra, ek donanım yükümlülüğü, daha önce de belirtildiği gibi, özellikle gelişmekte olan ülkelerde donanım fiyatlarından dolayı üretilen çözümün yaygınlaşmasına engel olabilmektedir. [1], [20],[26] ve [31] çalışmalarında olduğu gibi istemci sunucu mimarileri ile, deneysel dahi olsa, bunun önüne geçilebilmektedir. Bu çözümlerin deneysel boyuttan çıkarılıp son kullanıcılar arasında yaygınlaşması sunucu maliyetlerinin karşılanmasını gerektirmektedir. Günümüzde bu sunucu maliyetleri, çevrimiçi gerçek zamanlı oyunlar göz önünde bulundurulduğunda, reklam gelirleriyle ya da donanım temininin yanında çok küçük sayılabilecek ücretlerle karşılanabilmektedir [35]. Bunların haricinde, [7], [9] ve [12] çalışmaları için söz edilen ek donanım yükümlülüğü, bu çalışmalar iç mekan yönlendirmesine yönelik çalışmalar olduğu için, GPS sinyallerinin iç mekanlardaki zayıflığından kaynaklı bir yükümlülüktür. [8] çalışmasındaki gibi zemin çıkarımına dayanan bir yaklaşımla ya da GPS sinyalinin güçlü olduğu bir dış mekan noktasından çıkış alınarak gerçekleştirilen, jiroskop, ivmeölçer ve manyetometre duyargalarında üretilen açı mesafe bilgisi ile koordinat taşımaya dayanan çözümlerle (PDR-pedestrian dead reckoning) [36] bunun üstesinden gelinebilir.

Sağlanan çıktıların çeşitliliği açısından [1], [8] ve [26] çalışmaları, derinlik çıkarımı, nesne tanıma ve yönlendirme çıktılarının üçünü birden sunmalarıyla ön plana çıkmaktadırlar. Çıktının çeşitliliği kadar çıktılarının doğruluğu ve üretim hızları uygulamanın, GEB'in gerçek zamanlı kullanımına uygunluğu belirleyen önemli faktörlerdir. Derinlik

çıkartımı için en yüksek hız, doğruluk, çalışma menzili ölçünlerini [26] çalışması sağlamıştır. [26]'yı çok yakın ölçünlerle [1] takip etmektedir. Bu iki çalışmanın bu fonksiyonu diğerlerinden açık ara yüksek bir performansla sağlamaları, gerekli arka plan algoritmalarının koşumlarını uzak sunucuya aktarmak suretiyle GPU'ların yüksek hesaplama kapasitesi ve yüksek hızlarını mobil platformlarda kullanmaları sayesinde. Böylelikle, akıllı telefonlarda kullanılması depolama alanı ve işlemci güçlerinin yetersizliği sebebiyle zor olan DL modellerinin problem çözme kapasitelerinden faydalanılmıştır. Derinlik çıkarımı başarımında bu iki çalışmayı [20] takip etmektedir. Güvenli çalışma menzilinün düşük ve hatasının yüksek olması 2.2 başlığı altında açıklanan varsayım sebebiyledir.

En yüksek nesne tanıma doğruluğu ölçünü [8] çalışması sağlamıştır. Fakat bu doğruluğu hangi hızda sağladığına dair bilgi çalışmada sunulmamıştır. En yüksek nesne tanıma hızını ise [1] çalışması 27 ms gibi bir süreyle sağlamıştır. Bu değer, saniyede 37 kareye (37 fps) karşılık gelmektedir. Bu 30 fps hızında görüntü alan ortalama bir akıllı telefon üzerinde, gerçek zamanlı kullanım için yeterlidir. [1] çalışmasındaki nesne tanıma doğruluğunun kendisine yakın çalışmalara göre düşük olmasının en önemli nedeni, YOLO modelinin 91 nesne sınıfı barındıran COCO (Common Objects In Context) [37] veri kümesi ile eğitilmiş bir ön eğitilmiş sürümünü doğrudan uygulamaya uygulanmış olmasıdır. Kaykay, zürafa, çörek gibi GEB için gereksiz birçok nesne sınıfının bulunması, modelin hesaplaması gereken fazladan parametre ve eğitim verisi anlamına gelmektedir. Öyle ki, [26] çalışmasında böyle bir nesne sınıfı sayısı kısıtlamasına gidilerek, YOLO modeli GEB için yeterli 7 nesne sınıfı ile yeniden eğitilerek daha yüksek doğruluk sağlanmıştır. Benzer bir kısıtlamaya [32] çalışmasında da gidilerek, YOLO mimarisi yalnızca “yaya” sınıfı ile eğitilmiştir. Böylelikle referans alınan modelin eğitildiği COCO veri kümesindeki uygulama kapsamındaki sınıflar çıkarılmış fazla parametre hesaplamasının önüne geçilmiştir. [31] çalışmasında ise sağlanan nesne tanıma işleminin doğruluğuna ilişkin bir bilgi verilmemiştir. Fakat [1] çalışması ile aynı ön eğitilmiş modelin kullanıldığı düşünülürse, aynı doğruluk değeri kabul edilebilir. Ayrıca bu çalışmanın, yalnızca nesne tanıma ile sınırlı kalmayıp, görüntü üzerinden tanınan nesnelerin aralarındaki uzamsal ilişkiyi kullanarak sahne yorumlama sağlaması sebebiyle, uygulamanın yanıt süresi ortalama 441 ms dolaylarındadır.

İç mekan yönlendirmesi sağlayan çalışmalar içerisinde, herhangi bir ek donanım ya da işaretçi tesisi gerektirmeyen bir uygulama olması sebebiyle tercih [8] yönünde olmalıdır. Dış mekan yönlendirmesi sunan [1] ve [14] uygulamalarına bakıldığında ise, dış mekan yönlendirmesinin yanı sıra birçok çıktı sağlaması sayesinde tercih [1] çalışması olacaktır. Elbette ki bu seçim, kullanıcı tercih ya da önceliklerine göre farklılık gösterebilir.

### 3. Tartışma

Çalışmanın “Uygulamaların Kıyaslanması” başlığı altındaki irdemelere bakıldığında, istemci sunucu mimarilerine dayanan web tabanlı uygulamaların ek donanım ve bir platforma bağımlılık gerektirmemeleri sebebiyle ön plana

çıkartıldığı görülmektedir. Bu noktada, web tabanlı uygulamaların da sürekli internet kullanımı sebebiyle yüksek veri ücretleri ödenmesine yol açacağı dezavantajı aklı gelebilir. Günümüzde, destekleyici uygulamalarla sınırlı olmaksızın çoğu mobil uygulama yalnızca bir istemci uygulaması formunda olup arka plan işlemlerini uzak sunucuda gerçekleştirmektedir ve anlık veri tüketimi hayatımızın artık parçası haline gelmiştir. Bu uygulamada irdelenen destekleyici uygulamalar gibi yüksek hesap yükü gerektiren algoritmaların da uzak sunucu üzerinde değil kullanıcı cihazında koşması yüksek enerji tüketimine sebep olacak ve yüksek kapasiteli taşınabilir şarj istasyonlarına mutlak ihtiyaç doğuracaktır. Ayrıca, web tabanlı mimariler bu algoritmalarındaki bir güncellemenin anında tüm kullanıcılara, kullanıcıların bir kurulum yapmasını gerektirmeden, sunulabilmektedir. Algoritmadaki güncelleme sunucu üzerinde gerçekleştiği için istemci doğrudan güncel uygulama ile iletişime geçmektedir. Böylesi uygulamaların çevrimdışı olarak kullanıcı cihazında koşması durumunda, algoritmadaki her bir güncelleme için kullanıcının yeniden kurulum yapması gerekmektedir. Eğer ki gelecekte teknoloji, [32]'dekine benzer sunuculardakine denk yüksek işlem kapasiteli GPU'lara sahip atanmış dizgeleri uygun maliyetlerle son kullanıcıyla buluşturabilirse ve veri aktarım hızları, algoritmalarındaki güncellemelerin atanmış dizgeye yüklenmesi web tabanlı kullanımdakine denk seviyelere getirilebilirse, bu çalışmadaki ve benzeri çalışmalardaki değerlendirmelerin tekrar gözden geçirilmesi gerekecektir. Ayrıca, irdelenen çalışmaların birçoğunda kullanıcı için ortalama bir maliyet bilgisi verilmemiştir. Özellikle ülkemiz Türkiye gibi gelişmekte olan ülkelerde maliyet konusu birçok konuda belirleyici unsurdur.

Tüm bunların haricinde, destekleyici teknolojiler alanı yeniliklere çok açık ve hızlı ilerleyen bir alan olup, her yıl onlarca özgün çalışmayı karşımıza çıkarmaktadır. Bu çalışmada irdelenen uygulamaların belki de 1 yılı bulmayacak bir sürede daha yüksek işlevli alternatiflerinin ortaya çıkması muhtemeldir. Bu nedenle gelecekte, teknolojik gelişmelerin ve kullanıcı ihtiyaçlarının dikkate alınarak, değerlendirmelerin yeniden gözden geçirilmesi gerekecektir.

#### 4. Sonuç

Yukarıdaki incelemeler için seçilen uygulamaların üstün ve zayıf özellikleri açıklanmış, zayıf yönlerin geliştirilmesi için önerilerde bulunulmuştur. Bununla birlikte kullanım amaçlarına göre en uygun uygulama seçimlerinde bulunulmuştur. Şunu belirtmekte fayda vardır ki, ilgili alanda ortaya konulmuş çalışmalar bu çalışmada incelenenler sınırlı olmayıp; incelemeye tabi tutulan uygulamalar kullandıkları yöntemlerin özgünlüğü bakımından benzerlerinden sıyrılmaları sayesinde bu çalışmaya konu olmuşlardır. Yapılan irdelenmeler ışığında, kullanıcı aygıtlarının kurulumunda ve ek donanımlardan bağımsız, özellikle nesne tanıma ve derinlik çıkarımı fonksiyonlarında, yüksek performans sağlayan bir çözüm geliştirilmesi için tercihin web tabanlı istemci sunucu mimarileri yönünde olması gerektiği rahatlıkla söylenebilir. Hesaplama yükü barındıran ve koşum zamanı ölçünlü donanımlarda çok uzun sürecek algoritmaların sonuçları, yüksek işlem gücü barındıran sunucular sayesinde alt ve orta

maliyetli mobil cihazlarda yüksek hızda değerlendirilebilir olmaktadır.

**Çizelge-1. Uygulamaların özgün değerlerinin özet çizelgesi. Sol sütunda uygulamanın, eğer verilmişse, özel ismi, sağ sütunda ise özgün değerinin kısa ifadesi yer almaktadır.**

[1] (Akın ve Cömert, 2023)	Anlık görüntü üzerinden derinlik çıkarımı ve nesne tanıma özellikleri sunan bir dış mekan yönlendirme uygulaması.
[7] ARIANNA	İç mekanlarda zemin işaretçileri üzerinde yönlendirme sağlayan bir iç mekan yönlendirme uygulaması.
[8] ARIANNA +	İç mekanlarda zemin yüzeyinin ve engellerin ToF ve lidar duyargalar yardımıyla algılanıp bireyin yönlendirmesini sağlayan bir yönlendirme uygulaması.
[9] NavCog	BLE cihaz ağı ile anlık konum kestirimi yapılarak bireyin yönlendirilmesini sağlayan bir iç mekan yönlendirme uygulaması.
[12] SUGAR	UWB cihaz ağı ile anlık konum kestirimi yapılarak bireyin yönlendirilmesini sağlayan bir iç mekan yönlendirme uygulaması.
[13] Navibaston	Çevrimiçi harita altlıkları üzerinden dış mekan yönlendirme sağlayan bir mobil uygulama.
[14] WeWALK	Engel algılama ve dış mekan yönlendirme sağlayan bir akıllı baston uygulaması.
[20] (Lin vd., 2017)	Derinlik çıkarımı ve nesne tanıma ile engel algılama hizmeti sunan bir mobil uygulama
[23] Watch Your Step	Kızılötesi duyarga ile engel algılama sağlayan bir mobil uygulama
[26] (Bauer vd., 2020)	Anlık görüntü üzerinden derinlik çıkarımı ve nesne tanıma özellikleri sunan bir mobil uygulama.
[31] (Venkat Ragavan vd., 2023)	Anlık görüntü üzerinden nesne tanıma, yazı okuma ve sahne yorumlama sağlayan bir uygulama
[32] (Maya-Martinez vd., 2023)	Taşınabilir atanmış dizge üzerinden nesne tanıma işleminin koşum zamanı değerlendirmesi

**Tablo 2.Uygulamaların kıyaslama tablosu**

	[7]	[9]	[12]	[13]	[14]	[8]	[20]	[23]	[26]	[1]	[31]	[32]
Platform bağımsızlığı	+	iOS	+	+	+	iOS	+	Android	+	+	+	-
Ek donanım yükümlülüğü	+	+	+	-	+	-	-	-	+	-	-	+
Sağlanan çıktı çeşitliliği (sayısı)	1	1	1	1	2	3	2	1	3	3	2	1
Derinlik Çıkarımı Hatası (metre)	-	-	-	-	-	-	<2,5	-	0,67	0,84	-	-
Derinlik Çıkarımı Menzili (metre)	-	-	-	-	2,50	-	10,00	3,00	20,00	20,00	-	-
Derinlik Çıkarımı Hızı (milisaniye)	-	-	-	-	-	-	-	-	136	140	-	-
Obje Tanıma Doğruluğu	-	-	-	-	-	0,90	0,60	-	0,74	0,63	-	0,73
Obje Tanıma Hızı (milisaniye)	-	-	-	-	-	-	-	-	106	27	441	110
İç mekan yönlendirmesi	+	+	+	-	-	+	-	-	-	-	-	-
Dış mekan yönlendirmesi	-	-	-	+	-	-	-	-	-	+	-	-
Geri bildirim türü	Titreşim	Ses	Ses	Ses	Ses + Titreşim	Ses + Titreşim	Ses	Ses + Titreşim	Ses + Titreşim	Ses	Ses	-

## Kaynakça

- [1] Akin, A.T. and Cömert, Ç., "The development of an augmented reality audio application for visually impaired persons.", *Multimedia Tools and Applications*, 82(11), 17493-17512, (2023).
- [2] Elmannai, W., and Khaled E., "Sensor-based assistive devices for visually-impaired people: Current status, challenges, and future directions.", *Sensors*, 17.3:565, (2017).
- [3] Csapó, Á., et al., "A survey of assistive technologies and applications for blind users on mobile platforms: a review and foundation for research.", *Journal on Multimodal User Interfaces*, 9.4: 275-286 (2015).
- [4] Croce, D., et al., "Enhancing tracking performance in a smartphone-based navigation system for visually impaired people.", *24th Mediterranean Conference on Control and Automation (MED)*, IEEE, (2016).
- [5] Tapu, R., Bogdan M., and Titus Z., "Wearable assistive devices for visually impaired: A state of the art survey.", *Pattern Recognition Letters*, 137: 37-52, (2020).
- [6] Maidenbaum, S., Sami A., and Amir, A.i., "Sensory substitution: closing the gap between basic research and widespread practical visual rehabilitation.", *Neuroscience & Biobehavioral Reviews*, 41: 3-15, (2014).
- [7] Gallo, P., et al. "ARIANNA: pAth recognition for indoor assisted navigation with augmented perception.", *arXiv preprint, arXiv:1312.3724*, (2013).
- [8] Lo Valvo, A., et al., "A navigation and augmented reality system for visually impaired people.", *Sensors*, 21.9: 3061, (2021).
- [9] Ahmetovic, D., et al. "NavCog: a navigational cognitive assistant for the blind.", *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, (2016).
- [10] Alghamdi, S., van Schyndel R., and Ibrahim K., "Accurate positioning using long range active RFID technology to assist visually impaired people.", *Journal of Network and Computer Applications*, 41: 135-147, (2014).
- [11] Plikynas, D., et al., "Indoor navigation systems for visually impaired persons: Mapping the features of existing technologies to user needs.", *Sensors*, 20.3: 636, (2020).
- [12] Martinez-Sala, A., et al., "Design, implementation and evaluation of an indoor navigation system for visually impaired people.", *Sensors*, 15.12: 32168-32187, (2015).
- [13] <http://gorengoz.aile.gov.tr>, "Gören Göz", (Erişim: 18 Mayıs 2022).
- [14] <https://wewalk.io/tr/>, "Dünyanın En Akıllı Bastonu ve Mobil Uygulaması", (Erişim: 18 Mayıs 2022).
- [15] Wang, W., "Understanding augmented reality and ARKit.", *Beginning ARKit for iPhone and iPad*, Apress, Berkeley, CA, 1-17, (2018).
- [16] Du, R., et al., "DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality.", *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, (2020).
- [17] Liu, W., et al., "Ssd: Single shot multibox detector.", *European conference on computer vision*, Springer, Cham, (2016).
- [18] Szegedy, C., et al., "Going deeper with convolutions.", *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2015).
- [19] Oufqir, Z., El Abderrahmani A., and Satori K., "ARKit and ARCore in serve to augmented reality.", *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*, IEEE, (2020).
- [20] Lin, B.-S., Lee C.-C., and Chiang P.-Y., "Simple smartphone-based guiding system for visually impaired people.", *Sensors*, 17.6: 1371, (2017).
- [21] Redmon, J., et al., "You only look once: Unified, real-time object detection.", *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2016).
- [22] Girshick, R., "Fast r-cnn.", *Proceedings of the IEEE international conference on computer vision*, (2015).
- [23] Sun, M., et al. "'Watch Your Step': Precise Obstacle Detection and Navigation for Mobile Users Through Their Mobile Service.", *IEEE Access*, 7: 66731-66738, (2019).
- [24] Marder-Eppstein, E., "Project Tango", <https://dl.acm.org/doi/10.1145/2933540.2933550>, 25-25, (2016).
- [25] <https://www.theverge.com/2017/12/15/16782556/project-tango-google-shutting-down-arcore-augmented-reality>, "Google's Project Tango is shutting down because ARCore is already here", (Erişim: 18 Mayıs 2022).
- [26] Bauer, Z., et al. "Enhancing perception for the visually impaired with deep learning techniques and low-cost wearable sensors.", *Pattern recognition letters*, 137: 27-36, (2020).
- [27] Laina, I., et al. "Deeper depth prediction with fully convolutional residual networks.", *2016 Fourth international conference on 3D vision (3DV)*, IEEE, (2016).
- [28] Lee, J. H., et al. "From big to small: Multi-scale local planar guidance for monocular depth estimation.", *arXiv preprint, arXiv:1907.10326*, (2019).
- [29] Akin, A. T., and Cömert Ç.. "Testing of a Deep Learning Model Providing Monocular Depth Estimation on Mobile Devices via Web Service.", *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, (2021).
- [30] Kastanakis, B., "Mapbox Cookbook", Packt Publishing Ltd, (2016).
- [31] Venkat R., Tarun, A. H., Yogeeshwar, S., Vishwath Kumar, B. S., and Sofana Reka, S., "A realtime portable and accessible aiding system for the blind—a cloud based approach.", *Multimedia Tools and Applications*, 1-14, (2023).
- [32] Maya-Martínez, S. U., Argüelles-Cruz, A. J., and Guzmán-Zavaleta, Z. J., "Pedestrian detection model based on Tiny-Yolov3 architecture for wearable devices to visually impaired assistance.", *Frontiers in robotics and AI*, 10., (2023).
- [33] Fahme, M.U.S., and Khan T. H., "Choose Your Arsenal." *How to Make a Game*. Apress, Berkeley, CA, 31-40, (2021).
- [34] Tandel, S., and Jamadar, A., "Impact of progressive web apps on web app development.", *International Journal of Innovative Research in Science, Engineering and Technology* 7.9: 9439-9444, (2018).
- [35] Metzger, F., et al. "The Prospects of Cloud Gaming: Do the Benefits Outweigh the Costs?.", (2016).
- [36] Kang, W., and Han, Y., "SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization.", *IEEE Sensors journal* 15.5: 2906-2916, (2014).
- [37] Lin, T. Y., et al., "Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V* 13., 740-755, Springer International Publishing., (2014).
- [38] Explore connected papers in a visual graph, <https://www.connectedpapers.com/>, Son Erişim: 31/05/2023

## (1057, 31)-Aralıklı Dizilerden Üretilen Boole Fonksiyonlar

### Boolean Functions Generated from (1057, 31)-Interleaved Sequences

Selçuk KAVUT

Balıkesir Üniversitesi, Mühendislik Fakültesi

Bilgisayar Mühendisliği

Balıkesir, Türkiye

skavut@balikesir.edu.tr

ORCID: 0000-0002-9460-1418

#### Öz

1983 yılında keşfedildikten itibaren günümüzde halen bilinen en yüksek doğrusal olmama değerine (16276) sahip olan 15 değişkenli Patterson-Wiedemann (PW) fonksiyonlarının, özel bir yapıda bulunan (151, 217)-aralıklı dizilerden üretilen döngüsel simetrik Boole fonksiyonları (DSBF'ler) olarak yorumlanabildiği bilinmektedir. İlgili yayınlarda, aynı doğrusal olmama değerine ulaşan başka bir inşa veya arama yöntemi bilinmemekle birlikte, tam arama veya sezgisel arama yöntemleri ile (151, 217)- ve (217, 151)-aralıklı dizilerden, büyük-bağlaşım sınırını (16256) aşan doğrusal olmama değerine sahip genelleştirilmiş DSBF'lerin elde edilebildiği gösterilmiştir. Ancak, bahsedilen yöntemlerle ulaşılan en iyi doğrusal olmama değeri 16268'i aşamamıştır. Bu çalışmamızda, (1057, 31)-aralıklı dizilerden üretilen DSBF'ler araştırılmış ve sezgisel arama yöntemi ile 16272 doğrusal olmama değerine ulaşılmıştır.

**Anahtar Sözcükler:** Aralıklı dizi, döngüsel simetrik Boole fonksiyonu (DSBF), doğrusal olmama

#### Abstract

It is known that Patterson-Wiedemann (PW) functions with 15-variables, which still have the highest known nonlinearity value (16276) since their discovery in 1983, can be interpreted as rotation-symmetric Boolean functions (RSBFs) produced from (151, 217)-interleaved sequences which are in the form of a special structure. In the related literature, though no other search/construction method achieving the same nonlinearity value is known, it has been shown that

generalized RSBFs with nonlinearity exceeding the bent-concatenation bound (16256) can be obtained from (151, 217)- and (217, 151)-interleaved sequences by using exhaustive or heuristic search methods. However, the best nonlinearity value reached by these methods could not exceed 16268. In this study, RSBFs produced from (1057, 31)-interleaved sequences are investigated and the nonlinearity value of 16272 is attained by a heuristic search method.

**Keywords:** Interleaved sequence, nonlinearity, rotation-symmetric Boolean function (RSBF)

#### 1. Giriş

Simetrik kriptografide, bir kriptosistemde kullanılan Boole fonksiyonları önemli yapı taşlarıdır ve en önemli kriptografik özelliklerinden birisi, (akan şifrelerde) en iyi afin yaklaşıklama saldırısına [1] ve (blok şifrelerde) doğrusal kriptanalize [2] karşı dayanıklılığın sağlanması için yüksek olması gereken doğrusal olmama değeridir.  $n$  değişkenli bir Boole fonksiyonu  $f: F_2^n \rightarrow F_2$ , burada  $F_2 = \{0,1\}$ ,  $n$  bit girişi 1 bit çıkışa dönüştüren bir fonksiyon olarak tanımlanır. Çift  $n$  sayıları için,  $n$  değişkenli Boole fonksiyonların alabileceği en yüksek doğrusal olmama değeri  $2^{n-1} - 2^{n/2-1}$  olup, bu değeri alan Boole fonksiyonlara büyük fonksiyonlar denilmektedir. Tek  $n$  sayıları için ise, en yüksek doğrusal olmama değerinin üst sınırı  $2 \times [2^{n-2} - 2^{n/2-2}]$  olarak verilmektedir [3].  $n$  bir tek sayı,  $g$  ve  $h$  fonksiyonları ( $n-1$ ) değişkenli büyük fonksiyonlar olmak üzere, bu fonksiyonların bağlaşımı olarak tanımlanan  $n$  değişkenli  $f$  fonksiyonu  $f(x_0, x_1, \dots, x_{n-1}) = x_0 g(x_1, x_2, \dots, x_{n-1}) \oplus (1 \oplus x_0) h(x_1, x_2, \dots, x_{n-1})$ , büyük bağlaşım sınırı olarak bilinen  $2^{n-1} - 2^{(n-1)/2}$  doğrusal olmama değerine sahiptir.  $n \leq 7$  tek sayıları için,  $n$  değişkenli Boole fonksiyonların alabileceği en yüksek doğrusal olmama değeri bahsedilen büyük-bağlaşım sınırındır. Bununla birlikte,  $n > 7$  herhangi bir tek sayı olmak üzere, yayınlarda  $n$  değişkenli Boole fonksiyonların ulaşabileceği en yüksek

doğrusal olmama değeri bilinmemektedir. Bükük bağlaşım sınırından yüksek doğrusal olmama değerine sahip Boole fonksiyonları, yayınlarda Patterson-Weidemann (PW) fonksiyonları olarak bilinen  $16276 (= 2^{15-1} - 2^{(15-1)/2} + 20)$  doğrusal olmama değerine sahip ve birbiri ile afin ilişkili olmayan iki 15 değişkenli Boole fonksiyonun 1983 yılında Patterson ve Wiedemann tarafından keşfedilmesiyle [4] ortaya çıkmıştır. Bunun bir sonucu olarak, PW fonksiyonlarından bir tanesi ve bir bükük fonksiyonun dolaysız toplamı ile,  $n > 15$  tek sayıları için,  $2^{n-1} - 2^{(n-1)/2} + 20 \cdot 2^{(n-15)/2}$  doğrusal olmama değerine sahip  $n$  değişkenli Boole fonksiyonlar elde edilebilmektedir. Diğer taraftan, Kavut ve Yücel tarafından 2007 yılında  $242 (= 2^{9-1} - 2^{(9-1)/2} + 2)$  doğrusal olmama değerine sahip 9 değişkenli Boole fonksiyonların ortaya çıkarılmasıyla [5],  $n > 9$  tek sayıları için,  $2^{n-1} - 2^{(n-1)/2} + 2 \cdot 2^{(n-9)/2}$  doğrusal olmama değerine sahip  $n$  değişkenli Boole fonksiyonlar, bahsedilen dolaysız toplam yöntemi ile elde edilebilir olmuştur. Ancak bu doğrusal olmama değerinin  $n \geq 15$  için PW fonksiyonlarından elde edilen doğrusal olmama değerinden düşük olduğuna dikkat edilmelidir.

Bükük bağlaşım sınırını aşan doğrusal olmama değerine sahip Boole fonksiyonlarının inşası simetrik kriptografide karşılaşılan en zor problemlerdendir ve ilgili yayınlarda bu tür fonksiyonları ortaya çıkaran çalışmalar temel olarak, doğrusal olmama yönünden zengin kriptografik alt uzayların belirlenmesi ve bu alt uzaylarda tam veya sezgisel arama yöntemlerinin uygulanmasına dayanır. Patterson ve Wiedemann, keşfettikleri Boole fonksiyonları  $F_{2^5}^* \cdot F_{2^3}^*$  döngüsel grubu aksiyonu altında değişmez özelliğe sahip eş güçlü fonksiyonların oluşturduğu  $2^{11}$  büyüklüğündeki arama uzayında tam arama yürüterek ortaya çıkarmışlardır [4]. Bu fonksiyonların özel bir yapıda bulunan (151, 217)-aralıklı dizilerden elde edilen DSBF'ler olarak yorumlanabileceği, Gangopaphyay ve ark. tarafından gösterilmiştir [6]. Kavut ve Yücel, doğrusal olmama değeri bükük-bağlaşım sınırını aşan 9 değişkenli Boole fonksiyonları arama uzayı  $2^{104}$  büyüklüğündeki (genelleştirilmiş 3-döngüsel simetrik Boole fonksiyonların (3-DSBF'lerin) doğrusal olmama yönünden zengin bir alt uzayı olan) dihedral simetrik Boole fonksiyonları için yürüttükleri sezgisel arama algoritması ile elde etmişlerdir [5]. Yakın zamanda, (151, 217)-aralıklı dizilerden üretilen ve arama uzayı büyüklükleri sırasıyla  $2^{28.2}$  ve  $2^{47.85}$  olan 3- ve 5-DSBFs'ler için tam arama yapılarak, her iki arama uzayında da bükük bağlaşım sınırından yüksek doğrusal olmama değerine sahip yeni PW türü Boole fonksiyonlar tespit edilmiştir [7]. Daha sonra, PW inşa yönteminin genelleştirildiği [8] çalışmasında ise (217,151)-aralıklı dizilerden üretilen ( $k$ -) DSBF'ler için arama yapılmıştır. Özel olarak, arama uzayının büyüklüğü sırasıyla  $2^{15.3}$  ve  $2^{44.1}$  olan DSBF'ler ve 3-DSBF'ler için tam arama, arama uzayının büyüklüğü  $2^{88.1}$  olan 5-DSBF'ler için ise sezgisel arama yöntemleri ile bükük bağlaşım sınırından yüksek doğrusal olmama değerine sahip yeni Boole fonksiyonlar elde edilmiştir. Bununla birlikte, bahsedilen [7, 8] çalışmalarında ulaşılan en yüksek doğrusal olmama değeri 16268 olmuştur.

Çalışmamızda, (1057, 31)-aralıklı dizilerden üretilen DSBF'ler ele alınarak, bu DSBF'lerin oluşturduğu  $2^{73}$  büyüklüğündeki

arama uzayında, daha önce 21 değişkenli PW türü fonksiyonlar için kullanılan [9] sezgisel arama algoritması bu çalışmada ele alınan duruma uygulanmıştır. Bunun sonucunda, [7, 8] çalışmalarında ulaşılan en iyi doğrusal olmama değerinden (16268) daha yüksek bir doğrusal olmama değeri (16272) elde edilmiştir. Bu çalışmada ve daha önceki [4, 5, 7-9] çalışmalarında elde edilen bükük-bağlaşım sınırından yüksek doğrusal olmama değerine sahip Boole fonksiyonları dengesiz olduklarından (diğer bir ifadeyle, doğruluk tablolarındaki 0'ların sayısı 1'lerin sayısına eşit olmadığından) herhangi bir kripto sistemde doğrudan kullanılamazlar. Bununla birlikte, bu fonksiyonlar esneklik, cebirsel derece, cebirsel bağımsızlık, mutlak gösterge ve doğrusal olmama gibi kriptografik özellikler bakımından güçlü Boole fonksiyonların tasarımı için temel yapıtaşları olarak kullanılmaktadır [10-12]. Özel olarak, [12] çalışmasında PW fonksiyonlarının komşuluğunda kaba kuvvet arama ile dengeli Boole fonksiyonlar elde edilmiş olup, bu çalışmada ise bükük-bağlaşım sınırını aşan doğrusal olmama değerine sahip, aralıklı dizi yapısında ve dengeli olmayan DSBF'ler sezgisel arama yöntemi ile üretilmektedir. Bahsedilen kriptografik özellikleri taşıyan Boole fonksiyonları, özellikle donanım kapasitesinin kısıtlı olduğu hafif sıklet kripto sistemlerde, yaygın şekilde kullanılan kombinasyon üretici ve filtre üretici tasarımlarına sahip akan şifreler için doğrusal olmayan birleştirici olarak doğrudan kullanılabilirler. Ayrıca, çalışmamızda elde edilen Boole fonksiyonları döngüsel simetrik olduğundan ve DSBF'ler sahip oldukları (sadece yörünge temsilcilerine karşılık gelen çıkış bitleri ile ifade edilebilmelerini sağlayan) cebirsel özellik nedeniyle verimli bir şekilde gerçekleştirilebildiğinden [13, 14], bulunan DSBF'lerden türetililecek güçlü kriptografik özelliklere sahip fonksiyonların verimli bir şekilde gerçekleştirme olanağına sahip olduğu düşünülmektedir.

## 2. Temel Bilgiler

$f: F_{2^n} \rightarrow F_2$ ,  $n$  değişkenli bir Boole fonksiyon olsun.  $F_{2^n}$  sonlu cisminin elemanları, derecesi  $n$  olan bir indirgenemez polinom ile elde edilen ve  $n$  bit ile gösterilen ikili polinomlar olduğundan,  $f$  fonksiyonu eşdeğer olarak  $F_2^n \rightarrow F_2$  biçiminde düşünülebilir. Her  $\alpha \in F_{2^n}$  için,  $f(\alpha) = f(\alpha^2)$  koşulunu sağlayan fonksiyonlara eş güçlü fonksiyonlar denilmektedir. Burada, eğer bahsedilen indirgenemez polinom bir primitif polinom olarak seçilir ve normal taban kullanılırsa, eş güçlü fonksiyonlar DSBF'lere [15, 16], her  $\alpha \in F_{2^n}$  için  $f(\alpha) = f(\alpha^{2^k})$  koşulunu sağlayan fonksiyonlar ise, burada  $k$  bir sabit ve  $k|n$ , (genelleştirilmiş)  $k$ -DSBF'lere [5] karşılık gelir. Yayınlarda keşfedilen bükük bağlaşım sınırından yüksek doğrusal olmama değerine sahip fonksiyonların ( $k$ -)DSBF'ler sınıfına ait olmaları, bu sınıfın doğrusal olmama yönünden zengin olduğunun göstergesidir.

Her  $x \in F_{2^n}$  için  $Tr(x) = x \oplus x^2 \oplus x^{2^2} \oplus \dots \oplus x^{2^{n-1}}$  ve  $c \in F_2$  olmak üzere,  $n$  değişkenli bir afin fonksiyon  $Tr(ax) \oplus c$  biçimde tanımlanır.  $h_\alpha(x) = Tr(ax) \oplus 1$  ve  $l_\alpha(x) = Tr(ax)$  olsun, burada  $l_\alpha(x)$  doğrusal fonksiyon olarak isimlendirilir.  $n$  değişkenli iki fonksiyon  $f$  ve  $g$  için, aralarındaki uzaklık  $d(f, g)$ ,  $2^n$  elemanlı doğruluk tabloları arasındaki Hamming uzaklığı olarak tanımlanır. Diğer bir ifadeyle,  $d(f, g) = |\{x \in F_{2^n}: f(x) \neq g(x)\}|$ . Herhangi bir

Boole fonksiyon  $f$  için doğrusal olmama değeri, tüm afin fonksiyonlara (yani,  $h_\alpha(x)$  ve  $l_\alpha(x)$  fonksiyonlarına) olan Hamming uzaklıkların en küçüğü olarak ifade edilmektedir.

$p, q > 2$  birer asal sayı olmak üzere  $n = pq$  ve  $f: F_{2^n} \rightarrow F_2$  fonksiyonu  $F_{2^p}^*$  döngüsel grubu altında değişmez olsun. Bu durumda,  $f$  fonksiyonu aşağıda verilen ve her bir sütunu tümü 0 veya tümü 1 olan bir  $(d, r)$ -aralıklı dizi biçiminde yorumlanabilir:

$$\begin{bmatrix} f(\omega^0) & f(\omega^1) & f(\omega^2) & \dots & f(\omega^{d-1}) \\ f(\omega^d) & f(\omega^{d+1}) & f(\omega^{d+2}) & \dots & f(\omega^{2d-1}) \\ f(\omega^{2d}) & f(\omega^{2d+1}) & f(\omega^{2d+2}) & \dots & f(\omega^{3d-1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f(\omega^{(r-1)d}) & f(\omega^{(r-1)d+1}) & f(\omega^{(r-1)d+2}) & \dots & f(\omega^{rd-1}) \end{bmatrix}$$

burada  $r = 2^p - 1$ ,  $d = (2^n - 1)/r$  ve  $\omega, F_{2^n}$  sonlu cisminin bir primitif elemanıdır. Bu gösterimde,  $f(0)$  değeri keyfi olarak 0 veya 1 alınabilir; çalışmamızda  $f(0) = 0$  alınmıştır. Verilen aralıklı dizinin satırlarını 0'dan  $r - 1$ 'e ve sütunlarını 0'dan  $d - 1$ 'e numaralandıralım. Eğer  $f$  fonksiyonu eş güçlü ise, negatif olmayan bir  $s$  tamsayısı için

$$i \sim j \text{ ancak ve ancak } i \equiv j2^s \pmod{d}$$

olarak tanımlanan eşdeğerlik ilişkisine göre sütunların eşdeğerlik sınıflarına bölüntüleneceğine dikkat edilmelidir, burada  $i, j \in \{0, 1, \dots, d - 1\}$ . Diğer bir ifadeyle,  $(d, r)$ -aralıklı dizisi ile temsil edilen eşgüçlü bir  $f$  Boole fonksiyonunun, aynı eşdeğerlik sınıfındaki bütün sütunları ya tümü 1 ya da tümü 0 olan sütunlardan oluşur. Bu eşdeğerlik ilişkisini  $\rho_d$  ile gösterelim. Çalışmamızda ele aldığımız  $n = 15$  durumu için,  $\rho_{1057}$  eşdeğerlik ilişkisi ile  $(1057, 31)$ -aralıklı dizilerden elde edilen eşgüçlü fonksiyonların sayısının (diğer bir ifadeyle, arama uzayının büyüklüğünün)  $2^{73}$  olduğu görülebilir.

Aralıklı dizi gösterimine sahip herhangi bir  $f$  fonksiyonu için, tümü 1 olan sütunların  $\ell$  tane olduğunu kabul edersek, bu fonksiyonun tüm afin fonksiyonlara olan uzaklıkları aşağıdaki gibi hesaplanabilir [4]:

$$\begin{aligned} d(f, \mathbf{0}) &= \ell(2^p - 1), & d(f, \mathbf{1}) &= 2^n - \ell(2^p - 1), \\ d(f, l_\alpha) &= 2^{n-1} - 2^p t(\alpha) + \ell, \\ d(f, h_\alpha) &= 2^{n-1} + 2^p t(\alpha) - \ell, \end{aligned} \quad (1)$$

burada,  $\alpha \in F_{2^n}^*$  olmak üzere,  $t(\alpha)$  değeri  $f$  ve  $h_\alpha$  fonksiyonlarının aynı pozisyonlarındaki tümü 1 olan sütunlarının sayısını,  $\mathbf{0}$  ve  $\mathbf{1}$  ise sırasıyla tümü 0 ve tümü 1 olan  $2^n$  uzunluğundaki ikili dizileri göstermektedir. Doğrusal olmama tanımından,  $f$  fonksiyonunun büyük bağlaşımlarından yüksek doğrusal olmama değerine sahip olabilmesi için, (1) ile verilen eşitlikler vasıtasıyla, aşağıdaki eşitsizliklerin sağlanması gerektiği görülmektedir:

$$\begin{aligned} \kappa^- < \ell < \kappa^+, \\ \frac{1}{2^p} (\kappa^- - 2^{\frac{n-1}{2}}) < t(\alpha) < \frac{1}{2^p} (\kappa^+ + 2^{\frac{n-1}{2}}), \end{aligned} \quad (2) \quad (3)$$

burada  $\kappa^\pm = (2^{n-1} \pm 2^{\frac{n-1}{2}})/(2^p - 1)$ .  $\ell$  için verilen (2) koşulu,  $f$  fonksiyonunun Hamming ağırlığını (eşdeğer olarak, tümü 1 olan sütunların sayısını) belirlediği için, ağırlık koşulu olarak adlandırılmaktadır. (3) ile verilen koşulun sağlanması

için  $2^n - 1$  eşitsizliğin sağlanması gerektiği düşünülebilir; ancak, Gangopadhyay ve ark. tarafından gösterildiği [6] gibi,  $\rho_d$  eşdeğerlik ilişkisi ile elde edilen tüm eşdeğerlik sınıflarının sayısı kadar eşitsizlikten oluşan bir eşitsizlik sisteminin çözülmesi, bu koşulun sağlanması için yeterlidir.

Bölüm 3'te verilen arama algoritmasında,  $F_2^{15}$  sonlu cisminin gerçekleştirilmesi için  $x^{15} + x + 1$  primitif polinomu kullanılmıştır.

Bu bölümde anlatılanların örneklenilmesi amacıyla,  $F_{2^2}^*$  döngüsel grubu altında değişmez olan 4 değişkenli Boole fonksiyonları ele alalım ve  $F_2^4$  sonlu cisminin gerçekleştirilmesi için  $x^4 + x + 1$  primitif polinomu kullanalım. Örnekteki Boole fonksiyonlar, tümü 1 veya tümü 0 olan sütunlardan oluşan  $(5, 3)$ -aralıklı diziler ile temsil edilebilir ve bu durumda ( $\mathbf{0}$  doğrusal fonksiyonu dışında) tüm doğrusal fonksiyonlar, daha önce bahsedilen aralıklı dizinin tanımından, aşağıdaki matrisler ile ifade edilebilir:

$$\begin{aligned} I_0 &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}, I_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \\ I_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}, I_3 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}, \\ I_4 &= \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}, I_5 = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \\ I_6 &= \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, I_7 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \\ I_8 &= \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}, I_9 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \\ I_{10} &= \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}, I_{11} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \\ I_{12} &= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}, I_{13} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \\ I_{14} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}, \end{aligned}$$

burada  $I_i$  matrisi,  $F_2^4$  sonlu cisminin bir primitif elemanı  $\omega$  olmak üzere,  $\alpha = \omega^i$  için  $l_\alpha(x) = Tr(\alpha x)$  doğrusal fonksiyonuna karşılık gelen aralıklı dizidir.  $h_\alpha(x) = Tr(\alpha x) \oplus 1$  fonksiyonlarına karşılık gelen aralıklı dizilerin ise yukarıda verilen matrislerin tümleyenleri olarak (diğer bir ifadeyle, matrisleri oluşturan bitlere mod 2'ye göre 1 ekleyerek) elde edilebildiği görülmektedir.

Örneğin, aşağıdaki aralıklı diziyeye sahip bir  $f$  Boole fonksiyonunun doğrusal olmama değerini (1) eşitliklerini kullanarak bulalım:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$n = 4$ ,  $p = 2$  ve tümü 1 olan 2 tane sütun olduğundan ( $\ell = 2$  anlamına gelir),  $d(f, \mathbf{0}) = 2(2^2 - 1) = 6$  ve  $d(f, \mathbf{1}) = 2^4 -$

$2(2^2 - 1) = 10$  bulunur. Verilen aralıklı dizide ikinci ve dördüncü sütunlar tümü 1 sütunlarıdır.  $t(\alpha)$  değeri,  $h_\alpha$  fonksiyonunu temsil eden aralıklı dizinin bu pozisyonlarda bulunan tümü 1 olan sütunlarının sayısıdır. Böylelikle  $t(\alpha)$  değerleri  $\alpha = \omega^i$  olmak üzere,  $i = 0, 1, \dots, 14$  için sırasıyla 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1 olarak bulunur. Bu değerlere karşılık gelen  $d(f, l_\alpha)$  ve  $d(f, h_\alpha)$  uzaklıkları ise sırasıyla 10, 10, 6, 10, 6, 10, 10, 6, 10, 6, 10, 10, 6, 10, 6 ve 6, 6, 10, 6, 10, 6, 6, 10, 6, 10, 6, 6, 10, 6, 10 olarak elde edilir. Doğrusal olmama değeri tüm afin fonksiyonlara olan en küçük uzaklık olarak tanımlandığından,  $f$  Boole fonksiyonunun doğrusal olmama değerinin 6 olduğu görülmektedir. En yüksek doğrusal olmama değerine sahip olduğundan, burada  $f$  bir bükük fonksiyondur.

Şimdi  $F_{2^2}^*$  döngüsel grubu altında değişmez olan 4 değişkenli tüm bükük fonksiyonları bulalım. Bu durumda, (1) ile verilen uzaklıkların tümü 5'ten büyük olmalıdır:

$$\begin{aligned} d(f, \mathbf{0}) &= 3\ell > 5, & d(f, \mathbf{1}) &= 16 - 3\ell > 5, \\ d(f, l_\alpha) &= 8 - 4t(\alpha) + \ell > 5, \\ d(f, h_\alpha) &= 8 + 4t(\alpha) - \ell > 5. \end{aligned}$$

Burada,  $n = 4$  için bükük fonksiyonların doğrusal olmama değeri 6 olduğundan, yukardaki uzaklıklar doğrudan 6'ya eşit alınabilirdi. Bununla birlikte, örneğimizin çalışmamızda izlenen yöntemle paralel olması amacıyla uzaklıklar 5'ten büyük olarak yazılmıştır (Çalışmamızda ele alınan  $n = 15$  durumu için en yüksek doğrusal olmama değeri bilinmemektedir ve (1) ile verilen uzaklıklar bükük-bağlaşım sınırından büyük olarak seçilmiştir.).  $\ell$  ve  $t(\alpha)$  birer tamsayı olmak zorunda olduklarından, elde edilen bu eşitsizlikler aşağıda gibi ifade edilebilir:

$$2 \leq \ell \leq 3,$$

$$0 \leq t(\alpha) \leq 1 \ (\ell = 2 \text{ için}) \text{ ve } 1 \leq t(\alpha) \leq 1 \ (\ell = 3 \text{ için}).$$

$\ell$  değeri tümü bir olan sütunların sayısı olduğundan, bu sütunların 2 veya 3 tane olması gerektiği görülmektedir.

Dolayısıyla bu koşulu sağlayan aralıklı dizilerin sayısı  $\binom{5}{2} + \binom{5}{3} = 20$  olarak bulunur.  $t(\alpha)$  için elde edilen diğer koşullar ise Gangopadhyay ve ark. tarafından PW inşası için verilen algoritmanın [6] örneğimize uygulanması ile aşağıdaki eşitsizlik sistemlerine indirgenebilir:

$$0 \leq s_j \leq 1 \ (\ell = 2 \text{ için}) \text{ ve } 1 \leq s_j \leq 1 \ (\ell = 3 \text{ için}),$$

burada  $s_j = f(\omega^j)$ ,  $j \in \{0, 1, 2, 3, 4\}$ ,  $f$  Boole fonksiyonuna karşılık gelen aralıklı dizinin  $j$ . sütununun ilk elemanını temsil etmektedir.  $\ell = 3$  için bulunan sistem  $\{s_j = 1 \ \forall j \in \{0, 1, 2, 3, 4\}\}$ , aralıklı dizinin tüm sütunlarının tümü 1 olan sütunlar olması gerektiğini gösterir; bu ise tümü 1 olan sütun sayısının 3 olması koşulu ile çeliştiğinden,  $\ell$  değerinin 3 olamayacağı anlamına gelir (dolayısıyla, yukarda bulunan aralıklı dizilerin sayısı 20'den 10'a düşer). Diğer taraftan,  $\ell = 2$  için bulunan sistem  $\{0 \leq s_j \leq 1 \ \forall j \in \{0, 1, 2, 3, 4\}\}$  herhangi bir kısıtlamaya neden olmadığından,  $f$ 'nin bükük fonksiyon olması için tek koşul  $\ell = 2$  ağırlık koşuludur; diğer bir ifadeyle,  $f$ 'nin aralıklı dizisinde herhangi 2 sütunun tümü 1 ve diğer sütunların tümü 0 olan sütunlar olmasıdır. Bunun sonucunda,  $\binom{5}{2} = 10$  tane bükük fonksiyon elde edilir. Bir

Boole fonksiyonunun doğruluk tablosundaki tüm bitlerin tümleyeni alındığında doğrusal olmama özelliği değişmeyeceğinden, bu şekilde elde edilen Boole fonksiyonlar da hesaba katıldığında toplam olarak 20 bükük fonksiyon bulunmuş olur.

Örneğimizde eşgüçlü bükük fonksiyonların var olmadığı da görülebilir. Özel olarak,  $\rho_5$  eşdeğerlik ilişkisine göre aralıklı dizinin sütunları bölüntülendiğinde, ilk sütun dışındaki tüm sütunlar aynı eşdeğerlik sınıfına ait olmaktadır. Bu durum ağırlık koşulunun sağlanmasını olanaksız kıldığından,  $F_{2^2}^*$  döngüsel grubu altında değişmez olan 4 değişkenli bükük fonksiyonların içinde eşgüçlü bir Boole fonksiyon olmadığı sonucuna varılır.

$f: F_2^n \rightarrow F_2$ ,  $n$  değişkenli bir Boole fonksiyon olsun. Aynı değişken sayısına sahip bir  $g$  fonksiyonu için,  $f(x)g(x) = 0 \ \forall x \in F_2^n$  koşulu sağlanıyorsa,  $g$  fonksiyonuna  $f$  fonksiyonunun sıfırlayıcısı denir.  $f'(x) = f(x) \oplus 1 \ \forall x \in F_2^n$  olmak üzere,  $f$  ve  $f'$  fonksiyonlarının sıfırdan farklı tüm sıfırlayıcılarının oluşturduğu kümeleri sırasıyla  $AN(f)$  ve  $AN(f')$  ile gösterelim.  $f$  fonksiyonunun cebirsel bağışıklığı  $AN(f) \cup AN(f')$  kümesindeki fonksiyonların en küçük cebirsel derecesi olarak tanımlanır.  $n$  değişkenli bir Boole fonksiyonu için cebirsel bağışıklığın en fazla  $\lfloor n/2 \rfloor$  olduğu bilinmektedir [17]. Doğrusal olmama, mutlak gösterge ve cebirsel derece gibi diğer kriptografik özelliklerin tanımları örneğin [18] çalışmasında bulunabilir. [18] çalışmasında  $n$  biti  $m$  bite gönderen ve çok çıkışlı Boole fonksiyonları olarak tanımlanan S-kutuları (yerleştirme kutuları) ele alınırken, bu çalışmada  $n$  biti 1 bite gönderen (tek çıkışlı) Boole fonksiyonların ele alındığına dikkat edilmelidir.

### 3. Arama Algoritması

Ağırlık koşulunu kullanarak, (2) ile verilen eşitsizlikten  $n = 15$  ve  $p = 5$  için  $525 \leq \ell \leq 532$  elde edildiğinden, (1057, 31)-aralıklı dizisinin 1057 sütunundan en az 525 ve en fazla 532 tanesi tümü 1 olan sütunlar olmak zorundadır. Diğer taraftan,  $\rho_{1057}$  eşdeğerlik ilişkisine göre 70 tanesi 15'er, 2 tanesi 3'er ve 1 tane 1 sütundan oluşan 73 eşdeğerlik sınıfının bulunduğu görülebilir. Bu nedenle, ağırlık koşulunun sağlanması için tümü 1 olan sütunların, her biri 15 sütun içeren (70 eşdeğerlik sınıfından) 35 eşdeğerlik sınıfına ait tüm sütunlar olması gerektiği sonucuna varılır. Bu durumda, tümü 1 olan sütunların sayısı  $15 \times 35 = 525$  olduğundan, geriye kalan (3 sütunlu ve 1 sütunlu) 3 eşdeğerlik sınıfının herhangi birisinde bulunan sütunlar rastgele tümü 1 veya tümü 0 olan sütunlar olabilir. Böylelikle, gerçekleştirdiğimiz arama ağırlık koşulunu sağlayan Boole fonksiyonlar için yürütüldüğünden,  $2^{73}$  olan arama uzayının büyüklüğü  $2^{69.6} \left( \approx \binom{70}{35} \times 2^3 \right)$  büyüklüğüne düşmektedir. Bu çalışmada, daha önce  $n = 21$  durumu için önerilen [9] sezgisel arama algoritması Şekil 1'de sunulduğu gibi  $n = 15$  durumuna uyarlanarak, bahsedilen arama uzayı için yürütülmüştür.

Kullanılan arama algoritması, Boole fonksiyon tasarımında uygun maliyet fonksiyonu seçimiyle güçlü kriptografik özellikleri ortaya çıkardığı kanıtlanmış [19, 20] en dik iniş prensibine dayalı arama algoritmasıdır. Tavlama benzetimi ve tepe tırmanma gibi benzer arama algoritmaları ile



karşılaştırıldığında, bu algoritmanın tavlama benzetiminden ve tavlama benzetiminin ise genetik arama ve tepe tırmanmadan daha verimli sonuçlar ürettiği sırasıyla [19] ve [21] çalışmalarında gösterilmiştir. En dik iniş prensibine dayalı arama yönteminin bahsedilen diğer yöntemler ile karşılaştırması ve daha iyi sonuçlar üretebilmesinin sebepleri [18] çalışmasında tartışılmıştır.

(1057, 31)-aralıklı dizisi ile temsil edilen bir Boole fonksiyonun doğrusal olmama değeri, karşılık gelen doğruluk tablosu elde edilerek ve Walsh-Hadamard dönüşümü hesaplanarak ile bulunabilir. Ancak,  $n$  değişkenli bir Boole fonksiyonu için Walsh-Hadamard dönüşümünün hesaplama karmaşıklığı en iyilenmiş durumda  $O(n2^n)$  olduğundan, bu yöntem oldukça maliyetlidir. Bunun yerine, önceki bölümde bahsedildiği üzere Gangopadhyay ve ark. tarafından gösterildiği [6] gibi, ağırlık koşulunu sağlayan bir Boole fonksiyonun bükük-bağlaşım sınırından yüksek doğrusal olmama değerine sahip olup olmadığını belirlemek için,  $\rho_d$  eşdeğerlik ilişkisinin ürettiği eşdeğerlik sınıflarının sayısı kadar (bizim durumumuz için bu sayı 73) eşitsizlikten oluşan bir eşitsizlik sisteminin sağlanıp sağlanmadığına bakılabilir. Gangopadhyay ve ark.'nın PW inşası (diğer bir ifadeyle, (151, 217)-aralıklı diziler ile temsil edilen eş güçlü Boole fonksiyonları için eşitsizlik sistemini üreten algoritması, bu çalışmada ele aldığımız (1057, 31)-aralıklı diziler ile temsil edilen eş güçlü Boole fonksiyonları için uyarlanmış ve bükük-bağlaşım sınırını aşan doğrusal olmama değerine ulaşmak için sağlanması gereken eşitsizlik sistemi (aşağıda verilen (5) ile gösterildiği biçimde) elde edilmiştir.

**Girdi:** Ağırlık koşulunu sağlayacak şekilde rastgele üretilen kısaltılmış doğruluk tablosu  $s_{aday}$

**Çıktı:** Bükük-bağlaşım sınırını aşan doğrusal olmama değerini veren yineleme çıktısı  $s_{min}$

```

s ← saday;
for K = 0 to N - 1 do {
    k ← 0;
    //s vektöründe 15 elemanlı eşdeğerlik sınıflarına
    //karşılık gelen 70 bitten
    y0 ← 0'ların pozisyonları; // (35 tane)
    y1 ← 1'lerin pozisyonları; // (35 tane)
    y2 ← Geriye kalan bitlerin pozisyonları; // (3 tane)
    for i = 0 to 34 do
        for j = 0 to 34 do {
            sy0[i] ve sy1[j]'i birbiri ile değiştir;
            SET[k] ← sdeğişen; M[k] ← maliyetdeğişen;
            k ← k + 1; s ← saday;}
    for i = 1 to 7 do
        for j = 0 to 2 do {
            sy2[j] = sy2[j] ^ ((i & (1 << j)) >> j);
            SET[k] ← sdeğişen; M[k] ← maliyetdeğişen;
            k ← k + 1; s ← saday;}
    maliyetmin ← M dizisinin en küçük maliyetdeğişen değeri;
    smin ← Karşılık gelen SET dizisinin sdeğişen elemanı;
    while smin ∈ S do {
        maliyetmin değerini M dizisinden çıkart;
        maliyetmin ← M dizisinin en küçük değeri;
        smin ← Karşılık gelen SET dizisinin sdeğişen elemanı;}
    NLsmin ← smin 'den elde edilen doğrusal olmama değeri;
    if NLsmin > 16256 then
        return smin;
    s ← smin;
    saday ← s;
    S[K] ← s;}

```

**Şekil 1.** Sezgisel arama algoritması

(1057, 31)-aralıklı dizisinin (0'dan 1056'ya kadar numaralandırılan) sütunlarını  $\rho_{1057}$  eşdeğerlik ilişkisine göre bölüntülediğimizde ortaya çıkan 73 eşdeğerlik sınıfının her birini, eşdeğerlik sınıfını gösteren sütun numaralarının en küçüğü ile temsil edersek, bu temsilciler 0, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 65, 71, 73, 75, 77, 81, 85, 89, 97, 99, 103, 105, 107, 109, 113, 115, 119, 121, 123, 129, 131, 151, 161, 163, 165, 171, 173, 177, 179, 181, 185, 193, 195, 197, 225, 227, 229, 243, 245, 259, 353, 361 ve 453 olarak bulunur. Böylelikle, (1057, 31)-aralıklı dizi ile temsil edilen eş güçlü bir Boole fonksiyonu  $f$ , uzunluğu 73 olan kısaltılmış doğruluk tablosu

$$s = (s_0, \dots, s_{72}) = (f(\omega^0), \dots, f(\omega^{361}), f(\omega^{453})) \quad (4)$$

ile gösterilebilir. Bu durumda, elde ettiğimiz eşitsizlik sistemi

$$13 \leq \sum_{j=0}^{72} A_{i,j} s_j \leq 20 \quad (5)$$

biçimindedir. Burada eşitsizliğin alt ve üst sınırları (3) koşulundan hesaplanmaktadır ve  $73 \times 73$  büyüklüğündeki  $[A_{i,j}]$  katsayı matrisini elde eden algoritmanın MATLAB kodu [22]'de verilen bağlantıda sunulmuştur.

Şekil 1'de görülen algoritma, ağırlık koşulunu sağlayan rastgele bir aday çözüm  $s_{aday}$  ile başlamakta ve her yinelemede  $35 \times 35 + 7 = 1232$  komşuluk üretilmektedir. Herhangi bir yineleme girdisi  $s$  için komşuluklar ( $y_0$  ile verilen) 0 değerine sahip 35 pozisyondan birisinin ( $y_1$  ile verilen) 1 değerine sahip 35 pozisyondan birisi ile karşılıklı yer değiştirmesi ve geriye kalan 3 pozisyondaki bitlerin olası tüm farklı değerleri almasıyla elde edilmektedir. Her bir komşuluk için hesaplanan maliyet değeri (*maliyet<sub>değişen</sub>*)  $M$  dizisine ve karşılık gelen komşuluk (*s<sub>değişen</sub>*) ise *SET* dizisine kaydedilir. Sonrasında, daha önce yineleme çıktısı olarak üretilmemiş ve en düşük maliyete sahip komşuluk (*s<sub>min</sub>*) için doğrusal olmama değeri  $NL_{s_{min}}$  hesaplanır; eğer bu değer bükük-bağlaşım sınırından (16256) yüksekse algoritma çıktısı olarak verilir. Gerçekleştirdiğimiz aramada, en aza indirgenmeye çalışılan ve Şekil 2'de verilen algoritma ile hesaplanan maliyet değeri, herhangi bir komşuluk  $s = s_{değişen}$  için (5) ile gösterilen eşitsizlik sisteminin alt ve üst sınırlarından sapmalarının kareleri toplamıdır. Maliyet değeri sıfır olan 73 bit uzunluğundaki her çözüm, bükük-bağlaşım değerinden yüksek doğrusal olmama değerine sahip bir Boole fonksiyonu temsil etmektedir.

**Girdi:** Kısaltılmış doğruluk tablosu  $s$

**Çıktı:**  $s$ 'nin maliyet değeri *maliyet<sub>s</sub>*

```

for i = 0 to 72 do {
    c ← 0;
    for j = 0 to 72 do
        c ← c + Ai,j sj;
    if (c < 13)
        maliyets = maliyets + (c - 13)2;
    else if (c > 20)
        maliyets = maliyets + (c - 20)2;
    return c;
}

```

**Şekil 2.** Maliyet değerinin hesaplanması

**Çizelge-1: Bükük-bağlaşım sınırını aşan doğrusal olmama (DO) değerine sahip kısaltılmış doğruluk tabloları (CD: Cebirsel Derece, MG: Mutlak Gösterge, CB: Cebirsel Bağışıklık)**

DO	CD	MG	CB	Kısaltılmış doğruluk tablosu
16272	10	544	7	(011101100100100101101011100001001011100100000100111110010111010111000100)
16272	10	544	7	(0010110111110110001110111010111011000001110010001000011110100010000101001)
16271	15	548	7	(111101100100100101101011100001001011100100000100111110010111010111000100)
16271	15	548	7	(1010110111110110001110111010111011000001110010001000011110100010000101001)
16270	10	536	7	(101111011100011010111001110010011010010110001101000100010010001010110100)
16270	10	536	7	(1101101111111010110110001101010101100001110100101000001100000010001010110)
16269	15	292	7	(011101111110110001111000001010010001101111000010110001010001000010010110)
16269	15	532	7	(0011111011100011010111001110010011010010110001101000100010010001010110100)
16269	15	380	7	(01101101110111010110011000001111111001101000110001101010000100001000101)
16269	15	532	7	(0101101111110101101100011010101100001110100101000001100000010001010110)
16269	15	228	7	(0110101001010111100110101100101100100101100100011101011011100000011010001001)
16269	15	532	7	(0011111100100101111110100110101000100101000110011111000001001100001011101)
16268	10	376	7	(111011101110111010110011000001111111001101000110001101010000100001000101)
16268	10	232	7	(1110101001010111100110110101100101101100100011101011011100000011010001001)
16268	10	536	7	(1011111100100101111110100110101000100101000110011111000001001100001011101)
16268	10	776	7	(11100101110110001000111110000100010011100010010101110100011110111000101)

Daha önce [18] çalışmasında, S-kutularının tasarımı için kullanılan en dik iniş prensibine dayalı arama algoritmasının zaman karmaşıklığı elde edilmiştir. Benzer argümanlarla, eşdeğerlik sınıflarının sayısı  $e \left( \approx \frac{2^n - 1}{n(2^p - 1)} \right)$  olmak üzere bir yinelemedeki komşuluk sayısı yaklaşık olarak  $\left(\frac{e}{2}\right)^2$  kabul edilirse, her bir komşulukta (Şekil-2'den verilen) maliyet değerinin hesaplanması için  $e^2$  tane çarpma işlemi gerektiğinden, bu çalışmada gerçekleştirilen en dik iniş prensibine dayalı arama algoritmasının zaman karmaşıklığı sabit bir yineleme sayısı ( $N$ ) için  $O(e^4)$  olarak bulunabilir. Ayrıca, yineleme çıktılarını kaydetmek için ihtiyaç duyulan bellek miktarının  $N \times e$  bit olduğu görülmektedir.

#### 4. Bulgular

Arama algoritması C programlama dilinde yazılmış ve yineleme sayısı  $N = 4000$  seçilerek, Intel Xeon E5-1650 3.5GHz işlemci ve 16 GB belleğe sahip bir bilgisayarda 12000 kere koşulmuştur. Yaklaşık iki gün süren bu arama sonucunda bükük-bağlaşım sınırından yüksek doğrusal olmama değerine sahip 35 Boole fonksiyon bulunmuş, ancak bu Boole fonksiyonlardan sadece 16 tanesinin (kısaltılmış) doğruluk tablolarının birbirinden farklı oldukları gözlenmiştir. Bunlardan 4 tanesi 16268, 6 tanesi 16269 ve ikişer tanesi 16270, 16271, 16272 doğrusal olmama değerlerine sahiptir. Bulunan 16 Boole fonksiyonun kısaltılmış doğruluk tabloları Çizelge-1'de sunulmakta ve karşılık gelen  $2^{15}$  uzunluğundaki doğruluk tabloları, eşitsizlik sistemini üreten algoritmanın kodu ile birlikte [22]'de verilmektedir.

Çizelge-1'de, elde edilen Boole fonksiyonların diğer kriptografik özellikleri cebirsel derece, cebirsel bağışıklık ve mutlak gösterge verilmektedir. Patterson ve Wiedemann tarafından keşfedilen [4] 16276 doğrusal olmama değerine sahip iki PW fonksiyonu bulunmaktadır ve her ikisinin de mutlak göstergesi 160 olmakla birlikte, birinin cebirsel derecesi ve bağışıklığı sırasıyla 9 ve 6 iken diğerinin cebirsel

derecesi ve bağışıklığı sırasıyla 8 ve 7'dir. PW fonksiyonlarından başka mutlak gösterge değeri 160 veya

daha iyi olan 15 değişkenli Boole fonksiyonlar literatürde bilinmemektedir. Ayrıca, tek sayıda değişkene sahip Boole fonksiyonların mutlak gösterge değerleri için bildiğimiz kadarıyla genel bir alt sınır da bilinmemektedir. Bununla birlikte, 9, 11 ve 21 değişkenli Boole fonksiyonlar için bilinen en iyi mutlak gösterge değerleri sırasıyla 24 [20], 56 [20] ve 1564'tür [6, 23]. Bunlardan ilk ikisi dengeli, sonuncusu ise dengeli olmayan Boole fonksiyonlara aittir.

PW fonksiyonları ile karşılaştırıldığında, Çizelge-1'de verilen Boole fonksiyonların cebirsel derece bakımından daha iyi oldukları, buna karşın mutlak gösterge ve doğrusal olmama açılarından ise PW fonksiyonlarının daha iyi oldukları görülmektedir. Bunun yanı sıra, Çizelge-1'de başarılı en yüksek doğrusal olmama değeri (16272), daha önce [7, 8] çalışmalarında elde edilen sonuçtan (16268) daha iyidir. Doğrusal olmama değeri tek sayı olan  $n$  değişkenli bir Boole fonksiyonun cebirsel derecesinin  $n$  olduğu bilinmektedir. Bu nedenle, Çizelge-1'deki 16271 ve 16269 doğrusal olmama değerlerine sahip Boole fonksiyonların cebirsel dereceleri 15'tir. Ayrıca, tek sayıda değişkene sahip bir Boole fonksiyonun optimum cebirsel bağışıklığa sahip olabilmesi için dengeli olması gerekmektedir [24]. Bu nedenle, PW fonksiyonları ve Çizelge-1'deki Boole fonksiyonları dengeli olmadıkları için optimum cebirsel bağışıklığa sahip değildirler. Bahsedilen fonksiyonlar dengeli olmadıklarından esneklik özelliği göstermezler ve bu fonksiyonların hiçbirinin korelasyon bağışıklığı özelliğine sahip olmadığı gözlenmiştir.

Çalışmamızın doğal bir uzantısı olarak (1057, 31)-aralıklı diziler ile temsil edilen 3-DSBF'ler ve 5-DSBF'ler için de arama yapılmış, fakat arama uzaylarının çok büyük olması (sırasıyla  $2^{217}$  ve  $2^{353}$ ) nedeniyle bükük-bağlaşım sınırını aşan sonuçlar elde edilememiştir. Daha verimli arama algoritmalarının tasarlanması veya arama uzaylarının küçültülmesine olanak

sağlayan yeni yöntemlerin geliştirilmesi, literatürde ulaşılamayan doğrusal olmama değerlerinin elde edilebilmesi için önemli açık problemlerdir.

## 5. Sonuç

Bu çalışmada, (1057,31)-aralıklı diziler ile temsil edilen DSBF'lerin oluşturduğu ve büyüklüğü  $2^{69.6}$  olan arama uzayında sezgisel arama yürütülerek, daha önce (151,217)- ve (217,151)-aralıklı diziler ile temsil edilen genelleştirilmiş DSBF'lerin oluşturdukları arama uzaylarında yapılan tam arama ve sezgisel arama yöntemlerinin ulaştığı [7, 8] doğrusal olmama değeri iyileştirilmiştir. Ayrıca, büyük-bağlaşım sınırını aşan Boole fonksiyonları literatürde esneklik, cebirsel bağışıklık, cebirsel derece ve mutlak gösterge gibi diğer kriptografik özellikler açısından güçlü Boole fonksiyonların tasarımı için kullanıldığından [10-12], elde ettiğimiz sonuçların bu yönde katkı sağlayabilecek nitelikte olduğu düşünülmektedir.

## Kaynakça

- [1] Ding, C., Xiao, G., Shan, W. The stability theory of stream ciphers, Springer, Berlin, 1991.
- [2] Matsui, M. Linear cryptanalysis method for DES cipher, Springer, Berlin, 1994, EUROCRYPT 1993, LNCS, vol. 765, pp. 386-397.
- [3] X.-D. Hou. On the norm and covering radius of the first order Reed-Muller codes, IEEE Trans. Inf. Theory, 1997, 43(3), pp. 1025-1027.
- [4] Patterson, N. J., Wiedemann, D. H. The covering radius of the (215, 16) Reed-Muller code is at least 16276, IEEE Trans. Inf. Theory, 1983, 29(3), pp. 354-356.
- [5] Kavut, S., Yücel, M. D. 9-variable Boolean functions with nonlinearity 242 in the generalized rotation symmetric class, Inf. Comput., 2010, 208(4), pp. 341-350.
- [6] Gangopadhyay, S., Keskar, P. H., Maitra, S. Patterson-Wiedemann construction revisited, Discret. Math., 2006, 306(14), pp. 1540-1556.
- [7] Kavut, S. New Patterson-Wiedemann type functions with 15 variables in the generalized rotation-symmetric class, Turk. J. Electr. Eng. Comp. Sci., 2017, 25(6), pp. 4901-4906.
- [8] Kavut, S. A Modified Patterson-Wiedemann Construction Having Nonlinearity Greater Than Bent Concatenation Bound, Rostock, Germany, 2022, WCC 2022.
- [9] Kavut, S., Maitra, S. Patterson-Wiedemann type functions on 21 variables with nonlinearity greater than bent concatenation bound, IEEE Trans. Inf. Theory, 2016, 62(4), pp. 2277-2282.
- [10] Zhang, W.G. High-meets-low: construction of strictly almost optimal resilient Boolean functions via fragmentary Walsh spectra, IEEE Trans. Inf. Theory, 2019, 65(9), pp. 5856-5864.
- [11] Sarkar, S., Maitra, S. Idempotents in the neighbourhood of Patterson-Wiedemann functions having Walsh spectra zeros, Des. Codes Cryptogr., 2008, 49, pp. 95-103.
- [12] Kavut, S. Improved cryptographic properties of Boolean functions obtained from the neighbourhood of Patterson-Wiedemann functions. Cryptogr. Commun., 2023, 15, pp. 433-442.
- [13] Stănică, P., Maitra, S. Rotation symmetric Boolean functions – Count and cryptographic properties, Discret. Appl. Math., 2008, 156(10), pp. 1567-1580.
- [14] Pieprzyk, J., Qu, C. X. Fast hashing and rotation-symmetric functions, J. Universal Comput. Sci. 1999, 5(1) pp. 20-31.
- [15] Filiol, E., Fontaine, C. Highly nonlinear balanced Boolean functions with a good correlation immunity, Springer, Berlin, 1998, EUROCRYPT 1998, LNCS, vol. 1403, pp. 475-488.
- [16] Fontaine, C. On some cosets of the first-order Reed-Muller code with high minimum weight, IEEE Trans. Inf. Theory, 1999, 45(4), pp. 1237-1243.
- [17] Courtois, N. T., Meier, W. Algebraic attacks on stream ciphers with linear feedback, Springer, Berlin, 2003, EUROCRYPT 2003, LNCS, vol. 2656, pp. 345-359.
- [18] Kavut, S. Bazı alt uzaylarda kriptografik açıdan eniyilenmiş büyük S-kutuları, EMO Bilimsel Dergi, 2022, 12(1), pp. 43-51.
- [19] Kavut, S., Yücel, M. D. Güçlü kriptografik özelliklere sahip Boole işlevleri tasarımında yeni bir algoritma, Ankara, Türkiye, 2005, 1. Ulusal Kriptoloji Sempozyumu, Bildiriler Kitabı, pp. 95-105.
- [20] Kavut, S., Maitra, S., Yücel, M. D. Search for Boolean functions with excellent profiles in the rotation symmetric class, IEEE Trans. Inf. Theory, 2007, 53(5), pp. 1743-1751.
- [21] Clark, J. A., Jacob, J. L. Two-stage optimisation in the design of Boolean functions, Springer, Berlin, 2000, ACISP 2000, LNCS, vol. 1841, pp. 242-254.
- [22] Kavut, S. Truth tables and system of inequalities for (1057, 31)-interleaved sequences, GitHub, URL: [https://github.com/Selcuk-kripto/1057\\_31](https://github.com/Selcuk-kripto/1057_31) (Erişim tarihi: 20.11.2022).
- [23] Kavut, S. Correction to the paper: Patterson-Wiedemann construction revisited, Discret. Appl. Math., 2016, 202, pp. 185-187.
- [24] Dalai, D. K., Gupta, K. C., Maitra, S. Results on algebraic immunity for cryptographically significant Boolean functions, Springer, Berlin, 2004, INDOCRYPT 2004, LNCS, vol. 3348, pp. 92-106.



# Türkçe Konuşmadan Metne Dönüştürme için Ön Eğitimli Modellerin Performans Karşılaştırması: Whisper-Small ve Wav2Vec2-XLS-R-300M

## Performance Comparison of Pre-trained Models for Speech-to-Text in Turkish: Whisper-Small and Wav2Vec2-XLS-R-300M

Öykü Berfin MERCAN

AdresGezgini A.Ş.  
Ar-Ge Departmanı  
İzmir, Türkiye

oykumercan@adresgezgini.com  
ORCID: 0000-0001-7727-0197

Sercan ÇEPNİ

AdresGezgini A.Ş.  
Ar-Ge Departmanı  
İzmir, Türkiye

sercancepni@adresgezgini.com  
ORCID: 0000-0002-3405-6059

Davut Emre TAŞAR

Dokuz Eylül Üniversitesi  
Yönetim Bilişim Sistemleri  
İzmir, Türkiye

davutemre.tasar@ogr.deu.edu.tr  
ORCID:0000-0002-7788-0478

Şükrü OZAN

AdresGezgini A.Ş.  
Ar-Ge Departmanı  
İzmir, Türkiye

sukruozan@adresgezgini.com  
ORCID: 0000-0002-3227-348X

### Öz

Bu çalışmada konuşmadan metne çeviri için önerilmiş ve çok sayıda dille ön eğitilmiş iki model olan Whisper-Small ve Wav2Vec2-XLS-R-300M modellerinin Türkçe dilinde konuşmadan metne çevirme başarıları incelenmiştir. Çalışmada açık kaynaklı bir veri kümesi olan Türkçe dilinde hazırlanmış Mozilla Common Voice 11.0 versiyonu kullanılmıştır. Az sayıda veri içeren bu veri kümesi ile çok dilli modeller olan Whisper-Small ve Wav2Vec2-XLS-R-300M ince ayar yapılmıştır. İki modelin konuşmadan metne çeviri başarıları değerlendirilmiş ve Wav2Vec2-XLS-R-300M modelinin 0,28 WER değeri Whisper-Small modelinin 0,16 WER değeri gösterdiği gözlemlenmiştir. Ek olarak modellerin başarıları eğitim ve doğrulama veri kümesinde bulunmayan çağrı merkezi kayıtlarıyla hazırlanmış sınama verisiyle incelenmiştir.

**Anahtar sözcükler:** Konuşmadan Metne Çeviri, Whisper, Wav2Vec2-XLS-R, WER, Common Voice, İnce Ayar

### Abstract

In this study, the performances of the Whisper-Small and Wav2Vec2-XLS-R-300M models which are two pre-trained multilingual models for speech to text were examined for the Turkish language. Mozilla Common Voice version 11.0 which is prepared in Turkish language and is an open-source data set, was used in the study. The multilingual models, Whisper-Small and Wav2Vec2-XLS-R-300M were fine-tuned with this data set which contains a small amount of data. The speech to text performance of the two models was compared. WER values are calculated as 0.28 and 0.16 for the Wav2Vec2-XLS-R-300M and the Whisper-Small models respectively. In addition, the performances of the models were examined with the test data prepared with call center records that were not included in the training and validation dataset.

**Keywords:** Speech to Text, Whisper, Wav2Vec2-XLS-R, WER, Common Voice, Fine-tuning

### 1. Giriş

Konuşmadan Metne Dönüştürme (SM), Doğal Dil İşlemede (DDİ) aktif bir araştırma alanıdır. İnsan-makine etkileşimini

amaçlayan sesten metne dönüştürme alanına ise yoğun ilgi gösterilmektedir. Özellikle salgın döneminde evden çalışma biçiminin yaygınlaşmasıyla mesai saatleri içerisinde iş ve eğitim başta olmak üzere iletişim kurmak amacıyla mesajlaşma ve çevrimiçi olarak sesli ya da video görüşmeleri gerçekleştirilmektedir. Gerçekleşen görüşmeler esnasında konuşulanlar not edilmediğinde ya da kayıt altına alınmadığında tekrar erişmek mümkün olmamaktadır. Görüşme kayıtlarının tekrar tekrar incelenmesi toplantı süresi kadar bir zaman kaybı yaratarak verimsiz bir yöntem olmakla birlikte görüşme sırasında katılımcılar tarafından not alınsa dahi bu durum kişinin not alma becerisi ve dikkati ile kısıtlıdır. Çevrimiçi toplantılar haricinde halihazırda müşteri ilişkilerinde aktif olarak çağrı merkezi kullanan firmalar için hizmet ve kalite düzeyi takibi amacıyla görüşmelerin kalite yöneticisi tarafında kontrolünü sağlamak için konuşmadan metne çeviri önem arz etmektedir.

Bu çalışma, Türkçe Konuşmadan Metne Dönüştürme alanında son teknoloji modellerin performanslarının sistematik bir karşılaştırılmasını sunarak literatüre önemli katkılar sağlamaktadır. Özellikle, yayınlarda yer alan güncel, derin öğrenme tabanlı iki farklı modelin Sözcük Yanlış Oranı (Word Error Rate, WER) başarımını Türkçe konuşma tanıma bağlamında değerlendiren bu çalışma, model seçimi ve uygulama süreçlerinde araştırmacılar ve uygulayıcılar için önemli bir rehber niteliği taşımaktadır.

## 2. Benzer Çalışmalar

Literatürde, farklı MS model yaklaşımlarını öneren çalışmalar yapılmıştır. Örneğin, araştırmacılar telefon görüşmelerinin metin şeklinde depolanmasını sağlamak, farklı dillerdeki konuşmaları yazıya dökmek ve çevirmek, işitme engelli kişiler için ses içeriğine erişilebilirliği artırmak, konuşmayı gerçek zamanlı olarak yazıya dökmek, analiz etmek ve konuşmaya dayalı sanal asistanlar geliştirmek için STT modellerini kullanmışlardır [1, 2, 3]. Aktif olarak telefon görüşmelerinin gerçekleştiği çağrı merkezlerinde, görüşmelerdeki sahtekarlık tespiti üzerine odaklanmıştır. Çağrı merkezi kayıtları ilk olarak konuşma tanıma sistemine verilip konuşmalar metinlere dönüştürülmüş ardından metin sınıflandırma gerçekleştirilerek sahtekarlık tespiti gerçekleştiren sistem geliştirilmiştir [1]. Konuşmayı Hint işaret diline çeviren otomatik bir sistem geliştirerek işitme engelliler ile işiten insanlar arasında iletişim kopukluğuna çözüm getirilmesi amaçlanmıştır [2]. Bir diğer çalışmada, ALS kaynaklı dizartrisi olan kişilerin, yaygın olarak kullanılan üç akıllı telefon tabanlı asistanlar tarafından ne ölçüde anlaşıldığı ve tutarlı cevaplar alabildiği araştırılmış ve özellikle İtalyanca dizartrik konuşmanın tanınması üzerine odaklanılmıştır [3]. Bu çalışmalar, STT modellerinin çeşitli alanlardaki potansiyelini göstermiş ve bu alanda sürekli araştırma ve geliştirmenin önemini vurgulamıştır [4]. Bu pratik uygulamalara ek olarak, STT modellerinin doğruluğunu ve hızını geliştirmeye yönelik çalışmalar da yapılmıştır. Araştırmacılar, evrişimli sinir ağları (CNN'ler), tekrarlayan sinir ağları (RNN'ler) ve dönüştürücüler dahil olmak üzere çeşitli derin öğrenme mimarilerini araştırmış ve STT modellerinin performansını artırmak için transfer öğrenme ve çok görevli öğrenme gibi yeni eğitim yaklaşımları geliştirmişlerdir [5].

STT alanındaki bir başka aktif çalışma alanı da gürültüyü, aksanları ve konuşmadaki diğer varyasyonları işleyebilen modellerin geliştirilmesidir. Bu modeller, STT sistemlerinin başarısını artırmayı ve konuşma kayıtlarının düşük kalitede olabileceği pratikteki uygulamalara uygun hale getirmek üzere çalışmaları içermektedir [6]. STT teknolojisinin geliştirilmesinde dikkate alınması gereken bir diğer önemli husus gizlilik ve güvenlidir. Konuşma verileri son derece kişisel ve hassas olduğundan, STT sistemlerinin gizlilik ve güvenlik göz önünde bulundurularak tasarlandığından emin olmak önemlidir. Araştırmacılar, bu endişeleri gidermek için gizliliği koruyan makine öğrenimi ve birleşik öğrenme gibi çeşitli teknikleri araştırmaktadır [7].

Dikkat tabanlı kodlayıcı ve kot çözücü modellemesi konuşmadan metne görevi için önemli gelişmeler sağlamıştır. Dikkat tabanlı tekrarlayan sinir ağlarını konuşma tanıma için uygulanabilir kılan eklentiler sunulmuştur [8]. Üretilen her fonem için, bir dikkat mekanizması, giriş dizisindeki potansiyel olarak tüm zaman adımlarında eğitilmiş bir öznitelik çıkarma mekanizması tarafından üretilen sinyalleri seçer veya ağırlıklandırır. Ağırlıklı özellik vektörü daha sonra çıktı dizisinin bir sonraki öğesinin oluşturulmasını koşullandırmaya yardımcı olur. Bir başka çalışmada ağ bileşenlerinin ön eğitiminin etkisini gösterilmiş ve kodlayıcı ile kod çözücü arasına ek bir katman ekleyerek verimli bir şekilde nasıl birleştirileceği araştırılmıştır. Ekstra katmanın daha iyi ortak öğrenme sağladığı ve performans artışı gösterdiği gözlemlenmiştir [9]. [10]'de otomatik konuşma tanıma ve konuşma çevirisi görevleri için metin verilerinden yararlanmak üzere genel bir çoklu görev öğrenme çerçevesi sunulmuştur. Otomatik konuşma tanıma bir gürültü giderici otomatik kodlayıcı göreviyle birlikte eğitilirken, makine çevirisi görevi konuşma çevirisi göreviyle paralel verilerle ortaklaşa eğitilmiştir.

Yayınlarda yer alan Türkçe dili özelinde gerçekleştirilen konuşmadan metne çeviri çalışmaları araştırılmış, son yıllarda kaydedilen ilerlemeler ve performans sonuçları incelenmiştir. Çalışmalarda Türkçenin biçim birime dayalı biçim bilimsel yapısı dikkate alınmıştır. Destek Vektör Makineleri (SVM) tabanlı Türkçe konuşmayı metne dönüştürme sistemi geliştirilmiştir [11]. Türkçe konuşmanın özelliklerini çıkarmak için Mel Frequency Cepstral Katsayıları (MFCC) uygulanmış ve biçim birimleri sınıflandırmak için SVM tabanlı sınıflandırıcı kullanılmıştır. Başka bir çalışmada geleneksel Gauss karışım model saklı Markov modelinden daha iyi sonuçlar veren mobil kayıtlı bir Türkçe veri kümesi üzerinde Türkçe dilinde konuşmadan metne çeviri için derin sinir ağı modeli önerilmiştir [12]. Yine benzer bir çalışmada derin öğrenme tabanlı konuşmadan metne çevirinin Gauss karışım model-saklı Markov modeline göre başarılı sonuçlar vermiş olup çalışma [12]'de sunulan çalışma ile benzerlik göstermektedir [13]. [12]'den farklı olarak [13]'de farklı eğitim verisi ve ileri beslemeli sinir ağı yerine derin inanç ağları kullanılmıştır. [14]'de LSTM ve GRU tabanlı sistemler ile geleneksel yöntemler ve derin sinir ağı tabanlı sistemlerin konuşma tanıma performansı karşılaştırılmıştır. Konuşma tanıma çalışmalarının artmasıyla birlikte etiketli veri sayısının az olması temel problem haline gelmiştir. Bu sorunun çözümü için daha az etiketli veri gerektiren denetimsiz öğrenme

yaklaşımları önerilmiştir. [15]'de YouTube'dan seçilen 6,5 bin saatlik etiketsiz ses verisi ile ön eğitilmiş kendi kendine denetimli öğrenmenin avantajlarından yararlanarak Türkçe dilinde konuşma tanıma yaklaşımı geliştirmişlerdir.

Sınırlı kaynaklarla son teknoloji bir konuşma çevirisi sistemi oluşturmak için [16]'de önceden eğitilmiş konuşma tanıma ve metin çevirisi modelleri kullanılması önerilmiştir. Teknolojinin ilerlemesi ve konuşma tabanlı uygulamalara yönelik artan talep, STT'yi konuşma tanıma, transkripsiyon ve analiz için yaygın olarak kullanılan bir araç haline getirmiştir [17]. Wav2Vec2.0, HuBERT gibi kendi kendini denetleyen ses kodlayıcılar, yüksek kaliteli ses temsillerini öğrenir [15, 18]. Bu modeller denetimsiz ön eğitim yapısı nedeniyle, ses temsillerini kullanılabilir çıktılara dönüştürmek için uygun bir kod çözücü yapısına ihtiyaç duyarlar. Sesten metne çeviri görevi için modellerin ince ayarı gerekmektedir. Wav2vec2.0 [17] gibi kendi kendini denetleyen derin öğrenme modelleri, STT alanında dikkate değer sonuçlar göstermiştir. 2020'de Facebook AI Research tarafından geliştirilen wav2vec2.0 modeli, çok büyük bir konuşma verisi külliyatı üzerinde eğitilmiş transformer tabanlı bir model olarak sunulmuştur. Öte yandan, OpenAI tarafından 2021'de [19] tanıtılan Whisper, derin sinir ağı tabanlı bir mimari kullanan ve daha farklı bir yaklaşım kullanılarak eğitilmiş, gerçek dünya uygulamaları için uygun hale getirilmiş daha yeni bir STT modeli olarak sunulmuştur. [18]'de nöral tabanlı konuşmadan metne çeviri mimarilerinden Wav2Vec2.0 ve Whisper'ın ince ayarlı bir versiyonları üzerinde performans karşılaştırması yapılmış ve bu modeller İngilizce, İspanyolca, Almanca, Fransızca, İtalyanca, Portekizce ve Lehçe dahil olmak üzere yedi Hint-Avrupa dilinden alınan veriler üzerinde test edilmiştir.

Karşılaştırmalı bir analiz yapabilmek için Türkçe dilinde otomatik konuşma tanıma görevinde yapılmış olan ince ayar çalışmaları araştırılmıştır. Bir yüksek lisans tezi olarak yapılan çalışmada [20], Mozilla Common Voice veri setinin daha eski bir versiyonu olan 6.1 versiyonu ile bir wav2vec2.0 modeline ince ayar çalışması yapılmış ve 0,23'lük bir kelime hata oranı elde edilmiştir. Bir başka çalışmada [21] ise yine wav2vec2-XLSR-53 modelinin Common Voice, Youtube ve Ses kayıtları kullanılarak ince ayar yapıldığı görülmüştür. Whisper ile ilgili ise, Türkçe dili için herhangi bir ince ayar çalışması ile karşılaşılmamıştır. Tespit edilmiş olan bu çalışmalara ait sonuçlar ise aşağıdaki Çizelge-1'de gösterilmiştir.

**Çizelge-1: Benzer çalışmalar**

Model	Veri Seti	WER
Wav2Vec2.0-XLSR [20]	Common Voice 6.1	0,23
Wav2Vec2.0-XLSR-53 [21]	Common Voice Youtube Ses Kayıtları	0,32

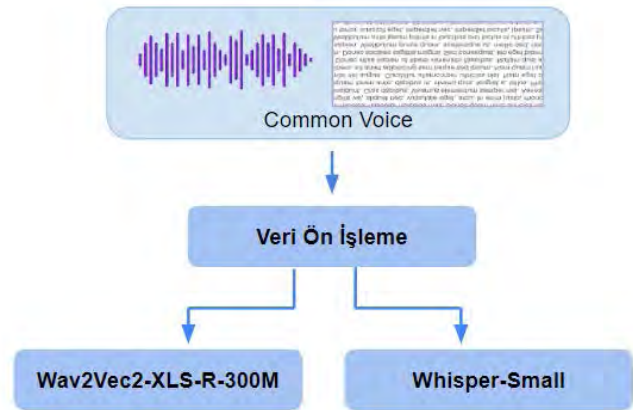
Bu çalışmada ise son teknoloji, derin öğrenme tabanlı iki model olan wav2vec2.0 [17] ve Whisper'ın [19] Kelime Hata Oranı (Word Error Rate (WER)) performansları Türkçe konuşma tanıma bağlamında karşılaştırılmıştır. Türkçe konuşma tanımada wav2vec2.0 ve Whisper'ın WER performansını karşılaştırmak için Mozilla Common Voice veri

kümesi [22] kullanılmıştır. Mozilla Common Voice veri kümesi, Türkçe dahil birçok dilde ses kaydı içeren, STT modelleri için yaygın olarak kullanılan bir veri kümesidir. Deneysel sonuçlar, bu makalenin aşağıdaki bölümlerinde ayrıntılı olarak sunulmaktadır.

Türkçe dahil olmak üzere birçok dille ön eğitilmiş olan iki modelin karşılaştırmasının sunulduğu bu çalışma, Türkçe STT alanındaki teknoloji ve metodolojinin mevcut durumunun kapsamlı bir değerlendirmesini sağlamaktadır. Önerilen çalışmanın sonuçları, özellikle Türkçe konuşma tanıma bağlamında, DDI ve STT alanındaki araştırmacılar ve uygulayıcılar için doğru model seçme konusunda fikir verecek bir çıktı içermektedir.

### 3. Tasarım ve Yöntem

Bu çalışmada, Türkçe dilinde konuşmadan metne çevirme amaçlanmış olup önerilen iki farklı modelin başarıları değerlendirilmiştir. Modellerin ince ayarı için açık kaynaklı veri kümesi Common Voice veri kümesi kullanılmıştır. Bu bölümde ise çalışmada kullanılan veri kümesi, veri kümesine uygulanan ön işlemler ve konuşmadan metne çeviri modelleri için ince ayar gerçekleştirilecek modeller açıklanacaktır. Deneysel akış diyagramı Şekil-1'de verilmiştir.



**Şekil-1: Akış diyagramı**

#### 3.1 Veri Kümesi

Konuşmadan metne çeviri modellerinin düşük kelime hata oranlarını ile başarılı sonuçlar üretebilmesi için çok sayıda konuşma-metin çiftinden oluşan veri kümesi ile eğitilmesi gerekmektedir. Mozilla Common Voice, ses teknoloji üzerine çalışan geliştiricileri destekleyici çok sayıda ses kaydı biriktirerek derin öğrenme mimarilerinin eğitimi için önemli ses kaynağı olma niteliği taşımaktadır. İlk veri kümesi 2017 yılında 500 saatlik İngilizce cümlelerin seslendirilmesi ile oluşturulan Common Voice veri tabanı 2022 yılı itibariyle 100 farklı dilde ses kaydı içermektedir. Önerilen çalışmada Türkçe dilinde sestem metne çeviri gerçekleştirmek amaçlanmış olup model eğitimleri Türkçe dilinde hazırlanmış Common Voice Corpus 11.0 veri kümesi ile gerçekleştirilmiştir. Veri kümesi, 1,82 GB boyutunda ve 1328 kişi tarafında seslendirilmiş 88 saatlik kayıt içermekte olup ses kayıtları MP3 (Motion Pictures Experts Group Layer 3) formatında kaydedilmiştir. Ses

kayıtları ve bu ses kayıtlarıyla eşleşen metinler konuşmadan metne çeviri modellerine girdi verisi oluşturacak şekilde ön işlemlerden geçirilmiştir.

### 3.2 Veri Ön İşleme

Model eğitimi ve doğrulama aşamasında kullanılacak veri kümesi ilk olarak bir dizi ön işlemde geçirilmiştir. Common Voice veri kümesinde ses kayıtlarına denk gelen metinler, büyük küçük harf kullanımı ve noktalama işaretleri içermektedir. Cümle içinde yer alan bu noktalama işaretleri konuşma içerisinde herhangi bir ses birimine karşılık gelmemektedir. Bu nedenle ses parçasını noktalama işaretine göre sınıflandırmak mümkün olmamaktadır. Bu durum göz önüne alınarak kelime anlamına katkısı olmayan ve bir sesle temsil edilmeyen noktalama işaretleri veri kümesinden kaldırılmıştır. Bir diğer düzenleme ise büyük harfler üzerinde gerçekleştirilmiştir. Veri kümesinde bulunan her harf için bir belirteç atanmaktadır. Aynı harfin büyük ve küçük halinin veri kümesinde bulunması o harfe ait iki belirteç tanımlanmasına neden olmaktadır. Veri kümesindeki tüm harfler küçük harfe dönüştürülerek normalize edilmiştir. Son olarak karakter çeşitliliğini azaltmak amacıyla kullanımı yaygın olmayan şapkalı harfler (â, î, ô, û) dönüştürülerek veri kümesi sadeleştirilmiştir. Whisper model eğitimi için sadece şapkalı harflerde düzenleme yapılmıştır. Bunun nedeni Whisper sestem metne dönüştürme işleminin yanında imla kurallarına uygun bir şekilde sonuç vermektedir.

### 3.3 Yöntem

Çalışmada, konuşmadan metne çeviri için Facebook AI ve OpenAI tarafından geliştirilmiş birçok farklı dilde ön eğitilmiş iki model olan Wav2Vec2-XLS-R-300M ve Whisper-Small modelleri önerilmiştir. Bu başlık altında önerilen bu modeller detaylı olarak açıklanacaktır.

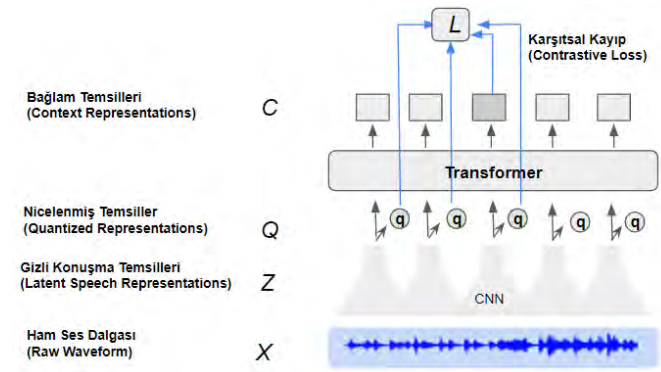
#### 3.3.1 Wav2Vec2-XLS-R-300M

2020 yılında Facebook AI denetimsiz ön eğitim tekniği ile geliştirdikleri STT modeli Wav2Vec2.0'i önermişlerdir [9]. Önerilen bu yöntem ile model, etiketli veriye ihtiyaç duymadan çok sayıda ham ses verisiyle ön eğitilir. Wav2Vec2.0 ardından Conneau vd. 53 farklı dilde ön eğitilmiş çok dilli bir model olan Wav2Vec2-XLS-R-53 modelini önermişlerdir [23]. XLSR modeli 8 dil içeren 50,7 bin saatlik çok dilli LibriSpeech, 36 dil içeren 3,6 bin saatlik Common Voice ve 17 dil içeren 1,7 bin saat uzunluğundaki Babel veri kümesiyle ön eğitilmiştir. 53 dilde 56.000 saat etiketsiz veriyle ön eğitilmiş bu model ile birlikte çok sayıda dilde konuşma tanıma olanağı sunulmuştur. 2021 yılında ise 128 farklı dilde 436.000 saat eğitim verisi kullanılarak XLSR-53 modeli üzerinde önemli iyileştirmeler gerçekleştirilmiş ve Wav2Vec2-XLS-R modeli yayınlanmıştır [24]. XLSR-53 modelini temel alan bu model ile daha fazla dil içeren daha büyük miktarda etiketlenmemiş eğitim verisi (Multilingual LibriSpeech, Common Voice, Voxlingua107, Babel, VoxPopuli) kullanarak gelişmiş bir model oluşturmayı amaçlamışlardır. Çizelge-2'de eğitilebilir parametre sayısına göre Wav2Vec2-XLSR modelleri verilmiştir. Çizelgede M: Milyon adet parametre sayısını, B: Bin saat eğitim verisi süresini ifade etmektedir. Parametrelere dayalı eğitim süresi göz önüne

alındığında Wav2Vec2-XLS-R-2B, 2 milyar parametre ile en büyük ve yavaş modelken 300M en küçük ve hızlı modeldir. Donanım ve zaman düşünüldüğünde bu çalışmada 300M modelinin kullanılmasına karar verilmiştir.

Çizelge-2: Wav2Vec2-XLS-R modellerinin özellikleri tablosu

Model	Parametre	Süre (saat)	Çok Dilli
Wav2Vec2-XLS-R-300M	300M	436B	✓
Wav2Vec2-XLS-R-1B	1 milyar	436B	✓
Wav2Vec2-XLS-R-2B	2 milyar	436B	✓



Şekil-2: Wav2Vec2.0 mimarisi [18]

Wav2vec2.0 modeli, konuşmayı çok katmanlı bir evrişimli sinir ağı aracılığıyla kodlar ve ardından maskelenmiş dil modellemeye benzer şekilde ortaya çıkan gizli konuşma temsillerinin yayılma alanlarını maskeler (Şekil-2). Gizli temsiller, bağlamsallaştırılmış temsiller oluşturmak için bir dönüştürücü ağına beslenir ve model çeldiricilerden ayırt edeceği karşılaştırmalı bir görevle eğitilir. Karşılaştırmalı görevdeki gizli temsilleri temsil etmek için ayrık konuşma birimlerini öğrenir. Etiketlenmemiş konuşmaya ilişkin ön eğitimden sonra model, konuşma tanıma görevleri için kullanılmak üzere etiketli veriler ile ince ayar yapılır.

#### 3.3.2 Whisper-Small

Whisper, OpenAI tarafından geliştirilen ve konuşmaları, konuşulduğu dilde metne dönüştüren veya eğitildiği herhangi başka bir dile çevirebilen bir otomatik konuşma tanıma (ASR) sistemidir [19]. Sistem, aksanlara, arka plan gürültüsüne ve teknik dile karşı sağlamlık sağlayan web'den toplanan 680.000 saatlik çok dilli ve çok görevli denetimli veriden oluşan geniş ve çeşitli bir veri kümesi üzerinde eğitilmiştir [19]. Çizelge-3'de Whisper modelinin farklı büyüklüklerdeki özellikleri gösterilmektedir. M: Milyon adet parametre sayısını ifade etmektedir [25].

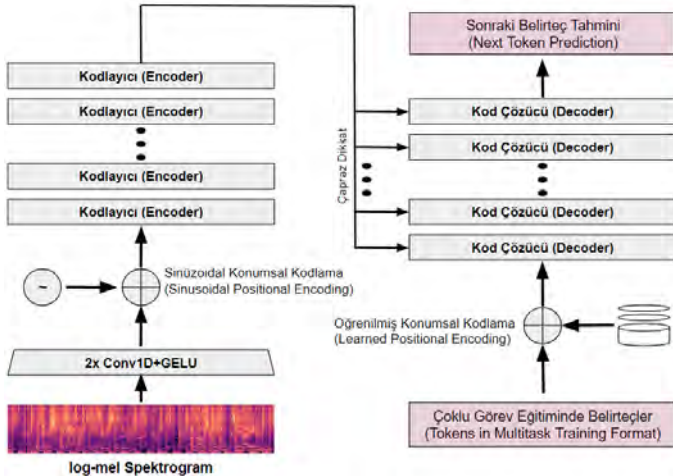
Çizelge-3: Whisper modellerinin özellikleri tablosu [25]

Boyut	Parametre	İngilizce	Çok Dilli
Tiny	39 M	✓	✓
Base	74M	✓	✓
<b>Small</b>	<b>244M</b>	✓	✓
Medium	769M	✓	✓
Large	1550M		✓

Whisper-Small modeli, 244 milyon parametre ile geliştirilen dokuz Whisper modellinden biridir. Orta büyüklükte olması ve kapasite olarak Wav2Vec2.0'da kullandığımız modele benzemesi nedeniyle bu çalışma için seçilmiştir.

Whisper mimarisi, bir kodlayıcı-kod çözücü Transformatörü olarak uygulanan uçtan uca basit bir yaklaşımdır. Giriş sesi 30 saniyelik parçalara bölünür ve bir log-Mel spektrogramına dönüştürülür. Spektrogram daha sonra bir kodlayıcıya iletilir ve bir kod çözücü karşılık gelen metin başlığını tahmin etmesi için eğitilir. Kod çözücü ayrıca, modeli dil tanımlama, tümce düzeyinde zaman damgaları, çok dilli konuşma transkripsiyonu ve konuşma çevirisi gibi görevleri gerçekleştirmeye yönlendiren özel belirteçleri karıştırır [19].

Şekil-3'de gösterildiği gibi, Whisper mimarisi bir kodlayıcı ve bir kod çözücünden oluşur. Kodlayıcı, sesin log-Mel spektrogramını girdi olarak alır ve daha sonra kod çözücüyü iletilen gizli bir temsili çıkarır. Kod çözücü, nihai transkripsiyonu veya çeviri çıktısını tahmin eder.



Şekil-3: Whisper mimarisi [19]

### 3.3.3 Başarı Metrikleri

Sesten metne çeviri modellerinin başarılarını değerlendirmek için başarı metrikleri kullanılmıştır. Model performanslarının analizi için popüler teknikler olan kelime hata oranı ve karakter hata oranı bu bölümde açıklanmıştır.

### 3.3.3.1 Kelime Hata Oranı

Kelime hata oranı (Word Error Rate (WER)) konuşma tanıma sisteminin performansını analiz etmek için kullanılan popüler bir tekniktir [26]. WER hesaplaması Levenshtein mesafesini temel alır. Levenshtein mesafesi iki kelime arasındaki farklılığı harf düzeyinde analiz eder. WER ise çeviri performansını kelime düzeyinde analiz etmekte olup sesten metne çeviri başarısı, sistem çıktısı ile referans kelimelerin eşleşmesine dayanmaktadır. WER değeri insan tarafından yazıya çevrilmiş referans kelime dizisi dikkate alındığında sesten metne çeviri dizisinde gözlemlenen silinen kelime, eklenen kelime ve değişen kelime sayısına göre hesaplanır. Eklenen kelime sayısı I, silinen kelime sayısı D, değişen kelime sayısı S olmak üzere toplam hatanın referans kelime dizideki kelime sayısı N'e oranıyla WER değeri hesaplanır. (Denklem 1)

$$WER = \frac{I+D+S}{N} \quad (1)$$

### 3.3.3.2 Karakter Hata Oranı

Karakter hata oranı (Character Error Rate (CER)), WER ile benzerlik göstermekte olup farklı olarak doğruluk karakter düzeyinde analiz edilmektedir [27]. Bu metrik ile yanlış tahmin edilen karakterlerin oranı elde edilmektedir. CER değeri ne kadar düşükse, konuşma tanıma sistemi o kadar başarılı demekle beraber '0' en iyi puandır. Eklenen karakter sayısı I, silinen karakter sayısı D, değişen karakter sayısı S, referans metindeki karakter sayısı N olmak üzere CER değeri Denklem 2'deki gibi hesaplanır.

$$CER = \frac{I+D+S}{N} \quad (2)$$

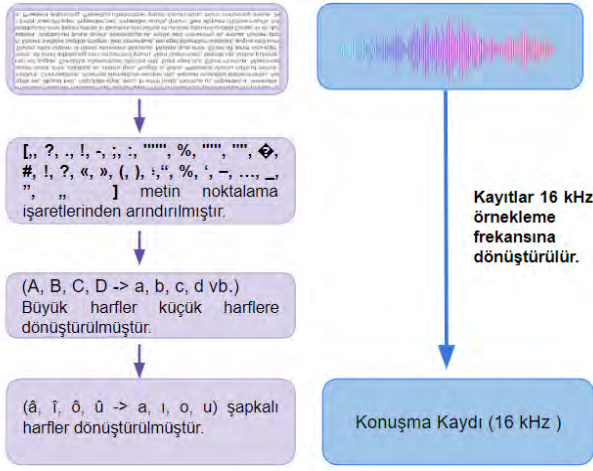
## 4. Deneysel Sonuçlar

Bölüm 2'de model eğitimleri ve doğrulamasında kullanılacak veri kümesi, veri kümesinde uygulanan ön işlemler, sesten metne çeviri için önerilen derin öğrenme mimarileri ve bu mimarilerin performans değerlendirmesinde kullanılan başarı metrikleri açıklanmıştır. Bu bölümde ise model eğitimleri sonucunda elde edilen veriler sunulacaktır.

Çalışmada Türkçe dilinde konuşmadan metne çeviri gerçekleştirmek amaçlanmış olup bu doğrultuda literatürde yer alan iki farklı model eğitilerek sonuçları değerlendirilmiştir. Modellerin eğitimi ve doğrulamasında Common Voice Corpus 11.0 kullanılmıştır. Veri kümesindeki 36125 adet veriyle model eğitimi, 10143 adet veriyle modelin doğrulaması gerçekleştirilmiştir. Her iki model için eğitimden önce veri kümesi ön işlemlerden geçirilmiş veri üzerinde bir takım düzenlemeler gerçekleştirilmiştir (Şekil-4). Kullandığımız veri kümesindeki metinler herhangi bir ses birimine karşılık gelmeyen noktalama işaretleri içermektedir. Wav2Vec2-XLS-R-300M modeline uygun olarak veri kümesindeki metinler noktalama işaretlerinden arındırılmıştır. Ek olarak Wav2Vec2-XLS-R-300M modeli veri kümesindeki her harf için bir belirtecin atandığı sözlük oluşturmayı gerektirmektedir. Bu nedenle veri kümesinde yer alan her harf tespit edilip her birine bir tam sayı atanmıştır. Harflere atanan belirteç haricinde kelimeleri ayırmak için '|', modelin karşılaşılabileceği veri kümesinde yer almayan karakterler için 'UNK' ve kelime içerisinde tekrar eden harfleri tespit etmek için 'PAD'



belirteçleri de tanımlanarak sözlüğü dahil edilmiştir. Ayrıca Common Voice 11.0 veri kümesinde MP3 formatındaki depolanmış kayıtların örnekleme frekansı model girişine uygun olarak 16 kHz'e dönüştürülmüştür.



Şekil-4: Veri ön işleme akış diyagramı

Whisper-Small model eğitim parametreleri, adım sayısı (step) 5000, öğrenme oranı (learning rate) 1e-05, batch boyutu 32 ve Adam optimizasyon algoritması olarak belirlenmiştir. Wav2Vec2-XLS-R-300M model eğitim parametreleri ise eğitim tur sayısı (epoch) 30, öğrenme oranı (learning rate) 3e-04, batch boyutu 64 ve Adam optimizasyon algoritması olarak belirlenmiştir. Her iki model için farklı parametrelerle çok sayıda deney gerçekleştirilmiş bu deneyler sonucunda en yüksek başarıların elde edildiği parametreler model parametreleri olarak belirlenmiştir.

Şekil-5 ve Şekil-6'da sırasıyla ince ayarlanmış Wav2Vec2-XLS-R-300M [28] ve Whisper-Small [29] modellerinin adım sayısı boyunca gözlemlenen kayıp grafikleri verilmiştir. Şekil-6'da Whisper-Small modelinin 2.000'inci adım sayısında en düşük doğrulama kaybına ulaştığı sonrasında aşırı öğrenme (overfitting) gösterdiği gözlemlenmiş ve grafikte siyah nokta ile belirtilmiştir. Şekil-5'de ise Wav2Vec2-XLS-R-300 modeli için böyle bir durumun söz konusu olmadığı kayıp grafiklerinde düşüşün devam ettiği gözlemlenmektedir. Bu nedenle Wav2Vec2-XLS-R-300M modelinin eğitimi devam ettirilmiştir. Wav2Vec2-XLS-R-300M ve Whisper-Small modelleri için sırasıyla Şekil-7 (a) ve Şekil-7 (b)'de eğitim ve doğrulama kayıp grafikleri tek bir grafikte birleştirilerek modellerin eğitim süreci daha anlaşılır kılınmıştır. Modellerin STT başarısını değerlendirmek amacıyla eğitim süresince en düşük WER değerinin gözlemlendiği adım sayısındaki model kaydedilmiştir. Çizelge-4'te modellerin en düşük WER değerinin elde edildiği adım sayısındaki eğitim, doğrulama kayıpları ve STT başarısını değerlendirmek için kullandığımız iki performans metriği olan WER ve CER değeri verilmiştir. Elde edilen sonuçlar incelendiğinde Whisper-Small modelinin başarılı sonuçlar verdiği gözlemlenmiştir. Ayrıca eğitim süresi göz önüne alındığında Whisper-Small model eğitimi daha az adım sayısında başarılı sonuçlar vermiş olup Wav2Vec2-XLS-R-300M modeline kıyasla eğitim daha kısa sürede sonuçlanmıştır.

Çizelge-4: Wav2Vec2-XLS-R-300M ve Whisper-Small modelleri sonuç tablosu

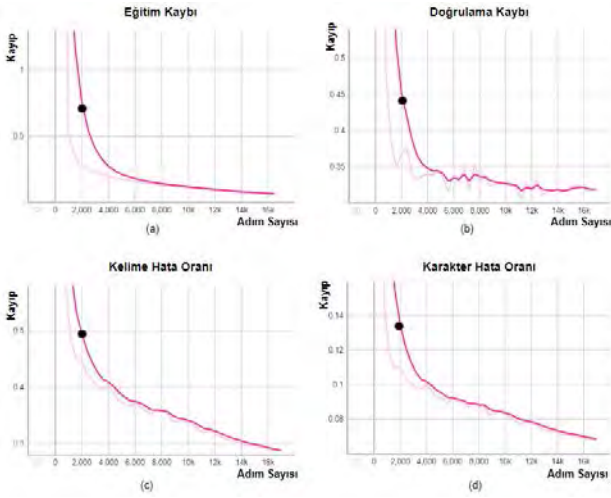
Model	Eğitim Kaybı	Doğrulama Kaybı	WER	CER	Adım Sayısı
Wav2Vec2-xls-r-300m	0,06	0,31	0,28	0,06	16800
Whisper-Small	0,001	0,25	0,16	0,04	4000

Modellerin performansını değerlendirmek için eğitim ve doğrulama veri kümesinde yer almayan kayıtlarla modeller test edilmiştir. Test verisi hazırlanırken AdresGezgin A.Ş. veri tabanında kayıtlı çağrı merkezi kaydı örneklerinden yararlanılmıştır. Farklı kişilerin konuştuğu çağrı kayıtlarından seçilen ve yaklaşık 3 saat uzunluğundaki test verisi yazıya çevrilerek her ses kaydına ait etiket oluşturulmuştur. Test veri kümesindeki her bir kayıt Common Voice veri kümesiyle en iyileştirmesini gerçekleştirdiğimiz Whisper-Small ve Wav2Vec2-XLS-R-300M modelleriyle sestem metne çevrilmiştir. İnsan tarafından etiketlenilmiş çağrı merkezi kayıtlarının konuşmadan metne çevirisi referans metin olarak kabul edilmiş ve model çıktılarıyla referans metinler kıyaslanarak modellerin test kayıtları üzerindeki WER değeri hesaplanmıştır. Test verileriyle gerçekleştirilen bu değerlendirme sonucunda Whisper-Small modelinin Wav2Vec2-XLS-R-300M modeline göre sestem metne çeviride daha yüksek başarı gösterdiği gözlemlenmiştir. Ancak Şekil-8'de de görüldüğü üzere çağrı merkezi kayıtlarıyla gerçekleştirilen testte her iki model de yüksek WER değerlerine sahiptir. Çağrı merkezi kayıtlarının net şekilde anlaşılmayan, temiz kayıtlar olmaması ayrıca eğitim veri kümesinde çağrı merkezi kayıtlarına hiç yer verilmemesi her iki modelin de bu kayıtlardaki başarısını düşürmüştür.

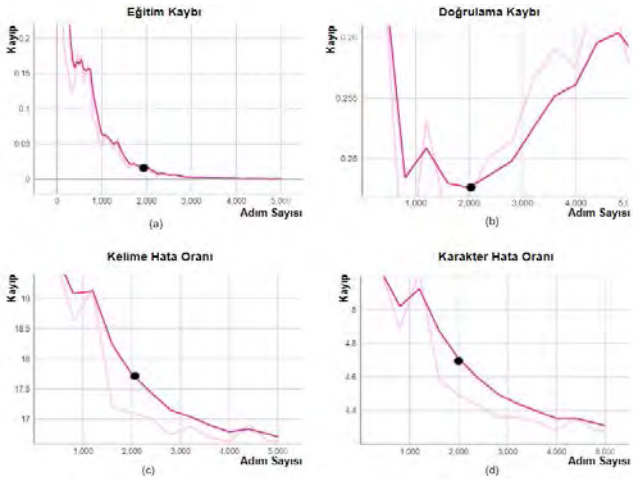
Ayrıca Wav2Vec2-XLS-R-300M modelinin STT çıktısı noktalama işaretleri ve küçük büyük harf uyumu bulunmayan metinler iken Whisper-Small modeli farklı olarak metin çıktısına noktalama işaretlerini eklemektedir. Fakat bu çalışmada modelleri aynı özellikteki birebir aynı veri kümesiyle ince ayar yaparak başarılarını analiz etmek amacıyla her iki model için de veri kümesi aynı ön işlemlerden geçirilmiş ve eğitim veri kümesi noktalama işaretleri, özel karakterler, büyük harflerden ve şapkalı harflerden arındırılmıştır. Bu nedenle bu çalışma sonucunda elde edilen ince ayarlanmış Whisper-Small modelinin konuşmadan metne çevirisi noktalama işaretleri içermemektedir. Bununla birlikte Şekil-8'de görüleceği üzere Whisper-Small modelinin genelleme yetisi seçili Wav2Vec2-XLS-R-300M modeline göre belirli ölçüde daha iyidir. Son olarak Çizelge-5'te eğitim ve doğrulama veri setine dahil edilmemiş Common Voice verisinden alınmış ses kayıtlarının konuşmadan metne çıktıkları verilmiştir. Referans metin ve model çıktıları karşılaştırıldığında Whisper-Small modelinin daha başarılı sonuçlar verdiği gözlemlenmektedir.

Önerilen çalışma Python programlama dili ile gerçekleştirilmiştir. Verilerin düzenlenmesi için Numpy ve Pandas kütüphanelerinden faydalanılmıştır. Eğitim sonuçlarının gösterilmesi için TensorBoard kullanılmıştır.

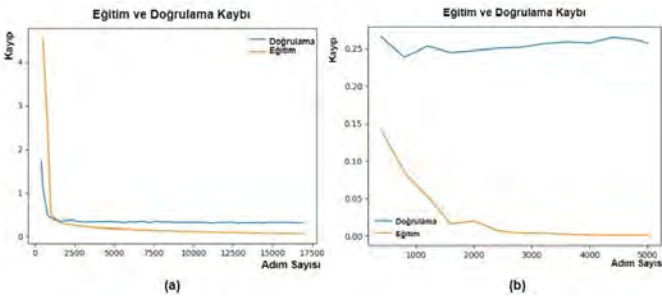
Model eğitimleri 12 çekirdekli 2.90 GHz Intel(R) i5-10400F CPU, 48 GB 3600MHz DDR4 RAM ve GeForce RTX 3060 12 GB GPU'ya sahip cihazda gerçekleştirilmiştir.



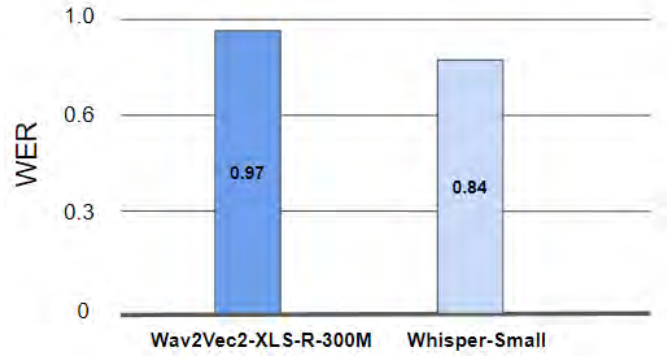
Şekil-5: Wav2Vec2-XLS-R-300M eğitim kaybı (a), doğrulama kaybı (b), model başarısını kelime düzeyinde analiz eden kelime hata kaybı (c), model başarısını karakter düzeyinde analiz eden karakter hata kaybı (d) grafikleri



Şekil-6: Whisper-Small eğitim kaybı (a), doğrulama kaybı (b), model başarısını kelime düzeyinde analiz eden kelime hata kaybı (c), model başarısını karakter düzeyinde analiz eden karakter hata kaybı (d) grafikleri



Şekil-7: Wav2Vec2-XLS-R-300M modeli eğitim kaybı ve doğrulama kaybı (a), Whisper-Small modeli eğitim kaybı ve doğrulama kaybı (b) grafikleri



Şekil-8: Wav2Vec2-XLS-R-300M ve Whisper-Small modellerinin çağrı merkezi kayıtlarında WER değerleri

Çizelge-5: Referans metin ile birlikte ince ayarlanmış Wav2Vec2-XLS-R-300M ve Whisper-Small modelleriyle konuşmadan metne çeviri örnekleri

Referans Metin	Wav2Vec2-XLS-R-300M	Whisper-Small
Son birkaç yılda nasıl bir değişim oldu	son birkaç yılda nasıl bir değişim olduk	son birkaç yılda nasıl bir değişim olduk
evrak işlerini halletmek iki üç ay sürerse yunanistana hangi şirket gelir	evrak işlerini halletmek iki üç ay sürerse yunanistan ahangı şirket gelir	evrak işlerini halletmek iki üç ay sürerse yunanistanı hangi şirket gelir
güneydoğu avrupa ülkeleri biyoyakıt konusunu değerlendiriyor	güneydoğu ağrup avülkeleri biryoyakıt konusunu değerlendiriyor	güneydoğu avrupa ülkeleri biyoyakıt konusunu değerlendiriyor
yine teknolojiye güvenip olanları kayıtsız kalmak mümkün mü	yine teknolojiye güvenip olanları kayıtsız kalmak mümkün mü	yine teknolojiye güvenip olanları kayıtsız kalmak mümkün mü
bu bizi her koşulda birleştiren bir şey	bu bizi her koşulda birleştirembi ir şey	bu bizi her koşulda birleştiren bir şey

## 5. Sonuçlar

Önerilen bu çalışma ile açık kaynaklı bir veri kümesi olan Türkçe dilinde hazırlanmış Common Voice 11.0 ile literatürde yer alan Whisper-Small ve Wav2Vec2-XLS-R-300M STT modelleri ince ayar yapılmıştır. İnce ayar ardından her iki modelin Türkçe dilinde sestem metne çeviri problemi üzerindeki başarıları incelenmiş ve modellerin performansı değerlendirilmiştir. Çalışmada Whisper-Small modelinin 2000. adım sayısında aşırı öğrenme gösterdiği dikkate alınarak Wav2Vec2-XLS-R-300M modelinin de aynı adım sayısındaki performansı incelenmiş fakat aşırı öğrenme gözlenmemesi nedeniyle eğitim devam ettirilmiştir. Her iki model için de en düşük doğrulama kaybının elde edildiği adım sayısındaki modeller kaydedilmiştir. En düşük WER değerinin elde edildiği adım sayısında Whisper-Small modelinde 0,16 WER değeri elde edilirken Wav2Vec2-XLS-R-300M modelinde WER değeri 0,28 olarak elde edilmiştir. Ayrıca şirket bünyesinde gerçekleştirilmiş çağrı merkezi kayıtlarıyla oluşturulmuş test verisiyle modeller test edilmiştir. Eğitim veri kümesinde çağrı merkezi kaydına yer verilmemesi nedeniyle her iki modelde

de yüksek WER değeri elde edilmiştir. Ancak yine de gerek daha hızlı bir şekilde eğitilebilmesi, gerekse uygun noktalama işaretlerini üretebilme gerekse genelleme kabiliyetleri açısından Whisper-Small modeli daha tercih edilebilir bir model olarak görünmektedir. Gerçekleştirilen bu çalışmada veri kümesinin tek bir kaynaktan oluşması ve az sayıda veri içerdiği göz önüne alındığında gelecek çalışmalarda veri kaynağı çeşitliliğinin oluşturulması ve çok sayıda veri ile gerçekleştirilecek model ince ayarıyla STT başarısının artırılması hedeflenmektedir.

## 6. Teşekkür

Bu çalışma TÜBİTAK TEYDEB 1501 kapsamında desteklenmekte olan 3210713 numaralı "Güncel Derin Öğrenme Mimarileri ile Türkçe Dili için Konuşmadan Metne Çeviri Yapabilen ve Hizmet Olarak Yazılım (SaaS) Modeli ile Çalışan Sistemin Geliştirilmesi" isimli proje kapsamında gerçekleştirilmiştir.

## Kaynaklar

- [1] Özlan, B., Haznedaroğlu, A., Arslan, L. M., *Automatic fraud detection in call center conversations*, In 2019 27th Signal Processing and Communications Applications Conference (SIU), 2019, pp. 1-4.
- [2] Dhanjal, A. S., Singh, W. *An automatic machine translation system for multi-lingual speech to Indian sign language. multimedia Tools and Applications*, 2022, pp.1-39.
- [3] Ballati, F., Corno, F., De Russis, L., *Assessing virtual assistant capabilities with Italian dysarthric speech*, In Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility, 2018, pp. 93-101.
- [4] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., Kingsbury, B., *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups*, IEEE Signal processing magazine, 2012, 29(6), pp.82-97.
- [5] Sainath, T. N., Vinyals, O., Senior, A., Sak, H. *Convolutional, long short-term memory, fully connected deep neural networks*, IEEE international conference on acoustics, speech and signal processing (ICASSP), 2015, pp. 4580-4584.
- [6] Alharbi, S., Alrazgan, M., Alrashed, A., Alnomasi, T., Almojel, R., Alharbi, R., Almojil, M., *Automatic speech recognition: Systematic literature Review*, IEEE Access, 9, 2021, pp.131858-131876.
- [7] Hellman, E., Nordstrand, M., *Research in methods for achieving secure voice anonymization: Evaluation and improvement of voice anonymization techniques for whistleblowing*, 2022.
- [8] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y., *Attention-based models for speech recognition*, Advances in neural information processing systems, 2015.
- [9] Bahar, P., Bieschke, T., Ney, H., *A comparative study on end-to-end speech to text translation*, Automatic Speech Recognition and Understanding Workshop (ASRU), 2019, pp. 792-799.
- [10] Tang, Y., Pino, J., Wang, C., Ma, X., Genzel, D., *A general multi-task learning framework to leverage text data for speech to text tasks*, In ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing, 2021, pp. 6209-6213.
- [11] Tombaloğlu, B., Erdem, H. A., *SVM based speech to text converter for Turkish language*, In 2017 25th Signal Processing and Communications Applications Conference (SIU), 2017, pp. 1-4.
- [12] Kimanuka, U. A., & Buyuk, O., *Turkish speech recognition based on deep neural networks*, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2018, pp.319-329.
- [13] Tombaloğlu, B., Erdem, H., *Deep Learning Based Automatic Speech Recognition for Turkish*, Sakarya University Journal of Science, 2020, pp.725-739.
- [14] Tombaloğlu, B., & Erdem, H., *Turkish Speech Recognition Techniques and Applications of Recurrent Units (LSTM and GRU)*, Gazi University Journal of Science, 2021, pp.1035-1049.
- [15] Safaya, A., Erzin, E., *HuBERT-TR: Reviving Turkish Automatic Speech Recognition with Self-supervised Speech Representation Learning*, 2022, arXiv preprint arXiv:2210.07323.
- [16] Li, Z., Niehues, J., *Efficient Speech Translation with Pre-trained Models*, 2022, arXiv preprint arXiv:2211.04939.
- [17] Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. *Wav2vec 2.0: A framework for self-supervised learning of speech representations*, Advances in Neural Information Processing Systems, 2020, pp. 12449-12460.
- [18] Vásquez-Correa, J. C., Álvarez Muniain, A., *Novel Speech Recognition Systems Applied to Forensics within Child Exploitation: Wav2vec2.0 vs. Whisper*, 2023, Sensors, 23(4), 1843.
- [19] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I., *Robust speech recognition via large-scale weak supervision*, 2022, arXiv preprint arXiv:2212.04356.
- [20] Taşar D.E., *An automatic speech recognition system proposal for organizational development*, 782089, Master's thesis, Dokuz Eylül University Management Information Systems, 2023.
- [21] Mercan, Ö. B., Özdil, U., Ozan, Ş., *Çok Dilli Sesten Metne Çeviri Modelinin İnce Ayar Yapılarak Türkçe Dilindeki Başarısının Artırılması Increasing Performance in Turkish by Finetuning of Multilingual Speech-to-Text Model*, 30th Signal Processing and Communications Applications Conference (SIU), 2022, pp. 1-4.
- [22] Arduengo, J., Köhn, A., *The Mozilla Common Voice Corpus. In Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH 2019)*, 2019, pp. 1823-1827.
- [23] Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M., *Unsupervised cross-lingual representation learning for speech recognition*, 2020, arXiv preprint arXiv:2006.13979.
- [24] Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Auli, M., *XLS-R: Self-supervised cross-lingual speech representation learning at scale*, 2021, arXiv preprint arXiv:2111.09296.
- [25] Openai. (2022, December 9). *Whisper/model-card.md at main · openai/whisper*. GitHub. Retrieved February 5, 2023, from <https://github.com/openai/whisper/blob/main/model-card.md>
- [26] Ali, A., Renals, S., *Word error rate estimation for speech recognition: e-WER*, In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 20-24.
- [27] Maas, A., Xie, Z., Jurafsky, D., & Ng, A. Y., *Lexicon-free conversational speech recognition with neural networks*, In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 345-354.
- [28] "wav2vec2-xls-r-300m-tr", <https://huggingface.co/Sercan/wav2vec2-xls-r-300m-tr>
- [29] "whisper-small-tr-2", <https://huggingface.co/Sercan/whisper-small-tr-2>



# Birleştirilmiş Derin Öznitelikleri Kullanarak BT ve X-Ray Görüntülerinden COVID-19 Tespiti

## COVID-19 Detection from CT and X-Ray Images Using Deep Feature Fusion

Asuman GÜNAY YILMAZ  
Karadeniz Teknik Üniversitesi  
Yazılım Mühendisliği  
Trabzon, Türkiye  
gunaya@ktu.edu.tr  
ORCID: 0000-0003-3960-5085

Emine AYAZ  
Karadeniz Teknik Üniversitesi  
Yazılım Mühendisliği  
Trabzon, Türkiye  
eminekoch@gmail.com  
ORCID: 0009-0008-3512-4640

Muammer TÜRKOĞLU  
Samsun Üniversitesi  
Yazılım Mühendisliği  
Samsun, Türkiye  
muammer.turkoglu@samsun.edu.tr  
ORCID: 0000-0002-2377-4979

### Öz

2019 yılında Çin'in Wuhan kentinde başlayan yeni tip Koronavirüs (COVID-19), solunum yolu enfeksiyonuna neden olan bulaşıcı bir virüsdür. Bu virüs dünyada kısa sürede etkili olmuş ve bir salgına dönüşmüştür. Bu tür bulaşıcı hastalıkların erken teşhisi ve gerekli tedavinin erken süreçte başlatılması çok önemlidir. X-ışını (X-Ray) ve Bilgisayarlı Tomografi (BT) tıbbi radyolojik görüntüleme yöntemleri ile derin öğrenme ve makine öğrenmesi tekniklerinin kullanılması bu hastalığın doğru ve hızlı tespitine yardımcı olmaktadır. Bu çalışmada; normal-COVID-19-pnömoni (zatürre) etiketli X-Ray ve normal-COVID-19 etiketli BT görüntülerini içeren 2 farklı veri kümesi kullanılmıştır. Bununla birlikte; InceptionResNetV2, VGG-16 ve DenseNet121 derin öğrenme mimarileri ve kNN ile SVM sınıflandırıcıları kullanılmıştır. Bu kapsamda 3 farklı çalışma yürütülmüştür. Öncelikle her bir ağın sınıflandırma başarımı incelenmiştir. Daha sonra ağların ürettiği öznitelik vektörleri ayrı olarak sınıflandırıcılarla işleme sokulmuştur. Son olarak ağların ürettiği öznitelik vektörleri birleştirilmiş ve sınıflandırma işlemi gerçekleştirilmiştir. Sonuç olarak göğüs BT veri kümesindeki COVID-19 ve normal görüntüleri için en yüksek sonuç %98,9 doğruluk ile birleştirilmiş öznitelikler ve kNN sınıflandırıcısı ile elde edilmiştir.

**Anahtar sözcükler:** COVID-19, derin öğrenme, sınıflandırma, öznitelik birleştirme

### Abstract

The new type of Coronavirus (COVID-19), which started in 2019 in Wuhan, China, is an infectious virus that causes respiratory tract infection. This virus became effective in the world in a short time and turned into an epidemic. Early diagnosis of such infectious diseases and initiation of the necessary treatment at an early stage are very important. The use of X-ray (X-Ray) and Computed Tomography (CT) medical radiological imaging methods and deep learning and machine learning techniques help in the accurate and rapid detection of this disease. In this study; Two different datasets were used, including X-Ray images labeled normal-COVID-19-pneumonia (pneumonia) and CT images labeled normal-COVID-19. Inception ResNetV2, VGG-16 and DenseNet121 deep learning architectures and kNN and SVM classifiers are used. In this context, 3 different experiments were carried out. First of all, the classification performance of each network was examined. Then, the feature vectors produced by the networks were separately processed with classifiers. Finally, the feature vectors produced by the networks were combined and the classification process was carried out. As a result, the highest result for COVID-19 and normal images in the chest CT dataset was obtained with the combined features and kNN classifier with 98.9% accuracy.

**Keywords:** COVID-19, deep learning, classification, feature fusion

### 1. Giriş

Koronavirüs (COVID-19); hem üst solunum yollarında hem de akciğerlerde enfeksiyona neden olan yeni bir virüsdür. Çin'in

Wuhan kentinde ortaya çıkan COVID-19, artan hareketlilik ve hızlı bulaşıcılık faktörlerine bağlı olarak kısa sürede küresel bir salgın haline gelmiş, tüm dünyada milyonlarca kişinin ölümüne neden olmuştur. COVID-19 tanısı, ilk aşamada PCR veya kan tahlilleri ile doğrulanmaktadır. Son araştırmalar göstermiştir ki, özellikle hastalığın erken döneminde veya viral yük düşük olduğunda, pozitif çıkması gereken PCR testleri negatif çıkabilmektedir. Ancak hastalık tedavisi ve kontrolü için erken teşhis büyük önem taşımaktadır. Bu sebeple X-Ray ve Bilgisayarlı Tomografi (BT) görüntüleme yöntemleri tercih edilmektedir. Çeşitli viral ve bakteriyel pnömoni (zatürre) görüntüleri COVID-19 ile benzerdir ve benzer özellikler içermektedir [1]. Radyologların viral ve bakteriyel pnömoni gibi benzer akciğer hastalıklarını COVID-19'dan ayırt etmesi zordur. Bu benzerlik yanlış tanıları yol açabilmektedir. Bu görüntülere derin öğrenme mimarileri ve makine öğrenmesi sınıflandırma tekniklerinin uygulanması ile hastalığın erken tespitinde çok yüksek oranda tahmin başarısı elde edilebilmektedir.

Bu çalışmada birleştirilmiş derin öznitelikler kullanılarak BT ve X-ray görüntülerinden COVID-19 tespiti gerçekleştirilmiştir. Bu amaçla deneylerde 3 sınıflı (normal, COVID-19, viral pnömoni) akciğer X-ray görüntülerini içeren "COVID-19 Radiography Database" ve 2 sınıflı (normal, COVID-19) BT görüntülerini içeren "SARS-COV-2 Ct-Scan Dataset" veri kümeleri kullanılmıştır. İlk aşamada Inception-ResNetV2, VGG-16 ve DenseNet121 CNN mimarilerinin her iki veri kümesindeki COVID-19 tespiti başarımları incelenmiştir. Daha sonra bu ağlarla üretilen derin özniteliklerin kNN ve SVM sınıflandırıcıları ile başarımları hesaplanmıştır. Son olarak bu derin öznitelikler birleştirilerek sınıflandırma gerçekleştirilmiştir. Bir sonraki bölümde yayınlarda bu konuda yapılan çalışmalar özetlenmektedir.

Yazının devamı şu şekilde planlanmıştır: COVID-19 tespiti ile ilgili yapılan çalışmalar 2. bölümde özetlenmiştir. Çalışmada kullanılan sistem 3. bölümde açıklanmıştır. 4. bölümde elde edilen Bulgular verilmiş, 5. bölümde sonuçlar tartışılmıştır.

## 2. Yayınlar Özeti

Yayınlarda COVID-19 vakalarının tespitinde çeşitli çalışmalar yapılmıştır. Erdaş ve Ölçer [2]; göğüs röntgeni görüntüleri kullanarak COVID-19 hastalarının teşhisinde CNN mimarilerinden AlexNet kullanmıştır. Çalışmada; halka açık platformlar olan Kaggle ve Github'dan derlenen 3 sınıflı (normal, COVID-19, bakteriyel pnömoni) 150 göğüs X-Ray görüntüsü kullanılmıştır. Sonuçta; pnömoni kaynağının bakteri veya COVID-19 virüsü olup olmadığını ayırt etmede %81,0 doğruluk elde edilmiştir.

Özbay ve arkadaşı [3]; COVID-19 vakalarının tespitinde çeşitli sınıflandırma algoritmalarının ve CNN başarımlarını yöntem karşılaştırmıştır. Çalışmada 2 sınıflı (COVID 19, normal) 2482 göğüs BT görüntüsü kullanılmıştır. Sonuçta CNN modelinin %98,1 doğruluk ile diğer gelişmiş sınıflandırma algoritmalarından daha iyi başarımlar gösterdiği belirlenmiştir. Kutlu ve Camgözlü [4]; COVID-19'un tespiti için CNN derin öğrenme modelleriyle birlikte kNN, SVM gibi sınıflandırıcıları kullanmıştır. Çalışmada "COVID-19 radiography database" veri kümesinde 3 sınıflı (COVID-19, normal ve viral pnömoni)

toplam 3886 X-Ray görüntüsünden yararlanılmıştır. Sonuçta, %99,53 doğruluk oranı ile CNN+SVM en iyi sonucu vermiştir.

Güraksın ve arkadaşları [5]; göğüs X-Ray görüntülerinden COVID-19 teşhisi için AlexNet ve GoogleNet ağlarını tercih etmişlerdir. Eğitim ve test için "Chest X-Ray Images (Pneumonia)" ve "COVID-19 Image Data Collection," açık kaynak veri kümelerinden alınan görüntüler bir araya getirilerek 4 sınıflı (normal, COVID 19, bakteriyel pnömoni ve viral pnömoni) 1670 adet göğüs X-Ray görüntüsünden oluşan bir veri kümesi kullanılmıştır. Çalışmada %98,77 doğruluk ile normal-COVID 19- bakteriyel pnömoni sınıflandırmada en yüksek başarı elde edilmiştir. Hemdan ve arkadaşları [6]; VGG19, DenseNet121, ResNetV2, InceptionV3, InceptionResNetV2, Xception ve MobileNetV2 gibi yedi CNN ağı ile X-Ray görüntülerinden COVID-19 tespiti gerçekleştirmiştir. Çalışmada, 2 sınıflı (COVID 19, normal) 50 adet göğüs X-Ray görüntüsü kullanılmıştır. Sonuçta VGG19 ve DenseNet modelleri, normal ve COVID-19 için sırasıyla %89 ve %91 F1 Değeri ile sınıflandırma başarımı göstermiştir.

Jia ve arkadaşları [7]; BT görüntülerinden COVID-19 teşhisinde ResNet ağını, X-Ray görüntülerinden COVID-19 teşhisinde ise MobileNet ağını kullanmıştır. Çalışmada 5 sınıflı (normal, COVID 19, viral pnömoni, bakteriyel pnömoni, tüberküloz) toplam 7592 X-Ray ve BT görünümü kullanılmıştır. Önerilen yöntemlerle beş sınıflı X-Ray görüntü veri kümesinde %99,6, BT görüntü veri kümesinde %99,3 test doğruluğuna ulaşılmıştır. Şahinbaş ve Çatak [8]; X-Ray görüntüleriyle COVID-19 tespitinde VGG-16, VGG19, ResNet, DenseNet ve InceptionV3 ağlarını kullanmıştır. Çalışmada değişik hastanelerden ve halka açık Github kaynağından elde edilen 2 sınıflı (normal, COVID 19) 140 X-Ray göğüs röntgen görüntüleri kullanılmıştır. En iyi başarımlar, %80'lik sınıflandırma doğruluğu ile VGG-16 modelinde elde edilmiştir.

Sethy ve Behera [9]; COVID-19 tespiti için AlexNet, VGG-16, VGG19, GoogleNet, ResNet18, ResNet50, ResNet101, InceptionV3, InceptionResNetV2, DenseNet201 ve XceptionNet gibi derin öğrenme mimarilerini kullanmıştır. Bu çalışmalar için GitHub, Kaggle ve Open-i tabanlı ve 2 sınıflı (COVID 19 ve Normal) toplam 316 X-Ray görüntüsü kullanılmıştır. Bu derin öğrenme modellerinden elde edilen derin özellikler SVM ile sınıflandırılmış, ResNet50 + SVM sınıflandırma yöntemi ile %95,38 başarımlar sağlanmıştır. Asnaoui ve Chawki [10]; COVID-19 pnömonisinin tespiti ve sınıflandırılması amacıyla VGG-16, VGG19, DenseNet201, Inception\_ResNet\_V2, Inception\_V3, ResNet50 ve MobileNet derin öğrenme modellerinin karşılaştırmasını yapmışlardır. Bu çalışma için; 'chest X-ray & CT dataset' ve 'COVID Chest X-ray Dataset' 'den oluşan iki ayrı veri kümesindeki X-Ray(röntgen) ve BT(bilgisayarlı tomografi) görüntüleri birleştirilerek 3 sınıflı (COVID, normal ve viral pnömoni) toplamda 6087 görüntü kullanılmıştır. Çalışma sonucunda; Inception ResNetV2 ağı %92,18 doğruluk ile en başarılı model olarak belirlenmiştir.

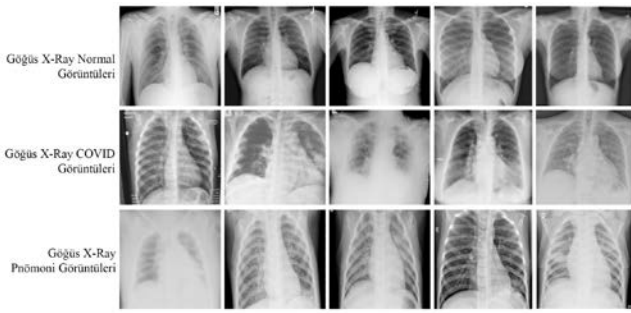
Adrakani ve arkadaşları [11]; COVID 19 ile sağlıklı hastaları ayırt etmek için AlexNet, VGG-16, VGG-19, SqueezeNet, GoogleNet, MobileNet-V2, ResNet-18, ResNet-50, ResNet-101 ve Xception modellerini kullanmıştır. Elde edilen başarımların değerlendirilmesinde ResNet-101 %99,51

doğruluk oranı ile en başarılı model olarak belirlenmiştir. Kart ve Başçıftçı [12]; makine öğrenmesi algoritması olan kNN, YSA, RO ve SVM algoritmaları ile COVID-19 hastalığına ait görüntüleri sınıflandırmıştır. Bu çalışmalar için açık kaynaklı 2 sınıflı (normal, COVID-19) "Posteroanterior Chest X-ray (PACXR)" veri kümesindeki 1500 X-Ray görüntüsü kullanılmıştır. Sonuçta en yüksek test başarı oranı %99,8'lük bir oranla SVM algoritmasından elde edilmiştir.

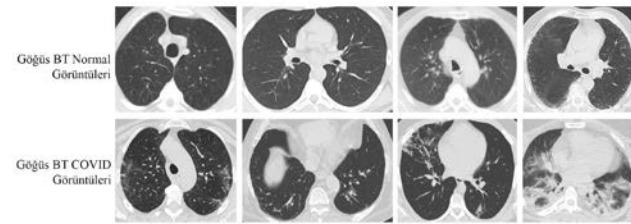
### 3. Materyal ve Yöntem

#### 3.1 Veri Kümesi

Bu çalışmada; 3 sınıflı X-Ray görüntüleri içeren "SARS-COV-2 Ct-Scan Dataset" [13] ile 2 sınıflı BT görüntüleri içeren "COVID-19 Radiography Database" [14] veri kümeleri kullanılmıştır. Veri kümelerindeki sınıflara ait örnek görüntüler sırasıyla Şekil 1 ve 2'de görülmektedir.



Şekil-1: Akciğer X-Ray görüntüleri



Şekil-2: Akciğer bilgisayarlı tomografi görüntüleri

SARS-COV-2 Ct-Scan Dataset, 1252 COVID, 1230 normal olmak üzere toplam 2.482 BT görüntüsünden oluşan halka açık bir COVID-19 tarama veri setidir. Bu veriler Brezilya, Sao Paulo'daki hastanelerdeki gerçek hastalardan toplanmıştır. Bu veri kümesinin amacı, bir kişinin BT taramalarının analizi yoluyla COVID-19 ile enfekte olup olmadığını belirleyebilen yapay zeka yöntemlerinin araştırılmasını ve geliştirilmesini teşvik etmektir.

COVID-19 Radiography Database veri kümesinde 10.192 normal, 6012 COVID ve 1345 Viral Pnömoniden oluşan 17.549 X-Ray görüntüsü mevcuttur. Çeşitli ülkelerden araştırmacıların birlikte oluşturduğu geniş bir veri kümesidir. Çalışmanın uygulama ortamı Google Colab'da veri kümesinin büyük olması sorunlara neden olduğu için çalışmada bu veri kümesindeki her sınıftan rastgele 1000 adet olmak üzere toplam 3.000 X-Ray görüntüsü alınmıştır. Çalışmada kullanılan, X-Ray ve BT görüntü sayıları Çizelge 1'de verilmiştir. Deneylerde toplam 5.482 adet görüntü kullanılmıştır.

### 3.2 Derin Öğrenme

Derin öğrenme; çok katmanlı yapay sinir ağları ile insan beynini taklit eden çalışma prensibine sahiptir. Derin

Çizelge-1: Veri setleri görüntü sayıları

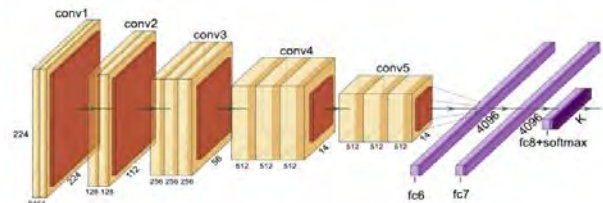
	X-RAY GÖRÜNTÜLER	BT GÖRÜNTÜLER
Normal	1.000	1.230
COVID	1.000	1.252
Viral Pnömoni	1.000	-
Toplam	3.000	2.482

öğrenme; makinelerin konuşulanları algılama, tercüme etme, nesnelere tanımlama ve karar verme yeteneği kazandırma gibi insan denetimi olmadan öğrenmesini sağlar. Sağlık hizmetleri, finansal hizmetler, havacılık ve savunma, sürücüsüz araçlar, sanal yardımcıları gibi bilgisayarla görme, ses tanıma ve doğal dil işleme alanlarında kullanılmaktadır.

Yapay sinir ağlarındaki katman sayılarının artırılmasıyla kurulan Evrişimli Sinir Ağları (CNN), Tekrarlayan Sinir Ağı (SRN), Uzun Kısa Vadeli Memory (RNN), Kısıtlanmış Boltzmann Makineleri (RBM), Derin İnanç Ağları (DBN) ve Derin Oto-kodlayıcılar (DAE) gibi çok farklı türde derin öğrenme mimarileri bulunmaktadır. CNN, bir girdi verisini alarak, bu verideki farklı birçok özelliği birbirinden ayırabilen bir derin öğrenme algoritmasıdır. AlexNet, ResNet, GoogleNet/InceptionNet, MobileNetV1, VGG-16,VGG19, Inception v2, Inception v3,V4 , Inception-ResNet, DenseNet, Xception, Channel Boosted CNN, Residual Attention CNN Attention Based CNNs, Feautre-Map based CNNs ve Squeeze and Excitation Networks gibi birçok farklı mimarileri vardır. Mevcut çalışmada, üç farklı yapıya ve yüksek başarıma sahip VGG-16, InceptionResNetV2 ve DenseNet121 derin mimariler kullanılmıştır.

#### 3.2.1 VGG-16 Mimarisi

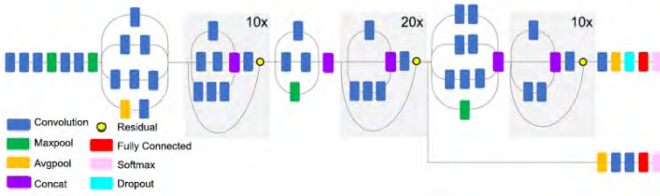
VGG-16 modeli 13 konvolüsyon, 3 tam bağlaşımlı katman olmak üzere 16 katmandan oluşur (Şekil 3). Modele 224x224x3 boyutlarında veri girişi yapılır. Bu ağ oldukça büyük bir ağıdır, ReLU aktivasyon fonksiyonu kullanılmıştır ve yaklaşık 138 milyon parametreye sahiptir [15].



Şekil-3: VGG-16 yapısı [15]

#### 3.2.2 Inception ResNet V2 Mimarisi

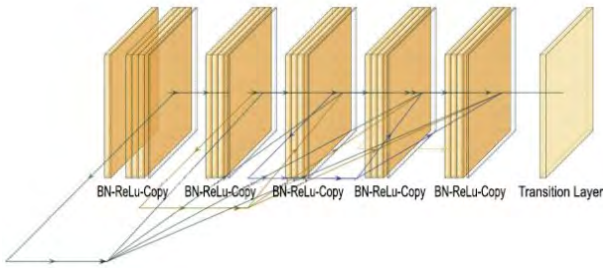
Inception-ResNet modeli, Inception mimarisinde yer alan Inception modülleri ile ResNet mimarisinin başarımından esinlenilerek oluşturulan melez bir modeldir [16]. Inception V3 modelinden daha derin bir mimariye sahiptir (Şekil 4).



Şekil-4: Inception ResNet V2 mimari yapısı [16]

### 3.2.3 DenseNet121 Mimarisi

Yapay sinir ağları eğitilirken, evrişim ve alt örnekleme süreçleri nedeniyle öznitelik haritaları azalır. Aynı zamanda katmanlar arası geçişlerde görüntü özelliğinde kayıplar olmaktadır. Görüntü bilgilerini daha etkin kullanmak için Huang ve arkadaşları DenseNet sistemini geliştirmiştir. Sistemde her katman diğer katmanlara ileri doğru beslenir. Bu sayede herhangi bir katman, kendisinden önceki tüm katmanların özellik bilgilerine erişebilir (Şekil 5). [17]



Şekil-5: DenseNet Mimarisi [17]

## 3.3 Sınıflandırma

### 3.3.1. k-En Yakın Komşu Algoritması

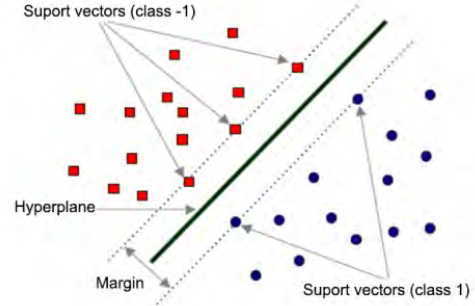
kNN algoritması, tüm makine öğrenme algoritmaları arasında en basit ve kullanışlı denetimli sınıflandırma algoritmasıdır. Yüz tanıma, el yazısı ve resim tanıma, videodaki nesnelerin algılanmasında sıkça kullanılmakta olup görüntüleme teknikleri ve tahlillerden hastalıkların teşhisine olanak sağlamaktadır. Bu algoritmada, sınıf özelliği daha önceden belirlenmiş verilerden faydalanılır. kNN algoritması ile sınıflandırma yapabilmek için, öznitelik vektörlerinin önceden oluşturulmuş olması gereklidir.

kNN sınıflandırma yöntemi, uygulanmasının basit, gürültülü verilere karşı etkili olması ve büyük eğitim setlerinde uygulanabilmesi avantajlarına sahiptir. Her yeni örnek eklendiğinde tüm uzaklık hesaplamalarının yinelenmesi açısından hesaplama maliyeti oldukça yüksektir. k parametresine ihtiyaç duyması, yüksek miktarda bellek alanlarına ihtiyaç duyması, hesaplamada birçok farklı fonksiyon kullanılabilmesi ve hangisinin kullanılması gerektiği karmaşası içermesi dezavantajlarına sahiptir.

### 3.3.2. Destek Vektör Makineleri

SVM farklı sınıflara ait verileri karar sınır çizgisi ile birbirinden en uygun şekilde ayırmak için kullanılan denetimli sınıflandırma algoritmasıdır. Günümüzde yüz algılama, protein çaprazlama, el yazısı tanıma, görüntü ve sınıflandırma, biyoinformatik, çevre bilimleri ve tahmine dayalı kontrol gibi birçok alanda kullanılmaktadır.

Ayırım için gerekli karar sınırı, sınıflar arasındaki en iyi boşluğu bularak düzlem üzerine yerleştirilmiş noktaları ayırmak için bir doğru (hiper düzlem sınır çizgisi) çeker (Şekil 6). En iyi yerden geçebilecek çizginin belirlenmesini kolaylaştıran ve karar sınırına en yakın noktalara veya değerlere "destek" denir. Bu değer veya noktalar değiştiğinde karar sınırları da değişir. Destek vektör ismini buradan almaktadır.



Şekil-6: SVM küme ayrışımı

## 4. Deneyler ve Bulgular

Bu çalışma, Python programlama diline dayalı bir bulut hizmeti olan Google Colab kullanılarak gerçekleştirilmiştir. Çalışmada OpenCV, pandas, keras, tensorflow, numpy, matplotlib, random, Python kütüphaneleri kullanılmıştır. Deneysel çalışmanın yapıldığı bilgisayar, Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz 2.00 GHz işlemcili, NVIDIA GeForce GT 550 ekran kartlı ve 4GB birincil belleğe sahiptir. Deneysel çalışmada, veriler %80 eğitim, %20 test olarak ayrılmıştır. 2.482 BT verilerinin 1.986'sı eğitim, 496'sı test için kullanılmıştır. 3.000 X-Ray verilerinin 2.400'ü eğitim, 600'ü test için kullanılmıştır.

Denetimli öğrenme tekniklerinin kullanıldığı sınıflandırma problemlerinde, model başarımını değerlendirebilmek için en çok kullanılan yöntemlerden biri, hedef niteliğin sınıflarına ait gerçek ve tahmin değerlerinin bir arada gösterdikleri karmaşıklık tablosundan hareketle hesaplanan değerlerdir. Şekil 7'de bu tablo gösterilmiştir.

DURUM	GERÇEK			
	POZİTİF	POZİTİF	NEGATİF	TOPLAM
TAHMİNİ	POZİTİF	Doğru Pozitif (dp)	Yanlış Pozitif (yp)	tPoz
	NEGATİF	Yanlış Negatif (yn)	Doğru Negatif (dn)	tNeg
	TOPLAM	poz	neg	m

Şekil-7: Karmaşıklık tablosu

Yapılan çalışmada modellerin başarımlarını değerlendirmek için karmaşıklık tablosundan hareketle hesaplanan başarı ölçütleri doğruluk, tutturma, bulma ve F1 ölçüsüdür. Bu ölçütler aşağıdaki denklemlerle hesaplanmaktadır.

$$\text{Doğruluk (Accuracy)} = \frac{N_{TN} + N_{TP}}{N_{TN} + N_{FN} + N_{TP} + N_{FP}} \quad (1)$$

$$\text{Tutturma (Precision)} = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad (2)$$

$$\text{Bulma (Recall)} = \frac{N_{TP}}{N_{TN} + N_{FP}} \quad (3)$$

$$\text{F1 Ölçüsü} = \frac{2 * N_{TP}}{2 * N_{TP} + N_{FP} + N_{FN}} \quad (4)$$

**Çizelge-2: VGG16, InceptionResNetV2 ve DenseNet121 ağlarının COVID-19 tespiti başarımları**

Veri kümesi	BT (COVID 19, Normal)				X-RAY (COVID 19, Normal, Pnömoni)			
	Doğruluk	F1	Bulma	Tutturma	Doğruluk	F1	Bulma	Tutturma
VGG-16	%87,5	%87,6	%79	%98,2	%79,1	%72	%84,4	%80,6
InceptionResNetV2	%94,5	%94,2	%89,5	<b>%99,5</b>	%92,1	%92,2	%92,2	%92,3
DenseNet121	<b>%98,1</b>	<b>%97,9</b>	<b>%98,1</b>	%97,7	<b>%97</b>	<b>%97</b>	<b>%97</b>	<b>%96,9</b>

#### 4.1. VGG16, InceptionResNetV2 ve DenseNet121 ağları ile COVID-19 Teşhisi Sonuçları

Her iki veri kümesi 1000 sınıf ve yaklaşık 15 milyon görüntüye sahip ImageNet te önceden eğitilmiş derin öğrenme mimarileri olan VGG-16, InceptionResNetV2 ve DenseNet121'de ayrı ayrı eğitilip başarımlar hesaplanmış ve kıyaslanmıştır. 3 derin öğrenme modeli için BT verilerinin eğitimi 9 dk 50 s, X-Ray verilerin eğitimi 8 dk 29 s sürede tamamlanmıştır. Eğitimler Epok=25, batch=32 değerleri ile gerçekleştirilmiştir. Elde edilen sonuçlar Çizelge 2'de verilmiştir. Çizelgeden görüldüğü gibi BT verileriyle VGG-16 eğitimi çalışmasında; eğitim aşamasında 16.813.206, sınamada 512 olmak üzere toplamda 16.813.718 parametre kullanılmıştır. Yapılan başarımlar ölçümleri sonucu %87,5 doğruluk, %87,6 F1, %79 özgülük ve %98,2 tutturma değerleri elde edilmiştir. X-Ray verileriyle VGG-16 eğitimi çalışmasında, eğitim aşamasında 15,895,959, sınamada 512 olmak üzere toplamda 15.896.471 parametre kullanılmıştır. Yapılan başarımlar ölçümleri sonucu %79,1 doğruluk, 76,1 F1, %62,1 Bulma ve %98,3 tutturma değerleri elde edilmiştir.

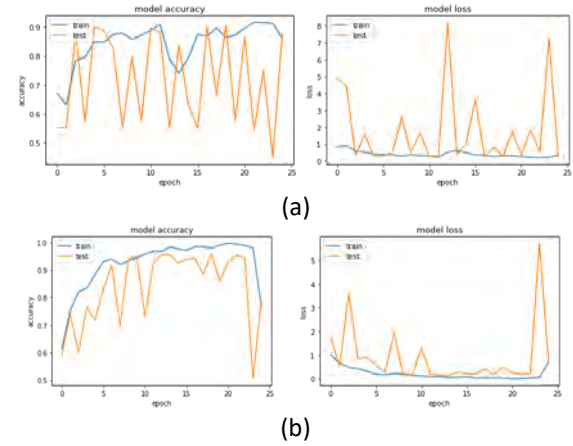
BT verileriyle InceptionResNetV2 eğitimi çalışmasında, eğitim aşamasında 54.673.846, sınamada 64.128 olmak üzere toplamda 54.737.974 parametre kullanılmıştır. Yapılan başarımlar ölçümleri sonucu %94,5 doğruluk, %94,2 F1, %89,5 bulma ve %99,5 tutturma değerleri elde edilmiştir. X-Ray verileriyle InceptionResNetV2 eğitimi çalışmasında, eğitim aşamasında 54.674.103, sınamada 64.128 olmak üzere toplamda 54.738.231 parametre kullanılmıştır. Yapılan başarımlar ölçümleri sonucu %92,1 doğruluk, %92,2 F1, %92,2 bulma ve %92,3 tutturma değerleri elde edilmiştir.

BT verileriyle DenseNet121 eğitimi çalışmasında, eğitim aşamasında 7.219.414, sınamada 86.208 olmak üzere toplamda 57.305.622 parametre kullanılmıştır. Yapılan başarımlar ölçümleri sonucu %98,1 doğruluk, %97,9 F1, %98,1 bulma ve %97,7 tutturma değerleri elde edilmiştir. X-Ray verileriyle DenseNet121 eğitimi çalışmasında, eğitim aşamasında 7.219.671, testte 86.208 olmak üzere toplamda 7.305.879 parametre kullanılmıştır. Yapılan başarımlar ölçümleri sonucu %97 doğruluk, %97 F1, %97 bulma ve %96,9 tutturma değerleri elde edilmiştir.

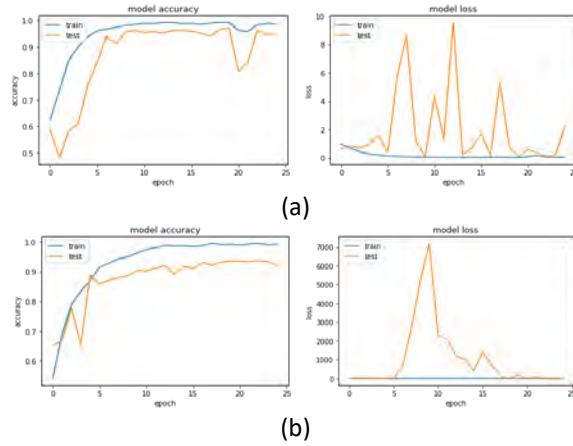
VGG-16, InceptionResNetV2 ve DenseNet121 ağlarının BT ve X-Ray görüntüleri üzerindeki başarımlar grafikleri sırasıyla Şekil 8, 9 ve 10 da verilmiştir.

BT verilerle yapılan deneylerde DenseNet121 mimarisi doğruluk, F1 ve bulma ölçütlerinde, InceptionResNetV2

mimarisi tutturma ölçütünde en yüksek başarımlar değerlerini sağlamıştır. X-Ray görüntülerde ise DenseNet121 mimarisi tüm ölçütlerde en yüksek başarımlara sahiptir.

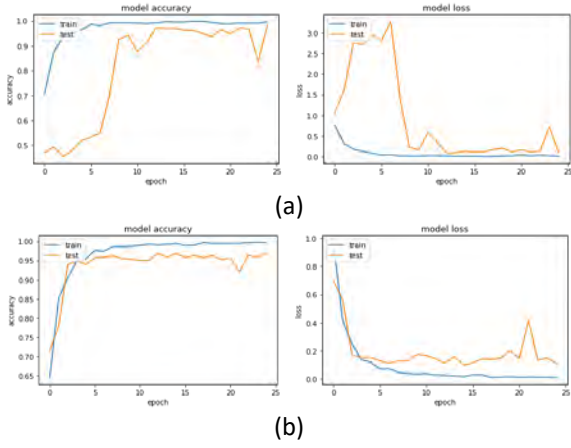


**Şekil-8: VGG-16 Modelinin (a) BT ve (b) X-Ray verileriyle başarımlar grafiği**



**Şekil-9: InceptionResNetV2 Modelinin (a) BT ve (b) X-Ray verileriyle başarımlar grafiği**





Şekil -10: DenseNet121 (a) BT ve (b) X-Ray verileriyle başarımlar grafiği

#### 4.2. Derin Özniteliklerin kNN ve SVM ile Sınıflandırılması ile COVID-19 Tanısı Sonuçları

Çalışmanın ikinci aşamasında, VGG16, InceptionResnetV2 ve Densenet121 ağlarının sırasıyla “fc6”, “predictions”, ve “fc1000” olmak üzere tam bağlı katmanları kullanılarak derin özellikler çıkarılmıştır. Daha sonra, bu derin özellikler, kNN ve SVM sınıflandırıcı yöntemlerinin girişine verilmiştir. BT verilerinin eğitimi 16 dk 22 s, X-Ray verilerin eğitimi 11 dk 57 s sürede tamamlanmıştır. Elde edilen sonuçlar Çizelge 3’te verilmiştir. BT verileriyle yapılan çalışmada başarımlar ölçümlerinde; doğruluk sonuçlarında %97,9, F1 sonuçlarında %97,7 ve bulma sonuçlarında %99 ile DenseNet121’den elde edilen özniteliklerin kNN ile sınıflandırılması en yüksek başarımlar elde edilmiştir. Tutturma sonucunda ise %96,8 ile

VGG-16’den elde edilen özniteliklerin kNN ile sınıflandırılması en yüksek başarımlara sahiptir. X-Ray verileriyle yapılan çalışmada ise, doğruluk sonuçlarında %90,5, F1 sonuçlarında %90,4, bulma sonuçlarında %90,7 ve tutturma sonucunda %90 ile InceptionResNetV2 ağından elde edilen özniteliklerin SVM ile sınıflandırılması en yüksek başarımları vermiştir. Her iki veri kümesinin sonuçlarından, BT verileriyle elde edilen sonuçların X-Ray sonuçlarından yüksek olduğu belirlenmiştir.

#### 4.3. Birleştirilmiş Derin Özniteliklerin kNN ve SVM ile Sınıflandırılması ile COVID-19 Teşhisi Sonuçları

Çalışmanın bu aşamasında her iki veri kümesinde ayrı ayrı VGG16, InceptionResnetV2 ve DenseNet121 ağlarında üretilen öznitelikler birleştirilmiş ve birleşik öznitelik vektörleri kNN ve SVM ile sınıflandırılmıştır. VGG-16’dan 4.096, InceptionResNetV2’den 1.000 ve DenseNet121’den 1.000 olmak üzere birleştirilmiş öznitelik vektörünün boyutu 6.096’dır. Deneysel çalışmanın bu aşamasında sınıma verilerinden sınıflandırma başarımlarının değerlendirilmesinde Karmaşıklık Matrisi’nden yararlanılmıştır. Önerilen VGG-16, InceptionResNetV2 ve DenseNet121 ağlarından elde edilen özniteliklerin kNN ve SVM ile sınıflandırılması sonucu oluşan karmaşıklık matrisleri Şekil 11 ve 12’de verilmiştir. Çizelge 4’te ise birleştirilmiş özniteliklerle elde edilen sonuçlar görülmektedir. Elde edilen tüm sonuçlara göre (Çizelge 5) BT görüntüleri için en yüksek değerler %98,9 doğruluk, %98,8 F1, %99 bulma %98,6 tutturma ile birleştirilmiş özniteliklerin kNN ile sınıflandırılmasında elde edilmiştir. X-Ray görüntülerinde ise DenseNet121 ağı bütün ölçütlerde tek başına daha iyi bir başarımlar göstermektedir.

Çizelge -3 : Derin Özniteliklerin kNN ve SVM ile Sınıflandırılması ile COVID-19 Teşhisi Sonuçları

	BT (COVID 19, Normal)				X-RAY (COVID 19, Normal, Pnömoni)			
	Doğruluk	F1	Bulma	Tutturma	Doğruluk	F1	Bulma	Tutturma
VGG-16+kNN	%97,7	%97,5	%98,1	%96,8	%89,8	%89,8	%90,1	%89,8
VGG-16+SVM	%81,2	%80,7	%75,0	%87,4	%82,8	%82,7	%83,1	%82,8
InceptionResNetV2 +kNN	%95,9	%95,4	%96,3	%94,6	%83,5	%83,5	%83,8	%83,5
Inception ResNet V2 +SVM	%91,3	%90,7	%86,8	%95,0	%90,5	%90,4	%90,7	%90,0
DenseNet121+kNN	%97,9	%97,7	%99,0	%96,4	%87,8	%87,7	%87,0	%87,8
DenseNet121+SVM	%92,3	%91,7	%88,7	%95,0	%89,0	%88,9	%89,1	%89,0

Gerçek Sınıf	KNN	
	COVID19	Normal
	COVID19	272
Normal	3	220
		Tahmini Sınıf

Gerçek Sınıf	SVM	
	COVID19	Normal
	COVID19	240
Normal	13	210
		Tahmini Sınıf

Şekil-11: BT veri kümesi için karmaşıklık matrisleri

Gerçek Sınıf	KNN			Gerçek Sınıf	SVM		
	COVID19	Normal	Pnömoni		COVID 19	Normal	Pnömoni
COVID19	169	6	10	COVID19	151	21	13
Normal	2	194	10	Normal	1	193	2
Pnömoni	35	3	181	Pnömoni	38	0	181
	Tahmini Sınıf				Tahmini Sınıf		

Şekil-12: X-Ray veri kümesinde karmaşıklık matrisleri

Çizelge-4: Öznitelik birleştirilmesi sonucu sınıflandırma başarımları

	BT (COVID 19, Normal)				X-RAY (COVID 19, Normal, Pnömoni)			
	Doğruluk	F1	Bulma	Tutturma	Doğruluk	F1	Bulma	Tutturma
Birleştirilmiş Derin Öznitelikler +kNN	%98,9	%98,8	%99,0	%98,6	%90,6	%90,6	%91,1	%90,6
Birleştirilmiş Derin Öznitelikler +SVM	%90,5	%89,9	%86,0	%94,1	%87,5	%87,4	%87,6	%87,5

Çizelge-5: Çalışmada elde edilen sonuçlar

Veri kümesi	BT (COVID 19, Normal)				X-RAY (COVID 19, Normal, Pnömoni)			
Ağ Mimarisi	Doğruluk	F1	Bulma	Tutturma	Doğruluk	F1	Bulma	Tutturma
DenseNet121	%98,1	%97,9	%98,1	%97,7	<u>%97</u>	<u>%97</u>	<u>%97</u>	<u>%96,9</u>
DenseNet121+kNN	%97,9	%97,7	%99,0	%96,4	%87,8	%87,7	%87,0	%87,8
Birleştirilmiş Derin Öznitelikler +kNN	<u>%98,9</u>	<u>%98,8</u>	<u>%99,0</u>	<u>%98,6</u>	%90,6	%90,6	%91,1	%90,6

## 5. Sonuçlar

Bu çalışmada, COVID-19 vakalarını sınıflandırmak için BT ve X-Ray olmak üzere 2 ayrı veri kümesi ile ayrı ayrı çalışılmış, BT'de 2 ve X-Ray'de 3 sınıflı etiketleme yapılmış, 3 derin öğrenme mimarisi ve 2 makine öğrenmesi sınıflandırıcısı kullanılarak başarımlar karşılaştırılmıştır. BT görüntülerden elde edilen başarımların X-Ray görüntülerde daha yüksek olduğu gözlenmiştir. Bununla birlikte, derin öğrenme mimarileri arasından Densenet121 mimarisi ve makine öğrenmesi sınıflandırıcısından kNN algoritması daha başarılı olmuştur. Son olarak, genel olarak farklı yapıya sahip derin mimarilerden elde edilen derin özelliklerin birleştirilmesi, sınıflandırma başarımlarını artırdığını açıkça göstermiştir.

## Kaynakça

- [1] Bozkurt, F. "Derin Öğrenme Tekniklerini Kullanarak Akciğer X-Ray Görüntülerinden COVID-19 Tespiti", Avrupa Bilim ve Teknoloji Dergisi, no. 24, Apri 2021, pp. 149-156, <https://doi:10.31590/ejosat.898385>
- [2] Erdaş, Ç. B., Detection and differentiation of COVID-19 using deep learning approach fed by x-rays, International Journal of Applied Mathematics, 8(3), 2020, pp. 097-101. <https://dergipark.org.tr/en/download/article-file/1308359>
- [3] Özbay, E., Özbay, F. A., COVID-19 Detection from CT images with Deep Learning and Classification Approaches, DÜMF Mühendislik Dergisi 12(2), 2021, pp. 211-219. <https://dergipark.org.tr/tr/download/article-file/1352635>
- [4] Kutlu, Y., Camgözlü, Y., Detection of coronavirus disease (COVID-19) from X-ray images using deep convolutional neural Networks, Natural and Engineering Sciences, 6(1), 2021, pp. 60-74. <https://doi.org/10.28978/nesciences.868087>
- [5] Güraksın, G. E., COVID-19 Diagnosis Using Deep Learning, Düzce University Journal of Science & Technology, 9, 2021, pp. 8-23. <https://doi.org/10.29130/dubited.866124>
- [6] Hemdan, E. E., A. Shouman, M., Karar, M.E., COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images, Cornell University Electrical Engineering and Systems Science -Image and Video Processing, <https://doi.org/10.48550/arXiv.2003.11055>
- [7] Jia, G., Lam, H. K., Classification of COVID-19 chest X-Ray and CT images using a type of dynamic CNN modification method, Computers in Biology and Medicine, 134, 2921, pp. 104425. <https://doi.org/10.1016/j.compbiomed.2021.104425>
- [8] Şahinbaş, K., Çatak, F. Ö., Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images, Data Science for COVID-19, 2021, pp. 451-466. <https://doi.org/10.1016/B978-0-12-824536-1.00003-4>

- [9] Sethy, P.K., Behera, S.K., Detection of Coronavirus Disease (COVID-19) Based on Deep Features, Preprints, 2020, 2020030300, <https://doi.org/10.20944/preprints202003.0300.v1>
- [10] Asnaoui, K., Chawki, Y., Using X-ray images and deep learning for automated detection of coronavirus disease, Journal of Biomolecular Structure and Dynamics, 39(10), 2021, pp. 3615-3626. <https://doi.org/10.1080/07391102.2020.1767212>
- [11] Ardakani, A.A., Kanafi, A. R., Acharya,U. R. , Khadem,N. ,Mohammadi, A., Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural Networks, Computers in Biology and Medicine, 121, 20220, pp. 103795. <https://doi.org/10.1016/j.combiomed.2020.103795>
- [12] Kart, Ö., Başçiftçi, F., Makine Öğrenmesi Algoritmalarıyla Akciğer Tomografi Görüntülerinden COVID-19 Tespiti, Avrupa Bilim ve Teknoloji Dergisi Özel Sayı, 28, 2021, pp. 630-637. <https://doi.org/10.31590/ejosat.1009611>
- [13] <https://www.kaggle.com/datasets/tawsifurrahman/COVID19-radiography-database>
- [14] <https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset>
- [15] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ArXiv Prepr., 2014, pp. 1409e1556
- [16] [https://www.researchgate.net/figure/Schematic-diagram-of-InceptionResNetV2-model-compressed-view\\_fig9\\_326421398](https://www.researchgate.net/figure/Schematic-diagram-of-InceptionResNetV2-model-compressed-view_fig9_326421398)
- [17] Huang, G. Liu, Z. Maaten L. Van Der, Weinberger, K.Q. Densely connected convolutional networks,in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700e4708

# Görüntülerden veya Çizimlerden Otomatik Kot Oluşturma Teknikleri: Bir Derleme Çalışması

## Automatic Code Generation Techniques from Images or Sketches: A Review Study

Musa Selman KUNDURACI  
Bursa Teknik Üniversitesi  
Bilgisayar Mühendisliği  
Bursa, Türkiye  
mskunduraci@gmail.com  
ORCID: 0000-0001-9823-3387

Turgay Tugay BİLGİN  
Bursa Teknik Üniversitesi  
Bilgisayar Mühendisliği  
Bursa, Türkiye  
turgay.bilgin@btu.edu.tr  
ORCID: 0000-0002-9245-5728

### Öz

Bir yazılımın geliştirilmesi sürecinde, tasarım ve öncül üretim en önemli ve zaman alıcı aşamalardır. Kullanıcılar yazılımların görsel arayüzlerine ve tasarımlarına oldukça önem vermektedir. İyi bir görsel arayüz tasarımına sahip bir yazılım daha iyi işleve sahip olup fakat arayüzü kullanışsız olan benzerinden daha fazla tercih edilmektedir. Görsel arayüz tasarımı sürecinde geliştiriciler öncelikle kâğıt üzerinde tasarım gerçekleştirip ardından görsel arayüz tasarım programları ile dijital tasarıma dönüştürürler. Sonraki aşamada, tasarımın çeşitli biçimlendirme dilleriyle (xml, html, css vb.) veya doğrudan programlama dilleriyle kodlanması gerekmektedir. Otomatik kot üretme yaklaşımlarında amaç minimum yazılım geliştirici maliyeti ile kısa zamanda verimli ve hızlı uygulamalar geliştirmektir. Bu çalışmada, çeşitli yöntemleri kullanarak otomatik kot üretimi gerçekleştiren çalışmalarını içeren geniş bir yayın taraması oluşturulmuştur. İncelenen makalelerde çoğunlukla derin öğrenme, görüntü işleme, yapay sinir ağları veya makine öğrenmesi yöntemleri kullanılmıştır. Bu derleme çalışması ile bu alanda çalışma yapacak araştırmacılara rehber olunması amaçlanmıştır.

**Anahtar sözcükler:** Otomatik Kot Üretimi, Derin Öğrenme, Makine Öğrenmesi

### Abstract

In the process of developing a software, design and prototyping are the most important and time-consuming stages. Users attach great importance to the visual interfaces and designs of the software. A software with a good visual interface design is preferred more than a similar one with better functionality but an unusable interface. In the process of visual interface design, developers first design on paper and

then turn it into digital design with visual interface design progra

ms. In the next step, the design needs to be coded with various markup languages (xml, html, css etc.) or directly with programming languages. The aim of automatic code generation approaches is to develop efficient and fast applications in a short time with minimum software developer cost. In this study, a large literature review was created that includes studies that perform automatic code generation using various methods. In the reviewed articles, mostly deep learning, image processing, artificial neural networks or machine learning methods were used. With this review study, it is aimed to guide researchers who will work in this field.

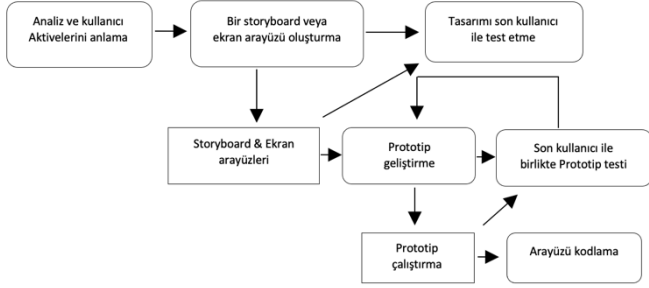
**Keywords:** Automatic Code generation, Deep Learning, Machine Learning

### 1. Giriş

Kullanıcı arayüzü diğer bir adıyla kullanıcı arabirimi, kullanıcıların bir makine veya cihazı kullanabilmelerine olanak sağlayan yöntemlerin bileşkesi olarak tanımlanabilir. Kullanıcı arayüzü sayesinde kullanıcılar bir sistemin işleyişini değiştirebilir veya sistem, kullanıcının yaptığı değişikliklerin sonuçlarını üretebilir. Görsel arayüz tasarımı bilgisayarlardan akıllı saatlere, arabalardan uçaklara kadar birçok üründe kullanılmaktadır. Görsel arayüz tasarımlarının karmaşıklığı arttıkça kullanıcıların onu kullanabilmeleri için gerekli bilgi ve beceri ihtiyacı da artmaktadır. Bu nesnelere göre Android Tasarlanan arayüzü kullanmak için pilotluk eğitimi gerekirken, akıllı saati kullanmak için kullanım kılavuzunu okumak yeterli olacaktır.

Kullanıcı arayüz tasarımı süreci Şekil-1'de gösterilmiştir. Kullanıcı arayüz tasarımında ilk aşama Analiz ve Kullanıcı Aktivelerini anlamadır. Kullanıcıların arayüz kullanım alışkanlıkları, benzer uygulamaların arayüzlerinin analizi yapılarak kullanıcıların uygulamayı en iyi şekilde nasıl

kullanabileceği anlaşılmalı çalışılır. Tasarlanan arayüzlerin nasıl bir sıra veya olay ile görüntüleneceği bir sonraki aşamada gerçekleştirilir. Bu aşamadan sonra son kullanıcı tarafından test edilir ve ekran arayüzleri için aynı işlemler tekrarlanır. Ekran arayüz sayısı birden fazla ise bir prototip geliştirilerek test edilir. Eğer prototipin çalışmasında bir problem yoksa arayüz kodlanabilir.



**Şekil-1:** Kullanıcı Arayüzü Tasarım süreci [1]

İyi bir kullanıcı arayüzü kullanıcıların yapacağı işleri kolaylaştırılmalıdır. İyi bir arayüz tasarımı için de uzun bir arayüz tasarım süreci gerekebilir. Arayüz tasarım süreçlerinin uzaması yazılım şirketleri için maliyetlerin artması anlamına gelmektedir [1]. GUI çizimleri veya eski çalışmalarının koda dönüştürülmesi programcılar tarafından yaygın olarak gerçekleştirilen bir görevdir. Bu görevin gerektirdiği zaman nedeniyle uygulamanın arka plan kodlarına daha az zaman kalmaktadır. Bu durum, programın istenilen kalitede olamamasına hatta istenilen sürede tamamlanamamasına yol açabilmektedir.

Serbest el kullanıcı arabirimi (UI) çizimini sıkı bir şekilde entegre etmek, günümüzün yazılım geliştirme sürecinde önemli bir boşluğu dolduracaktır. Özellikle, kağıt tabanlı prototipler üzerinde yineleme yaptıktan sonra, bugün UI tasarımcıları prototipleri Photoshop veya bir IDE gibi zahmetli ve maliyetli (GUI oluşturuculara bile) "yüksek doğrulukta" araçlarda manuel olarak yeniden oluşturmak zorundadır. Bu boşluk aynı zamanda UI tasarımcılarının ekiplerinin sonunda ürettiği koddan ayırır ve bu da genellikle programlama yapılarında uygulanması zor olan UI tasarımlarına yol açar [2].

Bunun önüne geçmek için bu çizimlerin makine öğrenmesi, yapay zeka veya görüntü işleme teknikleri ile arayüz koduna otomatik olarak dönüştürülmesi, bu aşamada harcanacak sürenin minimuma inmesini sağlayacaktır.

Son yıllarda bu alanda yapılan çalışmaların artması bu süreci daha da hızlandıracak ve doğruluk oranını artıracaktır. Farklı yöntemlerin denenerek diğer yöntemlere göre avantaj ve dezavantajların belirlenmesi, bu konuda çalışma yapacak araştırmacılar yol gösterecektir. Bu konuda yapılacak çalışmaların artması karmaşık arayüz tasarımlarının bile rahatlıkla arayüz koduna dönüştürülmesi sağlayarak son kullanıcı açısından daha kullanışlı programların ortaya çıkmasıyla neticelenebilir.

## 2. Otomatik Kod Oluşturma için Kullanılan Yöntem ve Teknikler

### 2.1 Makine Öğrenmesi

Makine öğrenmesinin tanımına bakıldığında; bir bilgisayarın doğrudan yönergeler olmadan öğrenmesine yardımcı olmak için matematiksel modelleri kullanma biçimi olduğu ifade edilmektedir. Makine öğrenmesi esasında yapay zekanın bir alt kümesi olarak değerlendirilir [3]. Yapay zeka insan zekasını taklit eden sistemler veya makineler olarak tanımlanabilir. Makine öğrenimi ve yapay zeka genellikle bir arada değerlendirilir. Kimi durumlarda birbirinin yerine kullanılır ancak aynı anlama gelmezler. Tüm makine öğrenimi çözümleri yapay zekâ iken tüm yapay zeka çözümlerinin makine öğrenimi olmaması önemli bir ayrımdır [4].

Denetimli Makine Öğrenimi ve Denetimsiz Makine Öğretimi olmak üzere iki tür makine öğretimi vardır. Denetimli makine öğrenimine lojistik regresyon, destek vektör makineleri (SVM) ve çoklu sınıf sınıflandırma gibi algoritmalar örnek verilebilir [4].

Denetimsiz Makine Öğreniminde ise belirli bir etikete sahip olmayan verilerin sınıflandırılması veya kümelenmesi durumu vardır. K-NN gibi algoritmalara buna örnek verilebilir [4].

### 2.2 Derin Öğrenme

Derin öğrenme bir veya daha fazla gizli katman içeren yapay sinir ağları ve benzeri makine öğrenme algoritmalarını kapsayan çalışma alanıdır. Derin öğrenme, özünde, makinelerle insan zekasını taklit etmeyi öğretmek için yinelemeli yöntemlere dayanır. Yapay bir sinir ağı, bu yinelemeli yöntemi birkaç hiyerarşik düzey aracılığıyla gerçekleştirir. İlk seviyeler, nöronların basit bilgileri öğrenmesine yardımcı olur ve seviyeler arttıkça bilgi birikmeye devam eder. Her yeni seviye ile nöronlar daha fazla bilgi toplar ve bunları son seviyede öğrendikleriyle birleştirir. Sürecin sonunda, elde edilen bilgi bir nevi mantıksal düşünce örneği gibidir. Derin öğrenme yöntemleri kullanılarak tıbbi cihazlardan elde edilen görüntülerden hastalık tespiti, araçlarda plaka okumanın yanında marka model tespiti, hayvan, böcek, bitki, kuş ya da mikro düzeyde canlıların sınıflandırılması gerçekleştirilebilir [5][6].

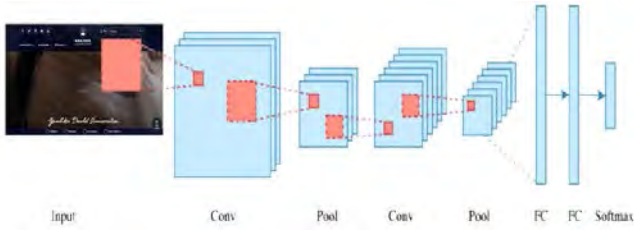
#### 2.2.1 Evrimsel Sinir ağları

Evrimsel sinir ağları (Convolutional neural network - CNN), derin öğrenmenin bir alt dalıdır ve genellikle görsel bilginin analiz edilmesinde kullanılır. CNN, convolution ve pooling operatörlerini kullanır.

Bir CNN üç temel katman türüne sahiptir:

- Convolutional katmanı
- Pooling katmanı
- Fully-connected katmanı

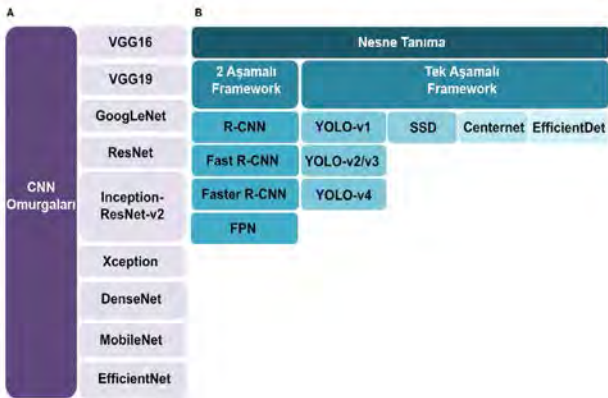
Art arda çok sayıda convolution+pooling katmanı olabilir. Bunu takiben birkaç tane fully connected katman bulunur. Çok etiketli sınıflandırma problemlerinde, en sonda softmax katmanı bulunur.



Şekil-2: Evrişimsel sinir ağı yapısı [7]

Şekil-2’de görüldüğü gibi pooling katmanının görevi, kayma boyutunu ve ağı içindeki parametreleri ve hesaplama sayısını azaltmaktır. Bu sayede ağıdaki uyumsuzluk kontrol edilmiş olur. Fully-connected katmanı üç boyutlu girişi tek boyuta indirgeyerek alır ve bir sınıf etiketi elde eder. Softmax katmanı çikış sınıflarının olasılık dağılımını hesaplar.

CNN Omurgaları ve Nesne Tanıma Algoritmaları Şekil-3’de gösterilmiştir. A kısmında yer alanlar CNN omurgalarıdır. Bunlar Nesne tanıma özellik çıkarımı elde etmek için kullanılabilirler. B kısmında yer alanlar ise Nesne Tanıma Algoritmalarıdır. Bunlar iki şekilde ele alınabilir. İki aşamalı çözümlere bakıldığında R-CNN, Fast R-CNN, Faster R-CNN ve FPN olduğu görülmektedir. Bu framework’ler hız açısından tek aşamalı çözümlere göre daha yavaş olsa da doğruluk açısından daha başarılı olabilmektedir [8]. Tek aşamalı çözümler ise YOLO versiyonlarını [9], SSD [10], CenterNet ve EfficientDet vardır [11]. Bunlar hızlı çalışırlar fakat doğruluk oranları daha düşük olabilmektedir.

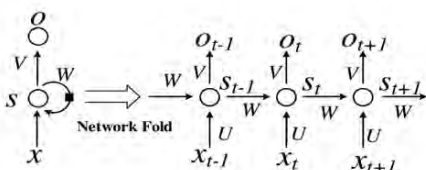


Şekil-3: Evrişimsel sinir ağı omurgaları [12]

Makaleler incelendiğinde CNN omurga ve algoritmalarının kullanıcı arayüz tasarım ya da çizimlerinden nesne tespiti yapmak ya da sınıflandırma amacıyla kullanıldığı görülmektedir.

### 2.2.2 Yinelemeli Sinir ağıları (RNN)

Yinelemeli Sinir Ağıları (Recurrent Neural Network=RNN) bir sonraki adımı tespit etmek için kullanılan Derin Öğrenme yapılarıdır. Diğer derin öğrenme yapılarından farkı “hatırlama” yapısıdır. Yani diğer sinir ağılarında her girdi



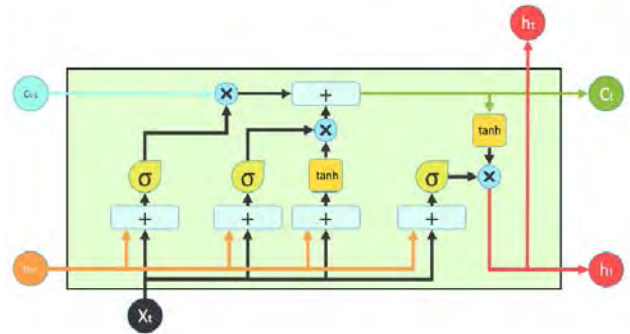
Şekil-4: Temel bir Yinelemeli sinir ağı yapısı [14]

birbirinden bağımsız iken RNN’lerde her girdi birbiri ile ilişkilidir. Bir sonraki adımı tahmin etmek için girdiler arasındaki ilişkiyi kullanır[13].

RNN yapıları kullanıcıların el yazılarını, görüntü, konuşma tanıma gibi birçok alanda kullanılmaktadır.

### 2.2.3 Uzun Kısa Süreli Bellek (LSTM)

Yinelemeli Sinir Ağlarının kısa süreli olan bellekleri bazı problemlerde iyi sonuç vermemektedir. Bu sorunu oradan kaldırmak için araştırmacılar Sepp Hochreiter ve Jürgen Schmidhuber LSTM modelini önermişlerdir. Bir RNN yapısında tek bir katman yer alırken LSTM’nin yapısında birbiriyle iletişim halinde olan 4 katman (Unutma Kapısı-Giriş Kapısı-Çıkış Kapısı-Hücre Durumu) bulunmaktadır[15].



Şekil-5: Uzun Kısa Süreli Bellek Sinir Ağı'nın (LSTM) yapısı [16].

Unutma yapısı hangi bilgilerin unutulacağı veya tutulacağına karar veren kapıdır. Giriş kapısı Hücre Durumu güncellemesi yapar. Çıkış kapısı ise sonraki girişteki veriyi belirler. Bu aşamadan sonra hücre durumuna karar verilir.

## 3. Araştırma Yöntemi

Bu derleme makalesinde otomatik kod oluşturma teknikleri ile ilgili bilimsel çalışmalar geniş bir yayın taraması ile tespit edilmiş ve bu çalışmaların analizi yapılmıştır. Makale ve bildiriler, bilimsel makale arama motorlarında ve veri tabanlarında önceden belirlenen anahtar sözcüklere göre araştırılmıştır. Bu anahtar sözcükler, eş anlamları ile birlikte Çizelge 1’de verilmiştir.

Çizelge-1: Anahtar Sözcükler

Anahtar Sözcükler	Eş anlamlıları
Sketch	Çizim, mockup, screenshot
Deep learning	CNN, derin öğrenme
User interface	UI, GUI
Machine learning	Makine öğrenmesi
Software	App, mobile, web, css, html, android, ios
Code generation	Automatic code generation, Kot üretimi, otomatik kod üretimi

Çalışma kapsamında taranan veri tabanları; Google Scholar, ACM Digital Library, Scopus, IEEE Xplore Digital Library ve arXiv.org olarak belirlenmiştir. Veri tabanlarında yukarıdaki

çizelgede belirtilen anahtar sözcükleri barındıran çalışmalardan, 2017 ve sonrasında yayınlananlar ayıklanmıştır.

### 3.1 Derleme Çalışmasının Amacı

Bu derleme çalışması ile aşağıdaki araştırma sorularının cevabı aranmaktadır;

**Q1.** Görüntüden otomatik kod oluşturan çalışmalarda hangi yaklaşımlar mevcuttur?

**Q2.** Kullanılan yaklaşımlar ne tür bir platform ve girdi/çıkıtları verileri içindir?

**Q3.** Hangi veri setleri kullanılmıştır?

**Q4.** Yaklaşımlarda kullanılan yöntemlerin karakteristikleri nelerdir?

**Q5.** Çalışmaların başarımları nasıldır?

Bu araştırma sorularına göre çalışmalar incelenmiş ve analizi yapılmıştır.

## 4. Bulgular ve Tartışma

Bu bölümde incelenen makalelerin özeti, giriş-çıkış karakteristikleri, teknik analizleri, veri kümesi analizleri, başarımlar ölçütlerine göre analizleri yapılmıştır.

### 4.1 İncelenen Çalışmalar

Otomatik kod üretimi ile ilgili son 5 yıldaki çalışmalar incelenmiştir. Çalışmalarda farklı veri setleri, yöntemler ve ölçünler kullanılmıştır. Bu bölümde bu çalışmalarda farklılıklar ortaya çıkarılmaya çalışılmıştır. Ayrıca incelenen çalışmaların benzer, ortak noktaları belirlenerek çalışmaların yönelimi gösterilmiştir.

İncelenen çalışmalar giriş yöntemine göre gruplandırılarak incelenmiştir. Giriş yöntemi olarak; Eskiz, Çizim, GUI Tasarımı, Ekran Görüntüsü veya Mockup görüntüsü kullananlar olarak gruplandırılmıştır.

#### 4.1.1 Eskiz veya Çizim Kullanan Çalışmalar

Bir yazılım arayüzünün oluşturulmasındaki ilk aşama kağıt çizimi ya da eskiz olarak nitelendirilebilir. Çünkü arayüz tasarımı oluşturulurken fikir değişiklikleri çok sık olmaktadır. Ayrıca kağıt ve kalem her an her yerde bulunabilecek materyaller olduğundan kolaylıkla herkes istenilen bir zamanda, herhangi bir program kullanma bilgisi olmasa bile oluşturabilir[17]. Yapılan çalışmaların amaçları incelendiğinde birçok çalışmada amacın eskiz tasarımından kod kısmına olan süreci hızlandırmaya yönelik olduğu görülmektedir. Bazı çalışmalarda ise eğitimde arayüz tasarımını öğretilmesine yönelik olduğu görülmektedir [18]. Eskiz veya çizim kullanan çalışmalara bakıldığında; bu çalışmalar ürettikleri çıktı kodlarına göre 2 alt grupta inceleyebildiği görülmüştür. Bu alt gruplar Web kodu üretenler ve Mobil ya da Android kodu üretenler olarak tespit edilmiştir.

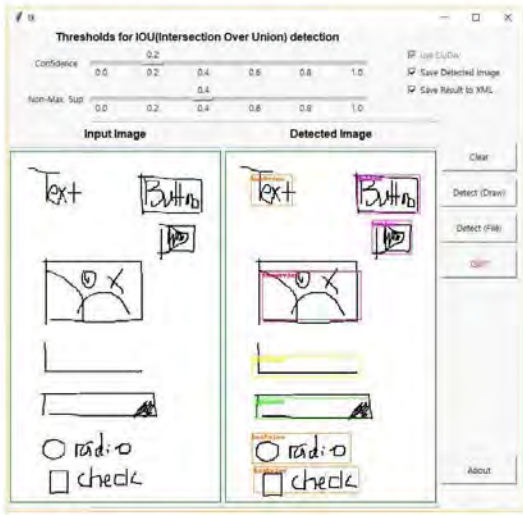
Han ve ark. 2018 yılındaki CSSSketch2Code adını verdikleri çalışmalarında çizimden otomatik HTML kodu üreten bir çalışma gerçekleştirmişlerdir. Bu çalışmada diğerlerinden

farklı olarak CSS kodu da üretilmiştir [19]. Şekil-6'da CSSSketch2Code çalışmasının girdi resmi olan el çizimi ve çıkış olarak koda dönüştürdüğü hali görülmektedir. Aşıroğlu ve ark. 2019 yılındaki çalışmalarında makine öğrenmesi yöntemlerini kullanarak Çizimden otomatik Web kodu üreten bir çalışma gerçekleştirmişlerdir. Çalışmada elle çizilen tasarımdan Web öğeleri tespit edildikten sonra HTML kodu üretilip konumuna göre yerleştirme işlemi gerçekleştirilmektedir[20]. 2022 yılındaki bir başka çalışmada kullanıcıların çizdiği web sitesi şablonunu Derin öğrenme çerçevelerinden VGG kullanarak HTML ve CSS koduna çeviren bir yaklaşım uygulanmıştır. Bu çalışmada özellikle web sitelerinin navbar kısımlarının koda dönüştürülmesi üzerinde durulmuştur [21]. Vitkare ve ark. elle çizilen bir arayüz tasarımını otomatik HTML koduna çeviren bir çalışma yapmışlardır. Bu çalışmada görüntü işleme teknikleri ve CNN modeli kullanmışlardır. Veri kümesi olarak Pix2Code veri kümesini kullanmışlardır[22].



**Şekil-6:** Han ve ark. CSSSketch2Code adını verdikleri çalışmada çizimden koda dönüştürme [19].

Adefris 2020 yılında gerçekleştirdiği çalışmasında derin öğrenme yöntemleri kullanarak düşük doğruluk oranına sahip grafik arayüz çizimlerinden otomatik kod oluşturan bir yaklaşım sunmuştur [23]. Yun ve ark. 2018 yılındaki çalışmalarında derin ağ yaklaşımlarını kullanarak çizimleri otomatik olarak Android uygulamaların arayüz tasarım dili olan XML'e çevirmişlerdir [24]. Yun ve ark. 2019 yılında bir başka çalışmada önceki çalışmalarını geliştirmişlerdir [25]. Mohian ve Csaallner 2020 yılındaki çalışmalarında da Çizimden otomatik Android kodu üretmişlerdir. Doodle2App adını verdikleri bu yazılımda derin öğrenme yöntemleri kullanmışlardır [2]. Jisu Park ve ark. da 2020 yılında diğer çalışmalar gibi Çizimden otomatik Android kodu üreten bir çalışma gerçekleştirmişlerdir [26]. 2020 yılındaki diğer bir çalışma olan Rahmadi ve Sudaryanto'nun çalışmalarında Derin Öğrenme Tekniği Kullanılarak Grafik Kullanıcı Arayüzü Bileşenlerinin Görsel Olarak Tanınması gerçekleştirilmiştir [27]. Jain ve ark. 2019 yılındaki çalışmalarında kendi oluşturdukları veri kümesi ile derin öğrenme yöntemlerinden ResNet50 kullanarak çizimden mobil uyumlu web koda dönüştüren bir yazılım geliştirmişlerdir [28].



**Şekil-7:** Yun ve ark. 2018 yılındaki çalışmalarından bir görüntü [24]

Birçok çalışmada derin öğrenme yöntemleri kullanılırken görüntü işleme ve OCR kullanan çalışmalar da bulunmaktadır. Bu çalışmalardan S. Kim ve arkadaşlarının çalışmasında çizimler üzerinde kenar belirleme ve OCR gibi teknikler kullanılarak nesne tespiti yapılmıştır ve bu nesnelere göre Android Arayüz XML kodu üretilmiştir [29]. Ge 2019 yılında Elle çizilen eskizleri derin öğrenme yöntemleri kullanarak eskize en yakın GUI görüntüsünün kodunu oluşturan bir çalışma gerçekleştirmiştir. Çalışmada Android arayüz tasarımlarında kullanılan 7 widget, geliştirilen yöntem ile tespit edilip kodu üretilmektedir [30]. Baule ve ark. eskiz görüntülerinden otomatik Android uygulama tasarım koduna çeviren bir çalışma yapmışlardır. Sketch2aia adını verdikleri yazılım çizimleri Applinventor platformunda bir kullanıcı arayüzüne dönüştürmektedir. Yaptıkları çalışmanın amacını K-12 düzeyindeki öğrencilerin mobil uygulama öğrenmesini kolaylaştırmak olarak bildirmişlerdir. Yöntem olarak derin öğrenme modellerinden Yolo-Darknet kullanmışlardır. Yaptıkları testlerde yaklaşık %87 sınıflandırma başarısı elde etmişlerdir[18]. Jadhav, Gaikwad ve Gawande elle çizilen arayüz çizimlerinden DSL kodu üreten bir çalışma gerçekleştirmişlerdir.

#### 4.1.2 GUI Tasarımı ya da Ekran Görüntüsü Kullanan Çalışmalar

Graphical User Interface (GUI) bir yazılımın ekran görüntüsü ya da bir tasarımcının bir yazılım için oluşturduğu arayüz tasarımı olarak adlandırılabilir. GUI kullanıcılar ile etkileşim açısından büyük öneme sahiptir[31]. Tasarımcılar da bunu dikkate alarak kullanıcıların kolay anlayabileceği ve kullanışlı arayüzler oluşturmaya çalışırlar[32]. Bununla birlikte kullanıcıların belirli arayüz alışkanlıkları da olabilmektedir. Yazılım firmaları da bu alışkanlıkları da göz önünde bulundurarak özgün bir tasarım geliştirmeye çalışırlar[33]. Bu sürecin daha hızlı ilerleyebilmesi için mevcut uygulamaların GUI tasarımı veya ekran görüntülerinden otomatik bir tasarım oluşturma yaklaşımı ortaya çıkmıştır. Bu alanda çalışma yapanlardan; Pang ve ark. 2020 yılındaki çalışmalarında giriş olarak GUI tasarımı kullanarak dikkate dayalı derin sinir ağı ile yeni bir söz dizimine duyarlı otomatik grafik kodu oluşturan

SGui2Code ve HGui2Code adında iki model sunmuşlardır. Bu modellerde LSTM ve CNN kullanmışlardır. Görüntü tespiti için CNN kullanılırken, DSL kodunu çözmek için LSTM modelini kullanmışlardır [34]. DSL kodu üreten diğer bir çalışma Yang Liu vd.'nin 2019 yılındaki çalışmasıdır. Bu çalışmada Mobil uygulamalar için otomatik platform bağımsız GUI kodu üreten bir yazılım geliştirmişlerdir [35]. C. Chen ve ark. 2018 yılında benzer şekilde GUI tasarım resminden GUI DSL iskeletine çeviren ve sinir ağlarını kullanan bir uygulama gerçekleştirmişlerdir [36]. 2019 yılında ise bu çalışmalarını geliştirerek Gallery D.C. adını verdikleri bir yazılım ile tasarımı algılayıp otomatik arama gerçekleştiren bir çalışma gerçekleştirmişlerdir [37]. Xiao ve ark. IconIntent adını verdikleri, Android Uygulamaları için Simge Sınıflandırmasına dayalı olarak Hassas UI Widget'larının Otomatik Tanımlanması çalışmasını gerçekleştirmişlerdir [38]

Sethi, Kumar ve Swami 2019 yılında Mix-NLP kullanarak tema algılama ve otomatik web kodu oluşturan bir çalışma gerçekleştirmişlerdir [39]. Benzer bir çalışma da Kolthoff, müşterinin istediği tasarımı NLP teknikleriyle analiz edip veri kümesindeki GUI tasarımlarında derin öğrenme teknikleri ile yeni bir GUI öncül tasarım oluşturan bir sistem geliştirmiştir [40]. Nguyen ve arkadaşları 2018 yılında Mobil uygulamaların derin öğrenme kullanıcı arayüzü tasarım kalıpları sunan bir çalışma gerçekleştirmişlerdir [41]. Mobil uygulama arayüz kodu dönüşümü gerçekleştiren diğer bir çalışmada Yun ve ark. CNN kullanarak GUI öğelerinin tespitini yapmışlardır [25]. Bir başka çalışmada J.Chen ve ark. Derin Öğrenme ve Bilgisayarlı görü teknikleri karşılaştırılarak kendi yöntemlerini sunmuşlardır [42]. Xie ve ark. 2020 yılındaki çalışmalarında CNN nesne tespiti ile GUI ekran görüntüsündeki nesnelere algılayıp bunları Android XML koduna çeviren bir çalışma ortaya koymuşlardır [43]. Mohian ve Csallner 2022 yılındaki çalışmalarında benzer olarak derin öğrenme yöntemleri kullanarak GUI ekran görüntüsünden Android arayüz kodu elde eden bir yaklaşım sunmuşlardır [44]. Chen ve ark. ise dikkate dayalı kodlayıcı-kod çözücü modeli aracılığıyla bir grafik kullanıcı arabiriminden kod oluşturma yaklaşımı sunmuşlardır. Giriş verisi olarak verilen mobil uygulamanın arayüz tasarımı ya da bir web sitesinin tasarım görüntüsü kullanılmaktadır. Bu giriş görüntüsü VGG-16 derin öğrenme modeli kullanılarak DSL kodu oluşturulmaktadır. Yapılan testler sonucunda hata oranları karşılaştırılarak benzer çalışmalardan web, Android ve iOS için de daha başarılı olduklarını göstermişlerdir [45]. Saravanan ve ark. web görüntüleri üzerinden HTML kodu üreten bir çalışma gerçekleştirmişlerdir. Bu çalışmada LSTM, Canny kenar tespiti, CNN VGG-19 yöntemlerini birleştiren hibrit bir model ortaya koymuşlardır[46]. Zhao ve ark. GUIGAN adını verdikleri çalışmada genetik algoritmalarını kullanarak otomatik GUI tasarımı oluşturmuşlardır. Fakat bu tasarımı koda dönüştüren bir yapı kurmamışlardır [47]. Xu ve ark. web arayüz resimlerinden otomatik HTML, CSS ve emmet kodu üreten bir çalışma yapmışlardır. Bu çalışmada derin öğrenme obje tespiti ile web arayüzündeki bileşenleri tespit etmişlerdir. Derin öğrenmede; CNN modeli olarak Faster R-CNN, RNN modeli olarak LSTM kullanmışlardır. Çalışmada image2emmet ile resimden emmet koduna dönüşüm işlemi yapılırken, image2html ile ise resimden HTML ve CSS koduna



dönüştürme yapılmaktadır. Diğer çalışmada farklı olarak kod oluşturucunun doğruluğunu sınıflandırmak için NLP yöntemleri kullanmışlardır. Oluşturulan kod ile olması gereken kod arasındaki benzerlik oranını ölçmek için Terim frekans-Tersine Doküman Frekansı (TF-IDF) hesaplamışlardır [48]. Wu ve ark. tersine mühendislik ile UI görüntülerini ayrıştırma çalışması yapmışlardır. Bu çalışmada UI benzerlik araması, erişilebilirlik geliştirmesi ve UI'den kod oluşturma olmak üzere 3 uygulama geliştirmişlerdir. UI bileşenlerinin tespiti için Faster R-CNN modelini ResNet-50 algoritmasıyla birlikte kullanmışlardır. Çalışmada iOS uygulamaların ekran görüntülerini barındıran AMP veri kümesini ve Android uygulamaların ekran görüntülerini barındıran RICO veri kümesini kullanmışlardır [49].

#### 4.1.3 Mockup Kullanan Çalışmalar

Mockup, bir servisin, bir ürünün veya bir arayüzün duylara hitap edecek şekilde modellenip, sinanabildiği ya da sunulabildiği bir modelleme yöntemidir. Bir başka tanımla mockup, yapılan tasarımın görsel olarak zenginleştirildiği ve statik bir şekilde sunulduğu bir tasarım modelleme yöntemidir. Mockup görüntüden otomatik kod oluşturan çalışmalara bakıldığında Moran, Li ve ark. 2018 yılındaki çalışmalarında uygulama marketlerindeki mockup görüntülerinden Android'te uygulama tasarımındaki kullanılan öğeleri tespit ederek hiyerarşik GUI iskeletine çeviren bir çalışma gerçekleştirmişlerdir [50]. Moran ve arkadaşları 2018 yılındaki başka bir çalışmalarında önceki çalışmalarında kullandıkları yapıyı tasarım ihlallerini tespit etmek ve raporlamak için kullanmışlardır [51]. Çıkış olarak Android kodu üreten bir başka çalışmada Abdelhamid ve arkadaşları derin öğrenme çerçevelerinden YoloV5 kullanarak nesne tespiti gerçekleştirmişler ve Android GUI kodu üretmişlerdir [52].

Giriş görüntüsü olarak Mockup kullanan çalışmalardan Bouças ve Esteves'in 2020 yılındaki çalışmalarında Derin Öğrenme ile bu mockup web görüntülerini HTML vs CSS

**Çizelge-2: Çalışmaların Girdi- Çıktı Karakteristikleri**

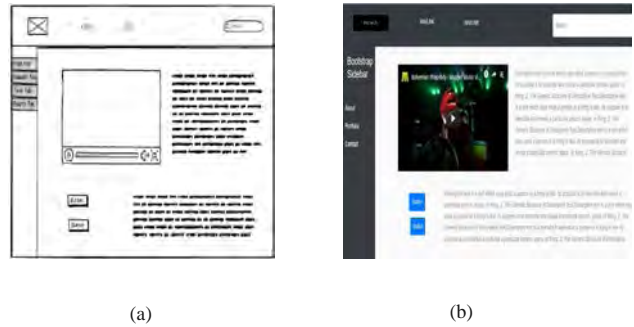
Referans	Platform	Girdi	Çıktı	Algılanan GUI öğeleri
[34]	Mobil	UI tasarımı	DSL kodu	-
[30]	Android	Çizim	Hiyerarşik GUI iskeleti	TextView, EditText, ImageView, Button, RadioButton, Switch ve Chechbox
[35]	Android, iOS ve Web	GUI ekran görüntüsü	İşaretleme benzeri DSL	-
[50]	Mobil	Mockup	Hiyerarşik GUI iskeleti	TextView, EditText, ImageView, Button, RadioButton, Switch ve Chechbox, RatingBar, Spinner
[39]	Web	Ekran Görüntüsü	HTML	Div, header, span
[23]	Web	Çizim	HTML	Image, checkbox, input form elemanlar
[40]	Mobil	NLP	Hiyerarşik GUI iskeleti	
[20]	Web	Çizim	HTML	textBox, dropdown, button ve checkbox
[24]	Android	Çizim	XML	-
[54]	Web	Mockup Resim	HTML, CSS	
[19]	Web	Çizim	HTML	-
[41]	Mobil	UI tasarımı	Android GUI kodu	
[52]	Mobil	Mockup çizimi	Android GUI kodu	

koduna dönüştüren bir çalışma gerçekleştirmişlerdir. Şekil 8'de çalışmada kullanılan giriş mockup görüntüsü (a) ve çıkış görüntüsü (b) gösterilmiştir [54].

#### 4.2 Giriş-Çıkış Karakteristikleri

Bu başlıkta, araştırma sorularından Q2'nin cevabı verilmektedir. İncelenen çalışmalar Web veya Mobil platformlara yöneliktir. İncelenen çalışmaların 13 tanesi sadece Web, 8 tanesi sadece Android, 14 tanesi hem iOS hem Android, 3 tanesi hem Web hem de Mobil platforma yöneliktir.

İncelenen çalışmalar giriş görüntüsü olarak Çizim, UI tasarımı, Ekran görüntüsü ya da Mockup tasarımı kullanmaktadırlar. Çıkış olarak ise çalışma hangi platforma yönelik bir çalışma ise ona uygun bir çıktı kodu üretilmektedir. Web platformuna



**Şekil-8:** Boçucas ve Esteves'in çalışmasından bir görüntü[53]

yönelik ise HTML, Mobil uygulama yönelik ise XML arayüz kodu vermektedir. Bazı çalışmalar ise hem Web hem de Mobil platforma yönelik olduğundan ona uygun bir çıkış sağlanmaktadır [27]. Çizelge 2'de incelenen çalışmaların girdi-çıkış karakteristikleri özetlenmiştir.

[25]	Mobil	GUI çizimi	Android GUI kodu	
[2]	Mobil	Çizim	Android Kodu	
[36]	Android	GUI tasarım resmi	DSL GUI framework	-
[37]	Android, iOS	GUI ekran görüntüsü	Android veya iOS kodu	-
[38]	Android	GUI ekran görüntüsü	Android Kodu	
[29]	Android	Çizim	Android Kodu	EditText, ImageView
[51]	Mobil	Mockup	GUI hiyerarşik iskeleti	
[42]	Mobil	GUI Ekran Görüntüsü	Android Kodu	
[44]	Mobil	GUI Ekran Görüntüsü	Android Kodu	
[28]	Android, iOS ve Web	Çizim	HTML	HTML etiketleri
[26]	Android	Çizim	Android Kodu	
[43]	Mobil	GUI Ekran Görüntüsü	Android Kodu	Edittext,
[27]	Mobil	Çizim	Android Kodu	
[21]	Web	Çizim	HTML vs CSS	Navbar
[45]	Android, iOS ve Web	GUI Ekran Görüntüsü	XML, HTML	
[46]	Web	GUI Ekran Görüntüsü	HTML	
[22]	Web	Çizim	HTML	
[18]	Android	Çizim	XML	
[47]	Android	GUI Ekran Görüntüsü	XML	
[48]	Web	GUI Ekran Görüntüsü	HTML, CSS, emmet	
[49]	iOS	GUI Ekran Görüntüsü	SwiftUI kodu	
[55]	Web	Çizim	DSL kodu, HTML	

### 4.3 Kullanılan Yöntem ve Teknikleri Analizi

Bu bölümde çalışmalarda kullanılan yaklaşım ve teknikler analiz edilmiştir. Ayrıca, derin öğrenme kullanan yaklaşımların varsa CNN modelleri ve kütüphaneleri belirlenmiştir.

**Çizelge-3: Çalışmalarda Kullanılan Yöntem ve Teknikler**

Referans	Yaklaşım	Kullanılan Teknikler	CNN model(ler)i ve Kütüphane	Programlama
[34]	Derin Öğrenme	CNN, RNN	LSTM	-
[30]	Derin Öğrenme	CNN	-	-
[35]	Derin Öğrenme	CNN, RNN	-	-
[50]	Hibrit	Bilgisayarlı Görü Teknikleri, CNN, kNN	kNN VE CNN'in beraber kullanıldığı kendi	MATLAB
[39]	Derin Öğrenme	CNN, NLP	-	-
[23]	Derin Öğrenme	CNN obje tespiti	Faster R-CNN, Tensorflow	Python
[40]	Derin Öğrenme	NLP, CNN	-	-
[20]	Hibrit	Filtreleme, CNN ve BiLSTM	Kendi CNN yöntemleri	
[24]	Derin Öğrenme	DNN	-	-
[54]	Hibrit	Makine Öğrenmesi ve CNN	YOLOv3, Darknet	Python
[19]	Derin Öğrenme	CNN obje tespiti, Bi-LSTM	CNN, Bi-LSTM, Mask R-CNN	-
[41]	Derin Öğrenme	RNN, GAN	-	-
[52]	Derin Öğrenme	CNN obje tespiti	YOLOv5	
[25]	Derin Öğrenme	CNN obje tespiti	YOLO	
[2]	Derin Öğrenme	RNN ile obje tespiti	RNN	
[36]	Derin Öğrenme	CNN, RNN	-	-
[37]	Hibrit	CNN, kenar belirleme	-	-
[38]	Görüntü İşleme, Bilgisayarlı	Resim Filtreleme, OCR	-	-
[29]	Görüntü İşleme, Bilgisayarlı	OCR, Kenar Belirleme	-	-
[51]	Hibrit	CNN, Bilgisayarlı görü teknikleri	-	-
[42]	Bilgisayarlı Görü ve Derin	OCR, CNN obje tespiti	Yolo, Centernet,	

[44]	Derin Öğrenme	CNN	Tensorflow	Python
[56]	Diğer	Tasarım Mining	-	-
[28]	Derin Öğrenme	CNN Obje Tespiti	SSD	C++
[26]	Derin Öğrenme	CNN Obje tespiti	Yolo, Darknet	
[43]	Hibrit	CNN, Görüntü İşleme teknikleri	Kendi Modelleri	
[27]	Derin Öğrenme	CNN Obje Tespiti	MobileNet	
[21]	Derin Öğrenme	CNN Obje Tespiti	VGG	
[45]	Derin Öğrenme	CNN Obje Tespiti	VGG-16	
[46]	Görüntü İşleme, Derin Öğrenme	CNN, LSTM, Canny Kenar Belirleme	VGG-19	
[22]	Görüntü İşleme, Derin Öğrenme	CNN, Kenar belirleme teknikleri		
[18]	Derin Öğrenme	CNN	Yolo, Darknet	
[47]	Derin Öğrenmesi	GAN, SeqGAN		
[48]	Derin Öğrenme	CNN, RNN - LSTM	Faster R-CNN	
[49]	Derin Öğrenme	CNN	Faster R-CNN, ResNet-50	
[55]	Derin Öğrenme	CNN		

Çalışmaların hangi platformlara yönelik olduğu ve hangi yöntemlere göre olduğu Çizelge 2 ve Çizelge 3'de gösterilmiştir. Çizelge 3'te ise yöntemlerin platformlara göre çalışma sayılarının bilgisi verilmektedir. Çizelge görüldüğü gibi Derin Öğrenme yöntemini kullanarak Mobil uygulama kodu üreten çalışmaların sayısı fazladır. Derin Öğrenme kullanan toplam 28 çalışma bulunmaktadır. Sadece Görüntü İşleme (IP) veya Bilgisayarlı Görü(CV) yöntemlerini kullanan 2 çalışma vardır. Bu iki yöntemi ya da Makine Öğrenmesi yöntemlerini de birleştirerek melez bir yöntem kullanan çalışma sayısı ise 8'dir.

**Çizelge-4: Platform ve Yöntemlere göre Çalışma Sayıları**

		Platform					Toplam
		Web	Mobil	Andorid	Mobil, Web	iOS	
Yöntem	Derin Öğrenme	7	8	7	3	1	27
	melez	3	5	0	0	0	8
	IP-CV	0	0	2	0	0	2
Toplam		11	13	9	3	1	38

Çizelge 4'de kolayca anlaşılacağı üzere çalışmaların büyük bir bölümünde derin öğrenme yöntemleri kullanılmıştır. Son yıllarda artan derin öğrenme yöntemi kullanımının bu durumun oluşmasında payı büyüktür.

#### 4.4 Veri Seti Analizleri

Bu bölümde, makalelerde girdi olarak kullanılan GUI görüntü veri setleri ile ilgili çalışmalar ele alınmış ve bu çalışmalarda veri setlerinin büyüklükleri karşılaştırılmıştır. Çizelge-5'de görüntü birleştirme problemi ile ilgili incelenen çalışmalara yer verilmiştir.

Deka ve ark. bu alanda çalışma yapan araştırmacılara yönelik oluşturduğu Rico veri kümesi bu konudaki araştırmalar için önemli bir materyal olmuştur [56]. Rico veri kümesinde 9.700 uygulamadan elde edilen 72.200 GUI ekran görüntüsü vardır.

Daha önce benzer amaçlarla oluşturulmuş ERICA veri kümesinde 18,600 GUI ekran görüntüsü, Shirazi ve ark. çalışmalarında 29,000 ekran görüntüsü vardır [57], [58]. Zhang ve ark. 4068 iPhone uygulamasından 77,637 adet ekran görüntüsü toplamış ve etiketlemişlerdir. AMP adını verdikleri bu veri kümesini obje tanımlamak için kullanmışlardır [59].

**Çizelge-5: Çalışmaların Veri Kümesi Analizi**

Referans	Veri kümesi Adı	Veri kümesi Büyüklüğü
[34]	Pix2code	1750 ekran görüntüsü
[30]	Rico dataset	66.261 GUI görüntüsü
[35]	Google Play, AppStore Uygulama ekran görüntüleri	1.842.580 ekran görüntüsü
[50]	REDRAW	14.382 GUI görüntüsü
[39]	Bilgi Yok	-
[23]	Kendi veri kümeleri	562 UI resmi 11.152 UI bileşeni
[40]	Bilgi yok	-
[20]	Bilgi yok	-
[24]	Bilgi yok	600 GUI görüntüsü
[54]	Balsamiq mockups	1100 resim 1000 eğitim,100 test için
[19]	Kendi veri kümeleri	1500 web script
[41]	Bilgi yok	-
[52]	Kendi veri kümeleri	490 resim 390 eğitim, 100 sınama için
[25]	Kendi veri kümeleri	600 GUI bileşeni içeren 50 resim
[2]	Kendi veri kümeleri	11.500 resim
[36]	Andorid Uygulama görüntüleri	185.277 GUI görüntüsü
[37]	Google Play, AppStore Uygulama ekran görüntüleri	1.842.580 ekran görüntüsü
[38]	Kendi veri kümeleri	1576 eğitim için
[29]	Kendi veri kümeleri	50 ekran görüntüsü
[51]	REDRAW	14.382 GUI görüntüsü
[42]	Rico dataset	66.261 GUI görüntüsü

[44]	Rico dataset	66.261 GUI görüntüsü
[28]	Kendi veri kümeleri	2001 örnek
[26]	Bilgi Yok	-
[43]	Rico dataset	66.261 GUI görüntüsü
[27]	GUI CORE	330 Bileşen resmi
[21]	Kendi veri kümeleri	3000 navbar
[46]	Pix2code	1750 ekran görüntüsü
[18]	Kendi veri kümeleri	279 resim
[48]	Kendi veri kümeleri	9000 UI Bileşen görüntüsü
[49]	AMP ve RICO veri kümesi	130000 iOS ekranı 80000 Android ekranı
[55]	Bilgi yok	-

#### 4.4 Başarım Ölçütleri Analizleri

Başarım ölçünleri akademik bir çalışmada bir yaklaşımın başarımını ölçmek için kullanılır [60]. Çalışmada kullanılan yöntemlere göre kullanılan ölçün değişkenlik gösterir. Derin Öğrenme kullanan çalışmalar giriş resimlerinden nesne sınıflandırmak için konfüzyon matrisi oluşturularak Doğruluk oranı (1), Precision (Tutturma)(2), Recall (Bulma)(3) ve F değeri(4) kullanmışlardır. Bu ölçünlerin hesaplanması için konfüzyon matrislerinden TP (True Positive), TN (True Negative), FP (False Negative), ve FP (False Positive) değerleri bulunmalıdır. Bu değerler bulunduktan sonra aşağıdaki formüllere göre bu ölçünler hesaplanır. Örneğin Liu ve ark. çalışmalarında Derin Öğrenme yöntemi kullandıklarından ölçün olarak doğruluk, F-Değeri, Bulma gibi ölçünler kullanmışlardır [35].

$$Accuracy(Doğruluk) = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision(Tutturma) = \frac{TP}{TP + FP} \quad (2)$$

$$Recall(Bulma) = \frac{TP}{TP + FN} \quad (3)$$

$$F \text{ Ölçüsü} = 2 \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

BLEU (BiLingual Evaluation Understudy), makine tarafından çevrilmiş metni otomatik olarak değerlendirmek için bir ölçümdür. BLEU puanı, makine tarafından çevrilmiş metnin bir dizi yüksek kaliteli referans çevirisine benzerliğini ölçen sıfır ile bir arasında bir sayıdır[61]. BLEU otomatik kot üreten çalışmalarda kodun doğru üretilip üretilmediğini kontrol etmek için kullanılan bir başarım ölçünüdür. İncelenen çalışmaların üç tanesinde BLEU skoru başarım ölçünü olarak kullanılmıştır.

Makine Öğrenmesi tekniklerinin başarımını ölçmek için kullanılan diğer ölçünler ise METEOR, ROGUE-L ve SUM'dur. METEOR (Metric for Evaluation of Translation with Explicit Ordering), makine çevirisi çıktısının değerlendirilmesi için bir

ölçüdür. Bu ölçün, unigram tutturma bulma değerlerinin harmonik ortalamasına dayalıdır ve bulma değeri, tutturma değerinden daha ağırlıklıdır[62]. ROUGE (Recall-Oriented Understudy for Gisting Evaluation), doğal dil işlemede otomatik özetleme ve makine çevirisi yazılımını değerlendirmek için kullanılan bir ölçüdür[63]. SUM (Single Usability Metric) sistemin kullanılabilirliğini ölçmek için kullanılır[64].

Modellerin tahmin başarımını ölçmek için kullanılan diğer ölçünler ise hatanın kareler ortalaması (Mean Squared Error=MSE) ve hatanın mutlak ortalamasıdır (Mean Absoulte Error=MAE). Chen ve ark. 2019 yılındaki çalışmalarında MAE, MSE gibi ölçünleri kullanmışlardır [37]. Bazı çalışmalarda ise BLEU değeri ölçün olarak kullanılmıştır.

Nesne tespiti ve sınıflandırılmasında tahmin başarımını ölçmek için kullanılan başarım ölçünlerinden bir tanesi de genel ortalama tahmindir (Mean Average Precision=mAP). mAP değeri nesne tespiti ve sınıflandırılması için geliştirilen CNN modellerinde de başarım ölçünü olarak kullanılmıştır[8]. İncelenen çalışmaların dördünde başarım ölçütü olarak mAP kullanılmıştır.

Wu ve ark. çalışmasında graf ve ağaç yapısı kullanılmıştır. Bu yüzden çalışmada ölçün olarak graf ve ağaçlarına uygun ölçünler kullanılmışlardır. Ağaç yapılarında yaprak düğümlü kenarların F1 skoru olarak F1 Leaves kullanılmıştır. GED (Graph Edit Distance), bir grafiği diğerine izomorf yapmak için gereken minimum "düzenleme" sayısını dikkate alan bir grafik benzerliği ölçümüdür[49].

Çizelge 6'da incelenen çalışmaların başarım ölçünleri ve sonuçları verilmiştir.

Çizelge 6. Çalışmaların Başarım Ölçün Analizi

Referans	Ölçün Adı	Sonuçlar
[34]	Hata oranı, F ve C değerleri	%3,4 hata oranı
[30]	Görsel olarak benzer uygulamaları bulma yaklaşımının potansiyeli	-
[35]	Doğruluk Tutturma Bulma	%85 Doğruluk
[50]	Doğruluk	%85 doğruluk oranı
[39]	Ölçüm yok	-
[23]	mAP	%95,5 mAP skoru
[40]	Ölçüm yok	-
[20]	Metot doğruluk ve validation accuracy	-
[24]	mAP	%87 mAP değeri
[54]	BLEU Değeri Doğruluk	%83 doğruluk
[19]	Farklı diğer yaklaşımlarla karşılaştırma: -METEOR -ROUGE-L -BLEU -SUM	BLEU 0,69 METEOR 0,51 ROUGE 0,78 SUM 0,65
[41]	Ölçüm yok	-
[52]	Tutturma – Doğruluk Bulma F-Değeri	13 sınıf için ayrı ayrı ölçülmüştür.
[25]	Ölçüm yok	-

[2]	Doğruluk Zaman ölçümü	%93,9 doğruluk Sınıflandırma süresi: 526 ms yerelde (Android)
[36]	Doğruluk Zaman, Benzerlik durumları	%60,28 doğruluk oranı
[37]	Diğer Makine öğrenmesi yöntemleriyle doğruluk oranı karşılaştırması MAE (Mean Absolute error) MSE (Mean squared error)	%85 doğruluk oranı
[38]	Tutturma Bulma F-Değeri	%88,2 tutturma değeri %87,3 bulma değeri %87,7 F-Skoru
[29]	Başarım sınaması yok	-
[51]	Detection Precision Classification Precision Recall	
[42]	Tutturma Bulma F-Değeri Doğruluk	0,490 0,557 0,524 0,91
[44]	Doğruluk	%88
[28]	Öge tespiti çıkarım zamanı	Çıkarım süresi 0,2 ila 1,7 saniye arasında sürmektedir.
[26]	mAP	Her bileşen için ayrı ayrı hesaplanmıştır.
[43]	F1 Değeri	0,524 F1 değeri
[27]	Doğruluk	%95 doğruluk oranı
[44]	Doğruluk	Sentetik Çizimlerde 0,91, Gerçek Çizimlerde 0,69 doğruluk
[46]	BLEU Değeri	Mean 0,89 Std 0,38
[18]	Doğruluk F1 Değer, mAP	%87 doğruluk 0,88 %87,72
[48]	Ortalama Doğruluk (AP), Recall, Precision	Her bileşen için ayrı hesaplanmış
[49]	F1 Değeri, GED, F1 Leaves	F1 0,60 F1 Leaves 0,67 GED 20,2
[55]	Doğruluk Oranı	%87

### 3. Sonuç ve Öneriler

Bu çalışmada bir çizimden veya ekran görüntüsünden otomatik kot oluşturan çalışmaların analizi yapılmıştır. Bu analizler 4 kategoride incelenmiştir. Çalışmaların giriş çıkış karakteristikleri, çalışmalarda kullanılan veri kümeleri, yöntemler ve kullanılan başarım ölçünlerine göre analizler gerçekleştirilmiştir.

İncelenen çalışmalarda kullanılan veri kümeleri genellikle daha önceki çalışmalarda oluşturulan veri kümeleridir. Bu alanda çalışma yapacak araştırmacılar da bu veri kümeleri kullanabilir ve geliştirebilir. Bilimin doğasına uygun olarak her bir araştırmacının sağladığı verilerle gelişen veri kümeleri sayesinde araştırmalardaki başarı artacaktır.

Metot açısından incelendiğinde otomatik kot üretimi tekniklerinin eski yıllarda görüntü işleme, bilgisayarlı görü yöntemlerini kullandıkları görülmüştür. Son yıllarda ise derin öğrenme ve makine öğrenimi yöntemlerine doğru bir eğilim gerçekleşmiştir. Derin öğrenmede ise CNN ile nesne tespiti kullanılmıştır. Farklı CNN çerçeveleri (framework) farklı sonuçlar vermektedir. Bazı çalışmalarda melez yöntemlerin daha etkili olduğu görülmüştür. Bu alanda çalışma yapacak araştırmacıların derin öğrenme ve bilgisayarlı görü tekniklerini ileri düzeyde bilmesi gerekmektedir.

Başarım metriği olarak derin öğrenme kullanan çalışmalar ağırlıkta olduğundan genellikle Tutturma, Bulma ve F-Değeri kullanılmıştır. Bu değerler her sınıf için ayrı hesaplandığı gibi ortalama olarak da hesaplanabilmektedir. Bu alanda çalışma yapan araştırmacılar bu ölçünleri öncelikli olarak tercih etmeleri tavsiye edilmektedir.

Sonuç olarak Çizimden, ekran görüntüsünden ya da Mockup görüntülerinden otomatik kot geliştirme çalışmaları gelişen bir çalışma alanıdır. Bu alanda çalışma yapacak araştırmacılar yayın taramasını özenli yaptığı takdirde nitelikli bir veri kümesi ve uygun bir yöntem ile başarılı sonuçlar elde edebilirler.

### Kaynakça

- [1] D. Stone, C. Jarrett, M. Woodroffe, and S. Minocha, *User interface design and evaluation*. Elsevier, 2005.
- [2] S. Mohian and C. Csallner, "Doodle2App: Native app code by freehand UI sketching," in *Proceedings - 2020 IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2020*, Jul. 2020, pp. 81–84. doi: 10.1145/3387905.3388607.
- [3] T. M. Mitchell and T. M. Mitchell, *Machine learning*, vol. 1, no. 9. McGraw-hill New York, 1997.
- [4] T. M. Mitchell and T. M. Mitchell, *Machine learning*, vol. 1, no. 9. McGraw-hill New York, 1997.
- [5] D. Ozdemir and M. S. Kunduraci, "Comparison of Deep Learning Techniques for Classification of the Insects in Order Level With Mobile Software Application," *IEEE Access*, vol. 10, pp. 35675–35684, 2022, doi: 10.1109/ACCESS.2022.3163380.
- [6] M. F. Kunduraci and H. K. Örnek, "Vehicle Brand Detection Using Deep Learning Algorithms," *International Journal of Applied Mathematics Electronics and Computers*, pp. 0–3, 2019.
- [7] M. Mandal, "Introduction to Convolutional Neural Networks (CNN)," *analyticsvidhya.com*, May 01, 2021.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," pp. 1–14, 2016.
- [9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [10] W. Liu, "SSD : Single Shot MultiBox Detector SSD : Single Shot MultiBox Detector," no. December, 2015.
- [11] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019, pp. 6105–6114.

- [12] R. Yang and Y. Yu, "Artificial Convolutional Neural Network in Object Detection and Semantic Segmentation for Medical Imaging Analysis," *Frontiers in Oncology*, vol. 11. Frontiers Media S.A., Mar. 09, 2021. doi: 10.3389/fonc.2021.638182.
- [13] L. R. Medsker and L. C. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001.
- [14] M. Gao, G. Shi, and S. Li, "Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network," *Sensors (Switzerland)*, vol. 18, no. 12, Dec. 2018, doi: 10.3390/s18124211.
- [15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [16] Y. Guo, X. Cao, B. Liu, and K. Peng, "El Nino index prediction using deep learning with ensemble empirical mode decomposition," *Symmetry (Basel)*, vol. 12, no. 6, Jun. 2020, doi: 10.3390/SYM12060893.
- [17] J. A. Landay and B. A. Myers, "Interactive sketching for the early stages of user interface design," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995, pp. 43–50.
- [18] D. Baulé, C. G. von Wangenheim, A. von Wangenheim, J. C. R. Hauck, and E. C. V. Júnior, "Automatic code generation from sketches of mobile applications in end-user development using Deep Learning," *arXiv preprint arXiv:2103.05704*, 2021.
- [19] Y. Han, J. He, and Q. Dong, "CSSSketch2Code: An automatic method to generate web pages with CSS style," in *ACM International Conference Proceeding Series*, Oct. 2018, pp. 29–35. doi: 10.1145/3292448.3292455.
- [20] B. Asiroglu et al., "Automatic HTML Code Generation from Mock-up Images Using Machine Learning Techniques," *IEEE*, 2019.
- [21] T. Calò and L. de Russis, "Style-Aware Sketch-to-Code Conversion for the Web," in *EICS 2022 - Companion of the 2022 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, Jun. 2022, pp. 44–47. doi: 10.1145/3531706.3536462.
- [22] G. Vitkare, R. Jejurkar, S. Kamble, Y. Thakare, and A. P. Lahare, "AUTOMATED HTML CODE GENERATION FROM HAND DRAWN IMAGES USING MACHINE LEARNING METHODS".
- [23] B. B. Adefris, "Automatic Code Generation From Low Fidelity Graphical User Interface Sketches Using Deep Learning," 2020.
- [24] Y. S. Yun, J. Park, J. Jung, S. Eun, S. Cha, and S. S. So, "Automatic Mobile Screen Translation Using Object Detection Approach Based on Deep Neural Networks," *Journal of Korea Multimedia Society*, vol. 21, no. 11, pp. 1305–1316, 2018, doi: 10.9717/kmms.2018.21.11.1305.
- [25] Y. S. Yun, J. Jung, S. Eun, S. S. So, and J. Heo, "Detection of GUI elements on sketch images using object detector based on deep neural networks," in *Lecture Notes in Electrical Engineering*, 2019, vol. 502, pp. 86–90. doi: 10.1007/978-981-13-0311-1\_16.
- [26] Jisu Park, Jinman Jung, Seungbae Eun, and Young-Sun Yun, "UI Elements Identification for Mobile Applications based on Deep Learning using Symbol Marker," *The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)*, vol. 20, no. 3, pp. 89–95, Mar. 2020, doi: <https://doi.org/10.7236/IIBC.2020.20.3.89>.
- [27] A. A. Rahmadi and A. Sudaryanto, "Visual Recognition Of Graphical User Interface Components Using Deep Learning Technique," Surabaya, Jan. 2020.
- [28] V. Jain, P. Agrawal, S. Banga, R. Kapoor, and S. Gulyani, "Sketch2Code: Transformation of Sketches to UI in Real-time Using Deep Neural Network," Oct. 2019, [Online]. Available: <http://arxiv.org/abs/1910.08930>
- [29] S. Kim et al., "Identifying UI Widgets of Mobile Applications from Sketch Images," 2018.
- [30] X. Ge, "Android GUI Search Using Hand-drawn Sketches."
- [31] W. O. Galitz, *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons, 2007.
- [32] R. Lal, *Digital design essentials: 100 ways to design better desktop, web, and mobile interfaces*. Rockport Pub, 2013.
- [33] D. Gavalas and D. Economou, "Development platforms for mobile applications: Status and trends," *IEEE Softw*, vol. 28, no. 1, pp. 77–86, 2010.
- [34] X. Pang, Y. Zhou, P. Li, W. Lin, W. Wu, and J. Z. Wang, "A novel syntax-aware automatic graphics code generation with attention-based deep neural network," *Journal of Network and Computer Applications*, vol. 161, Jul. 2020, doi: 10.1016/j.jnca.2020.102636.
- [35] Y. Liu, S. Chen, L. Fan, L. Ma, T. Su, and L. Xu, "Automated Cross-Platform GUI Code Generation for Mobile Apps," 2019.
- [36] C. Chen, T. Su, G. Meng, Z. Xing, and Y. Liu, "From UI design image to GUI skeleton: A neural machine translator to bootstrap mobile GUI implementation," in *Proceedings - International Conference on Software Engineering*, May 2018, pp. 665–676. doi: 10.1145/3180155.3180240.
- [37] C. Chen, S. Feng, Z. Xing, L. Liu, S. Zhao, and J. Wang, "Gallery D.C.: Design search and knowledge discovery through auto-created GUI component gallery," *Proc ACM Hum Comput Interact*, vol. 3, no. CSCW, Nov. 2019, doi: 10.1145/3359282.
- [38] X. Xiao, X. Wang, Z. Cao, H. Wang, and P. Gao, "IconIntent: Automatic Identification of Sensitive UI Widgets based on Icon Classification for Android Apps."
- [39] N. Sethi, A. Kumar, and R. Swami, "Automated web development: Theme detection and code generation using Mix-NLP," in *ACM International Conference Proceeding Series*, Jun. 2019. doi: 10.1145/3339311.3339356.
- [40] K. Kolthoff, "Automatic generation of graphical user interface prototypes from unrestricted natural language requirements," in *Proceedings - 2019 34th IEEE/ACM International Conference on Automated Software Engineering, ASE 2019*, Nov. 2019, pp. 1234–1237. doi: 10.1109/ASE.2019.00148.
- [41] T. T. Nguyen, P. M. Vu, H. V. Pham, and T. T. Nguyen, "Deep learning UI design patterns of mobile apps," in *Proceedings - International Conference on Software Engineering*, May 2018, pp. 65–68. doi: 10.1145/3183399.3183422.
- [42] J. Chen et al., "Object detection for graphical user interface: Old fashioned or deep learning or a combination?," in *ESEC/FSE 2020 - Proceedings of the 28th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Nov. 2020, pp. 1202–1214. doi: 10.1145/3368089.3409691.

- [43] M. Xie, S. Feng, Z. Xing, J. Chen, and C. Chen, "UIED: A hybrid tool for GUI element detection," in *ESEC/FSE 2020 - Proceedings of the 28th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Nov. 2020, pp. 1655–1659. doi: 10.1145/3368089.3417940.
- [44] S. Mohian and C. Csallner, "PSDoodle: Searching for App Screens via Interactive Sketching," Apr. 2022, doi: 10.1145/3524613.3527807.
- [45] W. Y. Chen, P. Podstreleny, W. H. Cheng, Y. Y. Chen, and K. L. Hua, "Code generation from a graphical user interface via attention-based encoder–decoder model," *Multimed Syst*, vol. 28, no. 1, pp. 121–130, Feb. 2022, doi: 10.1007/s00530-021-00804-7.
- [46] V. Saravanan, "Automated Web Design And Code Generation Using Deep Learning," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 6, pp. 364–373, 2021.
- [47] T. Zhao, C. Chen, Y. Liu, and X. Zhu, "Guigan: Learning to generate gui designs using generative adversarial networks," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 2021, pp. 748–760.
- [48] Y. Xu, L. Bo, X. Sun, B. Li, J. Jiang, and W. Zhou, "image2emmet: Automatic code generation from web user interface image," *Journal of Software: Evolution and Process*, vol. 33, no. 8, p. e2369, 2021.
- [49] J. Wu, X. Zhang, J. Nichols, and J. P. Bigham, "Screen Parsing: Towards Reverse Engineering of UI Models from Screenshots," in *The 34th Annual ACM Symposium on User Interface Software and Technology*, 2021, pp. 470–483.
- [50] K. Moran, B. Li, C. Bernal-Cárdenas, D. Jelf, and D. Poshyvanyk, "Automated reporting of GUI design violations for mobile apps," May 2018, pp. 165–175. doi: 10.1145/3180155.3180246.
- [51] K. Moran, C. Bernal-Cárdenas, M. Curcio, R. Bonett, and D. Poshyvanyk, "Machine learning-based prototyping of graphical user interfaces for mobile apps," *IEEE Transactions on Software Engineering*, vol. 46, no. 2, pp. 196–221, 2018.
- [52] A. A. Abdelhamid, S. R. Alotaibi, and A. Mousa, "Deep learning-based prototyping of android gui from hand-drawn mockups," *IET Software*, vol. 14, no. 7, pp. 816–824, Dec. 2020, doi: 10.1049/iet-sen.2019.0378.
- [53] T. Bouças and A. Esteves, "Converting web pages mockups to HTML using machine learning," 2020.
- [54] T. Bouças and A. Esteves, "Converting web pages mockups to HTML using machine learning," in *WEBIST 2020 - Proceedings of the 16th International Conference on Web Information Systems and Technologies*, 2020, pp. 217–224. doi: 10.5220/0010116302170224.
- [55] G. Jadhav, H. Gaikwad, and M. Gawande, "Generation Source Code from Hand Draw Image—A Machine Learning Approach," *Generation Source Code from Hand Draw Image—A Machine Learning Approach (February 25, 2022)*, 2022.
- [56] B. Deka et al., "Rico: A mobile app dataset for building data-driven design applications," in *UIST 2017 - Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, Oct. 2017, pp. 845–854. doi: 10.1145/3126594.3126651.
- [57] B. Deka, Z. Huang, and R. Kumar, "ERICA: Interaction Mining Mobile Apps," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, Oct. 2016, pp. 767–776. doi: 10.1145/2984511.2984581.
- [58] A. S. Shirazi, N. Henze, A. Schmidt, R. Goldberg, B. Schmidt, and H. Schmauder, *Insights into Layout Patterns of Mobile User Interfaces by an Automatic Analysis of Android Apps*. 2013.
- [59] X. Zhang, L. de Greef, and S. White, "Screen Recognition: Creating Accessibility Metadata for Mobile Applications from Pixels," in *Conference on Human Factors in Computing Systems - Proceedings*, May 2021. doi: 10.1145/3411764.3445186.
- [60] Y. Liu, Y. Zhou, S. Wen, and C. Tang, "A strategy on selecting performance metrics for classifier evaluation," *International Journal of Mobile Computing and Multimedia Communications (IJMCMC)*, vol. 6, no. 4, pp. 20–35, 2014.
- [61] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [62] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.
- [63] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.
- [64] J. Sauro and E. Kindlund, "A method to standardize usability metrics into a single score," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2005, pp. 401–409.

# nDPI Derin Paket İnceleme Aracı Üzerinde Bir Çalışma

## A Study on the nDPI Deep Packet Inspection Tool

Zehra Nur ÖZBAY  
Ege Üniversitesi  
Uluslararası Bilgisayar Enstitüsü  
İzmir, Türkiye  
zehra.nur.ozbay@ege.edu.tr  
ORCID: 0000-0001-5680-1227

Mehmet Emin DALKILIÇ  
Ege Üniversitesi  
Uluslararası Bilgisayar Enstitüsü  
İzmir, Türkiye  
mehmet.emin.dalkilic@ege.edu.tr  
ORCID: 0000-0003-3932-5155

### Öz

Derin paket inceleme (DPI), ağ paketlerini beş katmanlı ağ modelinde bulunan uygulama katmanına kadar analiz ederek ağda bulunan uygulama protokollerini tespit etmek için kullanılan ileri seviye bir paket tanımlama yöntemidir. Bu çalışmada DPI yöntemi kullanarak paket tanımlaması yapan açık kaynak kodlu nDPI kütüphanesi ele alınmıştır. Bu kütüphane üzerinden yeni uygulama protokolü tespiti ve eksikliği bulunan bazı protokollere de eklemeler sağlanarak alana katkıda bulunulması hedeflenmiştir. Ayrıca, nDPI tarafından yanlış kategorize edildiği tespit edilen ağ paketleri için yeni kural tanımlarının yapılarak bu gibi durumların düzeltilmesi sağlanmıştır. Tüm bunlar için, ağ trafiği Wireshark paket yakalama aracıyla kaydedilip paket içerikleri analiz edilerek yeni kurallar oluşturulmuş ve yeni bulunan uygulama protokolleri nDPI kütüphanesine eklenmiştir. Son olarak, nDPI'da uygulama protokolü tespiti için yapılan çalışmaların kısmi bir otomasyonu yazılmıştır.

**Anahtar sözcükler:** Derin paket inceleme, DPI, nDPI

### Abstract

Deep packet inspection (DPI) is an advanced packet identification method used to detect application protocols found in the network by analyzing network packets up to the application layer in the five-layer network model. In this study, the open source nDPI library, which uses DPI method to identify packets, is discussed. It is aimed to contribute to the field by detecting new application protocols and adding some missing protocols through this library. In addition, new rule definitions were made for network packets that were found to

be miscategorized by nDPI, and such cases were corrected. For all these, network traffic was recorded with Wireshark packet capture tool, packet contents were analyzed, new rules were created and newly found application protocols were added to the nDPI library. Finally, a partial automation of the work done for the application protocol detection in nDPI was written.

**Keywords:** Deep packet inspection, DPI, nDPI

### 1. Giriş

Her geçen gün yeni bir uygulama ağda yerini alırken tanımlanması ve sınıflandırılması gereken protokol sayısı da artmaktadır. Bunun yanı sıra internete bağlı cihazların sayısı da nesnelerin interneti (IoT) ile birlikte muazzam seviyelere ulaşmaktadır. 2030 yılına gelindiğinde yaklaşık 30 milyar IoT cihazının kullanımda olacağı tahmin edilmektedir [1]. Cihaz ve uygulama sayılarındaki bu kontrolsüz artış ağ güvenliği problemlerini de beraberinde getirmektedir. Güvenlik duvarları, Sızma Tespit Sistemleri (IDS) ve Sızma Önleme Sistemleri (IPS) ağa yönelik saldırılara karşı koymak için sıklıkla tercih edilen ve kullanılan ağ güvenliği cihazlarıdır. Genellikle iç ağ ile dış ağ arasındaki sınır bölgelere konumlandırılan güvenlik duvarları, iç ağdan dış ağa ya da tersi yönde hareket eden ağ paketlerinin başlık bilgilerini kontrol ederek kullanıcının önceden tanımladığı kurallar ile uyumlu paketlere geçiş hakkı verirken, uyum sağlamayanları düşürür. Güvenlik duvarları ağ giriş ve çıkışlarında klasik paket filtreleme için kullanılır. Fakat "ince taneli" saldırı paketlerini tespit edemeyeceği için bu filtreden içeriye sızma mümkündür. Diğer bir ağ güvenliği cihazı olan IDS ise dinleme yapan duyargalar aracılığı ile iç ağda görev yapar. IDS imza veri tabanını kullanarak iç ağdaki şüpheli ve düzensiz trafiği tespit ederek ağ yöneticisine rapor eder. Genellikle güvenlik duvarlarının tam arkasına konumlandırılan IPS, IDS'nin paketler üzerindeki pasif işlevinin aksine, ağda aktif rol alarak



paketleri engelleme ve düşürme gibi yetkilere de sahip olup IDS cihazlarının bir uzantısıdır. Tüm bu cihazlar kuşkusuz ağ güvenliğinin ana unsurlarını oluşturmaktadır. Fakat bu cihazlar tek başlarına ele alındığında yeni tip ağ saldırılarına karşı etkileri düşük olacaktır. Örneğin, güvenlik duvarlarının işlem gücü sınırlı olup büyük hacimli paketleri işlemek için yeterli değildir. Ayrıca, bir güvenlik duvarı durum bilgisi tutsa dahi beş katmanlı ağ modelinin son basamağı olan ve asıl veriyi içeren uygulama katmanındaki bilgileri göremez. Bu tıpkı elimize aldığımız bir kitabın içeriğinden haberdar olmadan sadece kitabın başlığını okuyarak kitabı değerlendirmemize benzer [2]. DPI ise bu noktada ağ trafiğini çözerek uygulama katmanı dâhil tüm katmanlarda inceleme yapmayı ve böylece paket içeriğinde yer alan veriyi de, şifrelenmemiş ise, okumayı sağlar. Diğer bir yandan, saldırganlar hedef ağa gönderecekleri paketleri daha küçük parçalara ayırıp saldırı niteliğindeki imzanın tek bir pakette yer almasını engelleyerek IDS tarafından tespitini imkânsız hale getirebilirler. DPI analizinde ise saldırı imzası birçok pakete ayrılarak izi silinmeye çalışılmış olsa bile bu paketler bir araya getirilerek imza tespit edilir. Dolayısıyla, DPI teknolojisi ağ güvenliği cihazlarının temel bir bileşeni haline gelmiştir. Bir DPI aracı ya da kütüphanesinin kullanımı ağ içerisinde dinleme yapan paket yakalama araçlarıyla birlikte çalışması şeklinde olabileceği gibi daha yaygın kullanımlar olan DPI tanımlı yeni nesil güvenlik duvarı (NGFW) ya da IDS/IPS cihazları şeklinde de olabilir.

Yayınlardaki çalışmalar incelendiğinde hem uygulama protokolü tespiti hem de uygulamayı engelleme amaçlı derin paket inceleme kullanan çalışmalar bulunmaktadır. Renals ve Jacoby tarafından yapılan çalışmada [3], DPI kullanılarak P2P bir mesajlaşma uygulaması olan Skype'ın nasıl engellendiği gösterilmiştir. Çalışmada, Skype paketlerini tanımlamak için Skype'ı diğer uygulamalardan ayırt eden özellikleri kullanarak bir kural kümesi oluşturulmuştur. Bu kural kümesini inşa edebilmek için önce -olası tüm durumları hesaba katmak üzere- farklı bilgisayarlar üzerinde, farklı zamanlarda ve farklı Skype hesapları ve versiyonlarında Wireshark ağ dinleme aracı kullanılarak veri toplanmıştır. Elde edilen Skype trafiğine ait veriler, aynı anda en fazla üç farklı oturum trafiğini karşılaştırabilen, Araxis Merge isiminde bir analiz programı kullanılarak karşılaştırılmıştır. Bu analiz Skype kurallarının, anahtar kelimeler, port numaraları ve içerik olmak üzere, üç kategoride şekillenmesiyle sonuçlanmıştır. Anahtar kelimeler içeriğinde "Skype" gibi belirli dizgiler içeren paketleri, port numarası bağlantı noktası 33033 olan TCP trafiğini ve içerik ise "16 03 01 00 00" veya "17 03 01 00 00" ASCII dizgilerini içeren paketleri tanımlar. Bu kurallar, içerisinde DPI tanımlı bir IPS cihazı olan açık kaynak kodlu Snort'ta tanımlanır. Bu kuralları uyguladıktan sonra Wireshark ile trafik yeniden kaydedilerek Skype'ın davranışı gözlemlenir. Skype'ın bağlantı kurmaya çalışıp yeni dizgiler kullandığı tespit edilir. Son olarak bu yeni dizgiler de kural tablosuna eklenerek Snort çalıştırıldığında bir önceki davranışından farklı bir yol denemeyen Skype'ın bloke olduğu gözlemlenmiştir. Diğer bir çalışmada ise [4], internette anonim kimlikle iletişimi sağlayan Soğan Yönlendirme (TOR) yazılımının DPI kullanılarak ağda kullanımının tespit edilmesi ve elde edilen analiz sonuçlarıyla bu uygulamanın nasıl engellendiği ele alınmıştır. Bunun için,

yeni bir TCP bağlantısı kurulduğu zaman bazı komutları çalıştıracak olan (olaya dayalı) ve DPI özellikli bir Bro-IDS sunucusu, Wireshark ve bir Squid vekil sunucusu kullanılmıştır. TOR trafiğini, TLS protokolü ile şifrelenmiş durumda olduğundan, karakterize edebilmek amacıyla TOR bağlantı kurulumu için gerekli olan ve veri alım ve gönderimi öncesi gerçekleştirilen iki süreç analiz edilmiştir. Birincisi TOR kullanıcısı ile TOR ağı arasındaki -SYN, SYN-ACK ve ACK paketlerinin iletimi olarak da bilinen- üç yönlü el sıkışma süreci, ikincisi ise TLS oturumu kurma sürecidir. Üç yönlü el sıkışma, TOR dışındaki diğer bağlantılar ile karşılaştırıldığında, standart bir trafik olarak seyretmiştir. Fakat TLS el sıkışması olarak da adlandırılan TLS bağlantı kurulumunda ise bazı farklılıklar tespit edilmiştir. Bunlardan istemcinin sunucuya gönderdiği "ClientHello" mesajlarında TOR'u karakterize eden iki özellik gözlemlenmiştir: Birincisi, şifreleme paketi kombinasyonlarının -yani istemci (TOR tarayıcısı) tarafından teklif edilen şifreleme algoritmalarının- hep aynı olması, diğeri ise sunucu adı formatının her zaman [www.<crastgele\\_dizgi>.com](http://www.<crastgele_dizgi>.com) veya "www.<crastgele\_dizgi>.net" şeklinde olmasıdır. Diğer bir farklılık, "ClientHello" mesajlarından sonra sunucunun gönderdiği sertifikadaki bilgilerinde de "ClientHello" mesajındaki aynı dizgi formatının yer almasıdır. TOR trafiğinin, çıkarılan bu özellikleri Bro-IDS cihazında kural olarak yazıldıktan sonra, bloke edilme süreci ise bir internet kullanıcısı ile vekil sunucu arasındaki trafiğin tam bir kopyasının Bro-IDS tarafından analiz edilmesiyle başlar. TOR karakterine sahip paketlerin bazı özellikleri çıkartılarak Bro-IDS'in kayıt dosyasına yazılır. Bro-IDS, bu yazım olayı gerçekleşince, ilgili paketlerin varış IP adreslerini bir erişim listesi dosyasına yazma ve vekil sunucuyu -bloke edilecek bu adresler ile- yeniden yükleme olmak üzere iki fonksiyonunu aktive eder. Böylece TOR trafiği engellenmiş olur. Son olarak, ağ trafiğindeki Adobe Creative Suite, Microsoft Sharepoint, Salesforce, Yammer ve Zendesk uygulamalarının nDPI kullanılarak tespit edilmesini ele alan bir çalışmada [5] otomatik bir betik yazılarak uygulamaların ilgili web sitelerinde gezinti yapılmış ve en çok kullanılan 20 IP adresi ve alan adı kaydedilmiştir. Daha sonra bu IP adreslerinin ve alan adlarının gerçekte kime ait olduğunu doğrulamak için bir whois aracı kullanılmıştır. Bazı IP adreslerinin iki uygulama tarafından da kullanımda olduğu bir durumda bu adresler, tek bir uygulamayı net bir şekilde ifade edemeyeceği için, kural kümesine dâhil edilmemiştir. Diğer yandan, bir IP adresinin sunuculuğu başka bir firmaya ait olup kullanım hakkı bakımından sahipliği uygulamaya aitse bu adresler dâhil edilmiştir.

Bizim çalışmamız ise açık kaynak kodlu nDPI kütüphanesi [6] ve Wireshark [7] ağ dinleme cihazının beraber kullanıldığı bir senaryo üzerinden gerçekleştirilmiştir. Kullanılan yöntem yukarıdaki çalışmalara benzer olup, Radityatama ve arkadaşlarının araştırmasına [5] daha yakındır. Fakat bu çalışmadan farklı olarak IP adresleri kullanılmadan yeni protokoller tespit edilip eklenmek suretiyle açık kaynak kodlu nDPI kütüphanesine katkı sağlanması hedeflenmiştir. IP adreslerinin neden eklenmediği kural çıkarma alt başlığında açıklanmıştır.

## 2. Temel Tanım ve Kavramlar

Akış kavramı, bir TCP/IP bağlantısını oluşturan aynı beşli değer kümesine sahip bir veya birden fazla paketi ifade eder. Bu beşli değer kümesinin elemanlarını; kaynak IP adresi/port numarası, hedef IP adresi/port numarası ve IP başlık bilgisinde yer alan protokol bilgisi (TCP, UDP, ICMP vb.) oluşturur. Bir TCP/IP bağlantısını eşsiz olarak bu beşli belirler. Bir diğer ifadeyle, aynı beşli değere sahip TCP/IP paketleri aynı akışa aittir.

Ağ trafiği sınıflandırması bir ağda dolaşımda olan uygulamaları tanımlamak ve analizini yapmak için kullanılan temel bir metottur. Bu teknik ağın performans ölçümü, güvenliği, daha iyi yönetimi ve trafik mühendisliği için özellikle internet servis sağlayıcıları (ISP) ve ağ operatörleri tarafından sıklıkla kullanılır. Böylece, ISP'ler uygulama veya abone bazında trafiği tanımlayabilir, izleyebilir ve kontrol edebilirler. Ağ trafiğinin, genel olarak web, bulut, multimedya, mesajlaşma, e-posta, oyun, müzik, veritabanı, dosya paylaşımı ve IP üzerinden ses (VoIP) gibi daha birçok sınıf tanımlanmaktadır. Örneğin, bu çalışmadaki uygulama protokolleri alışveriş ve eğlence kategorileri kapsamındadır.

### 2.1 DPI

Derin paket inceleme (diğer bir adıyla Tam Paket Denetimi veya Bilgi Çıkarma [8]); ağ yönetimi, güvenlik, filtreleme, yönlendirme, sansür, istatistik, hizmet kalitesi (QoS) ve deneyim kalitesi (QoE) gibi birçok farklı amaç doğrultusunda kullanılan; genellikle güvenlik duvarı, yönlendirici veya dağıtıcı gibi ağ düğümleri üzerinde önemli bir dişli olarak çalışan; paketlerin sadece başlık bilgilerine değil uygulama katmanında yer alan yük içeriklerine de bakarak "derinlemesine" analiz eden bir paket tanımlama tekniğidir. Kısaca, derin paket incelemesi içeriğe duyarlı işleme yaparak ağ trafiğinin doğru bir şekilde sınıflandırılmasını sağlar.

DPI metodu, kendi içerisinde dar ve geniş olmak üzere iki farklı kavram olarak ele alınır [9]. Dar anlam desen eşleştirme yöntemini ifade etmektedir. Bu yöntem de kendi içinde dizgi ve düzenli ifade eşleştirmesi olarak ikiye ayrılmaktadır. Bu tanıma işlemi, paket yükünün önceden tanımlanmış imzalarla eşleştirilmesi ile uygulanmaktadır. Geniş anlamda ise; istatistiksel analiz, port tabanlı eşleştirme ve protokol çözümü yöntemleri ele alınmaktadır. Bu çalışmada uygulanan da, dar kapsamında sayılan ve nDPI'da ndpi\_content\_match.c.inc isimli dosya üzerinden gerçekleştirilen dizgi eşleştirme yönteminin kullanılmasıdır.

DPI metodu, paket yükünü de öğrenmeye dayalı olduğundan yüksek doğruluk sağlasa da kullanıcı gizliliği ve güvenliği açısından sakıncalı bulunabilmektedir. Ağ trafiğinin şifreli yapıya geçişi de bu yöntemin üzerinde baskılayıcı bir rol oynamaktadır. Çünkü SSL/TLS trafiğinde şifre bilinmediği sürece sadece el sıkışma fazındaki anahtar değişiminden faydalanılabilir. Bu çalışmadaki uygulama protokollerinin hepsi şifreli bağlantı kullanılmaktadır. Dolayısıyla sadece SSL/TLS el sıkışma fazındaki anahtar değişiminden elde edilen veriler kullanılarak ağ trafiği tanımlaması gerçekleştirilmiştir.

Hem şifreli hem de gerçek zamanlı ağ trafiğinin işlenmesi ise üzerinde çalışılması gereken ayrıca bir konu olup literatürdeki boşluğunu korumaktadır [10]. Ağ katmanlarının en üstünde yer alan bu problemten dolayı belki de sınıflandırma işlemlerini ağ katmanının en alt basamağı olan fiziksel katmandaki bitlere kadar indirgeyerek trafiği direkt izleme imkânı sunacak bir ağ trafiği sınıflandırması gerçekleştirme bu alanın geleceğindeki en büyük zorluklarından biri olarak görülmektedir [11].

DPI teknolojisinin iki temel fonksiyonu vardır. Birincisi, tanımlama yapmak, diğeri ise bu tanımlama sonucuna göre bir eylemde bulunmaktır [9]. Tanımlama, ağ paketinin karakteristik özelliğini çıkarmaktır. Bu karakteristik; uygulama protokolü, multimedya uygulaması, bulut yazılımı gibi normal trafik akışını ya da kötü amaçlı yazılım, virüs, siber saldırı gibi zararlı içeriği işaret edebilir. Uygulanan eylem ise güvenlik araçlarıyla kurulan ilişkidir. Paket tanımlama tamamlandıktan sonra, IDS, IPS veya güvenlik duvarı aktif hale getirilerek; uyarı sinyali verme, bloke etme, yeniden yönlendirme veya günlüğe kaydetme aksiyonları gerçekleştirilir. Örneğin, bir paket DPI tarafından; güvenlik duvarındayken "zararlı yazılım" tanımlaması yapılmış ise düşürülebilir, IDS üzerinde "potansiyel tehlike" olarak bildirilmişse ağ yöneticisine rapor gönderilebilir veya IPS üzerindeyken "şüpheli" olarak tanımlanmış ise bağlantı bloke edilebilir. Bu çalışma tanımlama kısmını ele almaktadır.

#### 2.1.1 DPI Uygulamaları

DPI teknolojisinin birçok farklı amaca yönelik kullanım alanları bulunur. Bunlardan bazıları; ağ güvenliği, bant genişliği yönetimi, deneyim kalitesi, gözetim ve sansürdür. Xu ve arkadaşlarının çalışması [9] ise DPI kullanılarak yapılan uygulamaları devlet, ISP ve şirket olmak üzere üç grubun kullanım amaçları doğrultusunda daha detaylı olarak ele alır. Bu kullanım amaçları arasında, kullanıcı profili üzerinden reklam enjeksiyonu gerçekleştirmek de bulunur. İlk olarak Bendrath'ın çalışmasında [12] geçen reklam enjeksiyonu, bir kullanıcının internette gezindiği veya alışveriş yaptığı web sitelerinin takip ve analizi yapılarak bıraktığı ayak izine ait bir reklam profilinin oluşturulması ve ona özel çevrimiçi reklam sunulmasıdır. Bu araştırma çerçevesinde bahsedilen yedi şirketin web sitesinden toplanılan veriler incelendiğinde ağ trafiğini büyük ölçüde reklam ve takip amaçlı yazılımlara ait paketlerin doldurduğu gözlemlenmiştir.

#### 2.1.2 nDPI

nDPI ilk olarak 2013 yılında, ntop şirketi tarafından ağ trafiğini karakterize etmek için, C dilinde geliştirilen ve LGPL lisansına sahip bir DPI kütüphanesidir. nDPI, GPL lisansına sahip olan fakat artık güncelleme almayan OpenDPI yazılımından ilham alınarak geliştirilmiştir. nDPI, OpenDPI ile benzer olarak, hem Linux çekirdeğinde hem de kullanıcı programlarının bulunduğu kullanıcı alanında kullanılabilir. Linux dışında, Windows, MacOS X ve BSD ailesi gibi işletim sistemlerinde de kurulumu yapılabilir. En son sürümü olan nDPI 4.6 [13], Şubat 2023'te çıkmıştır. Bu çalışmada yer alan çalışmalar Linux tabanlı bir işletim sistemi olan Ubuntu üzerinde ve o esnada mevcut olan nDPI 4.2 sürümü kullanılarak yapılmıştır. nDPI, en son sürümüyle beraber, 300'ü aşkın uygulama protokolü

tanımlamanın yanı sıra TLS sertifikası, tarayıcı adı ve şifreleme paketi gibi bir akışla ilişkili meta verileri de raporlar.

nDPI'nin belli başlı özellikleri aşağıdaki gibidir:

- nDPI'da bir protokol, genelde, nDPI kütüphanesinde tanımlı ve C dilinde yazılan trafik ayrıştırıcısı ile tespit edilir. Fakat protokoller sadece ayrıştırıcı kullanılarak bulunmaz. Port numarası, IP adresi ve protokol özelliklerine göre de bulunabilir. Örneğin, Dropbox trafiği hem yerel alan ağı tabanlı bağlantılar için kullanılan ayrıştırıcı hem de sunucu adı bilgisinde ".dropbox.com" dizgisi geçen HTTP trafiğinin Dropbox olarak etiketlenmesiyle tanınır [14]. Bu yüzden, yeni bir akış trafiğinin yaşam döngüsü öncelikle paketlerin üçüncü ve dördüncü katmanlarının çözülmesi ve bu arada ayrıştırıcıların denenmesiyle başlar. Modülerlik ve genişletilebilirlik özelliklerini sağlaması için her biri ayrı bir c uzantılı dosyada kodlanan ayrıştırıcıların uygulanış sırası, trafik türüne göre akışla eşleşme olasılığı en yüksek olandan başlayarak gerçekleşir. Örneğin, TCP/80 için önce HTTP ayrıştırıcısı denenirken, 53 port numaralı TCP/UDP protokolleri için ise önce DNS ayrıştırıcısı denenir. Her akış, eşleşmeyen ayrıştırıcıların durum bilgisini gelecek olan iterasyonlarda atlamak için saklar. Analiz, bir eşleşme bulunana kadar veya belirli sayıda denemeden sonra sona erer. Bu çalışmada, ele alınan web sitelerinin hepsi TLS/HTTPS kullandığı ve nDPI'da da HTTP ayrıştırıcısı bulunduğu için ayrıca bir protokol ayrıştırıcısı yazılmamıştır.
- Deri ve arkadaşlarının çalışmasında [14], nDPI'da bir uygulama protokolünü tespit etmek veya bilinmeyen olarak etiketlemek için gerekli paket sayısının en fazla sekiz olduğu belirtilmiştir. Bunun anlamı, bir akış protokolü imzasının tanımlanabilmesi için bir araya getirilmesi gereken paket sayısının en fazla sekiz olmasıdır. Ayrıca, çalışma sürecinin ne zaman başlatıldığı da önemlidir. Örneğin, nDPI akış başladıktan sonra çalıştırıldığı bir durumda akışa ait ilk paketler analiz edilmediğinden o akış sınıflandırılmamış olarak işaretlenebilir.
- nDPI kütüphanesinde her uygulama protokolü benzersiz bir protokol numarası ve sembolik bir protokol adı ile tanımlanır. Yeni bir protokol kütüphaneye ekleneceği zaman o protokol için daha önce hiç kullanılmamış bir numara ve isim bulunmalıdır. nDPI dilinde sadece TLS, DNS gibi temel internet ağı protokolleri değil, Skype, Facebook veya Youtube gibi uygulamalar da protokol olarak adlandırılır. Bu yüzden bir protokol aslında ağ ve uygulama olmak üzere iki türlü tanımlanır. Bu protokol türleri sırasıyla majör ve minör olarak da ifade edilebilmektedir. Örneğin, ağ protokolü TLS ve uygulama protokolü Facebook olan bir TCP paketi proto: 91.119/TLS.Facebook olarak raporlanır. Buradaki 91 TLS'e, 119 ise Facebook'a özel tanımlanmış protokol numaralarıdır. Aynı şekilde, en çok karşılaşılan, 5 numara DNS'e aittir.

- nDPI, iki uç sistem arasındaki bir SSL bağlantısında ortak bir anahtar yardımıyla gerçekleştirilen şifreli iletişim başlamadan önceki ilk anahtar değişimi kısmı için kod çözümleyicisi bulundurur. Bu sayede bağlantı kurulan uç sistemin sunucu adını çıkarabilir ve şifreli paketlerin hangileri olduğunu, içeriklerini okuyamasa da, belirleyebilir. Sunucu adı bilgisi, tıpkı HTTP bağlantılarında ana bilgisayar adının HTTP başlık bilgisinden alınması gibi, nDPI'nin akışa ait meta verilerinde tutulur. Bu çözümleyici sayesinde hem sunucu adlarına göre protokoller belirlenebilir hem de kendinden imzalı SSL sertifikaları bulunabilir. Kendinden imzalı sertifikalar, kullanıcıların kendi adlarına verdiği, herhangi bir sertifika yetkilisi tarafından onaylanmamış ortak anahtar sertifikalarıdır. Bu tür imza içeren bir bağlantı güvenli sayılmadığından bu bilgi değerlidir.

- nDPI'da protokollerin güvenli, kabul edilebilir, potansiyel tehlikeli, tehlikeli gibi türlere ayrıldığı ndpi\_typedefs.h dosyası bulunur. Aynı dosyada, ayrıca, protokolleri anlamlı bir şekilde gruplamak için; alışveriş, oyun, dosya paylaşımı, reklamlar, çevrimiçi bulut hizmetleri, sosyal ağlar, anlık mesajlaşma uygulamaları, kötü amaçlı yazılım ve yasaklı web sitesi gibi soyut kategoriler tanımlıdır.

- nDPI/example dosyasında bulunan ndpiReader demo uygulaması, komut satırından çalıştırılan ve nDPI kütüphanesinin bazı özelliklerini gösteren bir test aracıdır. Bu uygulamayı kullanmak için veriler ndpiReader tarafından belirli bir süre canlı trafiğin yakalanmasıyla veya Wireshark gibi ağ dinleme cihazlarıyla daha önceden kaydedilen hazır pcap dosyalarının girdi olarak verilmesiyle sağlanabilir. Bu yüzden, ndpiReader aynı zamanda bir ağ dinleyicisi olarak da çalışır. ndpiReader aracının ihtiyaca göre kullanılacak çeşitli komut satırı seçenekleri bulunur. Örneğin, -i bayrağı ndpiReader aracına girdi olarak verilecek önceden kaydedilmiş bir pcap dosyasının veya canlı trafik yakalaması yapılacak ise de kullanılacak cihazın veya arayüzün isminin belirlenmesini sağlarken; -v <1|2|3|4> bayrağı 1 numara için normal ayrıntılı, 2 için daha çok ayrıntılı, 3 için port istatistikleri ve 4 için özet (hash) istatistiklerine göre görüntüleme yapmayı sağlar. Tüm diğer seçenekler ve bazı nDPI özellikleri yardım anlamına gelen -h bayrağı kullanılarak elde edilebilir.

- Akış analizi sonuçları -C bayrağı kullanılarak csv formatında bir dosyaya kaydedilebilir. Bu csv dosyasında; akış numarası, kaynak/varış IP numarası, kaynak/varış port numarası, eşsiz protokol numarası, sunucu ismi, TLS versiyonu, her iki yöndeki paket ve byte sayısı gibi daha birçok alan bulunur. Bu şekilde csv formatındaki veriler düzenlenip sıralanarak trafik analizi süreçlerine faydalı olabilir. Bu formatta bir çıktı elde etmek için komut satırından aşağıdaki komut yazılır:

```
ndpiReader -i <dosya_adi.pcapng> -C <dosya_adi.csv>
```

Bu şekilde elde edilen csv dosyası, linux komut satırına

SQL sorgulama yetisi kazandıran q-text-as-data uygulamasına [15] girdi olarak kullanılırsa, trafik hakkında bazı çıkarımlarda bulunmamızı sağlar. Örneğin, belirli bir web sitesinde en çok hangi kaynak IP numarasının veya istemcinin, zaman harcadığını bilmek istediğimizi varsayalım. Aşağıdaki kod yardımıyla giden ve gelen trafikteki toplam byte sayısına göre siteyle iletişim kuran tüm IP numaralarını q uygulamasıyla sıralayabiliriz:

```
q -H -d ',"select src_ip,
SUM(s_to_c_bytes+c_to_s_bytes)
from <csv_dosya_yolu> where
server_name_sni like '<sunucu_ismi>'
group by src_ip"
```

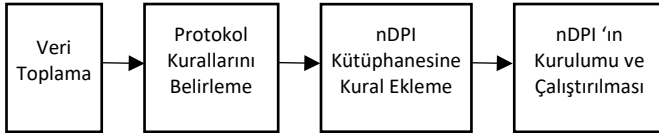
Burada -H bayrağı, dosyanın sütunları adlandırmak için kullanılan bir başlık satırı içerdiğini belirtir. -d bayrağı ise girdi sınırlayıcı olarak çalışır. Sunucu isminde "trendyol" geçen paketler için uygulanan bir örnek sorgunun çıktısı Şekil-2'de verilmiştir. Veriler kişisel masaüstü bilgisayar üzerinde toplandığı için karşılık gelen tek bir IP adresi, 155.223.40.20, görünmektedir. Bu şekilde, csv dosyasında bulunan diğer verilerden farklı farklı sorgular ile hedeflenen amaca yönelik ağ trafiği analizi gerçekleştirilebilir.

```
zehra@zehra:~/src/nDPI-4.2$ q -H -d ',"select src_ip,SUM(s_to_c_bytes
+c_to_s_bytes) from /usr/src/nDPI-4.2/trendyol_for_q_test.csv where
server_name_sni like '%trendyol%' group by src_ip"
155.223.40.20,1805491
```

Şekil-2: Bir q-text-as-data sorgusuna ait örnek çıktı

### 3. Gereç ve Yöntem

Çalışmanın genel çerçevesi Şekil 1'de verilmiştir.



Şekil-1: Çalışmanın genel çerçevesi

Bu kapsamda, tespit edilmesini istediğimiz uygulamanın gerçek zamanlı kullanımını sonucu ağda oluşan paketlerin Wireshark aracı yardımıyla yakalanıp pcap veya pcapng uzantısıyla kaydedilerek detaylı paket içerik analizlerinin yapılması için kurulumu yapılan nDPI aracına girdi olarak verilmiştir. Hem nDPI aracından elde edilen sonuçlar hem de kural çıkarma alt başlığında bahsedilecek olan diğer metotlar kullanılarak alan adlarından elde edilen bir kural listesi oluşturulmuştur. Son adımda nDPI cihazında eksik olan bu uygulama protokolleri kütüphaneye kural olarak eklenerek cihazın tespit edebildiği protokol sayısı artırılmıştır. Bu protokoller, Fast Company dergisinin 2020 Şubat ayında yayınlanan ve gelirini sadece internet üzerinden sağlayan Türkiye'deki en büyük 100 internet şirketi listesindeki [16] ilk yedi şirketin kullandığı web sitelerini kapsamaktadır. Bu şirketler büyüklük sırasına göre sırasıyla; Trendyol, Hepsiburada, Nesine.com, N11, Bilyoner, GittiGidiyor ve Yemeksepeti'dir. Protokol seçiminin bu listeden yapılmak

istenmesinin sebebi, bu şirketlerin kullanıcıyla etkileşiminin sadece internet üzerinden olması ve ilgili web sitelerinde yoğun veri akışı gerçekleşmesi sebebiyle ağ trafiğindeki tespitlerinin önemli olduğunun değerlendirilmesidir. Son olarak, yapılan çalışma Selenium web sürücüsü ve Python programlama dili kullanılarak kısmi olarak otomatize edilmiştir.

#### 3.1 Veri Toplama

Veriler iki farklı ana bilgisayar kullanılarak toplanmıştır: Bunlardan biri Windows 10 ve Ubuntu 20.04.5 LTS olmak üzere çift işletim sistemine sahip olan, Ethernet kablosuyla internete bağlanan ve dört çekirdekli 3.60 GHz Intel Core i7-4790 işlemciye sahip 64 bitlik bir masaüstü bilgisayardır. Diğeri ise, Windows Education serisi işletim sistemine sahip olan, internet ağına kablosuz bağlanan ve çift çekirdekli 2.50 GHz Intel Core i5-3210M işlemciye sahip 64-bitlik bir dizüstü bilgisayardır. Wireshark kullanılarak veri toplama amaçlı gerçekleştirilen web sayfası gezintilerinde, her iki bilgisayarda da hem Chrome ve hem de Firefox tarayıcısından faydalanılmıştır. Burada iki farklı bilgisayarın, işletim sisteminin, bağlantının ve tarayıcının kullanım amacı farklı sistem kombinasyonlarının yeni veri getirme olasılığını hesaba katmaktır. Fakat bu tercihlerden birinin diğerine göre fazladan veri sağladığı bir durum gözlemlenmemiştir. Bununla birlikte veri çeşitliliğine en çok katkı sağlayan fonksiyon, kuşkusuz, web sayfasında bulunan farklı web elemanlarına erişimin sağlanması olmuştur. nDPI kütüphanesine eklenecek kuralların belirlenebilmesi için öncelikle Trendyol, Hepsiburada, Nesine.com, N11, Bilyoner, GittiGidiyor ve Yemeksepeti'ne ait web sitelerinde Wireshark açılarak paket yakalaması başlatılmış ve ilgili web sitelerin her birinde bir süre gezinti yapılmıştır. Bu gezintilerde, bir web sayfasına ait tüm farklı alan adlarının elde edilebilmesi için, olabildiğince fazla sayı ve türdeki sayfa elemanlarına erişilmesi gözetilmiştir. Gezinti sonucu oluşan dosyalar protokol, tarayıcı ve bağlantı türüne göre çeşitli isimler verilerek .pcapng uzantısıyla işletim sistemi ayırımına göre bir klasöre kaydedilmiştir. Örneğin, n11\_chrome\_ethernet.pcapng veya trendyol\_firefox\_wifi.pcapng gibi. Bu şirketlerden eBay'ın sahibi olduğu GittiGidiyor 20 Haziran 2022 tarihinde Türkiye pazarından çekildiğini duyurarak 18 Temmuz 2022 itibarı ile de internet alışveriş sitesini kullanıma kapatmıştır. Bu nedenle, çalışmanın başlangıcında bu sitenin protokolü üzerinde çalışılmış ve kuralları çıkartılmış olsa da, daha sonra yazılan otomasyon sürecinde dâhil edilmemiştir.

#### 3.2 Kural Çıkarma

Çalışmanın başlangıcında trafik verilerini analiz etmeye ilk olarak yedi şirketin tarayıcıda açıkça görünen ve kullanıcılar tarafından da iyi bilinen alan adlarının IP adreslerine ulaşılması hedeflenmiştir. Bunun için, öncelikle bu alan adlarının kaydını gerçekleştiren kayıtçı kuruluşlar ve ilgili web adresleri whois veri tabanı sorgusuyla bulunmuştur. Daha sonra kayıtçı kuruluşların web adreslerinde tekrar whois sorgusu yapılarak şirketlerin kullanıcılara sunduğu web sayfalarının yetkili ad sunucusu bilgileri ve karşılık gelen IP adresleri listesine ulaşılacak istenmiştir. Yapılan sorgularda kayıtçı kuruluşların; yetkili ad sunucuları, kayıt olan

organizasyonun adı, sistem yetkilisinin adı ve iletişim bilgileriyle kuruluşun adresi gibi çeşitli bilgileri döndürmesine rağmen IP adreslerini vermediği gözlemlenmiştir. Bunun üzerine, bu adresleri IP adres aralıkları şeklinde elde etmeyi sağlayan başka bir yöntem denenmiştir. Bunun için, ARIN, APNIC ve RIPE NCC kuruluşlarının veri tabanında whois sorgusu yapılmıştır. Fakat burada da sağlıklı sonuçlar elde edilmemiştir. Örneğin, Trendyol şirketinin RIPE veri tabanındaki sorgusunda 159.20.112.0/24, 159.20.113.0/24, 159.20.114.0/24, 159.20.115.0/24 ve 159.20.116.0/24 IP adres aralıkları elde edilmiştir. Fakat Wireshark ile ağ dinlenip, ip.addr == 159.20.116.0/16 gibi, oldukça geniş IP aralığının varlığını kontrol eden birçok filtre uygulanmasına rağmen whois sorgusundan elde edilen yukarıdaki IP adreslerinden hiçbiri ağ trafiğinde tespit edilmemiştir. Fakat <https://dnschecker.org/ip-location.php> adresinden IP adresi-konum sorgusu yapıldığında bu IP adreslerinin İstanbul'un çeşitli ilçelerine ait olduğu ve organizasyon adı olarak da Trendyol'un sahibi olan DSM Grup Danışmanlık İletişim ve Satış Ticaret Anonim Şirketi'nin [17] isminin geçtiği de gözlemlenmiştir. Dolayısıyla üçüncü bir seçenek olarak, uzun ama daha sağlıklı ve kesin çözüm üretebilecek olan, Wireshark'ta toplanan trafik verilerindeki tüm DNS sorgu ve cevaplarının tek tek incelenmesi yolu izlenmiştir. DNS sorgu ve cevaplarına kolay ve hızlı bir şekilde ulaşabilmek için Wireshark filtreleme yöntemleri kullanılmıştır. Örneğin, bir akıştaki tüm DNS sorgu ve cevaplarını çekmek için dns.qry.class filtresi kullanılırken, paketler arasından "n11" dizgisini içeren DNS sorgu ve cevaplarını elde etmek için dns contains "n11" filtresi uygulanmıştır. Bu gibi sorgular sonucunda Wireshark'ta çıktı olarak sıralanan paketlerin içeriğindeki veriler incelenerek hem alan adları hem de karşılık gelen IP adresleri kaydedilmiştir. Ayrıca, Wireshark sonuçlarında çıkmamasına rağmen nslookup yöntemiyle elde edilen farklı alan adları ve IP adresleri olmuştur. Fakat bu yöntem tüm IP adreslerini vermediği için yüksek ihtimalle var olduğu düşünülen fakat Wireshark ile bulunamamış alan adı olduğu zaman başvurulmuştur. Örneğin, Trendyol için Wireshark üzerinden collect.trendyol.com, collect2.trendyol.com, collect3.trendyol.com ve collect5.trendyol.com gibi alan adları tespit edilmiştir. Bu nedenle, collect1.trendyol.com ve collect4.trendyol.com gibi alan adlarının da var olma olasılığı doğduğundan, komut satırından nslookup collect1.trendyol.com şeklinde sorgulama yapılmıştır ve gerçekten var oldukları tespit edilmiştir. Trendyol ve diğer şirketlerin tüm alan adları yukarıda anlatılan DNS sorgu ve cevaplarının çekilmesi yöntemiyle, nDPI-4.2 kütüphanesine eklemek üzere, bulunup kaydedilmiştir. Fakat diğer yandan, sadece Trendyol için -RIPE veri tabanından elde edilenler de eklenirse- toplam 2095 IP adresinden oluşan bir liste oluşmuştur. Bunun üzerine, IP adreslerinin hepsi ikinci bir kontrolden geçirilmek üzere IP adreslerinin gerçek sahibinin kim olduğunun sorgulanabildiği <https://dnschecker.org/ip-whois-lookup.php> adresinden araştırılmıştır. Bu adresteki sorgu aracı aynı zamanda bir whois sorgusu yaptığından IP adresinin hangi kayıtçı kuruluşu -ARIN, APNIC, RIPE NCC gibi- ait olduğu bilgisini de vermektedir. Dolayısıyla, adreslerin kime ait olduğu, son olarak kayıtçı kuruluşlarda tekrar whois sorgusu yapılarak

netleştirilmiştir. Tüm bu sorgulama sürecinin sonunda, Wireshark'tan elde ettiğimiz, Trendyol'un kullandığı IP adresleri listesindeki -RIPE'dan elde edilenler hariç- tüm adresler içerik dağıtım ağı (CDN) hizmeti sunmasıyla bilinen Alibaba Cloud ve CloudFlare gibi şirketlere ait çıkmıştır. Aynı şekilde, Hepsiburada, N11 ve Bilyoner şirketlerinin de CDN servisiyle ünlü Akamai şirketine ait IP adreslerini kullandıkları bulunmuştur. CDN hizmeti artık birçok firma tarafından tercih edildiği için birden fazla uygulamanın aynı adres aralığındaki farklı IP adreslerini kullandığı durumlarla karşılaşılabilir. Bu durumu en açık şekilde gösteren bir örnek Çizelge-1'de verilmiştir. Buna göre, nDPI-4.2 kütüphanesinde tanımlı TOR protokolünün belirleyicisi olarak kullanılan IP adresleri ile Trendyol'un kullandığı collect.trendyol.com alan adının Wireshark'tan elde edilen IP adres bilgilerinin hepsinin, APNIC kayıtçısındaki whois sorgularında, Alibaba şirketinin sağladığı Alibaba Cloud bulut hizmetinin IP adres aralığına denk geldiği anlaşılmaktadır. Dolayısıyla, bu bize ele aldığımız e-ticaret sitelerinin birden çok IP adresi kullanabileceğini ve bu IP adreslerinin de değişmeye müsait yapıda olduğunu göstermiştir. Bu durum, tespit etmek istediğimiz uygulama protokollerin belirleyicisi olarak IP adreslerinin seçilmek istenmemesinin temel nedenini teşkil etmiştir.

nDPI'nin 4.2'den hemen sonra çıkan 4.4 sürümünde yapılan daha sonraki incelemelerde TOR'un kullanmakta olduğu fakat Alibaba Cloud, Amazon gibi bulut servisi veren şirketlere ait IP adresinin kaldırıldığı fark edilmiştir.

**Çizelge-1: Aynı bulut servisinden faydalanan iki farklı uygulama: TOR ve Trendyol**

Trendyol Adresleri	TOR Adresleri
8.209.80.30	8.209.79.125
8.209.81.21	8.209.93.160
8.209.88.204	8.209.94.85
8.209.89.108	8.210.144.170
APNIC Sonuçları IP adres alanı aralığı: 8.209.64.0-8.209.127.255 Ağ Adı: ALICLOUD-DE	APNIC Sonuçları IP adres alanı aralığı: 8.209.64.0-8.209.127.255 Ağ Adı: ALICLOUD-DE

Paketlerin analizi yapılırken karşılaşılan ve IP adreslerinin yol açtığı somut bir problem ise ndpiReader aracının çıktılarında birçok alan adının yanlış kategori başlıkları altında etiketlendiğinin gözlemlenmesi olmuştur. Bunun sebebi, Cloudflare, Amazon AWS, Google ve Microsoft Azure gibi bulut hizmeti veren şirketlerin IP adreslerinden faydalanan organizasyonların alan adları nDPI-4.2 kütüphanesinde bulunmazken bulut hizmetini veren bu şirketlerin IP adreslerinin ise bulunmasıdır. Bu duruma birkaç örnek Çizelge-2'de verilmiştir.

**Çizelge-2: nDPI-4.2'de yanlış kategorize edilen alan adları ve buna sebep olan bazı bulut hizmeti uygulamaları**

Alan Adı ve IP Adresi	ndpiReader Çıktısı	Kayıtlı nDPI Protokolünün Adı ve IP Bilgileri
sync.srv.stackadapt.com 54.87.192.123	Proto: 91.265/TLS.Amazon AWS Cat: Cloud/13	NDPI_PROTOCOL _AMAZON_AWS 54.87.0.0/16

js.appboycdn.com 104.18.22.230	Proto: 91.220/TLS.Cloudflare Cat: Web/5	NDPI_PROTOCOL _CLOUDFLARE 104.16.0.0/12
api-js.mixpanel.com 107.178.240.159	Proto: 91.126/TLS.Google Cat: Web/5	NDPI_PROTOCOL _GOOGLE 107.178.192.0/18

Bu tabloya göre, .stackadapt.com ile başlayan alan adlarının reklam amaçlı veri toplayan StackAdapt yazılım şirketine, .appboycdn.com ile başlayan alan adlarının daha önce AppBoy olarak bilinen fakat şimdiki adı Braze olan bulut tabanlı yazılım şirketine ve .mixpanel.com ile başlayan alan adlarının ise kullanıcı etkileşimlerini izleyerek veri toplayıp raporlayan Mixpanel şirketine ait olduğu bulunmuştur. nDPI 4.2'de, bu örnekte olduğu gibi, yanlış kategorize edilen diğer tüm alan adları kütüphaneye doğru bir şekilde eklendikten sonra ağ paketlerinin cat: Advertisement/101 veya cat: Media/1 olarak tespit edilmesi sağlanmıştır.

Alan adlarının bulunabileceği ikinci bir yöntem ndpiReader uygulamasının kendisini kullanmaktır. nDPI'nin ayrıntılı modu seçildiği zaman, -v2, internet sunucu adı ile hizmet veren tüm sunucu adlarına ulaşılabilir. Otomasyon sürecinde bu yöntem kullanılmıştır.

Üçüncü ve son olarak, Türkiye'deki şirketlerin ticaret sicil numarası, ünvanı, adres bilgileri ve internet adresi gibi bilgilerine ulaşmayı sağlayan E-şirket şirket bilgi portalından [18] ve ilgili şirketin web adresinde hakkında veya biz kimiz gibi bölümlerinden alınan bilgiler incelenip karşılaştırıldığında, ağ trafiğinde o kuruma ait birkaç alan adının daha bulunması söz konusu olabilir. Örneğin, diğer şirketler için yapılan aramalarda bulunamasa da Trendyol için yapılan incelemede şirketin ticaret ünvanı DSM Grup Danışmanlık İletişim ve Satış Ticaret Anonim Şirketi olarak karşımıza çıkmaktadır. DSM grubu sadece Trendyol e-ticaret sitesinin sahibi olduğu için, Wireshark'ta dns contains "dsm" şeklinde filtrelediğimizde elde edilen aşağıdaki alan adları da Trendyol özelinde kaydedilmiştir:

- cdn.dsmcdn.com
- img-dsmncdn.mncdn.com
- dsmgrup.com
- static.dsmcdn.com

### 3.3 Kural Yazımı ve nDPI Kurulumu

nDPI kurulumundan önce yedi büyük şirket için kütüphaneye eklenmesi gereken iki kural kümesi vardır:

- /src/include klasöründeki ndpi\_protocol\_ids.h dosyasına her bir protokol için daha önce kullanımda olmayan kimlik numarası niteliğinde bir sayı yazılır. nDPI 4.2 versiyonunda en son 281 numarası kullanımda olduğu için, yedi sitenin protokolleri için sırasıyla karşılık gelecek şekilde, 282'den başlayarak 288'e kadar protokol adı ve kimlik numarası yazılır.
- İkinci olarak, her bir hedef protokolün ilgili alan adları /src/lib klasöründe yer alan ndpi\_content\_match.c.inc

dosyasına sırasıyla alan adı, tespit edilmesi halinde ndpiReader'da nasıl isimlendirileceği, protokol adı, protokol kategorisi ve protokol türü bilgilerini içerecek şekilde kural satırları olarak yazılır. Bu kuralların birkaç örneği aşağıda verilmiştir:

- {"trendyol.com", "Trendyol", NDPI\_PROTOCOL\_TRENDYOL, NDPI\_PROTOCOL\_CATEGORY\_SHOPPING, NDPI\_PROTOCOL\_SAFE, NDPI\_PROTOCOL\_DEFAULT\_LEVEL}
- {"nesine.com", "Nesine", NDPI\_PROTOCOL\_NESINE, NDPI\_PROTOCOL\_CATEGORY\_GAME, NDPI\_PROTOCOL\_FUN, NDPI\_PROTOCOL\_DEFAULT\_LEVEL}
- {"yemeksepeti.com", "Yemeksepeti", NDPI\_PROTOCOL\_YEMEKSEPETI, NDPI\_PROTOCOL\_CATEGORY\_SHOPPING, NDPI\_PROTOCOL\_SAFE, NDPI\_PROTOCOL\_DEFAULT\_LEVEL}

Bütün kural girdileri tamamlanınca artık nDPI kurulumu aşağıda yer alan komutlar sırasıyla çalıştırılarak gerçekleştirilebilir:

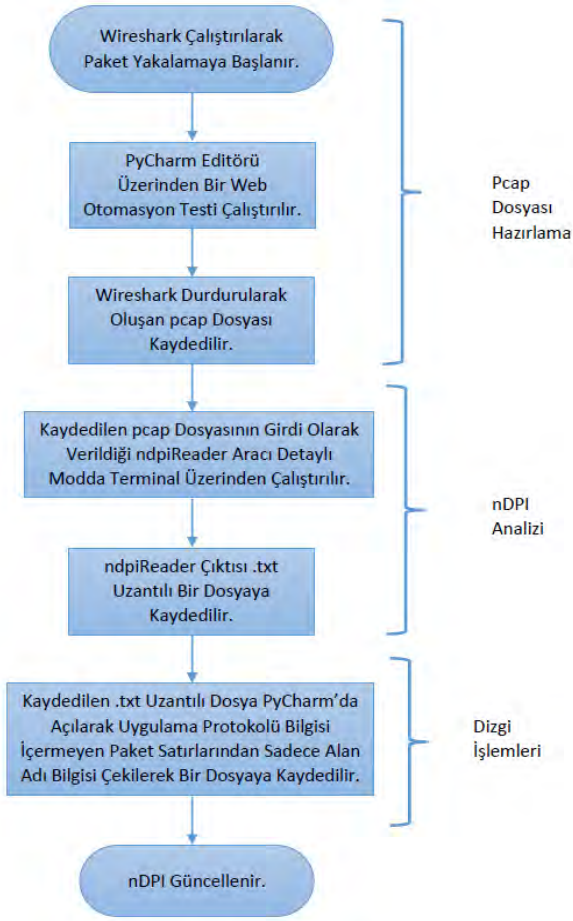
```
cd <nDPI_klasör_yolu>
sudo ./autogen.sh
sudo ./configure
sudo make
sudo make install
```

Kuralların eklenmesi ve nDPI kurulumunun yapılması sonrası ndpiReader aracının tespit ettiği uygulama protokollerine ait bir çıktının ekran görüntüsü Şekil-3'de verilmiştir.

```
Detected protocols:
Unknown      packets: 9          bytes: 816          flows: 1
DNS          packets: 832       bytes: 102819      flows: 409
HTTP        packets: 11        bytes: 2668        flows: 1
MDNS        packets: 14        bytes: 2616        flows: 2
SSDP        packets: 8          bytes: 1736        flows: 2
TikTok      packets: 441       bytes: 332078      flows: 6
Yahoo       packets: 68        bytes: 20535       flows: 6
STUN        packets: 6          bytes: 652         flows: 1
IGMP        packets: 1          bytes: 46          flows: 1
TLS          packets: 5374      bytes: 2539595     flows: 173
Facebook    packets: 884       bytes: 692461      flows: 16
Twitter     packets: 281       bytes: 112579      flows: 19
GoogleMaps  packets: 33        bytes: 9809        flows: 3
YouTube     packets: 138       bytes: 73019       flows: 11
Google      packets: 11148     bytes: 6877905     flows: 216
AppleTunes  packets: 4          bytes: 636         flows: 2
WindowsUpdate packets: 104      bytes: 35738       flows: 7
MSD         packets: 42        bytes: 29736       flows: 2
eBay        packets: 1787      bytes: 1466519     flows: 51
Pinterest   packets: 2          bytes: 358         flows: 1
QUIC        packets: 75        bytes: 34737       flows: 4
Instagram  packets: 4          bytes: 362         flows: 2
Cloudflare  packets: 4297      bytes: 371872      flows: 48
PlayStore   packets: 37        bytes: 12882       flows: 3
GoogleServices packets: 580      bytes: 343078      flows: 36
Z39.50     packets: 9          bytes: 1137        flows: 1
AmazonAWS   packets: 984       bytes: 428486      flows: 38
Azure       packets: 353       bytes: 242242      flows: 10
Trendyol    packets: 2667      bytes: 2288834     flows: 45
Hepsiburada packets: 9372      bytes: 8723725     flows: 73
Nesine      packets: 11756     bytes: 11670073    flows: 56
n11         packets: 6365     bytes: 6491823     flows: 58
Bilyoner   packets: 6338     bytes: 6387826     flows: 37
Gittigidiyor packets: 78        bytes: 18890       flows: 7
Yemeksepeti packets: 828      bytes: 694562      flows: 9

Protocol statistics:
Safe          22302912 bytes
Acceptable   11839730 bytes
Fun          19269392 bytes
Unrated      816 bytes
```

Şekil-3: Eklenen protokollerin nDPI tarafından tespit edilmesi



Şekil-4: Otomasyon akışı

#### 4. Otomasyon

Otomasyon sürecinin akış şeması Şekil 4’de verilmiştir. Buna göre, otomasyon üç bölümden oluşur; pcap dosyası hazırlama, nDPI analizi ve dizgi işlemleri. Bunlardan ilki, daha önce her bir web sayfasında manuel olarak yapılan gezintilerden elde edilen pcap veya pcapng uzantılı dosyaların web otomasyon testleri kullanılarak gerçekleştirilmesidir. Bunun için açık kaynak kodlu Selenium web sürücüsü ve Python programlama dili kullanılarak her bir web sayfasına ait bir adet test dosyası ve html raporunun yer aldığı bir Selenium web otomasyon çerçevesi oluşturulmuştur. Web otomasyon testleri çalıştırdıktan sonra Wireshark durdurulur ve oluşan pcap dosyası kaydedilir. Böylece, otomasyon akışının ilk aşaması olan pcap dosyası hazırlama bölümü biter.

İkinci aşama nDPI analizidir. Pcap dosyası, -i anahtarı kullanılarak, komut satırından çalıştırılan ndpiReader uygulamasına girdi olarak verilir. nDPI tarafından ayrıntılı paket sonuçlarına ulaşmak için ayrıntılı mod, -v2 anahtarı kullanılır. nDPI analizi bitince terminal çıktısı .txt uzantılı bir dosyaya kaydedilir. Tüm ikinci aşama işlemleri için aşağıdaki komut satırının yazılması yeterlidir:

```
ndpiReader -i <dosya_adi>.pcapng -v2 >> <dosya_adi>.txt
```

Üçüncü aşama dizgi işlemleridir. İkinci aşamada kaydedilen metin belgesi herhangi bir Python IDE’si kullanılarak açılır. Tüm paket satırları çekilerek bir metin dosyasına kaydedilir. Daha sonra, sadece paket bilgilerini içeren bu metin belgesi

açılıp nDPI tarafından uygulama protokolü bulunamamış olan paketler tespit edilir. Bunun için paket satırları taranarak TLS.<uygulama\_protokolü> ya da DNS.<uygulama\_protokolü> gibi dizgiler içermeyip bunun yerine [proto: 91/TLS] ya da [proto: 5/DNS] dizgilerini içeren

```
41 Hostname/SNI: guvendamgasi.org.tr
42 Hostname/SNI: bam.nr-data.net
43 Hostname/SNI: assets.human2.com
44 Hostname/SNI: sync.crowdctrl.net
45 Hostname/SNI: bilyoner.webinstats.com
46 Hostname/SNI: s.thebrighttag.com
47 Hostname/SNI: secure.adnxs.com
48 Hostname/SNI: vars.hotjar.com
49 Hostname/SNI: vars.hotjar.com
50 Hostname/SNI: se.semasio.net
51 Hostname/SNI: s.ad.smaato.net
52 Hostname/SNI: ad.360yield.com
53 Hostname/SNI: pixel.onaudience.com
54 Hostname/SNI: sslwidget.criteo.com
55 Hostname/SNI: pm.w55c.net
56 Hostname/SNI: sync.crowdctrl.net
57 Hostname/SNI: an.yandex.ru
58 Hostname/SNI: criteo-sync.teads.tv
59 Hostname/SNI: bam.nr-data.net
60 Hostname/SNI: ads.betweendigital.com
61 Hostname/SNI: creativecdn.com
```

satırlar varsa bu satırlardaki "Hostname/SNI:<alan\_adi>" dizgileri çekilir. Böylece, nDPI tarafından uygulama protokolü tespit edilmemiş olan paketler bulunarak yeni alan adları ayrı bir metin dosyasına kaydedilmiş olur. Dizgi işlemleri için yazılan betiğin örnek bir çıktısı Şekil-5’de verilmiştir. Bu şekle göre, 45. satırdaki bilyoner.webinstats.com alan adı yedi protokolden Bilyoner’e ait yeni bir alan adı olarak karşımıza çıkmaktadır.

Şekil-5: Dizgi işlemleri sonucu elde edilen çıktı bir örneği

Son olarak, yeni elde edilen alan adlarının kural olarak eklenip eklenmeyeceği belirlenerek nDPI’da güncelleme işlemleri yapılır. Burada önemli bir nokta, elde edilen yeni alan adları içerisinde, ndpi\_protocol\_ids.h dosyasında tanımlı eşsiz bir numaraya sahip olmayan -reklam veya CDN ağı gibi- uygulama protokollerine ait olan alan adlarının da bulunmasıdır. Bunun sebebi, bu tür protokollerin kategori bazlı tanımlanmasından dolayı paket içeriklerinde [proto: 91/TLS] ya da [proto: 5/DNS] dizgilerinin yer almaya devam etmesidir. Bu alan adları, nDPI’da yoksa, nDPI soyut kategorilerinden CUSTOM\_CATEGORY\_ADVERTISEMENT ve NDPI\_PROTOCOL\_CATEGORY\_MEDIA kullanılarak eklenebilir. Bu şekilde 6 adet CDN ağına ait, 74 adet reklam ve takip amaçlı yazılımlara ait alan adları bulunmuştur. Bu alan adları nDPI’in Şubat 2023’te çıkan son sürümü nDPI-4.6’da da bulunmamaktadır. Dolayısıyla, yanlış kategorize edilen alan adları probleminin devam etmemesi için bu çalışma kapsamında bulunan reklam ve takip amaçlı yazılım içeren alan adlarının nDPI kütüphanesine eklenmesine olan ihtiyaç devam etmektedir. nDPI-4.6’da –önceki 4.2 ve 4.4 sürümlerinin aksine- reklam, takip veya veri analitiği yapan şirketlere ait alan adları soyut kategori olarak ele alınmamıştır. Bu türdeki alan adlarının özelinde kullanılabilecek bir protokol adı ve buna karşılık bir kimlik numarası belirlenmiştir. Bu sebeple, yeni sürümde

NDPI\_PROTOCOL\_ADS\_ANALYTICS\_TRACK protokol adı ve NDPI\_PROTOCOL\_TRACKER\_ADS protokol türü kullanılarak aşağıdaki kural şablonuna göre ekleme yapılmalıdır:

```
{ "<alan_adi>", "ADS_Analytic_Track",  
NDPI_PROTOCOL_ADS_ANALYTICS_TRACK,  
CUSTOM_CATEGORY_ADVERTISEMENT,  
NDPI_PROTOCOL_TRACKER_ADS,  
NDPI_PROTOCOL_DEFAULT_LEVEL }
```

## 5. Tartışma

Bu çalışmada, Türkiye’de yüksek kullanıcı sayısına sahip ve oldukça popüler e-ticaret siteleri ele alınmıştır. Bu tür uygulamaların ağdaki varlığının tespitinin üç yönden önemli olduğu söylenebilir.

Birincisi, daha etkili bir bant genişliği yönetimi için bir yerel ağda bulunan uç sistemlerin bu tür sitelere erişiminde, belirli dönemlerde veya saatlerde, sistem yetkilileri tarafından ölçeklendirilmesi ihtiyacı doğabilir. Kayıt yenileme başvuru veya sonuç öğrenme dönemlerinde yüksek yoğunluk yaşayan bir kampüs ağı bu duruma örnek olarak verilebilir.

İkincisi, e-ticaret sitelerinin kişisel veya bölgesel kullanım oranlarına ait istatistiksel bilgiler çeşitli yönlerden faydalı olabilir. Örneğin, bir ISP’ye ait CDN ağında kullanımda olan uygulama protokollerinin hacmi göz önüne alınarak önceliklendirme tarifeleri uygulanabilir ve böylece hizmet kalitesi artırılabilir. Nitekim dört pcap dosyası üzerinden yapılan incelemelerde, Trendyol bağlantılarının akış sırası takip edildiğinde TCP bağlantı kurulumunun başında SYN ile başlayan üç yönlü el sıkışma ve hemen ardından onu takip eden TLS bağlantı kurulumu gerçekleştikten sonra bir CDN ağına bağlanıldığı tespit edilmiştir.

Üçüncü ve son olarak, gizlilik ve güvenlik gerekçeleriyle bu sitelerin yüksek güvenlik gereksinimi duyulan bazı ağlarda kısıtlanması ve hatta engellenmesi gerekebilir. Bu duruma en somut örneklerden birisi Yemeksepeti’ne ait bir web uygulama sunucusuna erişilerek gerçekleştirilen 18 Mart 2021 tarihli siber saldırıdır. Ele geçirilen veriler arasında ad, soyad, doğum tarihi, telefon numarası, eposta adresleri, adres bilgisi ve SHA-256 algoritması ile özeti alınmış şifreler bulunmaktadır. Şifrenin kendisi bilinmese bile bu tür veriler kötü niyetli kişiler için aslında oldukça değerli bir kaynaktır. Örneğin, Keeper Security firmasının Amerika’da tam zamanlı çalışan statüsünde bulunan ve işle ilgili çevrimiçi hesabına bir şifre yardımıyla giren 1000 kişi üzerinde yaptığı araştırmaya göre, bu çalışanların %44’ünün hem kişisel ve hem de iş ile alakalı hesaplarının şifresinin aynı olduğu sonucuna ulaşılmıştır [19]. Ayrıca, uzmanlara göre tüm internet kullanıcılarının yaklaşık %50’sinin bütün çevrimiçi hesaplarında hala aynı şifreyi kullandıkları tahmin edilmektedir [20]. Yemeksepeti’nde yaşanan veri hırsızlığı olayına dönecek olursak, özeti alınmış şifreleri elinde tutan bir saldırgan Yemeksepeti uygulamasına –kimlik doğrulama protokolünde de özeti alınmış şifrelerin oturumdan oturuma sabit tutulması gibi bir zayıflık varsa- Özet Geçirme (Pass the Hash - PtH) saldırısı gerçekleştirebilir. PtH saldırısı, bir saldırganın bir sistemden ele geçirdiği özeti alınmış şifreyi, sanki yeni bir oturum açıyormuş gibi, o sistemde yer alan kimlik doğrulama protokolüne kendi kimliğini doğrulamak

amacıyla kullanıp sisteme sızmasıdır. Şifre özetlerinin farklı oturumlarda sabit kaldığı zayıf bir sistemde şifre değişmedikçe özetler de değişmeyeceği için, şifre değişimine kadar, saldırganın sistemde fark edilmeden dolaşıp zaman kazanması mümkündür [21]. Böylece, özeti alınmış şifreleri deneyerek hesaplara, buradan da şifrenin açık haline ve belki de buradan da kritik yerlerde çalışan kişilerin iş yerlerinde kullandıkları hesaplara veya şirkete ait hassas verilerin bulunduğu sunuculara ulaşabilecektir. Diğer bir yandan, sadece doğum tarihi ve isim bilgilerinden bile şifreleri bulmaya yönelik bir saldırı yapılması mümkündür. Yapılan bir araştırmaya göre [22], Amerika’daki internet kullanıcılarının %60’a yakınının çevrimiçi hesaplarının parolasını bir isim veya doğum tarihi içeren bir dizgi oluşturmaktadır. Tüm bu saldırı metotları ile birlikte Yemeksepeti’nin KVKK’ya resmi olarak bildirdiği - 21 milyon 504 bin 83 rakamı ile Türkiye nüfusunun çeyreği eden- saldırıdan etkilenen kişi sayısı ve bu veri hırsızlığının saldırıdan ancak bir hafta sonra -25 Mart 2021’de- tespit edildiği göz önüne alındığında, bunun oldukça ciddi bir güvenlik açığı oluşturduğunu söylemek mümkündür. Diğer yandan, ele alınan Trendyol, N11 ve Hepsiburada şirketlerine ait web sitelerinde -sadece Yemeksepeti örneğindeki gibi ad, soyad, doğum tarihi, telefon numarası, eposta adresleri ve adres bilgisi gibi veriler değil- ürün arama, konum bilgisi, satın alma bilgileri gibi bir araya getirildiklerinde kişiler hakkında daha birçok çıkarımda (tercihleri, eğilimleri, sağlık durumu gibi) bulunmayı sağlayacak veriler de yer almaktadır. Elektronik ortamda gözetim ve takip amaçlı da toplanabilen bu veriler üçüncü parti şirketler ile de paylaşılabilirdiğinden önemli konumlarda çalışan kişiler ve çalıştıkları kurumlarda kısıtlayıcı önlemler alınması gerekebilir.

## 6. Sonuç

Bu çalışmada paketlerin sadece başlık bilgilerine değil beş katmanlı ağ modelindeki en üst katman olan uygulama katmanındaki yük içeriklerinin analiz edilerek uygulama protokollerinin tespit edilebildiği bir yapı olan derin paket inceleme yöntemi ele alınmıştır. Bu çalışmada, bu modeli kullanan açık kaynak kodlu nDPI aracına yeni protokoller eklenmiştir. Ele alınan protokoller şifreli bağlantı kullandıkları için sadece -SSL el sıkışma fazında bağlantı şifreli hale gelmeden önceki- sunucu adı ve ana bilgisayar adı bilgilerinin olduğu veriler kullanılarak Trendyol, Hepsiburada, Nesine.com, N11, Bilyoner, GittiGidiyor ve Yemeksepeti şirketlerinin kullandıkları web sitelerinin ağ trafiğinde tanımlanması sağlanmıştır. İleride oluşabilecek yeni ve güncel alan adlarının çekilebilmesi için kısmi bir otomasyon yazılmıştır. Bu web sitelerinde veriler toplanırken 6 adet CDN ağına ait, 74 adet reklam ve takip amaçlı yazılımlara ait alan adları bulunmuştur. Bu yazılımlar ilgili web sitesine bağlandıktan hemen sonra çalışmakta ve bağlantı sonuna kadar etkileşim devam etmektedir. Bu nedenle, internet üzerinden kullanıcı verilerinin işlenmesini ve belki de üçüncü parti şirketlere iletimini gerçekleştirme potansiyeline sahip bu tür sitelerin kullanımında, gizlilik ihtiyacı duyulan hassas verilerin yer aldığı kurumlar veya bu kurumlarda üst düzey yönetici pozisyonundaki yetkili kişiler için, kısıtlamaya gidilmesi ihtiyacı doğabilir. Bu kısıtlama neticesinde Yemeksepeti gibi olası saldırı durumları veya bu bilgilerin



doğrudan e-ticaret sitesinin kendisi tarafından kullanılması veya üçüncü parti şirketler ile paylaşması kaynaklı oluşabilecek güvenlik zafiyetlerinin önüne geçilmesi sağlanabilir. Bu çalışmada, ayrıca, nDPI tarafından yanlış kategorize edildiği tespit edilen ağ paketleri için yeni kural tanımlarının yapılarak bu gibi durumların düzeltilmesi sağlanmıştır.

Gelecek çalışmalarda ise sahte site ve link tuzaklarının DPI yöntemleri kullanılarak tespitinin sağlanması ile alana katkıda bulunulması hedeflenmektedir.

## Kaynakça

- [1] Vailshery, L. S., "IoT connected devices worldwide 2030", <https://www.statista.com/statistics/1183457/iot-connected-devicesworldwide/>, 2021.
- [2] Brook, C., "What is Deep Packet Inspection? How it Works, Use Cases for DPI, and More", <https://digitalguardian.com/blog/what-deep-packet-inspection-how-it-works-use-cases-dpi-and-more>, 2018.
- [3] Renals, P. and Jacoby, G. A., *Blocking skype through deep packet inspection*, 42nd Hawaii International Conference on System Sciences, 2009, pp. 1-5, doi: 10.1109/HICSS.2009.90.
- [4] Saputra, F. A., Nadhori, I. U. and Barry, B. F., *Detecting and blocking onion router traffic using deep packet inspection*, 2016 International Electronics Symposium (IES), 2016, pp. 283-288, doi: 10.1109/ELECSYM.2016.7861018.
- [5] Radityatama, G. A., Lim, C. and Ipung, H. P., *Toward full enterprise software support on nDPI*, 6th International Conference on Information and Communication Technology (ICoICT), 2018, pp. 1-6, doi: 10.1109/ICoICT.2018.8528792.
- [6] ntop, "nDPI", <https://github.com/ntop/nDPI.git>, 2023.
- [7] Wireshark, "About Wireshark", <https://www.wireshark.org/about.html>, 2023.
- [8] Wikibooks, "Intellectual Property and the Internet/Deep packet inspection", [https://en.wikibooks.org/wiki/Intellectual\\_Property\\_and\\_the\\_Internet/Deep\\_packet\\_inspection](https://en.wikibooks.org/wiki/Intellectual_Property_and_the_Internet/Deep_packet_inspection), 2023.
- [9] Xu, C., Chen, S., Su, J., Yiu, S. M. and Hui, L. C. K., *A survey on regular expression matching for deep packet inspection: Applications, algorithms, and hardware platforms*, IEEE Communications Surveys & Tutorials, 18(4), 2016, pp. 2991-3029, doi: 10.1109/COMST.2016.2566669.
- [10] Papadogiannaki, E. and Ioannidis, S., *A survey on encrypted network traffic analysis applications, techniques, and countermeasures*, ACM Computing Surveys, 2021, 54(6), pp. 1-35, <https://doi.org/10.1145/3457904>
- [11] Mellia, M., Pescapè, A. and Salgarelli, L., *Traffic classification and its applications to modern networks*, Computer Networks (Vol. 53), 2009, pp. 759-760, 10.1016/j.comnet.2008.12.007.
- [12] Bendrath, R., *Global technology trends and national regulation: Explaining variation in the governance of deep packet inspection*, International Studies Annual Convention, New York, 2009, pp. 15-18.
- [13] ntop, "nDPI 4.6 (Feb 2023)", <https://github.com/ntop/nDPI/releases/tag/4.6>, 2023.
- [14] Deri, L., Martinelli, M., Bujlow, T. and Cardigliano, A., *nDPI: Open-source high-speed deep packet inspection*, 2014 International Wireless Communications and Mobile Computing Conference, 2014, pp. 617-622, doi: 10.1109/IWCMC.2014.6906427.
- [15] Attia H. B., "q - Run SQL directly on CSV or TSV files", <http://harelba.github.io/q/>.
- [16] Fast Company Türkiye, "En büyük 100 internet şirketi", <https://fastcompany.com.tr/calisma-hayati/en-buyuk-100-internet-sirketi/>, 2020.
- [17] DSM Grup Danışmanlık İletişim ve Satış Ticaret A.Ş., <https://www.dsmgrup.com/>.
- [18] E-Şirket, <https://e-sirket.mkk.com.tr/esir/>.
- [19] Whitney, L., "How poor password habits put your organization at risk", <https://www.techrepublic.com/article/how-poor-password-habits-put-your-organization-at-risk/>, 2021.
- [20] Crafford, L., "7 Bad Password Habits to Break Now", <https://blog.lastpass.com/2021/01/7-bad-password-habits-to-break-now-2/>, 2021.
- [21] Delinea, "What is a Pass-the-Hash attack?", <https://delinea.com/what-is/pass-the-hash-attack-ptb>, 2022.
- [22] Google and The Harris Poll, "The United States of P@ssw0rd\$", <https://storage.googleapis.com/gweb-uniblog-publish-prod/documents/PasswordCheckup-HarrisPoll-InfographicFINAL.pdf>, 2019.

# Yazılım Hata Tahmininde Farklı Alt Örnekleme ve Üst Örnekleme Yöntemlerinin Karşılaştırılması

## Comparison of Different Oversampling and Undersampling Methods in Software Defect Prediction

Özge ŞEN KAYA  
Eskişehir Osmangazi Üniversitesi  
Bilgisayar Mühendisliği  
Eskişehir, Türkiye  
ozgeesenn1995@gmail.com  
ORCID: 0000-0002-4713-7536

Sinem OZKURT KESER  
Eskişehir Osmangazi Üniversitesi  
Bilgisayar Mühendisliği  
Eskişehir, Türkiye  
sbozkurt@ogu.edu.tr  
ORCID: 0000-0002-8013-6922

### Öz

Bilgisayarları ve makineleri çalıştırmak üzere belirli fonksiyonların işletilebilmesi için kullanılan komutlar bütünü yazılım olarak adlandırılmaktadır. Günümüzde birçok alanda yapılan faaliyetler ve kullanılan uygulamalar, içerisinde farklı algoritmalarla tasarlanmış yazılımlar barındırır. Bu yazılımların kusursuz ve ihtiyaçları karşılayacak şekilde olması büyük önem teşkil etmektedir. Yazılımın kalitesi, yazılımın içerisinde hata barındırmaması hem yazılımı geliştiren kişilerin hem de yazılımı kullanan son kullanıcıların önem verdiği konulardır. Yazılım hata tahmini doğası gereği dengesiz sınıf problemi içerir. Bu çalışmada, öncelikle dengesiz sınıf problemi çözülmeye çalışılmıştır. Bu doğrultuda, farklı alt örnekleme ve üst örnekleme yöntemleri, literatürde araştırmacıların kullanımına açık NASA'nın PROMISE veri deposundan alınan CM1, KC1, KC2, JM1 ve PC1 veri kümelerinin üzerinde uygulanmıştır. Yazılım hata tahmini aşamasında ise farklı sınıflandırma algoritmaları karşılaştırılarak her bir veri kümesi için en uygun algoritma belirlenmiştir. Deney sonuçlarında on farklı örnekleme yöntemi ile veri kümelerindeki dengesiz sınıf problemi giderilmiş; on üç farklı sınıflandırma algoritması ile sınıflandırma işlemi yapılmıştır. 0,92 oranında AUC ölçütü ile en iyi sınıflandırma sonucu PC1 veri kümesinde elde edilmiştir. Bu çalışma ile yazılım hata tahmininde örnekleme yöntemleri ve uygun sınıflandırıcılar ile hata tahmininin başarımının daha iyi olabileceği gösterilmiştir. Elde edilen sonuçlar, literatürde yapılan çalışmalar ile karşılaştırılarak önerilen yöntemin üstünlüğü ve etkinliği kanıtlanmıştır.

**Anahtar sözcükler:** Yazılım hata tahmini, Sınıf dengesizliği problemi, Alt örnekleme yöntemleri, Üst örnekleme yöntemleri, Sınıflandırma

### Abstract

The set of commands used to operate certain functions to operate computers and machines is called software. Today, activities and applications used in many fields contain software designed with different algorithms. It is of great importance that these softwares are perfect and in a way that meets the needs. The quality of the software and the absence of errors in the software are issues that both the developers of the software and the end users of the software attach importance to. Software defect prediction inherently involves an imbalanced class problem. In this study, first of all, the imbalanced class problem was tried to be solved. In this direction, different undersampling and oversampling methods were applied on the CM1, KC1, KC2, JM1 and PC1 datasets taken from NASA's PROMISE data repository, which is open to researchers in the literature. In the software defect prediction phase, different classification algorithms were compared and the most suitable algorithm was determined for each data set. In the experimental results, the imbalanced class problem in the datasets was resolved with ten different sampling methods; classification was done with thirteen different classification algorithms. With an AUC of 0.92, the best classification result was obtained in the PC1 dataset. With this study, it has been shown that the performance of defect prediction can be better with sampling methods and appropriate classifiers in software defect prediction. The results obtained were compared with the studies in the

*literature and the superiority and effectiveness of the proposed method were proven.*

**Keywords:** Software defect prediction, Class imbalance problem, Oversampling methods, Undersampling methods, Classification

## 1. Giriş

Bilgisayar ve bilişim teknolojileri günlük yaşamda birçok alanda kullanılan ve kullanımı da her geçen gün yaygınlaşan teknolojilerdir. Bu teknolojilerin işletilebilmesi ve içerisinde yer alan belirli fonksiyonlarının çalışabilmesi için kullanılan komutlar bütünü yazılım olarak adlandırılmaktadır. Her yazılım farklı ihtiyaçları karşılaması için farklı algoritmalar ile tasarlanır. Bu yazılımların kusursuz ve ihtiyaçları karşılayacak şekilde olması büyük önem teşkil eder. Yazılım kalitesi, yazılımın içerisinde hata barındırmaması hem yazılımı geliştiren kişilerin hem de yazılımı kullanan son kullanıcının önem verdiği konulardır.

Yazılım kalite standartları olarak adlandırılan ISO 25010, yazılım kalitesini belirlemede temel öğedir. Bir yazılım kalitesi standart içerisinde belli başlıklara bölünerek değerlendirilmiştir [1]. Taşınabilirlik, bakım yapılabilirlik, güvenilirlik, güvenlik, başarımlık, uyumluluk, işlevsel uygunluk ve kullanılabilirlik gibi sekiz kalite özelliğini içerisinde barındırır. Bu özellikler kapsamında bir yazılımın hatasız ve doğru çalışır durumda olması kalitesini göstermektedir.

Yazılımlar, içerisinde bazı mantıksal kusurlar barındırabilir. Bu kusurlar yazılımın kalitesini ve son kullanıcının yazılımı tercih etme düzeyini düşürebilir. Bir yazılımın kalitesi, yazılım son kullanıcıya ulaşmadan önce yapılan yazılım testleri ile ölçülür. Bu testler ile yazılımın ana fonksiyonlarının çalışabilirliği, yazılımda meydana gelebilecek hata durumları tespit edilir. Gerçekleştirilen yazılım sınamaları ile yazılımda ortaya çıkabilecek hatalı durumların erken aşamada önüne geçilmesi büyük önem taşımaktadır.

Yazılım hatalarını otomatik olarak tespit edebilmek amacıyla makine öğrenmesi yöntemleri önem teşkil etmektedir. Makine öğrenmesi yöntemleri birçok alanda kullanılan, literatür çalışmalarında da oldukça tercih edilen yöntemlerdir. Makine öğrenmesi yöntemleri insan aracılığı olmadan otomatik olarak büyük oranda veri analizini sağlamaktadır [2]. Yazılım test aşamasında makine öğrenmesi yöntemleri de birçok literatür çalışmasında incelenmiş oldukça popüler bir konudur. Literatür çalışmaları incelendiğinde farklı makine öğrenmesi yöntemleri ile yazılım hata tahmininde başarılar elde edilmiştir.

Bu çalışmada, yazılım hatalarının tespit edilmesi problemi üzerinde makine öğrenmesi yöntemleri ile çalışılmıştır. NASA'nın PROMISE veri deposunda bulunan CM1, KC1, KC2, JM1 VE PC1 veri kümelerinde yer alan yazılım hatalı ve hatasız yazılım kayıtları çalışmanın veri kaynağını oluşturmuştur. Bu veri kümelerinde hatalı ve hatasız kayıtların eşit dağılması sınıf dengesizliği problemini meydana getirmiştir. Dengesiz sınıf problemini çözmek için alt ve üst örnekleme yöntemleri uygulanmıştır. Bu doğrultuda Sentetik Azınlık Örnekleme Arttırma Yöntemi (SMOTE – Synthetic Minority Oversampling Technique), Rastgele üst örnekleme, Borderline SMOTE,

SVM- SMOTE, ADASYN, SMOTE + Tomek, SMOTE-NC, Rastgele alt örnekleme yöntemi, Küme Merkezli Örnekleme (Cluster Centroid), SMOTE+ENN yöntemleri kullanılmıştır. Sınıflandırma aşamasında ise K-En Yakın Komşu Algoritması, Destek Vektör Makineleri, Karar Ağacı Algoritması, Rastgele Orman Algoritması, Adaboost, Gradyan Artırma Algoritması, Ekstra Gradyan Artırma algoritması, Hafif Gradyan Artırma algoritması, Catboost algoritması, Naive Bayes Algoritması, Lineer Diskriminant Analizi, Kuadratik Diskriminant Analizi, Ekstra Ağaçlar algoritmaları ile deneyler gerçekleştirilmiştir. Doğruluk ve AUC/AUROC başarımlık ölçütleri kullanılarak algoritmalar değerlendirilmiştir. Deneylerde, her bir veri kümesi için en iyi başarımlık gösteren algoritmalar tespit edilmiştir.

## 2. Yayın İncelemesi

Yazılımda yer alan hataları tahmin etmek hem yazılımın kalitesinin artırılması hem de son kullanıcıya ulaşmadan önce doğru ve istenilen şekilde sunmak açısından önemlidir. Yazılım hata tahmininde sınıflandırma, derin öğrenme ve makine öğrenmesi gibi yöntemler kullanılır. Makine öğrenmesi algoritmaları kullanılarak sınıflandırma işlemi yapan birçok çalışma literatürde yer almaktadır.

Aydilek [3] SMOTE algoritması ile NASA PROMISE veri deposunda yer alan veri kümelerindeki sınıf dengesizliğinin iyileştirilmesi üzerinde çalışmıştır [1]. Bu çalışmada çeşitli karar ağacı tabanlı algoritmalar kullanarak iyileştirilmiş bilgi kazancı sonuçları elde etmiştir. Yazılım hata tahmininde kullanılan metrikler ile bilgi kazancı yönünden daha yüksek sayısal değerli sonuçları elde edilmiştir. Bu çalışma ile, kural tabanlı sınıflayıcı ve karar ağaçlarının başarımlarının yazılım hata tahmininde ön plana çıktığı ve bazı eksikliklerin de mevcut olduğu görülmüştür. Bu eksikliklerin ve problemlerin giderilmesi için bu çalışma yapılarak yüksek başarımlık sonuçlar literatüre katkı sağlamıştır.

Çetiner [4] yazılım hata tahmin sistemleri hakkında on farklı makine öğrenmesi algoritmasını kullanarak bu algoritmaların karşılaştırmalı analizini yapmıştır. Bu analiz için NASA'nın PROMISE veri kümesinden CM1, KC1, KC2, JM1 ve PC1 veri kümelerinden yararlanmıştır. Yapılan çalışmada veri setleri ön işleme sonrasında normalizasyon işleminden geçirilmiş daha sonra çapraz doğrulama yöntemi ile makine öğrenmesi algoritmaları eğitilmiştir. Çalışmada doğruluk başarımlık ölçünü kullanarak sonuçlar kıyaslanmıştır. Yapılan çapraz doğrulama yaklaşımı ile rastgele orman öğrenme modelinin birleştirilmesinde elde edilen sonuç, bu şekilde önerilen diğer modellere göre daha iyi bir doğruluğa ulaşmıştır.

Sun ve ark. [5] tarafından yazılım hata tahmini veri kümesinde yer alan sınıf dengesizliği problemi örnekleme, maliyete duyarlı örnekleme, torbalama ve arttırma gibi yöntemler ile modellenmiştir. Makine öğrenmesi algoritmaları ile yazılım hata tahmini yapmıştır. Deneylerde NASA veri kümesi üzerinde 4 farklı sınıflandırma algoritması karşılaştırılmıştır.

Önerilen yöntemin rastgele üst örnekleme, rastgele alt örnekleme, SMOTE, maliyete duyarlı öğrenme gibi yöntemlerden çok daha önemli ölçüde iyi başarımlık gösterdiği sonucu elde edilmiştir.

Eivazpour ve ark. [6] yazılım hata tahmininde veri kümesinin dengesiz dağılımının makine öğrenimi algoritmaları için zorlayıcı bir etkeni olduğunu belirterek, veri dengesizliği problemi üstünde durmuştur. Bu çalışmada, veri kümesindeki azınlık sınıftan örnekleme teknikleri ile artırılmış bir eğitim kümesi elde edilmiştir. Dengesiz veri kümesinin örnekleme için Rastgele Örnekleme, Sentetik Azınlık Örnekleme Arttırma Yöntemi (SMOTE), Borderline-SMOTE ve ADASYN gibi örnekleme teknikleri kullanılmıştır. Daha sonra önerilen yöntemde hangi modelin kullanılan tekniklerden hangisi ile daha iyi başarı elde ettiği analiz edilmiştir. Elde edilen sonuçlarda önerilen yöntemin, literatürdeki örnekleme yöntemlerine göre daha iyi başarı gösterdiği görülmüştür.

Malhotra ve ark. [7] NASA PROMISE veri kümesi üzerindeki sınıf dengesizliği problemi için SMOTE, SVM-SMOTE ve ADASYN yöntemlerini kullanmış, daha sonra elde edilen sonuçlarda Lineer Diskriminant Analizi yöntemi ile verilerini iyileştirmiştir. Bu veri kümesi üzerinde ise Naive Bayes ve Destek Vektör Makineleri (SVM) sınıflandırma algoritmaları ile yazılım hataları tahmin edilmiştir. Yapılan çalışmada elde edilen sonuçlar ile, SVM-SMOTE yönteminin Lineer Diskriminant Analizi ile iyileştirilmesi mekanizmasının yapılan tahmin kalitesini iyileştirmede önemli bir katkısı olduğu sonucuna varılmıştır.

Choirunnisa ve ark. [8] yapmış oldukları çalışmada NASA veri kümesi kullanılmış ve bu veri kümesinde ön işleme olarak veri dengeleme yöntemini kullanmışlardır. Ön işleme yapılmış veri kümesinde sınıf dengesizliği problemi için SMOTE, Borderline SMOTE, Safe-SMOTE ve A-SUWO (adaptive semi-supervised oversampling) yöntemleri uygulanmıştır. Veri kümesindeki sınıf dengesizliği problemi iyileştirildikten sonra K-En Yakın Komşu Algoritması, SVM, Rastgele Orman ve J48 sınıflandırma algoritmaları ile yazılım hata tahmininde bulunulmuştur. En gelişmiş veri dengeleme tekniği olarak nitelendirdikleri A-SUWO yöntemi ile yazılım hata tahmininde yeni bir yaklaşım önermişler ve bu yaklaşımı denetimli sınıflandırma algoritmaları ile birleştirmişlerdir.

Elahi ve ark. [9] çalışmalarında veri kümesindeki sınıf dengesizliği problemi için rastgele üst örnekleme ve komşu temizlik kuralı yöntemlerini önermişlerdir. Bu yöntemler ile veri kümesinin sınıf dengesizliği problemi çözüldükten sonra K-En Yakın Komşu Algoritması, Naive Bayes, Karar Ağacı, Lineer Regresyon, Ortalama Mutlak Hata (MAE) algoritmaları ile sınıflandırma işlemi yapılmıştır. Çalışmada elde edilen başarımların sınıf örtüşme yönteminin kullanıldığı veya kullanılmadığı ve yeniden örnekleme yönteminin de kullanılmadığı çalışmalara kıyasla çok daha iyi başarımlar gösterdiği açıklanmıştır.

Goyal [10] yazılım hata tahmininde sınıf dengesizliği problemini alt örnekleme yöntemi ile gidermiştir. Yapılan çalışmada eğitim kümesinde sınıf dengesizliği problemini gidermek için çeşitli alt örnekleme yöntemlerini kullanmıştır. Eğitim veri kümesinde uygulanan bu yöntemler ile sınıflandırma modeli eğitildikten sonra test veri kümesi kullanılarak sınıflandırma modelinin başarımları elde edilmiştir. Sınıflandırma aşamasında Yapay Sinir Ağları (ANN), Destek Vektör Makineleri (SVM), Karar Ağacı (DT), KNN, Naive Bayes

(NB) algoritmaları kullanılmıştır. Bu çalışmada sınıf dengesizliği problemini etkili bir şekilde ele alarak yeni bir alt örnekleme yöntemi önerilmiş ve literatüre katkıda bulunulmuştur. Bu yöntem 3 alt örnekleme tekniği ile kombinasyon yapılarak önerilmiştir. Bu yöntemler rastgele alt örnekleme, tomek-link örnekleme ve iskalama alt örnekleme yöntemleridir.

Jacob ve ark. [11] NASA Promise veri kümesindeki sınıf dengesizliği problemi için öncelikli olarak veri kümesine 2 farklı ön işleme uygulamıştır. Bu ön işlem yöntemleri "Wrapper-Based Feature Selection" ve "Heuristic-Based Feature Selection" yöntemleridir. Sınıf dengesizliği problemi, bu yöntemler ile çözüldükten sonra; Naive Bayes, Lineer Regresyon, Çok Katmanlı Perceptron, Destek Vektör Makineleri, K-En Yakın Komşu Algoritması, Torbalama, AdaBoost, Rastgele Orman (RF), Karar Ağacı algoritmaları ile sınıflandırma işlemi yapılmıştır. Bu çalışma ile oylama tabanlı topluluk modellerinin daha küçük kümeleri daha fazla etkiledikleri ve daha iyi başarımlar gösterdikleri çıkarımı yapılmıştır.

Çetiner ve ark. [12] NASA Promise veri kümesi içerisinde yer alan CM1, KC1, KC2, JM1 ve PC1 veri setleri ile 10 farklı makine öğrenmesi algoritmasından yararlanarak yazılım hata tahmininde makine öğrenmesini algoritmalarını kıyaslamıştır. Bu çalışmada SVM, KNN, NB, RF, Ekstra Ağaçlar, Adaboost, Gradyan Arttırma, Torbalama ve Multi Layer Perceptron algoritmalarını karşılaştırmışlardır. Yapılan karşılaştırma sonucunda algoritmaların yazılım hata tahmininde kullanılan veri setlerinden PC1 veri kümesi için daha ortalama doğruluk oranlarının elde edildiğini göstermiştir.

Pelayo ve ark. [13] yapmış oldukları çalışmada dengesiz veri kümelerinde makine öğrenmesi üzerinde durarak yazılım hata tahmini konusunu incelemiştir. Bu çalışmada SMOTE yöntemine odaklanılarak veri kümesinin iyileştirilmesi konusunda çalışılmıştır. Bu çalışmada, yazılım hata tahmininde çok az ilgi gören, dengesiz verileri öğrenmek için kullanılan katmanlaşma (stratification) incelenmiş; SMOTE yönteminin hata eğilimli modüllerin iyileştirilmesinde rolü araştırılmıştır.

Aleem ve ark. [14] makine öğrenmesi tekniklerinin yazılım hata tahmininde yararlı olduğunun üstünde durarak kullanıma açık veri setlerinde karşılaştırmalı analiz yapmışlardır. Makine öğrenmesi yöntemlerinin yazılım hata tahmininde iyi başarımlar gösterdiğini açıklamışlardır. Nöro-bulanık (neuro-fuzzy) teknikleri ve yazılım araçlarının test senaryolarını etkinliğini artırmada önemini ortaya koyarak doğru sonuçlar elde etmede birden fazla tekniğin birleştirilebileceği bu çalışmada önerilmiştir. Hata tespitlerinde destek vektör makineleri, MLP ve torbalama tekniklerinin de veri kümelerinde iyi başarımlar gösterdiği ortaya konmuştur.

Wang ve ark. [15] sınıf dengesizliği problemini ele alarak yazılım hata tahminine fayda sağlayıp sağlamayacağını tartışmışlardır. Yeniden örnekleme, topluluk algoritmaları olmak üzere farklı sınıf dengesizliği yöntemlerini araştırmışlardır. Bu çalışmada farklı sınıf dengesizliği yöntemlerinin araştırılarak incelenmesi yazılım hata

tahmininde literatüre katkı sağlamıştır.

İbrahim ve ark. [16] arama algoritmaları ve rastgele orman algoritmasını kullanarak yazılım hata tahmininde yüksek başarımla elde edebilmek için bir yaklaşım önermişlerdir. Önerilen bu yaklaşımın başarımla, yayın taramasında bahsedilmiş olan diğer yaklaşımlar ile değerlendirilmiş ve önerilen yaklaşımın literatüre göre çok daha iyi sonuçlar elde ettiği görülmüştür.

Akmal ve ark. [17] yapmış oldukları çalışmada, yazılım hata tahmininde yayınlardaki çalışmalarını inceleyerek açıklamışlardır. Hazırladıkları özet tabloda, yıl bazlı yapılan çalışmalarını kullandıkları yöntemleri, teknikleri yaklaşımları analiz etmişlerdir. Bu tabloda yapılan çalışmalarını sınıflandırarak ayırmışlardır. Sınıflandırma tabanlı yaklaşımlar, kümeleme tabanlı yaklaşımlar ve topluluk öğrenmesi tabanlı yaklaşımlar olmak üzere analizi yapmışlardır. Bu çalışmanın yazılım hata tahmini çalışmalarında, kaynakların tahsisine rehberlik edecek yapıda olması literatüre katkı sağlamıştır.

Naidu ve ark. [18] yapmış oldukları çalışmada, yazılım kusurlarını gruplandırmak için karar ağacı tabanlı bir sistem önermiştir. Bu sistem içerisinde ID3 ve C4.5 algoritmalarını kullanarak sınıflandırma işlemi yapmıştır. Yapılan sınıflandırma işlemi sonrasında model madenciliği tekniği kullanarak sınıflandırılan kusur modelleri ölçmüştür. Çalışmada zaman ve maliyeti azaltmak için toplam kusur sayısına odaklanılmış ve beş farklı parametre ile sınıflandırma yapılmıştır. Model madenciliği tekniği ile yazılım kusurları sınıflandırılmıştır.

Yazılım hata tahmininde örnekleme ve sınıflandırma algoritmaları ile NASA PROMISE veri kümesini kullanan son 4 yılda yapılan çalışmalar Çizelge-1 içerisinde özetlenmiştir. Bu çizelgede örnekleme yöntemleri, sınıflandırma algoritmaları

ve başarımla metrikleri detaylandırılmıştır.

### 3. Materyal ve Yöntem

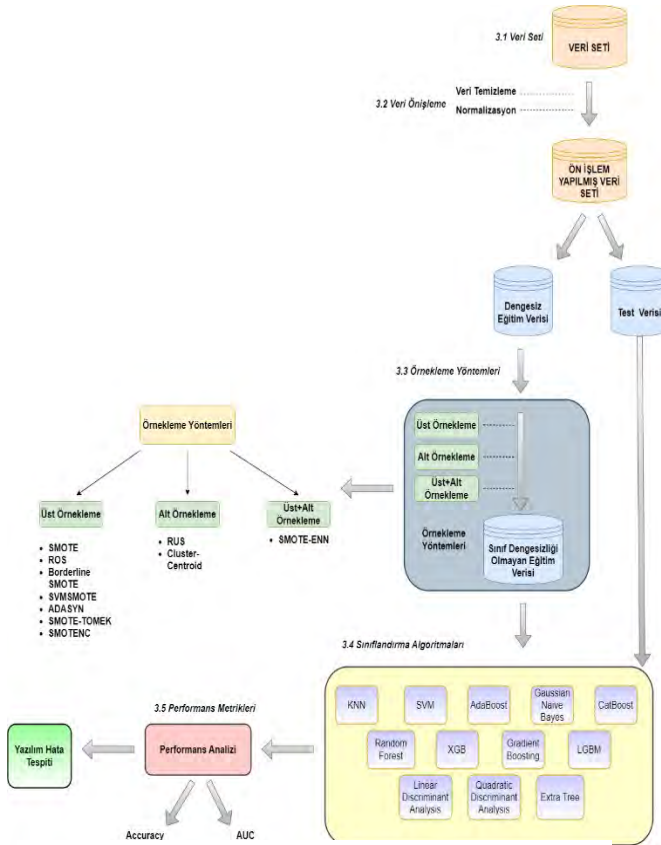
Bu bölümde çalışma kapsamında kullanılan veri kümesine ve model aşamalarına değinilip akış modeli ayrıntılı biçimde sunulacaktır. Gerçekleştirilen bu çalışmada genel anlamda veri ön işleme, örnekleme, sınıflandırma ve başarımla analizi aşamaları bulunmaktadır. Çalışmada kullanılan veri kümesi öncelikli olarak belirli ön işleme adımlarından geçirilmiştir. Bu ön işleme adımlarından sonra veri kümesi üzerinde çalışılabilecek duruma gelmiştir. Daha sonra veri kümesi eğitim ve test verisi olarak ikiye ayrılmıştır. Burada veri kümesinde bulunan sınıf dengesizliği problemi nedeniyle eğitim veri kümesinde örnekleme yöntemleri kullanılmıştır. Kullanılan örnekleme yöntemleri sınıf dengesizliği problemini gidermiş; veri setlerinden elde edilecek sonuçların daha doğru olmasını sağlamıştır. Seçilen farklı alt ve üst örnekleme yöntemleri ile sınıf dengesizliği problemi çözülen eğitim kümesine ve ana veri kümesinden alınan test veri kümesine 12 farklı makine öğrenmesi sınıflandırma algoritması ile sınıflandırma işlemi yapılmış, sınıf dengesizliği probleminde en çok kullanılan başarımla ölçünleri ile başarımla analizi yapılmıştır. Her veri kümesindeki hatalı yazılım kayıtları da göz önüne alınarak elde edilen değerler analiz edilerek kıyaslaması yapılmıştır. Veri kümelerinin hangi örnekleme yöntemi ve hangi sınıflandırma algoritması ile daha iyi başarımla gösterdiği tespit edilmiş sonuçları değerlendirilmiştir. Önerilen yöntemin model akışı Şekil-1 içerisinde ayrıntılı biçimde özetlenmiştir.

Çizelge -1: Özet Yayın Analizi

Referans	Yıl	Sınıf Dengesizliği için Kullanılan Yöntem	Veri Kümesi	Başarımla Ölçünleri	Kullanılan Algoritmalar
[3]	2018	SMOTE algoritması	NASA PROMISE	F-Ölçütü, Tutturma, Bulma, G-Ortalama, Özgüllük	JRip C4.5 PART
[4]	2020		NASA PROMISE	Doğruluk, Bulma, Tutturma	NB, Karar Ağacı, KNN, SVM, Gradyan Artırma, Artırma, RF, Ekstra Ağaç, MLP
[5]	2012	1-1, Orig, Rastgele alt örnekleme, Rastgele üst örnekleme, SMOTE, Cost, Torbalama, Artırma	NASA PROMISE	AUC	Rastgele Orman, C4.5, Ripper, Naive Bayes
[6]	2019	Rastgele üst örnekleme, SMOTE, Borderline-SMOTE, ADASYN	NASA PROMISE	AUC	Karar Ağacı, Rastgele Orman
[7]	2021	SMOTE, SVM-SMOTE, ADASYN	NASASDP	F-Ölçütü, AUC	Destek VekD Makineleri, Naive Bayes
[8]	2018	SMOTE, Borderline-SMOTE, SafeSMOTE, A-SUWO	NASA PROMISE	Doğruluk, TPR, FPR	KNN, SVM, Rastgele Orman, J48
[9]	2021	Rastgele üst örnekleme, NCL	NASA PROMISE	AUC	KNN, Naive Bayes, Karar Ağacı, Lineer Regresyon, MAE
[10]	2022	Çeşitli alt örnekleme teknikleri	NASA PROMISE	Doğruluk, AUC	Yapay Sinir Ağları (ANN), SVM, Karar Ağacı, KNN, Naive Bayes
[11]	2021	“Wrapper-Based Approach “ “Heuristic Based Approach “	NASA PROMISE	Doğruluk, Kesinlik, Duyarlılık, F-Ölçütü, AUC	Naive Bayes, Lineer Regresyon, MLP, SVM, KNN, Torbalama, AdaBoost, Rastgele Orman, Karar Ağacı

Çizelge-2: Veri Kümesi Parametreleri

Parametre	Açıklama/Formül
loc	Toplam Satır Sayısı
v(g)	Çevrimsel Karmaşıklık
ev(g)	Esas Karmaşıklık
iv(g)	Tasarımsal Karmaşıklık
N	Toplam Operatör ve İşlenen Sayısı
V	Hacim
L	Program Uzunluğu
D	Zorluk
I	Kabiliyet
E	Efor
B	Hata Sayısı
T	Zaman
IOCode	Kod Satırı Sayısı
IOComment	Yorum Satırı Sayısı
IOBlank	Boş Satır Sayısı
IOCodeAndComment	Yorumlu Kod Satır Sayısı
uniq_Op	Tekil Operatör
uniq_Opnd	Tekil İşlenen
total_Op	Toplam Operatör Sayısı
total_Opnd	Toplam İşlenen Sayısı
branchCount	Akış Grafiğindeki Dal Sayısı
Hata (Sınıf)	Evet / Hayır



Şekil 1: Model akış şeması

### 3.1 Veri Kümesi

Yazılım hata tahmininde modelleme yapabilmek için NASA PROMISE veri deposunda yer alan 5 veri kümesi kullanılmıştır. Bu çalışmada, NASA PROMISE veri deposunda yer alan 5 veri kümesinden yararlanılmıştır [19]. Veri deposu içerisinde CM1, JM1, KC1, KC2, PC1 veri kümeleri seçilmiştir. Bu veri kümelerine ait 22 parametrenin açıklaması Çizelge-2 içerisinde detaylandırılmıştır. Seçilen bu veri kümeleri içerisinde, hatalı ve hatasız olarak etiketlenmiş kayıtları barındırır. Hatalı kayıtlar ilgili kısımda değişiklik yapılması gerektiğini ifade eder. Kullanılan veri kümelerinden CM1, JM1 ve PC1 veri kümesi C yazılım dili ile; KC1 ve KC2 ise C++ yazılım dili ile geliştirilmiştir.

Çalışmada kullanılan veri kümeleri içerisinde toplam kayıt sayısı en fazla olan veri kümesi JM1 iken, toplam kayıt sayısı en az olan veri kümesi CM1'dir. Veri kümelerindeki hatalı kayıt sayıları incelendiğinde ise en fazla hatalı kayıt içeren veri kümesi JM1, en az hatalı kayıt içeren veri kümesi ise PC1'dir. Veri kümelerine ait kayıt sayıları ve hata oranları detaylı olarak Çizelge-3 içerisinde açıklanmıştır.

Çizelge-3: Veri Kümesi Hata Oranları

Veri Kümesi	Toplam Kayıt Sayısı	Hatasız Kayıt Sayısı	Hatalı Kayıt Sayısı	Hata Oranı
CM1	498	449	49	%9,83
JM1	10885	8779	2106	%19,35
KC1	2109	1783	326	%15,45
KC2	522	415	105	%20,50
PC1	1109	119	11	%8,46

### 3.2 Veri Ön İşleme

Bu çalışmada önerilen yöntem yazılım içerisinde yer alan hataların tespitini hedeflemektedir. PROMISE açık kaynaklı veri deposundan alınan veriler, örneklem yöntemleri ve makine öğrenmesi algoritmaları ile analiz edilmiş ve başarımlar elde edilmiştir. Yapılan çalışmada öncelikli olarak NASA PROMISE veri deposundan toplanan veriler analiz edilerek makine öğrenmesi yöntemlerine uygun olacak şekilde düzenlenmiştir. Daha sonra ikinci adım olarak veri kümelerine standart normalizasyon işlemi uygulanmıştır. Normalize edilmiş verilerde yer alan sınıf dengesizliği problemi incelenerek veri setlerine örneklem yöntemleri uygulanmıştır. Burada veri kümesi eğitim ve test verileri olarak 2'ye ayrılarak sadece eğitim kümesinde örneklem yöntemleri uygulanmıştır. Ön işlemden geçirilen ve eğitim/test olarak ayrılan veriler çalışmada kullanılan örneklem yöntemleri ve sınıflandırma algoritmalarına hazır hale getirilmiştir.

### 3.3 Örneklem Yöntemleri

Bir veri kümesinde sınıf etiketlerindeki gruplardan birinde bulunan gözlem sayısının diğer grupta bulunan gözlem sayısından fazla ya da az olması durumunda ortaya çıkan

problem sınıf dengesizliği problemidir. Veri kümesi içerisinde yer alan az sayıdaki veri grubu azınlık sınıfı diğer grup ise çoğunluk sınıfı olarak adlandırılır. Sınıf dengesizliği problemini çözebilmek için 3 farklı yaklaşım izlenebilir [20]. Bu yaklaşımlardan ilki verilerden yapay veriler üretmekle sınıflar arası dengesizliğin çözülmesidir. Diğer bir yaklaşım ise çoğunluk sınıfında yer alan verilerden azınlık sınıfına yakın olacak şekilde örnekler seçerek daha az örnekleme yapılmasıdır. Sınıflandırma işlemi esnasında, sınıfların eşit dağılmadığı dengesiz sınıf problemlerinde örnekleme yöntemleri kullanılmaktadır. Bir diğer yaklaşım ise azınlık sınıftan yeni azınlık sınıfları oluşturarak çoğunluk sınıfına denk gelecek şekilde verilerin artırılmasıdır. Bu çalışmada sınıf dengesizliği problemini gidermek için az örnekleme ve aşırı örnekleme yöntemleri kullanılmıştır.

### 3.3.1 Üst Örnekleme Yöntemleri

Sentetik Azınlık Örnekleme Arttırma Yöntemi (SMOTE – Synthetic Minority Oversampling Technique), en başarılı örnekleme yöntemlerinden birisi olarak bilinmektedir ve 2022 yılında geliştirilmiştir [21]. Bu yöntem, veri kümesindeki azınlık elemanlarından yeni örnekler oluşturarak eleman sayısını dengeli bir hale getirmeyi temel alır. Bu yöntem ile çoğunluk sınıfının eleman sayısında bir değişim yaşanmaz [22]. SMOTE yönteminde, her azınlık sınıfının bir örneği alınır ve bu örneğe ait k komşusunun herhangi birine ya da tümüne bakılarak sentetik örnekler oluşturulmaktadır [23]. Sentetik örnekler üretmek için interpolasyon tekniği kullanılır. Bu sentetik örnekler, sınıflandırıcının daha küçük ve daha spesifik bölümler yerine daha büyük ve daha az spesifik karar bölgeleri oluşturmasını sağlamaktadır [24].

Rastgele üst örnekleme yöntemi, sınıf dengesizliği probleminde kullanılan en eski ve en basit yöntemdir [25]. Sınıf dengesizliği içeren veri kümesinde azınlıktaki sınıftan rastgele farklı örneklerin seçilip tekrar azınlık sınıfına eklenmesi ve bu mantıkla veri kümesinin dengelenmesi akışını izleyen aşırı örnekleme yöntemlerinden biridir. Küçük bir veri kümesinde uygulanması aşırı öğrenme gibi durumlara yol açabilen rastgele örnekleme yöntemi, yaygın olarak kullanılan örnekleme yöntemlerinden birisidir.

Borderline SMOTE yöntemi, SMOTE yönteminden esinlenerek oluşturulmuş bir diğer üst örnekleme yöntemlerinden birisidir. Borderline SMOTE yönteminde, azınlıkta olan veri sınıfının sınır değerlerinde yer alan örnekler SMOTE yöntemi ile arttırılmaktadır. SMOTE yöntemi ile bu yöntem arasındaki temel fark ise, SMOTE yönteminde azınlık sınıfında yer alan tüm örnekler arttırılırken Borderline SMOTE yönteminde sınırlarda yer alan değerlerin arttırılmasıdır [26].

SVM-SMOTE yöntemi, Borderline SMOTE yönteminde olduğu gibi sınır çizgileri yakınında yer alan azınlık sınıfının örneklerini arttırmaya odaklanan bir yöntemdir. Ancak bu yöntemde SVM yöntemi ile sınır çizgilerinin yakınında yeni azınlık sınıfı örnekleri arttırılır.

SMOTE yönteminin bir varyasyonu olarak bilinen ADASYN yönteminde veri yoğunluğunun daha az olduğu alanlarda daha çok örnekler oluşturulur. Bu yöntemde sadece sınıf dengesizliği problemi çözülmeyip aynı zamanda öğrenilmesi

zor olan birimlere odaklanmak için karar sınırını uyarlamalı olarak değiştirir. Bu yöntem ile veri dağılımlarındaki öğrenme iki şekilde geliştirilmektedir. Bunlardan ilki sınıf dengesizliğinin getirdiği sapmanın azaltılması diğeri ise sınıflandırma karar sınırının zor örneklerle uyarlanabilir şekilde kaydırılmasıdır [27].

Tomek bağlantıları, Tomek tarafından geliştirilmiş yoğun en yakın komşu örnekleme tekniğinin değiştirilmiş bir biçimidir. SMOTE + Tomek yöntemi ise SMOTE ve Tomek bağlantılarının kombinasyonu olarak bilinmektedir. SMOTE yönteminin azınlık sınıfını için sentetik üretmesi ve Tomek bağlantıları olarak tanımlanan verilerin çoğunluk sınıfından kaldırılması işlemlerini birleştiren bir yöntemdir. SMOTE+Tomek, bir diğer tanımla, örnek uzayda dağıtılmış sınıfların her biri için örtüşen veri noktalarını temizlemeyi amaçlayan melez bir örnekleme tekniğidir. SMOTE yöntemi ile yüksek hızda örnekleme yapılmış sınıf kümeleri, birbirlerinin alanını işgal etmiş olabilirler. Bu durum ile SMOTE tarafından yapılan aşırı örnekleme azınlık sınıfı örneklerine Tomek bağlantıları uygulanır. Bu işleyle yöntem veri kümelerine uygulanır.

SMOTE yönteminin türlerinden biri olan SMOTE-NC hem nominal hem de sürekli özelliklere sahip veri kümeleri için tasarlanmış, verilerdeki sınıf dengesizliği probleminde çözüm olan aşırı örnekleme yöntemlerinden biridir. Veri kümesinde önemli sayıda nominal özellik olduğunda ve kategorik özelliklerle hedef sınıf arasında bir ilişki olduğunda bu yöntemin daha iyi tahmin sağladığı görülmektedir [28].

### 3.3.2 Alt Örnekleme Yöntemleri

Alt örnekleme, sınıf dengesizliği olan bir veri kümesinde azınlık sınıfındaki verileri tutarak, çoğunluk sınıfının boyutu küçültüp eşit olmayan çoğunluk ve azınlık sınıflarını dengelemeye odaklanan bir tekniktir. Potansiyel olarak verilerde yer alan önemli bilgilerin kaybı gibi dezavantajları olmasına rağmen birçok veri bilimcisi tarafından kullanılan tekniklerden birisidir. Rastgele alt örnekleme yöntemi, sınıf dengesizliği olan bir veri kümesinde çoğunluk sınıfında yer alan verilerden rastgele veriler kaldırarak sınıf dengesizliğini çözen örnekleme yaklaşımlarından birisidir. Bu yöntemde veri kümesinin sayısı azaltılacağından, sınıflandırma süresi de azalmış olur. Bu yöntemin en büyük dezavantajı ise sınıflandırmada önemli değere sahip olabilecek verilerin rastgele kaldırılma olasılığının olmasıdır.

Küme Merkezli Örnekleme, kümeleme yöntemleri ile merkezlere dayalı yeni bir küme oluşturarak alt örnekleme yapan bir tekniktir. Bu teknik en yakın komşu algoritmasının küme merkezine göre yeni bir küme oluşturmaktadır. Çoğunluk sınıfındaki örneklerden oluşan bir kümeyi en yakın komşu algoritmasının küme merkeziyle değiştirerek çoğunluk sınıfının altında örnekler oluşturan bir yöntemdir. Azınlık sınıfındaki değerler bu yöntemde korunurken çoğunluk sınıfındaki veriler en yakın komşu algoritması ile yeniden sentezlenir.

### 3.3.3 Alt ve Üst Örnekleme Yöntemleri

SMOTE+ENN yöntemi, 2004 yılında önerilen, SMOTE ve en yakın komşu kuralını birleştiren bir örnekleme yöntemidir. SMOTE aşırı bir örnekleme yöntemidir ve temelinde azınlık

sınıftan yeni örnekler oluşturarak sınıf dengesizliği problemini çözmeye çalışır. Burada sınıf dengesizliği problemi çözülsede yeni örneklerin yanında gürültü ve sınır örnekleri de oluşabilir. Daha iyi örnekler oluşturabilmek için ENN, sınıf etiketi en yakın üç komşusundan en az ikisinin sınıfından farklı olan herhangi bir örneği kaldıran bir veri temizleme yöntemi kullanır. Bazı çoğunluk sınıfındaki örnekler, azınlık sınıfının alanını işgal edebileceği için SMOTE+ENN sentetik örneklerin ortaya çıkardığı fazla uyurma olasılığını azaltır.

### 3.4 Sınıflandırma Algoritmaları

Yazılım hata tahmininde yapılan çalışmalarda farklı teknikler kullanılmıştır. Bu teknikler dikey hata sınıflandırması, makine öğrenmesi teknikleri, derin öğrenme teknikleri, yapay sinir ağları teknikleri olarak sıralanabilir. Çalışmalarda farklı birçok makine öğrenmesi algoritması kullanılmış ve başarımları sonuçları elde edilmiştir.

Yazılım hata tahmininde makine öğrenmesi algoritmaları ile sınıflandırma birçok çalışmanın konusu olmuştur. Makine öğrenmesi yöntemlerinin kullanılmasının nedeni, kolay anlaşılır ve iyi başarımları yüksek hızda elde edebilecek birçok algoritmayı içermesindedir. Bu çalışmada da kullanılan veri kümesindeki sınıf dengesizliği problemi örnekleme yöntemleri çözülmüş, örnekleme işlemi sonrasında ise veri kümesine 13 adet aşağıda yer alan sınıflandırma algoritması uygulanmıştır.

K-En Yakın Komşu Algoritması, sık kullanılan makine öğrenmesi algoritmalarından en yakın komşu algoritması, birbirine yakın durumda olan değerlerin benzer olacağı mantığına dayanır [29]. Öğrenme tabanlı algoritmalarından biri olan bu algoritmada, yeni karşılaşılan X bir değer, eğitim kümesindeki örnekler ile arasındaki benzerliğe göre sınıflandırılır [30]. Sınıflandırma işlemi esnasında bir değer diğer bir değere olan uzaklığı hesaplanır. Bir değer diğer bir değere uzaklığının k adedi kadar komşusu dikkate alınır. Bu en yakın komşular içerisinde en çok benzerlik gösterdiği sınıfa ataması yapılır ve algoritma bu şekilde sınıflandırma işlemini yapar [31].

Destek vektör makineleri, iki farklı sınıfı bir doğru, bir düzlem üzerinde birbirinden ayırmaya odaklanan sınıflandırma problemlerinde oldukça yaygın kullanılan algoritmalarından birisidir. İki sınıfı birbirinden ayırırken sınırdaki yer alan değerler ile bu işlemi gerçekleştirir. Ayrılma marjının net olduğu zamanlarda çok iyi başarımları gösteren algoritmalarından birisidir. Bu yöntem, iki farklı gruba göre etiketlenmiş bir eğitim verisi için geliştirilen algoritması ile yeni bir değer verildiğinde bu değer hangi gruba ait olması gerektiğini belirleyen bir model oluşturmaktadır ve bu grup ayırmasını yaparken hiper düzlem üzerinde çalışmaktadır. [32].

Karar ağacı algoritması, sınıflandırma işlemini bir ağaç yapısı oluşturarak yapan bir algoritmadır. Oluşturulan ağaç yapısında yapraklar sınıf etiketlerini temsil ederken, kökten yapraklara giden yollar ise özniteliklere göre dallanmaları temsil etmektedir [33]. Karar ağacı algoritması böl ve yönet yaklaşımdan yararlanarak sınıflandırma gerçekleştiren bir yöntemdir. Karar ve bitiş düğümlerinden oluşan bir ağaçta, her bir karar düğümü dalları etiketleyen ayrık bir fonksiyonu

gerçekleştirmektedir. Verilen değerin bu fonksiyonların ile sınıflandırma işlemi yapılmaktadır [34].

Rastgele orman algoritması, Karar ağaçlarının birleşmesine dayanan, öğrenme sürecinde her bir ağacın sınıflandırma işlemini rastgele seçimler yaparak yapan bir algoritmadır [35]. Rastgele orman algoritması, daha doğru ve daha kararlı tahminler elde edebilmek için daha fazla karar ağacını birleştirir. Denetimli öğrenme yöntemlerinden biri olan bu algoritmada kullanılan ağaç sayısı ile doğruluk arasında doğrusal bir ilişki vardır yani ağaç sayısı arttıkça algoritmanın doğruluk oranı artmaktadır [36].

Topluluk öğrenme algoritmalarından birisi olan Adaboost, "Boosting" yani verilerden çıkan zayıf sonuçları birleştirerek güçlü bir işlem elde etmeyi sağlayan bir algoritmadır [37]. Sınıflandırma problemi için birden fazla öğrenci ile eğitim gerçekleştirilerek işlem yapılmaktadır. Başlangıçta tüm verilere eşit ağırlık verip daha sonra oluşturulan sınıflandırıcılar arasından en zayıf sınıflandırıcıyı seçer [38].

Gradyan Artırma algoritması temel, güçlü bir tahminci elde edebilmek için zayıf tahminleri kademeleri olarak güçlü tahmincilere dönüştürmeye odaklanır. Karar ağacı, kök noktasından başlayarak aşağıya doğru dallanmalar yapar ve yapraklar oluşturur. Algoritmada ilk olarak oluşturulan karar ağacı ile bir tahmin elde edilir. Elde edilen tahmin ile hedef arasındaki fark hesaplanır. Her yeni yinelemede, hesaplanan fark ile yeni bir ağaç oluşturulur.

Ekstra gradyan artırma algoritması (XGBoost), gradyan artırma ve karar ağacı algoritmasına dayanan makine öğrenmesi algoritmalarından birisidir. Yüksek tahmin gücü olan bir algoritma olarak bilinen bu algoritma diğer algoritmalarından genel başarımları iyileştiren, fazla öğrenmeyi azaltan bir düzenleme içermektedir [39]. Bu yöntem karar ağacı tabanlı bir yöntem olduğu için, karar ağacı oluştururken maksimum derinlik değerini kullanmaktadır. Oluşturulan ağaç sayısı aşağıda yönde aşırı bir ilerleme gösterir ise budama gerçekleştirilerek aşırı öğrenmenin önüne geçilmektedir.

Hafif gradyan artırma makineleri algoritması (LightGBM), XGBoost algoritmasının eğitim süresinin başarımlarını arttırmak için geliştirilen başarımları yüksek bir algoritma türüdür. Yüksek doğruluk, düşük kaynak kullanımı hızlı ve verimli oluşu nedeniyle sık tercih edilen algoritmalarından birisidir. Bu yöntemde yaprak odaklı ağaç oluşturma sistemi olduğu için hata oranı azalır ve daha hızlı öğrenme işlemi gerçekleşir [40]. Bu yöntemi diğer gradyan artırma algoritmalarından ayıran yanı sıra, karar ağaçlarının eğitilmesi aşamasında kullandığı büyüme stratejisidir [41].

Catboost algoritması, gradyan artırma karar ağacına dayalı sınıflandırma algoritmalarından birisidir. Yüksek hızda öğrenme yeteneği olan bu algoritma kategorik ve sayısal verilerde daha hızlı çalışabilmektedir [42]. Veri hazırlığı aşamasını kısaltması, boş veriler ile başa çıkabilmesi önemli özelliklerindedir. Simetrik ağaç yapısı kurarak çok derin ağaç yapısı olmadan öğrenme işlemini gerçekleştirir.

Naive Bayes sınıflandırma algoritması şartlı olasılık kurallarına göre düzenlenmiş, hesaplama ve kullanım kolaylığı ile makine öğrenmesi algoritmaları arasında en sık kullanılan bir



algoritmadır [43]. Bu algoritma Bayes teoremine göre sınıflandırma işlemini hesaplamaktadır. Bir sınıfa ait öznitelik değerinin etkisinin, diğer öznitelik değerlerinden bağımsız olduğu varsayılır ve bu durum sınıf koşul bağımsızlığı olarak adlandırılır [44]. Naive Bayes yöntemi ulaşılmaması istenen hedef etiket ile problemde uygulanan giriş parametreleri arasındaki ilişkiyi tahmin etmeye uyum sağlayan sınıflandırma yöntemidir [45].

Lineer diskriminant analizi, denetimli sınıflandırma yöntemlerinde sık kullanılan bir boyut indirme tekniği olan bir yöntem olup iki sınıfı birbirinden ayırırken farklılıkları modeller. Daha yüksek boyutlu uzaydaki özellikleri daha düşük boyutlu bir uzaya yansıtmak için kullanılır. Bu algoritma sınıflar arasındaki varyansı en aza indirerek, sınıflar arasındaki mesafeyi en üst düzeye çıkarıp iz düşüm hiper düzlemini bulmayı hedeflemektedir [46]. Popüler sınıflandırma algoritmalarından biri olan Kuadratik Discriminant Analizi, farklı sınıfların bilinen bir olasılık fonksiyonuna ait olduğunu varsayan bir sınıflandırıcıdır [47]. Bu algoritmada sınıfların ortalama ve kovaryans matrisleri bulunur ve yeni örnek en yüksek olasılığa sahip olan sınıfa etiketlenir.

Rastgele orman algoritması ile aynı yapıda olan Ekstra Ağaçlar algoritması, veri setlerindeki kopyaları kullanarak modelleri eğitmektedir. Rastgele orman algoritmasının farklı bir mimarisini oluşturmaktadır. Rastgele ağaç algoritmasından farkı oluşturulan düğümlerin dallara ayrılması aşamasında karar alma ölçütü yerine rastgele dallanma yolunun seçilmesidir.

### 3.5 Başarım Ölçünleri

Bu çalışmada makine öğrenmesi algoritmaları ile yapılan sınıflandırmaların başarımlarını ölçmek için doğruluk (doğruluk) ve AUC/AUROC ölçütleri kullanılmıştır. Sınıflandırma algoritmalarının başarımlarını testinde karmaşıklık matrisi kullanılır. Bir karmaşıklık matrisi doğru pozitif (DP), yanlış pozitif (YP), doğru negatif (DN) ve yanlış negatif (YN) adı verilen dört değerden oluşmaktadır. DP ve DN değerleri karmaşıklık matrisinde gerçekten tahmin edilmiş hatalı ve hatasız kayıtları temsil eder. YP ve YN değerleri ise yanlış tahmin edilen hatalı ve hatasız kayıtların temsili gerçekleştirir. Makine öğrenmesi başarımlarını metriklerinden biri olan karmaşıklık matrisi sınıf etiketlerinin görselleştirilmesi için bir araç görevindedir.

Bu çalışmada kullanılan veri kümesinde sınıf dengesizliği problemi bulunmaktadır. Bu tür veri setlerinde sınıflandırma yapılırken en iyi başarımları verecek ölçünlerin seçimi önem teşkil etmektedir. Bu kapsamda sınıf dengesizliği probleminde başarımlarını karşılaştırması için genellikle doğruluk değeri kullanılmaktadır. Bir diğer yandan doğruluk başarımlarını değerlendirilmesinde tek ölçüt olmamalıdır [48]. Sınıf dengesizliği probleminde AUC/AUROC metriği ise sınıfları ayırt edebilmede başarımlarını göstermekte ve doğruluk metriği yanında kullanan metriklerden birisidir.

### 4. Deneysel Sonuçları

Bu çalışmada, yazılım hata tahmininde sınıf dengesizliği problemini çözebilmek ve yazılım hata tahminini sağlamak için mevcut algoritmalar ile bir model önerilmiştir. Üzerinde

çalışan bu model Kaggle platformunda yer alan veri setlerinden yararlanılarak, Python programlama dili ile analiz edilmiştir. Tüm analizler Intel Core i7 işlemcili, 16 GB RAM ve 500 GB disk kapasitesine sahip bilgisayarda gerçekleştirilmiştir. İlk aşamada, çalışmada kullanılan NASA veri deposundan alınan CM1, KC1, KC2, JM1, PC1 veri kümeleri için bir ön işlem uygulanmıştır. Veri ön işlem aşamasında, standart normalizasyon yöntemi ile veriler normalize edilmiştir. Veri kümeleri ön işlemlemeden geçirildikten sonra içerisinde yer alan hatalı ve hatasız kayıtlar arasında bir sınıf dengesizliği problemi olduğu görülmüştür. Literatürdeki çalışmalar incelenerek yazılım hata tahmininde bu veri kümesindeki sınıf dengesizliği probleminin çözümünde sadece eğitim kümesine örnekleme uygulanarak yöntemin daha doğru sonuçlara ulaştığı sonucuna varılmıştır. Bu doğrultuda veri kümesi eğitim ve test veri kümesi olarak ayrılıp sadece eğitim veri kümesine sınıf dengesizliği probleminde kullanılan örnekleme yöntemleri uygulanmıştır. Daha sonra eğitim veri kümesine çeşitli makine öğrenmesi algoritmaları ile sınıflandırma işlemi uygulanmıştır. En son aşama olarak test aşamasında sınıflandırma veri kümesine uygulanan algoritmalar ile başarımlarını karşılaştırması yapılmıştır. Kullanılan sınıflandırma algoritmaları ve örnekleme yöntemleri ile elde edilen başarımlarını analizi değerleri aşağıdaki şekiller içerisinde detaylı olarak gösterilmiştir.

PC1 veri kümesinde SMOTE örnekleme yöntemi ile en iyi AUROC değeri Ekstra Gradyan Arttırma sınıflandırma algoritmasında %89,94, en iyi doğruluk değeri ise Extra Tree algoritmasında 94.59% elde edilmiştir. PC1 veri kümesinde Rastgele Örnekleme yöntemi ile en iyi AUROC değeri Catboost algoritmasında %89,68, en iyi doğruluk değeri ise yine Catboost algoritmasında %94,14 olarak elde edilmiştir. PC1 veri kümesinde SMOTENN yöntemi ile en iyi AUROC değeri Catboost algoritmasında %91,83, en iyi doğruluk değeri Ekstra Ağaçlar algoritmasında %94,59 olarak elde edilmiştir. PC1 veri kümesinde Rastgele Örnekleme yöntemi ile en iyi AUROC değeri Ekstra Ağaçlar algoritmasında %90,2, en iyi doğruluk değeri Gradyan Arttırma algoritmasında %73,87 olarak elde edilmiştir. PC1 veri kümesi Küme Merkezli Örnekleme yöntemi ile en iyi AUROC değeri Ekstra Ağaçlar algoritmasında %87,73, en iyi doğruluk değeri KNN algoritması ile %80,63 olarak elde edilmiştir. PC1 veri kümesinde ADASYN yöntemi ile en iyi AUROC değeri Catboost algoritmasında %89,42, en iyi doğruluk değeri Ekstra Ağaçlar algoritmasında %94,14 olarak elde edilmiştir. PC1 veri kümesinde Borderline SMOTE yöntemi ile en iyi AUROC değeri Catboost algoritmasında %89,94, en iyi doğruluk değeri Ekstra Ağaçlar algoritmasında %95,04 olarak elde edilmiştir. PC1 veri kümesinde SVM-SMOTE yöntemi ile en iyi AUROC değeri Catboost algoritmasında %91,99, en iyi doğruluk değeri Ekstra Ağaçlar algoritmasında %94,59 olarak elde edilmiştir. PC1 veri kümesinde SMOTE-Tomek yöntemi ile en iyi AUROC değeri XGB algoritmasında %90,06, en iyi doğruluk değeri Ekstra Ağaçlar algoritmasında %94,59 olarak elde edilmiştir. PC1 veri kümesinde SMOTENC yöntemi ile en iyi AUROC değeri XGB algoritmasında %88,55, en iyi doğruluk değeri Ekstra Ağaçlar algoritmasında %93,69 olarak elde edilmiştir. Elde edilen sonuçlar Şekil-3'te özetlenmiştir.

JM1 veri kümesinde SMOTE örnekleme yöntemi ile en iyi AUROC değeri Extra Tree sınıflandırma algoritmasında %74,98, en iyi doğruluk değeri ise Extra Tree algoritmasında %80,30 elde edilmiştir. JM1 veri kümesinde Random Oversampling yöntemi ile en iyi AUROC değeri LGBM algoritmasında %73,84, en iyi doğruluk değeri Extra Tree algoritmasında %80,84 olarak elde edilmiştir. JM1 veri kümesinde SMOTENN yöntemi ile en iyi AUROC değeri Catboost algoritmasında %74,68, en iyi doğruluk değeri Extra Tree algoritmasında %81,35 olarak elde edilmiştir. JM1 veri kümesinde Random Undersampling yöntemi ile en iyi AUROC değeri Extra Tree algoritmasında %72,69, en iyi doğruluk değeri Gaussian Naive Bayes algoritmasında %68,53 olarak elde edilmiştir. JM1 veri kümesi Cluster Centroid yöntemi ile en iyi AUROC değeri Quadratic Discriminant Analysis algoritmasında %60,37, en iyi doğruluk değeri Gaussian Naive Bayes algoritması ile %61,64 olarak elde edilmiştir. JM1 veri kümesinde ADASYN yöntemi ile en iyi AUROC değeri Extra Tree algoritmasında %74,69, en iyi doğruluk değeri Catboost algoritmasında %79,74 olarak elde edilmiştir. JM1 veri kümesinde Borderline SMOTE yöntemi ile en iyi AUROC değeri Extra Tree algoritmasında %74,66, en iyi doğruluk değeri XGB algoritmasında %79,65 olarak elde edilmiştir. JM1 veri kümesinde SVM-SMOTE yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %75,21, en iyi doğruluk değeri Catboost algoritmasında %80,25 olarak elde edilmiştir. JM1 veri kümesinde SMOTE-Tomek yöntemi ile en iyi AUROC değeri Extra Tree algoritmasında %74,93, en iyi doğruluk değeri LGBM algoritmasında %80,43 olarak elde edilmiştir. JM1 veri kümesinde SMOTENC yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %75,24, en iyi doğruluk değeri Catboost algoritmasında %80,75 olarak elde edilmiştir. Elde edilen sonuçlar Şekil-4'te özetlenmiştir.

KC1 veri kümesinde SMOTE örnekleme yöntemi ile en iyi AUROC değeri Random Forest sınıflandırma algoritmasında %83,48, en iyi doğruluk değeri ise Catboost algoritmasında %85,31 elde edilmiştir. KC1 veri kümesinde Random Oversampling yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %82,67, en iyi doğruluk değeri yine Random Forest algoritmasında %85,07 olarak elde edilmiştir. KC1 veri kümesinde SMOTENN yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %84,12, en iyi doğruluk değeri Adaboost algoritmasında %86,49 olarak elde edilmiştir. KC1 veri kümesinde Random Undersampling yöntemi ile en iyi AUROC değeri Catboost algoritmasında %82,34, en iyi doğruluk değeri KNN algoritmasında %72,98 olarak elde edilmiştir. KC1 veri kümesi Cluster Centroid yöntemi ile en iyi AUROC değeri Quadratic Discriminant Analysis algoritmasında %74,39, en iyi doğruluk değeri Linear Discriminant Analysis algoritması ile %73,69 olarak elde edilmiştir. KC1 veri kümesinde ADASYN yöntemi ile en iyi AUROC değeri Catboost algoritmasında %81,78, en iyi doğruluk değeri Catboost algoritmasında %79,74 olarak elde edilmiştir. KC1 veri kümesinde Borderline SMOTE yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %82,86, en iyi doğruluk değeri Catboost algoritmasında %86,73 olarak elde edilmiştir. KC1 veri kümesinde SVM-SMOTE yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %83,39, en iyi doğruluk değeri LGBM algoritmasında %85,78 olarak

elde edilmiştir. KC1 veri kümesinde SMOTE-Tomek yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %83,42, en iyi doğruluk değeri LGBM algoritmasında %85,07 olarak elde edilmiştir. KC1 veri kümesinde SMOTENC yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %83,52, en iyi doğruluk değeri LGBM algoritmasında %85,54 olarak elde edilmiştir. Elde edilen sonuçlar Şekil-5'te özetlenmiştir.

KC2 veri kümesinde SMOTE örnekleme yöntemi ile en iyi AUROC değeri Quadratic Discriminant Analysis sınıflandırma algoritmasında %81,54, en iyi doğruluk değeri ise Adaboost algoritmasında %80,95 elde edilmiştir. KC2 veri kümesinde Random Oversampling yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %80,55, en iyi doğruluk değeri yine Random Forest algoritmasında %80,95 olarak elde edilmiştir. KC2 veri kümesinde SMOTENN yöntemi ile en iyi AUROC değeri Gradient Boosting algoritmasında %79,68, en iyi doğruluk değeri Catboost algoritmasında %80,95 olarak elde edilmiştir. KC2 veri kümesinde Random Undersampling yöntemi ile en iyi AUROC değeri XGB algoritmasında %82,14, en iyi doğruluk değeri Random Forest algoritmasında 77.14% olarak elde edilmiştir. KC2 veri kümesi Cluster Centroid yöntemi ile en iyi AUROC değeri Quadratic Discriminant Analysis algoritmasında %80,61, en iyi doğruluk değeri yine Quadratic Discriminant Analysis algoritması ile %80,95 olarak elde edilmiştir. KC2 veri kümesinde ADASYN yöntemi ile en iyi AUROC değeri Quadratic Discriminant Analysis algoritmasında %81,43, en iyi doğruluk değeri Gradient Boosting algoritmasında %82,85 olarak elde edilmiştir. KC2 veri kümesinde Borderline SMOTE yöntemi ile en iyi AUROC değeri Catboost algoritmasında %72,89, en iyi doğruluk değeri LGBM algoritmasında %89,00 olarak elde edilmiştir. KC2 veri kümesinde SVM-SMOTE yöntemi ile en iyi AUROC değeri Gradient Boosting algoritmasında %80,83, en iyi doğruluk değeri XGB algoritmasında %81,90 olarak elde edilmiştir. KC2 veri kümesinde SMOTE-Tomek yöntemi ile en iyi AUROC değeri XGB algoritmasında %83,04, en iyi doğruluk değeri yine XGB algoritmasında %82,86 olarak elde edilmiştir. KC2 veri kümesinde SMOTENC yöntemi ile en iyi AUROC değeri Quadratic Discriminant Analysis algoritmasında %81,70, en iyi doğruluk değeri XGB algoritmasında %81,90 olarak elde edilmiştir. Elde edilen sonuçlar Şekil-6'da özetlenmiştir.

CM1 veri kümesinde SMOTE örnekleme yöntemi ile en iyi AUROC değeri Extra Tree sınıflandırma algoritmasında %74,56, en iyi doğruluk değeri ise Random Forest algoritmasında %89,00 elde edilmiştir. CM1 veri kümesinde Random Oversampling yöntemi ile en iyi AUROC değeri Extra Tree algoritmasında %72,67, en iyi doğruluk değeri yine Extra Tree algoritmasında algoritmasında %90,00 olarak elde edilmiştir. CM1 veri kümesinde SMOTENN yöntemi ile en iyi AUROC değeri Gradient Boosting algoritmasında %72,67, en iyi doğruluk değeri LGBM algoritmasında %91,00 olarak elde edilmiştir. CM1 veri kümesinde Random Undersampling yöntemi ile en iyi AUROC değeri Catboost algoritmasında %74,44, en iyi doğruluk değeri Gradient Boosting algoritmasında %71,00 olarak elde edilmiştir. CM1 veri kümesi Cluster Centroid yöntemi ile en iyi AUROC değeri

Random Forest algoritmasında %72,83, en iyi doğruluk değeri KNN algoritması ile %73,00 olarak elde edilmiştir. CM1 veri kümesinde ADASYN yöntemi ile en iyi AUROC değeri Rastgele Orman algoritmasında %77,11, en iyi doğruluk değeri LGBM algoritmasında %88,00 olarak elde edilmiştir. CM1 veri kümesinde Borderline SMOTE yöntemi ile en iyi AUROC değeri Extra Tree algoritmasında %74,39, en iyi doğruluk değeri Random Forest algoritmasında 88.00% olarak elde edilmiştir.

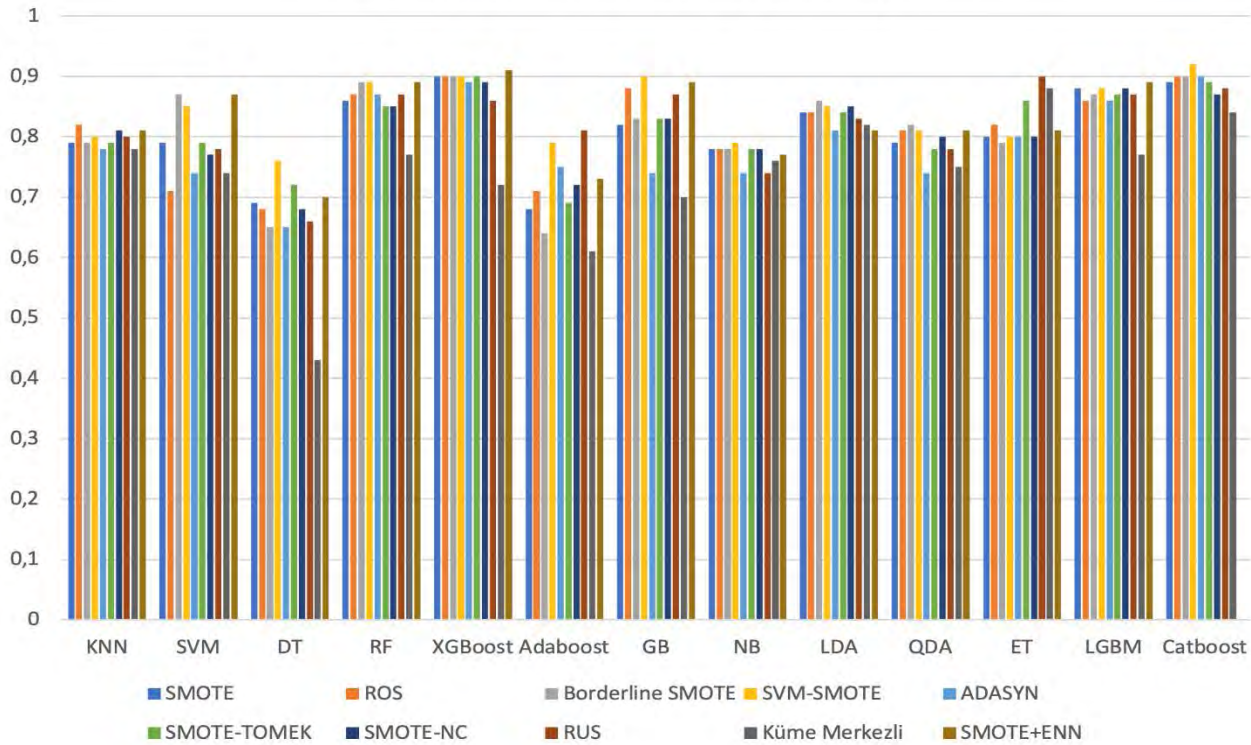
CM1 veri kümesinde SVM-SMOTE yöntemi ile en iyi AUROC değeri LGBM algoritmasında %75,44, en iyi doğruluk değeri Catboost algoritmasında %88,00 olarak elde edilmiştir. CM1 veri kümesinde SMOTE-Tomek yöntemi ile en iyi AUROC değeri Random Forest algoritmasında %83,41, en iyi doğruluk değeri XGB algoritmasında %85,07 olarak elde edilmiştir. CM1 veri kümesinde SMOTENC yöntemi ile en iyi AUROC değeri XGB algoritmasında %74,11, en iyi doğruluk değeri LGBM algoritmasında %86,00 olarak elde edilmiştir. Elde edilen sonuçlar Şekil-7'de özetlenmiştir.

Çalışma kapsamında incelenen literatür çalışmalarında eğitim ve test veri setlerini ayrı değerlendiren çalışmalarda elde edilen sonuçlar şu şekilde açıklanmıştır. Sun ve ark. tarafından yapılan çalışmada, Random Forest ve Naive Bayes algoritmaları ile SMOTE örnekleme yöntemi kullanılmıştır. Eğitim kümesinde SMOTE yöntemi ile örnekleme işlemi yapılmıştır. Elde edilen sonuçlar, bizim çalışmamız ile kıyaslandığında en iyi 0,85 AUROC değeri PC1 veri kümesinde SMOTE yöntemi ve Random Forest algoritmasının kombinasyonu ile elde edilmiştir. Bizim çalışmamızda ise; PC1

veri kümesinde SMOTE yöntemi ile XGB algoritması kullanılarak yaklaşık 0,90 değerinde AUROC değeri elde edilmiştir. Bu başarımla değeri ile literatür çalışmasında yer alan başarımla değerinden daha iyi bir sonuç elde edilmiştir. Eivazpour ve ark. tarafından yapılan çalışmada, Decision Tree ve Random Forest algoritmaları kullanılmış, sadece eğitim veri kümesinde SMOTE örnekleme yöntemi kullanılmıştır. Bu çalışmada KC2 veri kümesinde SMOTE yöntemi ve Karar Ağacı algoritması ile 0,74 AUROC değeri en iyi başarımla değeri olmuştur. KC2 veri kümesinde, bizim çalışmamızda Quadratic Discriminant Analysis algoritması ve SMOTE yönteminin birleşmesi ile 0,82 AUROC ile daha iyi bir başarımla değeri elde edilmiştir. Malhotra ve ark. tarafından yapılan çalışmada ADASYN örnekleme yöntemi ile SVM, Naive Bayes algoritmaları birleştirilip sınıflandırma işlemi yapılmıştır. Bu çalışmada KC1 veri kümesinde SVM algoritması ve ADASYN örnekleme yöntemi ile 0.70 AUROC değeri en iyi başarımla değeri olmuştur. Bizim çalışmamızda ise KC1 veri kümesinde ADASYN yöntemi ve Catboost algoritması birleştirilerek en iyi 0,82 AUROC başarımla değeri elde edilmiştir. Bu çalışmada SMOTE örnekleme yöntemine de yer verilmiştir. SMOTE örnekleme yönteminde KC1 veri kümesinde SVM algoritması ile 0,83 AUROC değeri elde edilen en iyi başarımla değeri olmuştur. Bizim çalışmamızda ise, KC1 veri kümesinde Random Forest algoritması ve SMOTE örnekleme yöntemi ile aynı oranda yani 0,83 AUROC değeri elde edilmiştir.

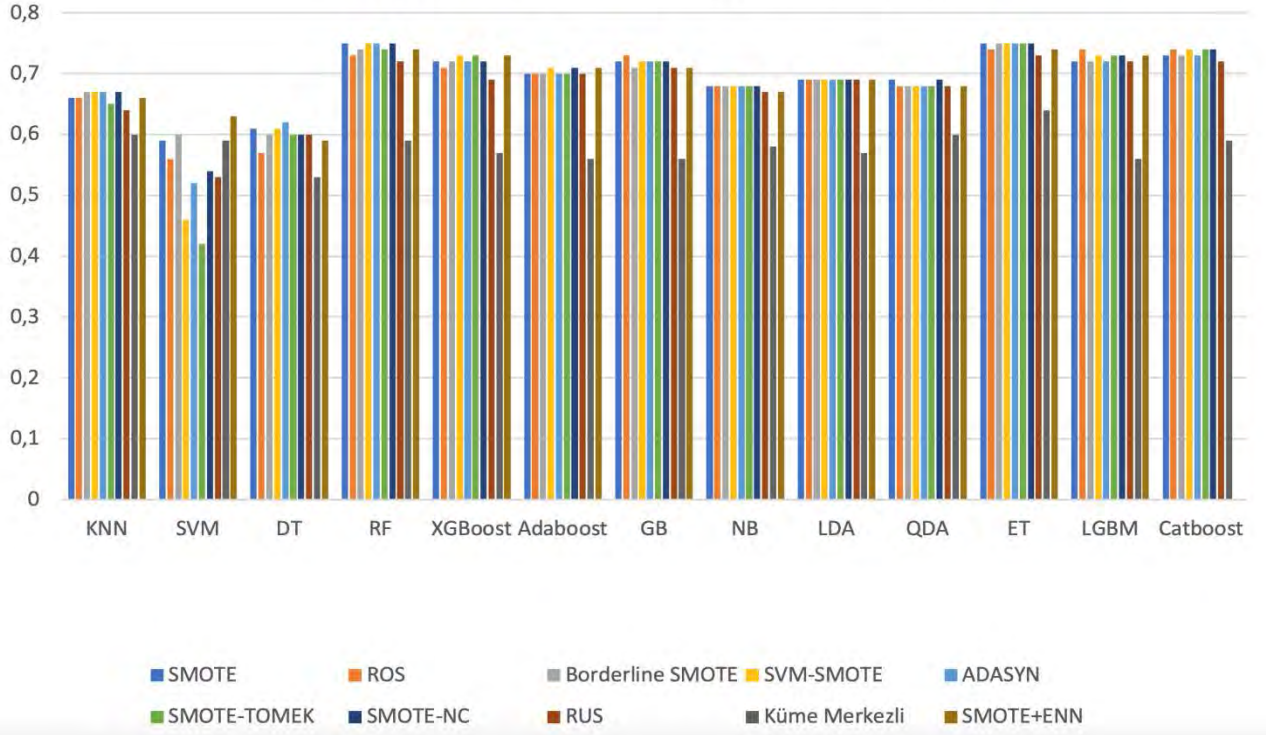
Tüm deneylerde veri kümesine bağlı olarak hangi örnekleme yönteminde hangi algoritma ile en iyi sonuçların elde edildiği Çizelge-5 içerisinde ayrıntılanmıştır.

### PC1 Veri Setinde AUROC Ölçütü Sonuçları



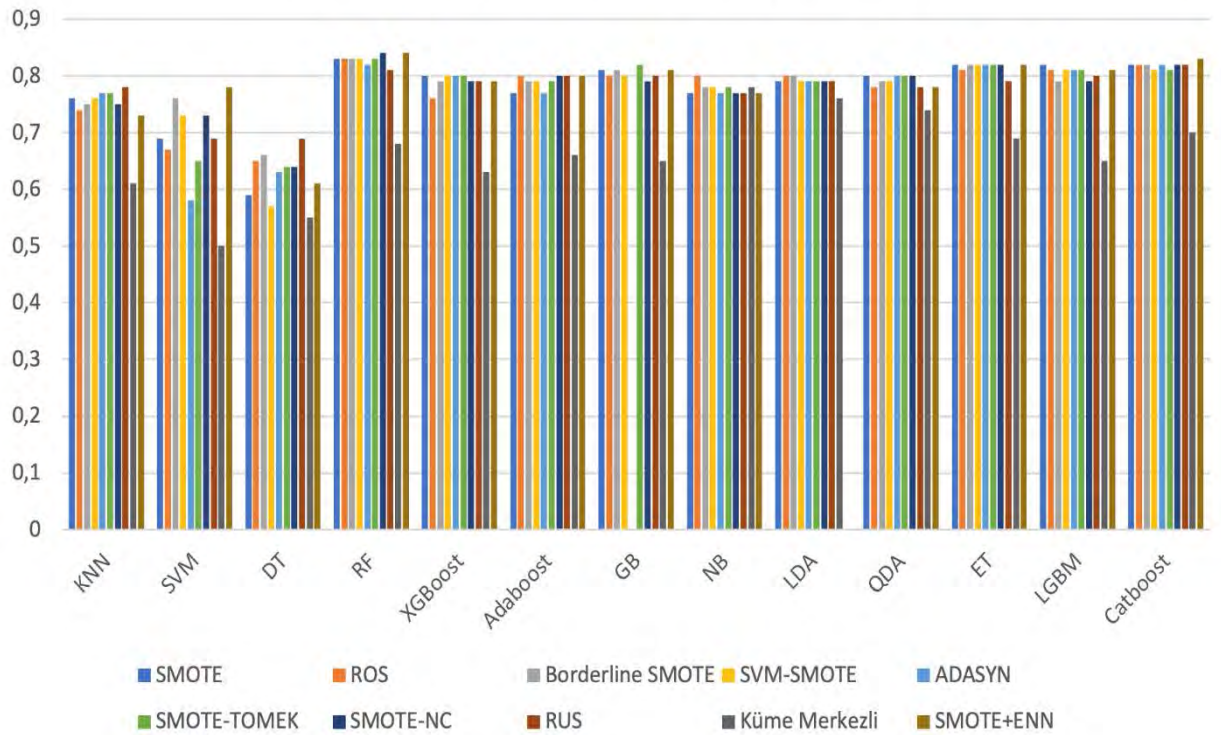
Şekil 3: PC1 veri kümesinde elde edilen AUROC başarımla sonuçları grafiği

### JM1 Veri Setinde AUROC Ölçütü Sonuçları



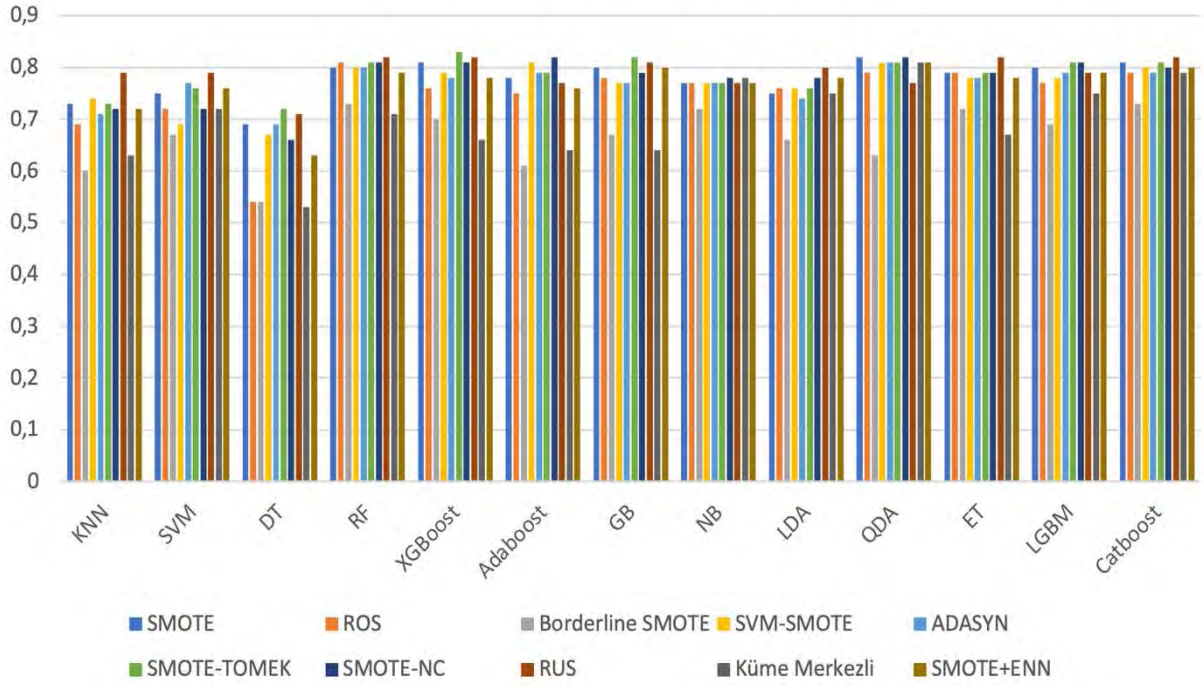
Şekil 4: JM1 veri kümesinde elde edilen AUROC başarımları grafiği

### KC1 Veri Setinde AUROC Ölçütü Sonuçları



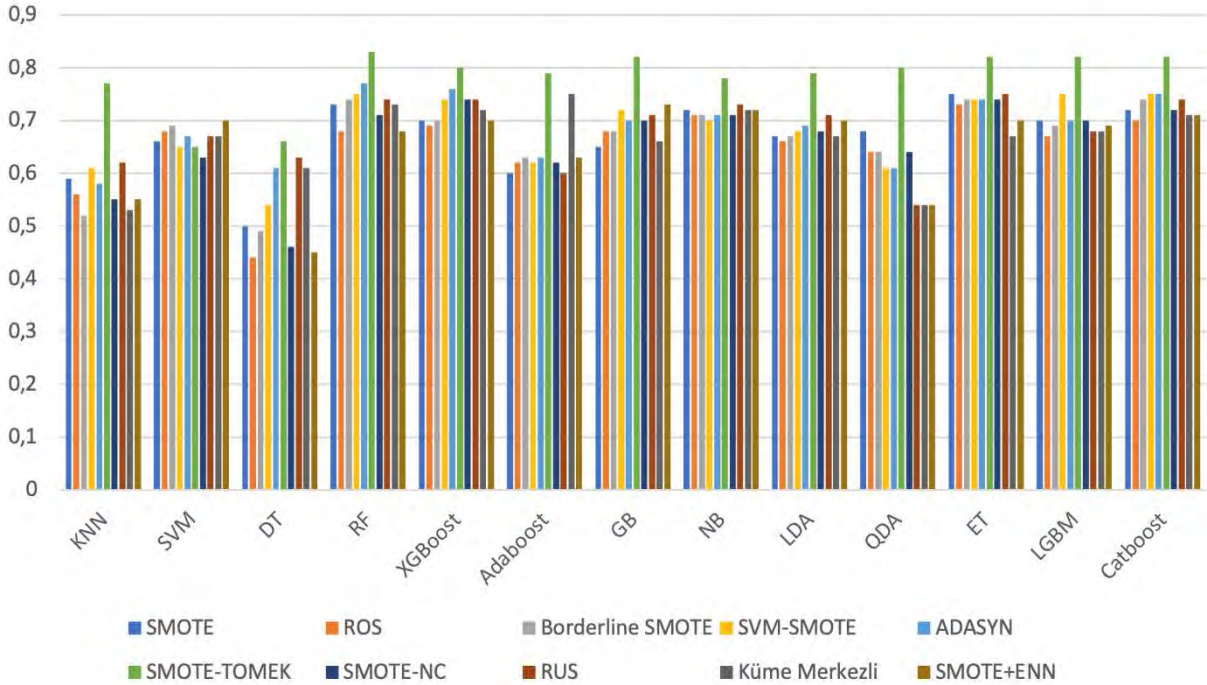
Şekil 5: KC1 veri kümesinde elde edilen AUROC başarımları grafiği

## KC2 Veri Setinde AUROC Ölçütü Sonuçları



Şekil 6: KC2 veri kümesinde elde edilen AUROC başarımları grafiği

## CM1 Veri Setinde AUROC Ölçütü Sonuçları



Şekil 7: CM1 veri kümesinde elde edilen AUROC başarımları grafiği

Çizelge-5: Veri Kümelerinde Örnekleme Yöntemleri ve Sınıflandırma Algoritmalarına göre en iyi Başarım Sonuçları

Veri Kümesi	Doğruluk	En İyi Doğruluk Değerinde Örnekleme Yöntemi ve Sınıflandırma Algoritması	AUROC	En İyi AUROC Değerinde Örnekleme Yöntemi ve Sınıflandırma Algoritması
PC1	0,95	Borderline-SMOTE ve Ekstra Ağaçlar Algoritması	0,92	SVM-SMOTE ve Catboost Algoritması
JM1	0,81	SMOTE-ENN ve Catboost Algoritması	0,75	SVM-SMOTE ve Random Forest Algoritması
KC1	0,87	Borderline-SMOTE ve Catboost Algoritması	0,84	SMOTE-ENN ve Random Forest Algoritması
KC2	0,89	Borderline-SMOTE ve Light GBM Algoritması	0,83	SMOTE-Tomek ve XGBoost Algoritması
CM1	0,91	SMOTE-ENN ve Light GBM Algoritması	0,83	SMOTE-Tomek ve Random Forest Algoritması

## 5. Sonuçlar ve Gelecek Çalışmalar

Birçok alanda aktif olarak kullanılan yazılımların kusursuz ve istenilen şekilde olması günümüzde oldukça önem teşkil eden bir konudur. Yazılımın kalitesinde başta gelen en mühim konu yazılımın içerisinde hata bulundurmamasıdır. Yazılım hata tahmininin makine öğrenmesi algoritmaları ile tespit edildiği birçok örnek yöntem bu çalışmada incelenmiştir. Yazılım hata tahmininde kullanılan veri kümelerinde sınıf dengesizliği problemi yer almaktadır. Bu problemin çözümü için örnekleme yöntemleri denenmiştir. Sınıf dengesizliği problemi çözülen veri kümelerinde birçok sınıflandırma algoritması ile sınıflandırma işlemi yapılmış; daha sonra örnekleme yöntemlerine bağlı olarak sınıflandırma başarımları karşılaştırılmıştır. Deney sonuçlarında PC1 veri kümesinde 0,92 AUC en iyi elde edilen başarımların değeri olmuştur.

Bu çalışma büyük yazılım projelerinin en çok hata alan kısımlarında referans olarak hataların tespit edilmesinde kullanılabilir. Güncel yazılım projelerinde en çok hata alınan modüllerin en çok hata alan bölümlerinde etiketler oluşturularak bu çalışmada olduğu gibi hatalar tespit edilebilir. Her bir yazılım projelerinde belirlenen metrikler ve en çok hata alınan modüller referans alınarak bu çalışma ile aynı mantıkta yazılım hata tahmini sistemleri oluşturulabilir.

Bu çalışma ile gelecekte yapılacak yazılım hata tahmininin makine öğrenmesi ve örnekleme yöntemleri yaklaşımları ile çözümleri için motivasyon oluşturabilir.

Farklı veri setlerinde yapılacak çalışmalarda, bu çalışmada kullanılan örnekleme yöntemleri ve algoritmalar bu alandaki araştırmacılar için farklı perspektifler sunacaktır.

## Kaynakça

- [1] Yalçın, N., & Şimşek Yağlı, B. Teknoloji mağazalarının ISO 25010 kalite modeline dayalı web sitesi kalite değerlendirmesinin çok kriterli analizi: Türkiye örneği, 2020.
- [2] Bulut, S. Makine öğrenmesi, Algoritmik Habercilik ve Gazetecilikte İşlevsiz İnsan Sorunsalı, Selçuk İletişim, 2020, 13(1), 294-313.
- [3] Aydılek, İ. Yazılım hata tahmininde kullanılan metriklerin karar ağaçlarındaki bilgi kazançlarının incelenmesi ve iyileştirilmesi, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 2018, 24(5), 906-914.
- [4] Çetiner, M. Makine öğrenmesi yöntemleri ile yazılım hata

tahmini, Master's thesis, İstanbul Kültür Üniversitesi/Lisansüstü Eğitim Enstitüsü/Bilgisayar Mühendisliği Ana Bilim Dalı/Bilgisayar Mühendisliği Bilim Dalı, 2020.

- [5] Sun, Z., Song, Q., & Zhu, X. Using coding-based ensemble learning to improve software defect prediction, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2012, 42(6), 1806-1817.
- [6] Eivazpour, Z., & Keyvanpour, M. R. Improving performance in software defect prediction using variational autoencoder, 5th Conference on Knowledge Based Engineering and Innovation (KBEI), 2019, pp. 644-649.
- [7] Malhotra, R., Agrawal, V., Pal, V., & Agarwal, T. Support Vector based Oversampling Technique for Handling Class Imbalance in Software Defect Prediction, 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 1078-1083.
- [8] Choirunnisa, S., Meidyani, B., & Rochimah, S. Software Defect Prediction using Oversampling Algorithm: A-SUWO, Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), 2018, pp. 337-341.
- [9] Elahi, E., Ayub, A., & Hussain, I. Two staged data preprocessing ensemble model for software fault prediction, International Bhurban Conference on Applied Sciences and Technologies (IBCAST), 2021, pp. 506-511.
- [10] Goyal, S. Handling class-imbalance with KNN (neighbourhood) under-sampling for software defect prediction, Artificial Intelligence Review, 2022, 55(3), 2023-2064.
- [11] Jacob, R. J., Kamat, R. J., Sahithya, N. M., John, S. S., & Shankar, S. P. Voting based ensemble classification for software defect prediction, IEEE Mysore Sub Section International Conference (MysuruCon), 2021, pp. 358-365.
- [12] Cetiner, M., & Sahingoz, O. K. A comparative analysis for machine learning based software defect prediction systems, 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-7.
- [13] Pelayo, L., & Dick, S. Applying novel resampling strategies to software defect prediction, NAFIPS 2007-2007 Annual meeting of the North American fuzzy information processing society, 2007, pp. 69-72.
- [14] Aleem, S., Capretz, L. F., & Ahmed, F. Benchmarking machine learning Technologies for software defect detection, arXiv preprint arXiv:1506.07563, 2015.
- [15] Ibrahim, D. R., Ghnemat, R., & Hudaib, A. Software defect prediction using feature selection and random forest algorithm, International Conference on New Trends in Computing Sciences (ICTCS), 2017, pp. 252-257.

- [16] Akmel, F., Birihanu, E., & Siraj, B. A literatüre review study of software defect prediction using machine learning techniques, *Int. J. Emerg. Res. Manag. Technol*, 2017, 6(6), 300-306.
- [17] Naidu, M. S., & Geethanjali, N. Classification of defects in software using decision tree algorithm, *International Journal of Engineering Science and Technology*, 2013, 5(6), 1332.
- [18] «<http://promise.site.uottawa.ca/>,» google, [Çevrimiçi]. <http://promise.site.uottawa.ca/SERepository/datasets-page.html> [Erişildi: 7 Ocak 2023].
- [19] Aydın Hakli, D. Sınıf Dengesizliği Sorununu Çözmek için Kullanılan Algoritmaların Farklı Sınıflandırma Yöntemlerinde Performanslarının Karşılaştırılması, 2018.
- [20] Yavaş, M., Güran, A. & Uysal, M. Covid-19 Veri Kümesinin SMOTE Tabanlı Örneklem Yöntemi Uygulanarak Sınıflandırılması, *Avrupa Bilim ve Teknoloji Dergisi Ejosat Özel Sayı 2020 (HORA)*, 2020, 258-264.
- [21] Çelik, Ö. & Kaplan, G. Yeniden Örneklem Teknikleri Kullanarak SMS Verisi Üzerinde Metin Sınıflandırma Çalışması, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi*, 2020, 36 (3), 433-442.
- [22] Akın, P. & Terzi, Y. Dengesiz Veri Setli Sağkalım Verilerinde Cox Regresyon ve Rastgele Orman Yöntemlerin Karşılaştırılması, *Veri Bilimi*, 2020, 3 (1), 21-25.
- [23] Öztürk, H. Dengesiz veri setlerinde farklı dengeleme algoritmalarının optimum denge oranlarının sınıflandırma ve regresyon ağaçları yöntemi ile incelenmesi: simülasyon çalışması, 2022.
- [24] Dal, A., Gümüş, İ. H., Güldal, S. & Yavaş, M. Dengesiz Veriler İçin Ağırlıklı Geometrik Ortalama Tabanlı Yeni Bir Yeniden Örneklem Yaklaşımı, *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, 2021, 8 (15), 343-352.
- [25] Topal, A., & Amasyalı, M. F. Yapay Örnek Üretimi Ne Zaman İşe Yarar? When does Synthetic Data Generation Work?
- [26] UYANIK, F., & KASAPBAŞI, M. C. Telekomünikasyon sektörü için veri madenciliği ve makine öğrenmesi teknikleri ile ayrılan müşteri analizi, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 2021, 9(3), 172-191.
- [27] Mukherjee, M., & Khushi, M. SMOTE-ENC: A novel SMOTE-based method to generate synthetic data for nominal and continuous features, *Applied System Innovation*, 2021, 4(1), 18.
- [28] Kaba, G. & Bağdatlı Kalkan, S. Kardiyovasküler Hastalık Tahmininde Makine Öğrenmesi Sınıflandırma Algoritmalarının Karşılaştırılması, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 2022, 21 (42), 183-193.
- [29] Taşçı, E., & Onan, A. K-en yakın komşu algoritması parametrelerinin sınıflandırma performansı üzerine etkisinin incelenmesi, *Akademik Bilişim*, 2021, 1(1), 4-18.
- [30] Şengül, Z. Makine öğrenmesi algoritmalarını kullanarak bitcoin fiyat tahmini, *Master's thesis, Trakya Üniversitesi Sosyal Bilimler Enstitüsü*, 2022.
- [31] Gumustekin Aydın, S. & Aydoğdu, G. Makine Öğrenmesi Algoritmaları Kullanılarak Türkiye ve AB Ülkelerinin CO2 Emisyonlarının Tahmini, *Avrupa Bilim ve Teknoloji Dergisi*, 2022, Ejosat Special Issue (ISAS 2022), 42-46.
- [32] Şenel Ahmet, F. Makine Öğrenmesi Algoritmaları Kullanılarak Kayısı İç Çekirdeklerinin Sınıflandırılması, 2020.
- [33] Kalaycı, T. E. Kimlik hırsız web sitelerinin sınıflandırılması için makine öğrenmesi yöntemlerinin karşılaştırılması, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 2020, 24 (5), 870- 878.
- [34] Yavuz, Ö. Ç., Karaman, E. & Yeşilyaprak, C. Makine öğrenmesi algoritmalarıyla astronomik gözlem kalitesi tahminine yönelik karar destek sistemi geliştirilmesi ve uygulanması, *Trends in Business and Economics*, 2022, 36 (3), 289-303.
- [35] Yıldırım, E. & Çalhan, A. Apache Spark ile Makine Öğrenmesi Destekli Diyabet Rahatsızlığı Tahmini, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 2022, 10 (3), 1107-1117.
- [36] Sahingoz, O. K., Çebi, C. B., Bulut, F. S., Fırat, H. & Karataş, G. Saldırı Tespit Sistemlerinde Makine Öğrenmesi Modellerinin Karşılaştırılması, *Erzincan University Journal of Science and Technology*, 2019, 12 (3), 1513-1525.
- [37] Güleş, Ş. Makine öğrenmesi yöntemleri ile zararlı yazılım tespiti, *Master's thesis, Konya Teknik Üniversitesi*, 2020.
- [38] YILDIRIM, E. Hızlandırılmış Makine Öğrenmesi Algoritmaları İle Türkçe Sahte Haber Tespiti, 2022.
- [39] Serbest, K., & Kılıç, S. A. Diz Eklemleri Momentinin Tahmini İçin Makine Öğrenmesi Yöntemlerinin İncelenmesi, *Academic Perspective Procedia*, 4(1), 341-349, 2021.
- [40] Üstüner, M., Abdikan, S., Bilgin, G. & Balık Şanlı, F. Hafif Gradyan Artırma Makineleri ile Tarımsal Ürünlerin Sınıflandırılması, *Turkish Journal of Remote Sensing and GIS, Academic Perspective Procedia*, 1 (2), 97-105, 2020.
- [41] Şahinbaş, K. Price Prediction Model for Restaurants In Istanbul By Using Machine Learning Algorithms, *Ekonomi İşletme ve Maliye Araştırmaları Dergisi*, 4 (2), 159-171, 2022.
- [42] Onan, A. Twitter Mesajları Üzerinde Makine Öğrenmesi Yöntemlerine Dayalı Duygu Analizi, *Yönetim Bilişim Sistemleri Dergisi*, 3 (2), 1-14, 2017.
- [43] Selimoğlu, M. & Yılmaz, A. Kredi Kartı Dolandırıcılık Tespitinin Makine Öğrenmesi Yöntemleri ile Tahmin Edilmesi, *Beykent Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 13 (2), 28-33, 2021.
- [44] Cihan, M., & Ceylan, M. Comparison of linear discriminant analysis, support vector machines and naive bayes methods in the classification of neonatal hyperspectral signatures, *29th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4), 2021.
- [45] Çetin, E. Yedinci Servikal Vertebra'nın Antropometrik Ölçümleri İle Makine Öğrenme Algoritmaları Kullanılarak Cinsiyet Tayini Üzerine Bir Çalışma (Doctoral dissertation), 2021.
- [46] Seçgin, Y. Pelvis bilgisayarlı tomografi görüntülerinden elde edilen parametreler ile makine öğrenme algoritmaları kullanılarak cinsiyet tahmini üzerine bir deneme (Master's thesis, Lisansüstü Eğitim Enstitüsü), 2020.
- [47] Kartal, E., ve Özen, Z. Dengesiz veri setlerinde sınıflandırma, *Mühendislikte Yapay Zekâ ve Uygulamaları*, 1st ed, 2017.

# İstatistiksel ve Derin Öğrenme Modellerini Kullanarak Hisse Senedi Fiyat Tahmini

## Stock Price Prediction Using Statistical and Deep Learning Models

Emre ALBAYRAK  
Çankaya Üniversitesi,  
Bilgisayar Mühendisliği Bölümü,  
Ankara, Türkiye  
c1771010@student.cankaya.edu.tr  
ORCID: 0000-0003-1352-6776

A. Nurdan SARAN  
Çankaya Üniversitesi,  
Bilgisayar Mühendisliği Bölümü,  
Ankara, Türkiye  
buz@cankaya.edu.tr  
ORCID: 0000-0002-7148-0457

### Öz

Borsa analizi, geleceğe yönelik tahminler yapmak için finansal, politik ve sosyal göstergeleri göz önünde bulundurarak borsayı inceler ve değerlendirir. Büyük veri ve derin öğrenme teknolojilerindeki gelişmelerin çığır açan sonuçları, araştırmacıların ve endüstrinin dikkatini bilgisayar destekli borsa analizine çekmektedir. Geleneksel makine öğrenimi ve derin öğrenme modellerini kullanarak borsa analizi konusunda çeşitli çalışmalar bulunmaktadır. Bu çalışmada, temel model olarak Otoregresif Entegre Hareketli Ortalama (ARIMA) yöntemini tekrarlayan sinir ağlarının üç farklı modeliyle karşılaştırılmıştır; Uzun Kısa Süreli Bellek (Long Short Term Memory- LSTM) ağları, Geçitli Tekrarlayan Birim (Gated Recurrent Unit- GRU), dikkat katmanlı LSTM modeli. Bu çalışmada literatürdeki diğer çalışmalardan farklı olarak 28 tane finansal indikatör kullanılarak Borsa İstanbul verileri üzerinde gün içi tahminler yaparken dört farklı modelin sonuçları karşılaştırılmıştır. İstatistiksel ve doğrusal bir model olan ARIMA, zaman serileri tahmini için doğrusal olmayan RNN modelleri ile karşılaştırılmıştır ancak 3 sinir ağı modelinden de yüksek ortalama hata oranına sahip olduğu görülmüştür. LSTM sonuçları GRU modeline çok yakın olsa da GRU diğerlerinden biraz daha iyi performans göstermektedir. Dikkat mekanizmalı sinir ağı diğer temel sinir ağlarından daha iyi sonuç vermemektedir.

**Anahtar Sözcükler:** Otoregresif Entegre Hareketli Ortalama, BIST, Geçitli Tekrarlayan Birim Modeli, Uzun Kısa Süreli Bellek Modeli

### Abstract

The stock market analysis examines and evaluates the

Gönderme, düzeltme ve kabul tarihi: 02.11.2022 - 10.01.2023 – 13.06.2023

Makale türü: Derleme

stock market by considering the financial, political, and social indicators to make future predictions. Breakthrough results of advancements in big data and deep learning technologies attract the attention of researchers and

traders to computer-assisted stock market analysis. There are several studies on stock market analysis using conventional machine learning and deep learning models. In this paper, we used Autoregressive Integrated Moving Average (ARIMA) as a base model and compared it with three different models of recurrent neural networks: Long Short-Term Memory (LSTM) networks, Gated Recurrent Unit (GRU), LSTM with an attention layer model. While making intraday forecasts on Borsa İstanbul data, the results of four different models have been compared. The statistical model, ARIMA, is used as a baseline model for comparison with neural networks, but it has higher mean absolute error than other neural network models. Even though the LSTM results are very close to the GRU model, GRU slightly outperforms the others. The attention neural network model does not give better results than other basic neural networks.

**Keywords:** ARIMA, BIST, GRU, LSTM

### 1. Giriş

Borsa analizi, kripto para birimleri, yüksek performanslı uygulamalar vb. ile günümüzün en ilgi çeken konularından biridir. İki çeşit analiz yönteminden bahsetmek mümkündür: temel ve teknik analiz. Temel analiz, borsa analiz sürecindeki verilerden yararlanmak için çok çeşitli veri kaynaklarını birleştirir. Veri kaynakları, siyasi haberler, hükümet politikaları gibi birçok farklı kaynak olabilir. Teknik analiz, hisse senedi çizelgeleri, mevcut piyasa eğilimleri, ticaret hacimleri vb. gibi finansal göstergelere dayanır. Teknik analiz, kısa vadeli hisse senedi davranışını anlamak için kullanılırken, uzun vadeli tahminler için temel analiz kullanılır [1]. Hisse



senedi tahmini için öncelikle istatistiksel yöntemler önerilmiştir; otoregresif bütünlümlü hareketli ortalama (ARIMA) [22] ve genelleştirilmiş otoregresif koşullu değişen varyans (GARCH) [23] gibi. ARIMA modeli, son teknoloji etkinliği ve sağlamlığı nedeniyle finansal tahmin uygulamalarında yaygın olarak kullanılan istatistiksel modellerden biridir [9]. ARIMA'nın tahmin ve karşılaştırma aşamalarında ayırt edici kullanımının yanı sıra veri işleme aşamalarında melez öznelilikler olarak derin öğrenme ile birleştirildiği çalışmalar vardır [12] [13] [14].

Ancak doğrusal olmayan, yüksek gürültülü finansal zaman serilerinde geleneksel istatistik modelleri bunları yüksek hassasiyetle tahmin etmekte güçlük çekmektedir [24]. Makine öğrenmesi modellerinin geliştirilmesiyle birlikte, regresyon ve destek vektör makinesi gibi yöntemler finansal zaman serileri tahmini için de uygulanmaktadır [25] [5] [6]. Destek vektör makineleri her ne kadar sınıflandırma yöntemleri için tasarlanmış olsa da zaman serileri tahmini içinde yayınlarda kullanılmıştır [26].

Günümüzde sinir ağları olarak da bilinen derin öğrenme algoritmaları, doğrusal olmayan yaklaşım yeteneği ve adaptif kendi kendine öğrenmesi sayesinde güncel bir tahmin aracı haline gelmiştir. Bu algoritmalar, bilgisayarla görme, doğal dil işleme ve konuşma tanıma alanlarında çığır açan sonuçların arkasında olmuştur. Hisse senedi fiyatlarını doğru bir şekilde tahmin etmenin karmaşıklığı ve ortaya çıkan büyük veri ve derin öğrenme teknolojileri, borsa analizlerinde bilgisayarların kullanılmasına yol açmıştır [2] [3]. Uygun fiyatlı emtia donanımı, hızla gelişen bilgi erişim teknolojileri ve artan veri hacimleri, anlamlı örüntüler ve ilişkiler çıkarmak için büyük hacimli verilerin işlenmesini mümkün kılar [4]. Derin öğrenmede en çok çalışılan yöntemlerden Evrişimli Sinir Ağları (CNN) zaman serisi verilerine az da olsa uygulanarak hisse senedi tahminleri için kullanılmıştır [1]. Ancak tekrarlayan sinir ağları (LSTM [7] [2], GRU gibi özellikle zaman serileri için tasarlandığı için hisse senedi tahminleri için literatürde sıklıkla kullanılmıştır. RNN ağları, zaman serisi gibi sıralı verileri işlemeye özellikle uygundur. Derin öğrenme yöntemleri, hisse senedi fiyatları hakkında doğru tahminler elde etmek için kullanılan hisse senedi verilerinin içindeki içsel doğrusal olmayan ilişkileri ve gizli faktörleri anlamamıza yardımcı olur [8].

2014'te Bahdanau ve ark. tarafından dikkat mekanizması [33] makine çevirisi için önerilmiş ancak sonra zaman serisi tahmininde ve hisse senedi tahmini için de kullanılmıştır [3]. Hisse senedi fiyatları gibi veri dizilerinin durağan olmama özelliği, zaman serisinin istatistiksel özelliklerinin (örneğin, ortalama ve varyans gibi) zaman içinde sabit olmaması olarak açıklanabilir [28]. Bu, zaman serisinin sürekli değiştiği anlamına gelir ve geçmiş verilere dayanarak uzun vadeli tahminler yapmak zorlaştırmaktadır. Dikkat mekanizması, modele tahmin yaparken girdi dizisinin belirli kısımlarına odaklanmasına izin verdiğinden zaman serilerinin durağan olmayan özelliğine rağmen modelin zaman içinde verilerdeki değişen desenlere uyum sağlamasına izin vermektedir [31]-[32].

Bu çalışmada temel amacımız, BIST verilerini kullanarak gün içi tahminler yapmak için istatistiksel modelleri ve tekrarlayan sinir ağları ile öğrenme yöntemlerini karşılaştırmaktır. Karşılaştırmalı bir çalışma ortaya çıkarmak için istatistiksel bir model olan ARIMA modeliyle, dikkat mekanizmalı LSTM gibi en popüler ve üst düzey derin öğrenme yöntemleri kullanılmaktadır. ARIMA hem istatistiksel hem de derin öğrenme modelleriyle karşılaştırma amacıyla yaygın olarak bir temel model olarak birçok çalışmada kullanılmıştır [10] [11]. Hisse senedi tahmini için ham fiyat ve hacim kullanımının yanı sıra, fiyat hareketlerinin tahminine temel ve teknik göstergeler kullanılarak yaklaşılabılır [26]-[27]-[29]. Dikkat mekanizmalı LSTM yönteminin finansal piyasa tahminine uygulanıp uygulanamayacağına ilişkin yayınlarda az çalışma gösterilmiştir. Bu nedenle, bu çalışma, Borsa İstanbul verileri üzerinde gün içi hisse senedi piyasasını tahmin etmek için 28 tane finansal göstergeler kullanılarak dikkat mekanizmalı LSTM yöntemi, GRU gibi derin öğrenme yaklaşımlarını istatistiksel bir yöntem olan ARIMA ile karşılaştırmaktadır.

Bu makale 4 bölümden oluşmaktadır. Bölüm 1, stok analizi, derin öğrenme konularına kısa bir giriş sağlar. Ayrıca ilgili çalışmaları özetler. Bölüm 2, modellerin ayrıntılı açıklamasını verir. Veri alma, ön işleme operasyonları ve özellikle mühendisliği süreçlerini açıklar. Bölüm 3'te model oluşturma aşamaları, sonuçlar ve ayrıntılı grafik ve sonuçları içeren tartışma bölümleri yer almaktadır. 4. bölüm çalışmamızın sonuç bölümüdür.

## 2. Yöntem

### 2.1. Veri

Veriler, Borsa İstanbul'daki 100 farklı hisse senedinin saatlik verileri olup, veriler Matrics Bilgi Dağıtım Hizmetleri A.Ş. aracılığıyla elde edilmiştir. Veriler 2001-2020 tarih aralığında olup, toplam 772533 satırdır.

Her dosyanın beş veri sütunu vardır: tarih saat, açılış fiyatı, yüksek fiyat, düşük fiyat, kapanış fiyatı ve hacim. Ön işleme sırasında 28 teknik gösterge ile hesaplama yapılmış ve ayrıca USD/TRY para birimi verilerini de eklenmiştir. Bu çalışmada, yer kaplamaması açısından sonuçları gösterirken alfabetik sıraya göre Borsa İstanbul'dan üç farklı hisse senedi seçilmiştir; ASELS, AKBNK ve AKENR. Ön işleme safhası sonrasında elde edilen teknik göstergeler ve formülleri Çizelge-1'de yer almaktadır.

**Çizelge-1 Teknik Göstergeler ve Açıklamaları**

Gösterge	Açıklama
SMA	Basit Hareketli Ortalama
RSI	Göreceli Güç Endeksi
CCI	Emtia Kanal Endeksi
DMI	Yönel Hareket Endeksi
ADX	Ortalama Yönel Endeksi
VPT	Hacim Fiyat Trendi
EFI	Elder Güç Endeksi
WOBV	Ağırlıklı Denge İşlem Hacmi
VZO	Hacim Bölgesi Osilatörü
PZO	Fiyat Bölgesi Osilatörü
TP	Tipik Fiyat

ADL	Birikim Dağıtım Göstergesi
SMMA	Yuvarlatılmış Hareketli Ortalama
TR	Gerçek Aralık
SAR	Bitiş ve Dönüş Göstergesi
VWAP	Hacim Ağırlıklı Ortalama Fiyat
SSMA	Yuvarlatılmış Basit Hareketli Ortalama
DEMA	Çift Üstel Hareketli Ortalama
TEMA	Üçlü Üstel Hareketli Ortalama
TRIX	Üçlü Üstel Hareketli Ortalama Osilatörü
CURRENCY	Dolar/TL Paritesi

Çizelge-1'deki teknik göstergelerin açıklamaları ve hesaplama yöntemleri aşağıda verilmiştir:

**Göreceli Güç Endeksi (RSI):** Hisse senedinin değişen fiyatının gücünü değerlendiren bir göstergedir. Hisse senedinin şimdiki ve zaman aralığındaki güç ve zayıflığını ölçer:

$$RSI = 100 - \frac{100}{H_t - L_t}$$

Burada, H: t zaman aralığında en yüksek ortalama, L: t zaman aralığında en düşük ortalama

**Basit Hareketli Ortalama (SMA):** En çok kullanılan momentum göstergelerindendir, belirli bir zaman aralığında endeksin kapanış fiyatları toplanarak zaman sayısına bölünmesiyle hesaplanır:

$$SMA = \frac{\sum_{i=0}^n C_i}{n}$$

Burada n zaman aralığı, C<sub>i</sub> kapanış değerleri

**Yuvarlatılmış Basit Hareketli Ortalama (SMMA):** Basit hareketli ortalamasının benzeri olan, ancak daha fazla ağırlığı son veri noktalarına veren bir teknik göstergedir. Önceki n veri noktalarının toplamı alınarak, sonra n+1 ile bölünerek hesaplanır.

$$SSMA_i = \frac{SSMA_{i-1} * (n - 1) + C_i}{n + 1}$$

**Üstel Hareketli Ortalama (EMA):** Basit hareketli ortalamasının üstel değişimidir, son fiyat değişikliklerine basit hareketli ortalamadan daha hızlı tepki verir [30]:

$$EMA_t = C_t * \frac{2}{n + 1} + EMA_y * \left(1 - \frac{2}{n + 1}\right)$$

Burada C<sub>t</sub> şimdiki kapanış, EMA<sub>t</sub> şimdiki değer, EMA<sub>y</sub> önceki değer (burada EMA<sub>0</sub> olarak SMA kullanılır). DEMA (çift üstel hareketli hesaplama), TEMA (üçlü üstel hesaplama) ve TRIX (üçlü üstel hareketli ortalama osilatörü) benzer şekilde hesaplanır.

**Emtia Kanal Endeksi (CCI):** Fiyatın mevcut ve geçmiş arasındaki ortalama farkıdır. [30].

$$CCI = \frac{(C - SMA)}{0.015 * Standard\_deviation}$$

**Denge İşlem Hacmi (OBV):** Endeks hacim akışını dikkate alan bir momentum göstergesidir.

$$OBV_t = \begin{cases} OBV_y + V_t & , if C_t > C_{t-1} \\ OBV_y & if C_t = C_{t-1} \\ OBV_y + V_t & if C_t < C_{t-1} \end{cases}$$

Burada C<sub>t</sub>, V<sub>t</sub> şimdiki kapanış ve hacim, C<sub>t-1</sub> önceki kapanış

**Tipik Fiyat (TP):** en yüksek, en düşük ve kapanış fiyatlarının ortalaması hesaplanır.

$$TP = \frac{H_t + L_t + C_t}{3}$$

**Direksiyonel Hareket İndeksi (DMI)** bir trendin yönünü belirlemek için kullanılan bir teknik göstergedir: pozitif yönel hareket indeksi (DI+) yükseliş eğilimini gösterirken, negatif yönel hareket indeksi (DI-) düşüş eğilimini gösterir.

$$DI+ = 100 * (H_t - H_{t-1}) / (H_t - L_t)$$

$$DI- = 100 * (L_t - L_{t-1}) / (H_t - L_t)$$

**Ortalama Yönel Endeksi (ADX):** Bir finansal enstrümanın eğiliminin güçlülüğünü ölçmek için kullanılan bir teknik analiz aracıdır.

$$ADX = (DI+ - DI-) / (DI+ + DI-)$$

**Hacim Fiyat Trendi (VPT):** Bir hisse senedinin fiyatı ve hacminin yüzdesel değişimlere dayalı olarak trend gücünü gösterir. Örneğin, bir enstrümanın fiyatı yükselirken hacim de artarsa, VPT değeri yükselecektir.

$$VPT_t = VPT_{t-1} + (C_t - C_{t-1}) * V_t$$

**Hareket Kolaylığı (EFI):** VPT gibi bir enstrümanın fiyatının hareket ettiği yönde hacim değişimlerini gösterir.

$$EFI = \frac{(C_t - C_{t-1}) * V_t}{(H_t - L_t) * \sum_t V_t}$$

**Birikim Dağılım Çizgisi (ADL):** Hisse senedi fiyatının birikimli ya da dağılımlı olup olmadığının bir göstergesidir.

$$ADL_t = ADL_{t-1} + ((C_t - L_t) - (H_t - C_t)) * V_t / (H_t - L_t)$$

**Gerçek Aralık (TR):** Bir varlığın volatilitasını ölçmek için kullanılan bir teknik göstergedir.

$$TR = \max(H_t - L_t, \text{abs}(H_t - C_{t-1}), \text{abs}(L_t - C_{t-1}))$$

**Bitiş ve Dönüş Göstergesi (SAR):** Potansiyel eğilim dönüşlerini belirlemek için kullanılan teknik bir göstergedir.

$$SAR_t = SAR_{t-1} + AF * (SAR_{t-1} - SAR_{t-2})$$

SAR(t) enstrümanın t. zaman için tahmini fiyatı gösterir ve AF ise hızlandırma faktörüdür ve genellikle 0,02 ile 0,20 arasında değişir.

**Hacim Ağırlıklı Ortalama Fiyat (VWAP):** Hacmi dikkate alarak, belirli bir zaman diliminde ortalama fiyatını belirlemek için kullanılan teknik bir göstergedir. Menkul kıymetteki tüm işlemlerin toplam değerinin menkul kıymetin toplam hacmine bölünmesiyle hesaplanır.

**Osilatör:** Finansal gösterge olarak kullanımları çok fazladır; henüz başlamamış olası bir eğilim değişimini işaret edebilecek öncü ve gecikmeli göstergeler olarak kabul edilirler.

**Hacim Bölgesi Osilatörü (VZO):** Şimdiki kapanış fiyatı önceki kapanış fiyatından yüksekse, hacim pozitif bir değere sahip olacaktır; aksi takdirde negatif bir değere sahip olacaktır.

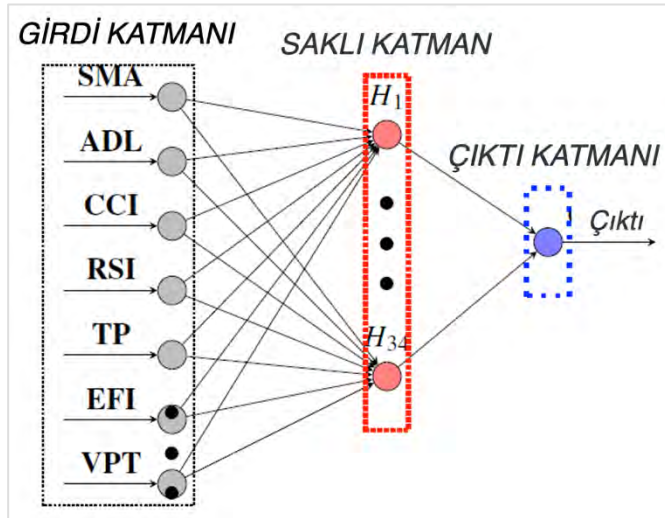
**Fiyat Bölgesi Osilatörü (PZO):** Mevcut fiyatı ortalama geçmiş fiyatlara karşı ölçen teknik bir göstergedir. Gösterge, iki üstel hareketli ortalamayı hesaplar ve aralarındaki orandır.

Tüm göstergelerde de varsayılan periyot 14 alınmıştır [34].

## 2.2. Modeller

Bu çalışmada BİST verileri üzerinde teknik analiz yapmak ve gün içi frekansta başarılı tahminler yapmak hedefi ile hem istatistiksel hem de derin öğrenme modelleri uygulanmıştır. İstatistiksel modeller açısından temel model olarak ARIMA, yapay sinir ağları açısından ise LSTM, GRU, dikkat mekanizmalı LSTM modelleri kullanılmıştır. ARIMA modeli, tüm verilerdeki alt kümelerin ortalamalarının hesaplanmasına ve gelecekteki eğilimler hakkında tahminlerde bulunulmasına dayanmaktadır. Sinir ağı modelleri, zaman serisi veri ile ilgili uygulamalarda yaygın olarak kullanılan tekrarlayan sinir ağları modelleridir. LSTM, Vanilla RNN'nin bir gelişimi olarak kabul edilebilir ve GRU, LSTM 'in bir çeşididir. Dikkat mekanizmasına sahip LSTM, dikkat mekanizmasını kullanan bir kodlayıcı-kod çözücü modelidir.

Yapay sinir ağları, insan beynini taklit eden ve bilgisayarla görme sorunları, konuşma tanıma, görüntü işleme vb. alanlarda yaygın olarak kullanılan özel makine öğrenme algoritmaları sınıfıdır. Yapay sinir ağlarının bir çeşidi olan tekrarlayan sinir ağları zaman içinde meydana gelen bilgileri anlamlandırabilmektedir. Sinir ağları, derin ağları eğitmeyi çok daha kolay hale getiren bir yeniliktir. Şekil 1 Yapay Sinir Ağlarının Genel Yapısı, bir YSA'nın giriş ve gizli katmanlarını göstermektedir.



Şekil-1: Yapay Sinir Ağlarının Genel Yapısı

Ölçünlü bir ileri beslemeli sinir ağında, bir nöron aldığı girdiyi toplar ve ardından bir çıktı üretmek için girdiyi bir etkilenim fonksiyonundan (ReLU, Sigmoid, Tanh, vb.) geçirir. Tekrarlayan bir sinir ağında, bir nöron çıktığı kendisine geri gönderir. Tekrarlayan bir nöron açılırsa, tekrarlayan nöronun bir önceki zamandaki (t-1) girişi t-1'de çıktı verir ve daha sonra t zamanındaki durumunda nörona geçirilir ve ardından t

zamanında bir çıktısı olur ve bu böyle devam eder. Tekrarlayan nöron, hem önceki bir zaman adımından girdileri hem de mevcut zaman adımından girdileri alır. Tekrarlayan bir sinir ağında, önceki zaman adımlarından girdi alan bu nöronlar "bellek hücreleri" olarak da bilinir. LSTM ve GRU, tekrarlayan bir sinir ağlarının alt sınıflarıdır ve aşağıdaki bölümlerde bu iki tür ayrıntılı olarak açıklanacaktır.

## Otoregresif Entegre Hareketli Ortalama (ARIMA)

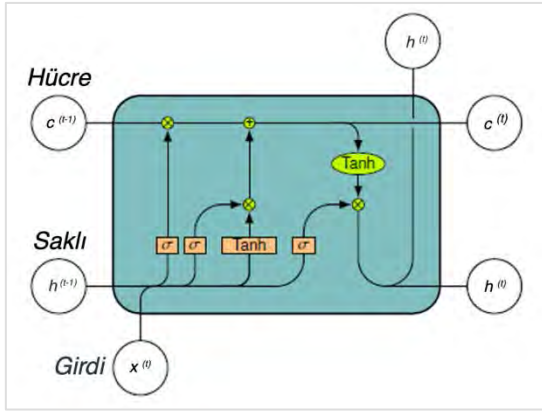
Bu model, gelecekteki eğilimleri tahmin etmek için geçmiş verileri kullanan zaman serisi veri analizinde kullanılan istatistiksel bir modeldir. ARIMA modeli, ARMA (Otomatik Regresif Hareketli Ortalama) modeline dayanmaktadır. ARMA modeli, AR (Otomatik Regresif) ve MA (Hareketli Ortalama) bölümlerinden oluşur. Oto regresif, mevcut değer geçmiş değerlere dayandığını düşünen bir modeli ifade eder. Hareketli ortalama, tüm veri setinin farklı alt kümelerinin ortalamalarından oluşan bir hesaplama. ARIMA ve ARMA arasındaki temel fark, ARIMA modelinin durağan olmayan verileri üzerinde çalışmaya başlamadan önce durağan hale getirmesidir. Akaike Bilgi Kriterleri (AIC), model performanslarını karşılaştırmak için istatistiksel bir modelde kullanılan bir ölçüm metriğidir [15].

## Uzun Kısa Süreli Bellek Ağları (LSTM)

Derin ağlarda, bazen kaybolan veya patlayan türevler (gradient) olarak adlandırılan geri yayılımdan kaynaklanan sorunlar ortaya çıkabilir. Alt seviye katmanlara veya giriş katmanlarına daha yakın olan ön katmanlara geri döndüğümüzde, türevler genellikle küçülür ve sonuçta nöronların ağırlıklarının bu alt seviyelerde asla değişmemesine neden olur. Farklı bir etkinleştirme işlevi veya toplu normalleştirme teknikleri, bir ölçekte kaybolan türev sorunlarına çözüm olarak önerilebilir. Ancak, uzun zaman serisi verileri gibi dizi girişinin uzunluğu nedeniyle, önerilen bu çözümler eğitimi yavaşlatabilir.

Tekrarlayan sinir ağlarının karşılaştığı bir başka sorun da, bu ağlarda geçen her adımda bilgi kaybolduğu için bir süre sonra ağın ilk girdileri "unutmaya" başlamasıdır. Bu noktada, tekrarlayan sinir ağları için uzun süreli bir bellek çözümüne ihtiyaç duyulmaktadır. Uzun Kısa Süreli Bellek hücresi, 1997'de kaybolan türev sorununun ele alınmasına yardımcı olmak için oluşturulmuştur [16].

LSTM hücresi, giriş kapısı, unutma kapısı ve çıkış kapısı olmak üzere üç farklı kapı içermektedir. Hücre durumunda neyin depolanacağına karar vermek için giriş kapısı, hücre durumundan hangi bilgilerin unutulacağını veya atılacağını belirlemek için unutma kapısı, hücre durumunda neyin güncelleneceğine karar vermek için çıkış kapıları kullanılmaktadır [16] [17].



Şekil-2 LSTM in Genel Yapısı

Şekil 2'de  $x_t$  geçerli giriş,  $c_{t-1}$  son LSTM biriminin belleği olan hücre durumu,  $h_{t-1}$  son LSTM biriminin çıkışı olan gizli durumdur.  $h_t$ , mevcut çıktıdır.  $c_t$  yeni güncellenen bellektir. Bir sonraki hücre durumu olarak kabul edilebilir.  $\sigma$  ve  $Tanh$ , sigmoid ve hiperbolik tanjant fonksiyonlarını,  $x$ , ölçekleme bilgisi işlemini ve  $+$  bilgi ekleme işlemini ifade eder.

Denklemlerde  $i_t$  giriş kapısı,  $f_t$  unut kapısı ve  $o_t$  çıkış kapısıdır.  $W$ , önceki gizli durum ile mevcut gizli durum arasındaki bağlantıdır.  $U$ , girdiler ve gizli katman arasındaki ağırlık matrisidir.  $\tilde{C}_t$ , mevcut girdinin ve önceki gizli durumun hesaplanması olan aday gizli durumdur.  $C_t$ , hücrenin dahili hafızasıdır.  $C_t$ , önceki belleğe dayalı olarak hesaplanır, unut kapısı, yeni gizli durum ve giriş kapısı ile çarpılır.

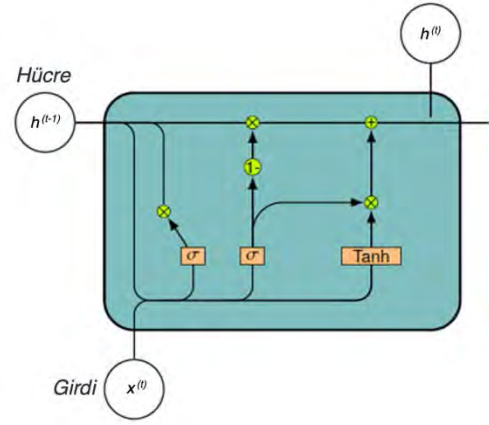
Ayrıntılı LSTM denklemleri aşağıdaki gibidir:

$$\begin{aligned} i_t &= \sigma(x_t U^i + h_{t-1} W^i) \\ f_t &= \sigma(x_t U^f + h_{t-1} W^f) \\ o_t &= \sigma(x_t U^o + h_{t-1} W^o) \\ \tilde{C}_t &= \tanh(x_t U^g + h_{t-1} W^g) \\ C_t &= \sigma(f_t \times C_{t-1} + i_t \times \tilde{C}_t) \\ h_t &= \tanh(C_t) \times o_t \end{aligned}$$

Kullandığımız LSTM modeli 3 katman LSTM ve dropout(0,2) fonksiyonları ve en on katmanda dense katmanından oluşturulmuştur.

### Geçitli Tekrarlayan Birimler (GRU)

LSTM hücresinin bir varyantı, 2014'te tanıtılan GRU olarak adlandırılır [18]. GRU, unut ve giriş kapılarını "güncelleme kapısı" adı verilen tek bir geçitte birleştirerek LSTM' in iç yapısını basitleştirir. Ayrıca hücre durumunu ve gizli durumu birleştirir. Güncelleme kapısı, hangi bilgilerin iletileceğine karar vermek için kullanılır, sıfırlama kapısı, hangi bilgilerin unutulacağını belirlemek için kullanılır. Bu kapılar sırasıyla uzun süreli ve kısa süreli belleği temsil eder.



Şekil 3 GRU nun Genel Yapısı

GRU, mevcut hücre durumunu ve LSTM' in gizli durumunu önceki aşamanın çıktısı olarak tek bir durumda birleştirir. Şekil 3'te girişler, mevcut giriş ve hücre durumudur.  $x^t$  mevcut girdi,  $h^{t-1}$  önceki aşamanın çıktısı,  $h^t$  mevcut çıktıdır. GRU, LSTM' in bir varyantı olarak LSTM' in işlevlerini ve işlemlerinin çoğunu korur.

Ayrıntılı denklemlerde  $z_t$ , güncelleme kapısını ve  $r_t$ , sıfırlama kapısı,  $\tilde{h}_t$  aday gizli durumu ve  $h_t$  gizli durumu,  $W$  ağırlık matrisidir. Ayrıntılı GRU denklemleri aşağıdaki gibidir:

$$\begin{aligned} z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\ r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\ \tilde{h}_t &= \tanh(W \cdot [r_t \times h_{t-1}, x_t]) \\ h_t &= (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \end{aligned}$$

Kullandığımız GRU modeli 3 katman GRU ve dropout(0.2) fonksiyonları ve en on katmanda dense katmanından oluşturulmuştur.

### Dikkat mekanizmalı LSTM

Derin öğrenmede dikkat mekanizması, eğitim sürecinde verilerin belirli özelliklerine odaklanır. Dikkat mekanizmasının en sık kullanılan durumlarından biri, kodlayıcı-kod çözücü modeli olarak da bilinen seq2seq modelleridir [18]. Sıralı derin öğrenme modelleri, doğal dil işleme, tanıma ve zaman serisi verilerini kullanan uygulamalarda yaygın olarak kullanılan tekrarlayan sinir ağları olarak kabul edilir. Kodlayıcılar ve Kod Çözücüler, LSTM veya GRU modelleri olabilir. Bu mekanizmada, kodlayıcı, dahili durumların girdi verilerini kod çözücüyeye iletir ve kod çözücü, kodlayıcıdan alınan girdi verilerinin dahili durumlarına dayalı olarak çıktı üretir. Dikkat mekanizmalı seq2seq modelinde, kod çözücü çıktı üretirken standart seq2seq model davranışı olan kodlayıcının son durumu yerine tüm dizinin iç durumlarını dikkate alır. Dikkat mekanizması, çıktıyı üretmek için belirli özelliklere odaklanırken kod çözücünün tüm verilere erişmesine izin verir.

[19]'da yazarlar, dikkat temelli sinir ağlarını global ve yerel dikkat modelleri olmak üzere iki geniş kategoriye ayırmaktadır. Bu iki model arasındaki fark, kodlayıcının çıktısını işleme biçimine bağlıdır. Global dikkatte, kodlayıcının tüm gizli durumları dikkate alınır, bu da tüm özelliklere önem verildiği anlamına gelir. Ancak, yerel dikkatte, çıktının üretilmesinde gizli durumların seçilmiş alt kümeleri kullanılır.

Bahdanau ve Luong, zaman serisi veri uygulama problemlerini ele almak için yaygın olarak kullanılan iki yaygın dikkat modelidir [3] [20] [18]. Bu iki mekanizma arasındaki temel fark, puan benzerliklerinin hesaplanma şeklidir. Luong'un dikkati, basit matris çarpımlarını kullanır ve böylece daha hızlı ve alan açısından daha verimli hale getirir. Her iki model de dikkat mekanizmasında "bağlam vektörü" olarak adlandırılan çıktıyı üretirken tüm gizli durum bilgilerini kullanan global dikkat modelleri olarak kabul edilebilir [19]. Buna ek olarak, Luong dikkati yerel dikkat mekanizması olarak da uygulanabilir.

Aşağıdaki denklemler, Bahdanau ve Luong'un puan benzerliklerini hesaplama yöntemi arasındaki farkı göstermektedir.  $h_s$ , kodlayıcıdan kod çözücüyeye iletilen giriş verilerinin gizli durumlarını belirtir.  $h_t$ , kod çözücünün önceki çıktısı,  $c_t$  bağlam vektörü ve  $W_t$  ağırlık matrisidir. Bu puanları, kodlayıcının tüm gizli durumları ile önceki kod çözücünün çıktısı arasındaki ilişkinin düzeyi olarak çıkarabiliriz. Bu puanlar, dikkat ağırlıklarının aşağıdaki denklemleri hesaplamak için kullanılır:

$$score(h_t, \tilde{h}_s) = \begin{cases} h_t^T W \tilde{h}_s & \text{Luong} \\ v_a^T \tanh(W_1 h_t + W_2 \tilde{h}_s) & \text{Bahdanau} \end{cases}$$

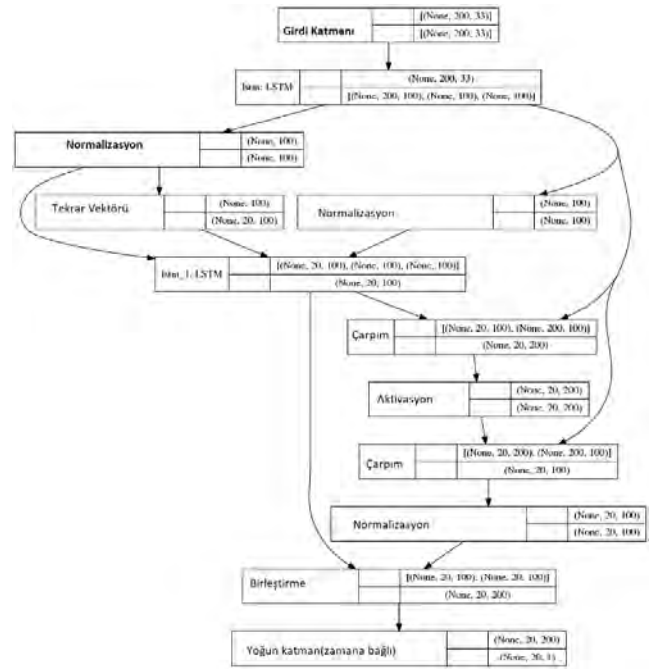
Bağlam vektörü, aşağıdaki gibi dikkat ağırlıkları kullanılarak hesaplanır:

$$a_{ts} = \frac{\exp(score(h_t, \tilde{h}_s))}{\sum_{s'=1}^S \exp(score(h_t, \tilde{h}_{s'}))}$$

Puan benzerliklerinin hesaplanması dışındaki aşamalar hem Bahdanau hem de Luong dikkat mekanizmaları için aynıdır.

$$c_t = \sum_s a_{ts} \tilde{h}_s$$

Bu çalışmada, 2015 yılında tanıtılan Luong küresel dikkat mekanizmasını kullanmıştır [19]. Dikkat mekanizması, ağdaki nöronlara ağırlık atarken diğer modellerden temel olarak farklıdır. Ağırlıkları ayarlayarak ağırlık verilerin özneliklerinin dikkate almasını sağlar [19]. 33 öznelik (adım sayısı:20) ile kullanılan dikkat mekanizmalı LSTM modeli Şekil 4 'te verilmiştir.



Şekil 4 Dikkat Mekanizmalı LSTM Modeli

Burada her katmanın ilk satırındaki 3-lü girdi, ikinci satırındaki 3-lü çıktı büyüklüğünü vermektedir. Normalizasyon için toplu normalizasyon, çarpım için nokta çarpım, birleştirme uç uca ekleme, yoğun katman çıktıya göre şekillendirme (dense) katmanı ifade etmektedir.

### 3. Sonuçlar ve Tartışmalar

#### Hiper Parametre Seçimi:

Arima için parametre seçimi: Çizelge 2 de verildiği üzere grid arama sonuçlarına göre görece uzun sürmesine rağmen en iyi performans veren ARIMA modeli ARIMA (2,1,0) olmuştur.

Burada ARIMA parametreleri (p, d, q) sırasıyla; p, verilerin geçmişteki değerleriyle ilişkisini belirtir, d (değişim miktarı), verilerdeki trendleri düzeltmek için kullanılır (seriyi durağan hale getirmek için gereken fark sayısıdır), q (Hareketli Ortalama) verilerdeki dalgalanmaları (hata terimleri) ölçen bir değişkendir.

Çizelge-2: Arima için Parametre Seçimi

Model Parametresi	AIC	Zaman
(0,1,0)	21592.2	0.20
(2,1,0)	<b>21638.4</b>	<b>0.84</b>
(1,1,0)	21590.2	0.25
(3,1,0)	21637.2	0.48
(2,1,1)	21637.0	0.45
(1,1,1)	21588.7	0.32
(3,1,1)	21635.2	0.55

**Yinelenen Sinir Ağları için Parametre Seçimi:** Yapay sinir ağlarında hiperparametrelerin farklı kombinasyonları eğitimin hızı ile verilerin geliştirilmesi arasında bir ödünleşime neden olur, bir başka deyişle, ağırlık çalışma hızını arttıran parametreler sistemin geliştirme yetkinliklerini azaltabilir. Bu nedenle hiperparametreler, test süreçlerimizde farklı

hiperparametre kombinasyonları dikkate alınarak ve en iyi sonuç veren parametreler elde edilmiştir. Bu çalışmada, Adam/Rmsprop optimizasyonu kullanılmıştır ve öğrenme katsayısı 0,001 seçilmiştir.

**Çizelge- 3: ASELS Hisse Senedi için Farklı Dönem Sonuçları**

Model	Dönem	Toptan	Adım Sayısı	MSE
LSTM	50	20	30	0,0014
GRU	50	20	30	0,0013
Dikkat Mekanizmalı	50	20	30	0,0471
LSTM	100	20	30	0,0011
GRU	100	20	30	0,0010
Dikkat Mekanizmalı	100	20	30	0,0460
LSTM	150	20	30	0,0009
GRU	150	20	30	<b>0,0008</b>
Dikkat Mekanizmalı	150	20	30	0,0456

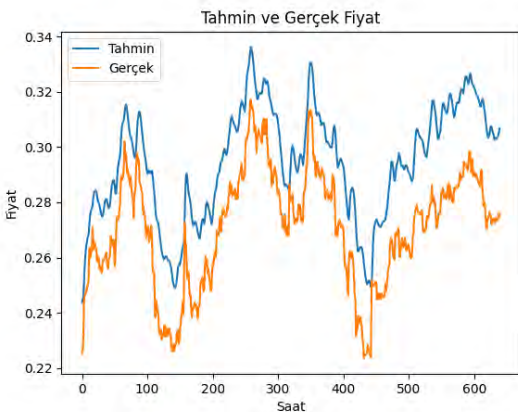
Çizelge-3'te farklı dönem (epoch) 50, 100, 150 değerleri için test sonuçları verilmiştir [21].

Çizelge-4'de görüldüğü gibi LSTM modelinde RELU, sigmoid fonksiyona göre daha iyi sonuçlar verdiği gözlemlenmiştir ve hiperbolik tanjant fonksiyonu en iyi sonuçları vermiştir.

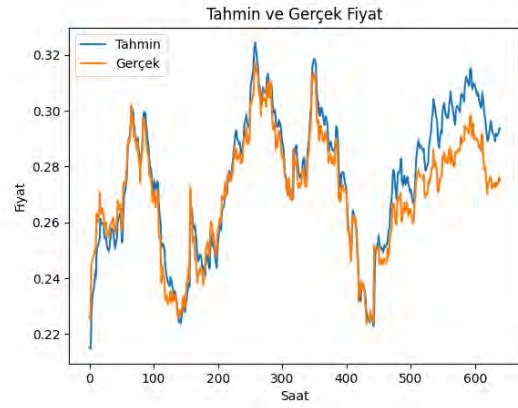
**Çizelge- 4 Tüm Hisse Senetleri için Aktivasyon Fonksiyonu Seçimi**

Aktivasyon	Hisse Senedi	MSE
Sigmoid	AKBNK	0,0097
RELU	AKBNK	0,0049
<b>Tanh</b>	AKBNK	<b>0,0042</b>
Sigmoid	AKENR	0,0080
RELU	AKENR	0,0042
<b>Tanh</b>	AKENR	<b>0,0036</b>
Sigmoid	ASELS	0,0444
RELU	ASELS	0,0274
<b>Tanh</b>	ASELS	<b>0,0266</b>

Modellerimizi BİST verilerinden üç farklı hisse senedine uygulanmıştır. ASELS hisse verileri için LSTM, GRU sonuçları (150 dönem-epoch), ARIMA (2,1,0) sırasıyla Şekil 5, Şekil 6 ve Şekil 7'da verilmektedir.



**Şekil 5 ASELS hisse senedi için LSTM sonuçları**



**Şekil 6 ASELS hisse senedi için GRU sonuçları**



**Şekil 7 ASELS hisse senedi için ARIMA (2,1,0) sonuçları**

Çizelge-5'te 3 hisse senedi için ARIMA sonuçlarını göstermektedir.

**Çizelge-5 ARIMA Karşılaştırma Çizelgesi**

Stock	MSE	MAE
ASELS	0,0105	0,0836
AKBNK	0,0194	0,1066
AKENR	1,4056	0,9692

Modellerin ortalama kare hata (MSE) ve ortalama mutlak hata (MAE) karşılaştırması Çizelge-6'da ASELS hisse senedi için, Çizelge-7 AKENR hisse senedi için,

Çizelge-8 AKBNK hisse senedi için verilmiştir. Burada ARIMA modeli için en iyi performans veren ARIMA (2,1,0) kullanılmış, derin öğrenme modelleri içinde 150 dönem, 20 toptan, 30 adım sayısı, Adam optimizasyonu 0.0001 öğrenme katsayısı kullanılmıştır.

**Çizelge-6 ASELS verileri için 4 modelin karşılaştırılması**

Model	MSE	MAE
ARIMA	0,0105	0,0836
LSTM	0,0009	0,0213
GRU	<b>0,0008</b>	<b>0,0205</b>
ATTENTION	0,0405	0,1670

Çizelge-7 AKENR verileri için 4 modelin karşılaştırılması

Model	MSE	MAE
ARIMA	1,4056	0,9692
LSTM	0,0019	0,0290
GRU	<b>0,0017</b>	<b>0,0284</b>
ATTENTION	0,0530	0,1959

Çizelge- 8 AKBNK verileri için 4 modelin karşılaştırılması

Model	MSE	MAE
ARIMA	0,0194	0,1066
LSTM	0,0021	0,0317
GRU	<b>0,0014</b>	<b>0,0257</b>
ATTENTION	0,0484	0,1817

Çizelgede görüldüğü üzere üç hisse senedi içinde derin öğrenme modeli olan LSTM ve GRU modelleri, istatistiksel model olan ARIMA modeline göre daha iyi sonuç vermektedir ancak GRU her üç hisse senedinde de en iyi sonuçlar vermektedir. Dikkat katmanına sahip LSTM, veri dizisinin uzun olmasına rağmen diğer modellerden daha iyi performans göstermediği görülmektedir.

Bu makalede, hata değerlerini göz önünde bulundurarak üç model karşılaştırılmıştır ve BIST100 deki tüm hisse senetleri içinde genel olarak en iyi sonuçları GRU modelinin verdiği gözlemlenmiştir.

#### 4. Sonuç

Deneylerimizi BİST' ten AKBNK, AKENR, ASELS olmak üzere üç farklı hisse senedi üzerinde gösterilmiştir. Verilerimizi dört modele beslenmektedir: LSTM, GRU, LSTM ile dikkat katmanı ve ARIMA. Model oluşturmada, dönem sayısı, parti boyutu, öğrenme oranı, etkilenim fonksiyonu, kayıp fonksiyonu, zaman adımı vb. seçimimize karar vermek için birçok farklı parametre kombinasyonu ile deneysel çalışmalar yapılmıştır. Etkilenim fonksiyonu olarak hiperbolik tanjant fonksiyonunu ve kayıp fonksiyonu olarak MAE ve MSE fonksiyonlarını kullanılmıştır. BIST verilerinin teknik analizini yaparak, seçilen hisse senetleri için bir sonraki saatin açılış fiyatını tahmin etmeyi amaçlanmıştır.

Çalışmamızda, üç farklı hisse senedi için dört modeli karşılaştırdık ve deneysel sonuçlar, GRU modeli BİST hisse senedi fiyatlarını tahmin etmede diğer modellerden daha iyi performans göstermektedir. ARIMA, istatistiksel modellerin etkilerini sinir ağları ile karşılaştırmak için bir temel model olarak kullanılmıştır. Hem LSTM hem de GRU, ARIMA modelinden daha iyi performans göstermiştir. Dikkat katmanı performansına sahip LSTM, diğer modellerin gerisinde kalmıştır. Bunun nedeni, uyguladığımız küresel dikkat modeli

olabilir. Çıktıyı üretecek iç yapısı nedeniyle uzun dizileri işlemede zorluklar yaşayabildiği gözlemlenmiştir. Genel olarak, GRU' nun performansı hem istatistiksel hem de diğer sinir ağı modellerinde en iyisidir.

Hisse senedi tahminlerinde verinin durağan olmamasından dolayı diğer makine öğrenmesi veri setlerinde olduğu gibi normalizasyon uygulanmamaktadır. Bu çalışmada çalışma da birçok finansal indikatör hesaplanmış olsa da özellik ayırıştırma optimizasyonu yapılması mümkündür. Çalışmanın ilerleyen versiyonlarında global dikkat mekanizması yerine yerel dikkat mekanizması kullanılarak dikkat modelinin performansı değerlendirilebilir.

#### Kaynakça

- [1] Gunduz, H., Yaslan, Y., And Cataltepe, Z., *Intraday prediction of borsa istanbul using convolutional neural networks and feature correlations*. Knowledge- Based Systems, 137:138–148, 2017.
- [2] Boronovkova, S. And Tsiamas, I., *An ensemble of lstm neural networks for high-frequency stock market classification.*, Journal of Forecasting, 38(6):600–619, 2019.
- [3] Qiu, J., Wang, B., And Zhou, C. , *Forecasting stock prices with long-short term memory neural network based on attention mechanism*. PLOS ONE, 15(1):1–15, 2020.
- [4] Hasan, A., Kalipsiz, O., And Akyoku, S., *Predicting financial market in big data: Deep learning*, International Conference on Computer Science and Engineering (UBMK), pages 510–515, 2017.
- [5] Kara, Y., Acar Boyacioglu, M., And Ömer Kaan Baykan, *Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange*, Expert Systems with Applications, 38(5):5311–5319, 2011
- [6] Rezaei, H., Faaljou, H., And Mansourfar, G., *Stock price prediction using deep learning and frequency decomposition*, Expert Systems with Applications, 169:114332, 2021.
- [7] Fischer, T. And Krauss, C., *Deep learning with long short-term memory networks for financial market predictions*, European Journal of Operational Research, 270(2):654–669, 2018.
- [8] Nguyen, T.-T. And Yoon, S., *A novel approach to short-term stock price movement prediction using transfer learning*, Applied Sciences, 9(22):4745, 2019.
- [9] Ariyo, A. A., Adewumi, A. O., And Ayo, C. K., *Stock price prediction using the ARIMA model*, 16th International Conference on Computer Modelling and Simulation. IEEE, 2014.
- [10] Ma, Q., *Comparison of ARIMA, ANN and LSTM for stock price prediction*, E3S Web of Conferences, 218:01026, 2020.
- [11] Adebisi, A. A., Adewumi, A. O., And Ayo, C. K., *Comparison of ARIMA and artificial neural networks models for stock price prediction*, Journal of Applied Mathematics, 2014:1–7, 2014.
- [12] Wang, J.-H. And Leu, J.-Y., *Stock Market Trend Prediction Using Arima-Based neural networks*, In Proceedings of International Conference on Neural Networks (ICNN'96), volume 4, pages 2160–2165 vol.4,1996.
- [13] Kumar, M. And Thenmozhi, M., *Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models*. 5(3):284,2014.
- [14] Rathnayaka, R. M. K. T., Seneviratna, D., Jianguo, W., And Arumawadu, H. I., *A hybrid statistical approach for stock market forecasting based on artificial neural network and ARIMA time series models*. IEEE, 2015.

- [15] Yang, Y., *Can the strengths of aic and bic be shared? a conflict between model identification and regression estimation*. Biometrika, 92(4):937–950,2005.
- [16] Banerjee, D., *Forecasting of indian stock market using time-series ARIMA model*. IEEE, 2014.
- [17] Almasarweh, M. And Wadi, S. A., *ARIMA model in predicting banking stock market data*,12(11):309.2018.
- [18] Cho, K., Merrienboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., And Bengio, Y., *Learning phrase representations using rnn encoder–decoder for statistical machine translation*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.
- [19] Luong, T., Pham, H., And Manning, C. D., *Effective approaches to attention- based neural machine translation*, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015.
- [20] Chen, S. And Ge, L., *Exploring the attention mechanism in lstm-based Hong Kong stock price movement prediction*, Quantitative Finance, 19(9):1507–1515, 2019.
- [21] Albayrak E. , *Stock Price Prediction Using Deep Learning Methods In High-Frequency Trading* ,MSc. Thesis, Çankaya University, 2021.
- [22] Hyndman, R. J., And Athanasopoulos G., *Forecasting: principles and practice*. Otexts, 2018.
- [23] Bollerslev, T., *Generalized autoregressive conditional heteroskedasticity.*, Journal of econometrics ,31.3, 307-327, 1986.
- [24] Bontempi, G., Souhaib B. T., And Yann-Aël L. B., *Machine learning strategies for time series forecasting.*, European business intelligence summer school, 62-77, Springer, Berlin, Heidelberg, 2012.
- [25] Alpaydin, E., *Introduction to machine learning*, MIT press, 2020.
- [26] Tay, F. E., and Lijuan C., *Application of support vector machines in financial time series forecasting.*,Omega 29.4, 309-317,2001.
- [27] Lo, A. W., Harry M., And Jiang W., *Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation*, The journal of finance 55.4: 1705-1765,2000.
- [28] Shintate, T., And Lukáš Pi., *Trend prediction classification for high frequency bitcoin time series with deep learning*, Journal of Risk and Financial Management 12.1: 17,2019.
- [29] Bao, W., Jun Y., And Yulei R., *A deep learning framework for financial time series using stacked autoencoders and long-short term memory.*, PloS one 12.7: e0180944,2017.
- [30] Raşo H., And Demirci M. *Predicting the turkish stock market bist 30 index using deep learning.*, International Journal of Engineering Research and Development 11.1: 253-265,2019.
- [31] Li Y., Zhu Z., Kong D., And Han H., & Zhao, Y., *EA-LSTM: Evolutionary attention-based LSTM for time series prediction*, Knowledge-Based Systems, 181, 104785, 2019.
- [32] Zhang, X., Liang X., Zhiyuli A., Zhang S., Xu R., And Wu, B., *AT-LSTM: An attention-based LSTM model for financial time series prediction*, IOP Conference Series: Materials Science and Engineering (Vol. 569, No. 5, p. 052037). IOP Publishing, 2019.
- [33] Bahdanau, D., Cho, K., And Bengio, Y., *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473, 2014.
- [34] Wilder, J.W., *New concepts in technical trading systems*, Trend Research, 1978.





## Bilimsel Yazılım Geliştirme Süreçleri için bir Yazılım Mühendisliği Yetkinlik Envanteri

### A Software Engineering Competency Inventory for Scientific Software Development Processes

Bilge SAY

Atılım Üniversitesi

Yazılım Mühendisliği Bölümü

Ankara, Türkiye

bilge.say@atilim.edu.tr

ORCID: 0000-0001-9276-729X

#### Öz

Araştırma amaçlı bilimsel yazılım geliştirme süreçleri, yazılım yaşam döngüsü açısından hem süreç hem girdi ve çıktılar bakımından iş veya endüstri amaçlı yazılımlardan bazı farklılıklar göstermektedir. Bilimsel yazılım geliştiricilerin, kod yazma deneyimi olsa da yapısal ve güncel yazılım mühendisliği yetkinliklerinin olmama olasılığı, bilimsel yazılımların gereksinimleri karşılaması ve sürdürülebilirliği açısından sorun yaratabilmektedir. Bu çalışmada literatürde bilimsel yazılım geliştirme alanında gereksinim duyulduğu belirtilen pratikler ile temelde IEEE Yazılım Mühendisliği Yetkinlik Modeli'nin (IEEE's Software Engineering Competency Model -SWECOM) yetkinlikleri eşleştirilmiş; ve bilimsel yazılım geliştirme açısından en gereksinim duyulan yazılım mühendisliği pratikleri belirlenmiştir. Yapılan sıklık analiziyle özellikle yazılım tasarımı ve yapımı sırasındaki detaylı tasarım ve planlama yetkinliklerine ihtiyaç duyulduğu belirlenmiştir. Üretilen envanter, Ar-Ge destekleri çerçevesinde eğitim programları geliştirmek ve iyileştirmek için kullanılabilir.

**Anahtar sözcükler:** Yazılım Mühendisliği Yetkinlikleri, Bilimsel Yazılım Geliştirme, SWECOM

#### Abstract

Scientific software development processes display some differences from business/industry-aimed software in terms of both processes and input/outputs. Although scientific software developers might have some coding experience, the fact that they may not possess formal and up-to-date software engineering competencies can affect negatively the

scientific software produced in terms of meeting the requirements and sustainability. A matching inventory has been created in the current work, pairing requirements for software engineering practices documented in scientific software development literature with competencies mostly from IEEE's Software Engineering Competency Model (SWECOM). Frequencies of the pairings have shown especially a need for software engineering competencies in software design and construction including detailed design and planning. The inventory may be used to generate and improve training programs within research and development support.

**Keywords:** Software Engineering Competencies, Scientific Software Development, SWECOM

#### 1. Bilimsel Yazılım Geliştirmede Yazılım Mühendisliği Yetkinlik Gereksinimlerine Giriş

Bilimsel yazılım geliştirme (araştırmalarda kullanılmak üzere yazılım geliştirme), endüstriyel yazılım geliştirme sürecinden bazı farklı özellikler gösterir. Örneğin, geliştirilen yazılım bir kuramın geçerliğinin sınanması veya modellenmesi hedefini taşıyorsa, araştırma sonuçları geldikçe değişim geçirecektir ve gereksinimlerin en baştan tam olarak bilinmesi zor olacaktır. Kuram, model ve o modelin temsilini sağlayan yazılım arasındaki sınırlar da her zaman net değildir [1–2]. Yazılımın geliştirilme süreci açısından bakılırsa da farklılıklar olabilir: lisansüstü öğrenciler ve araştırmacıların, kendi kullanımları için geliştirmeye başladıkları bir yazılım, ortak bir alanda etkileşim kuran küçük grubun dışında yaygın kullanıma tabi olduğunda kullanılabilirlik ve sürdürülebilirlik açısından sorunlar yaşanabilir. Bu ve benzeri yazılım mühendisliği pratikleri ve bilimsel yazılım geliştirme etkileşimleri konusunda, özellikle son yıllarda, karşılaşılan zorluklar ve kullanılan yöntemleri inceleyen artan sayıda çalışma yayınlanmıştır [3–6]. Ancak bu çalışmalarda ortaya çıkarılan

boşluk ve pratiklerin, yazılım mühendisliği güncel beceri ve yetkinlikleri envanterleri açısından standardizasyonuna olanak sağlayacak bir eşleştirme yapılmamıştır. Bu çalışmanın amacı, temelde SWECOM [7] kullanarak bir eşleştirme envanteri geliştirmektir. Eşleştirme, önce makale yazarının iki farklı listesinin madde madde üzerinden giderek bir temel eşleştirme yapması; daha sonra iki ayrı uzman görüşü alınarak geçerlik kontrolü ve gerekli değerlendirilen revizyonların yapılması şeklinde iki aşamalı olarak gerçekleştirilmiştir. Böyle bir eşleştirme, bilimsel yazılım geliştirme alanında çalışan ancak yazılım mühendisliği temeli olmayan araştırmacılara yönelik kısa veya uzun süreli eğitim müfredatlarının geliştirilmesi açısından anlamlı olacaktır.

Yazılım mühendisliği pratikleri ile bilimsel yazılım geliştirme alanında kullanılan yöntemler arasındaki “boşluk” yayınlarda temelde üç farklı açıdan yer almıştır denilebilir [3–6]. Birincisi, bu bölümde verilen örneklerde de bahsedildiği gibi, yazılım geliştirmenin doğasının, bir araştırma problemi için yazılım geliştirmekle, iş sektörü için yazılım geliştirme süreçlerindeki farkından kaynaklanmaktadır. İlk bölümde dile getirilenlere ek olarak, hızlı değişen ve küçük takımlarla yazılım geliştirme ve bu koşullarda proje yönetimi [8], bakımı yapılabilirlik ve taşınırılık gibi istenebilecek özelliklerin araştırma bütçesi kısıtları ve diğer gereksinimlerle çelişir olması [9] gibi bulgular bu kategoride sayılabilir.

Bir diğer “boşluk” sebebi, bilimsel yazılım geliştiricilerin ağırlıklı olarak başka disiplinlerden gelmeleri, programlama geçmişleri olmasına rağmen yazılım mühendisliği tekniklerine hakimiyetlerinin daha az olmasıdır. Bilgisayar Bilimleri ile ilgili eğitim görmüş olanlar, örneğin, sürüm kontrol ve bağımlılık belgeleme tekniklerini diğer bilimsel yazılım geliştiricilere göre daha sık kullanmaktadırlar [10]. Gereksinim yönetimi, güncel sınıma yöntemleri ve tasarım modelleme gibi konuların ise önemli bulunsalar dahi belgeleme ve sürüm yönetimine göre az bilinen pratikler olduğu belirlenmiştir [13]. Ayrıca belgeleme gibi görece sık uygulanan pratiklerin bile gereken yetkinlik düzeyinde uygulanmadığı ve dolayısıyla gerekli işlevi yerine getiremediği de örnek bilimsel yazılım incelemeleriyle belirlenmiştir [12].

Yukardaki bulguyla ilgili bir diğer nokta da bilimsel yazılım geliştiricilerin, sınav gibi bazı alanları önemli bulunsalar da yeterince bilgi sahibi olmamaları veya uygulamada eksik kalmalarıdır. Ayrıca, geliştiriciler, çevik yazılım geliştirme gibi araştırma süreçlerinin doğasında örtük olan süreçleri uyguladıklarının farkında olmayabilir. Dolayısıyla, bu konularda yazılım mühendisliği bilgi birikiminin olası katkılarından yoksun kalmaktadırlar [4,13]. Yazılım mühendisliği ile ilgili (kendi kendine öğrenme veya takım arkadaşlarından öğrenmeye ek olarak) kurumsal veya yapısal eğitimler almak ve büyük kullanıcı gruplarıyla veya büyük geliştirici takımlarıyla yazılım üretmek, yazılım mühendisliği yöntemleri farkındalığını uygulamaya geçirmede başarı etkenleri olarak değerlendirilmiştir [5,8].

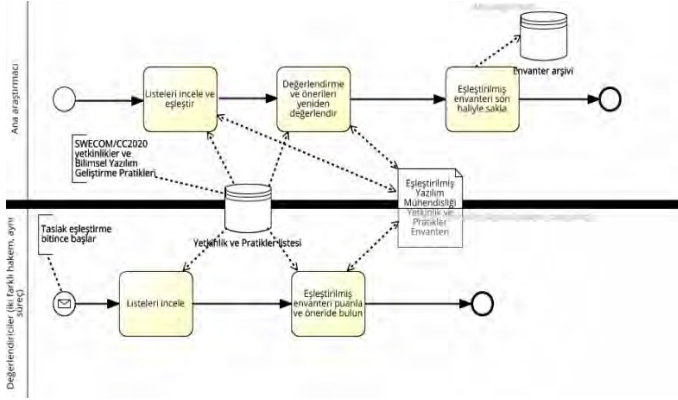
Bilimsel yazılım geliştirmede yazılım mühendisliği yöntemleri eğitimi ve topluluk desteğini yaygınlaştırmaya yönelik bazı girişimler başarıyla sürdürülmektedir. Örneğin, Yazılım Marangozluğu [14] ve Kod Rafinerisi [15] girişimleri,

gönüllülük ve açık kaynak esaslarına göre bilimsel yazılım geliştirmede süreç ve araç kullanımını iyileştirmeye yönelik eğitimler ve çalıştaylar düzenlemektedirler. Ancak yayınlarda temel alınan yazılım mühendisliği yetkinlik ve bilgi birikimi raporlarıyla, bilimsel yazılım geliştirmedeki gereksinimler, eksiklikler ve zorluklar üzerine yapılan çalışmaların bulguları üzerine bir eşleştirme yapılmamıştır; dolayısıyla mevcut girişimler de yazılım mühendisliğinin tüm yetkinlik birikimini kapsamamaktadırlar. Böyle bir çalışmanın gerçekleştirilmesi, bu makalenin ana motivasyonudur.

Makalenin kalan kısmı, aşağıda açıklandığı şekilde düzenlenmiştir: Bölüm 2’de bilimsel yazılım geliştirme gereksinimleri ve yazılım mühendisliği yetkinlik seçim yöntemi ve eşleştirme süreçleri aktarılmış; Bölüm 3’te bu eşleştirmeden doğan bulgu ve yorumlar, çalışmanın olası geçerlik tehditlerine karşı yapılanlar ve takip eden bir çalışmada yapılabilecekler beraber ele alınmıştır. Sonuç bölümünde ise literatürle karşılaştırılarak çalışmanın özgün değeri özetlenmiştir.

## 2. Yöntem

Bu çalışmada önce yazılım mühendisliği yetkinlik envanterleri araştırılmıştır. Daha sonra, güncel ve kapsayıcı bir liste çıkarılmaya çalışılmıştır. Çalışmanın bu kısmı Bölüm 2.1’de aktarılmaktadır. Sonrasında bilimsel yazılım geliştirmede yazılım mühendisliği pratikleri konusunda çalışmanın yapıldığı zaman itibarıyla güncel bir sistematik haritalama makalesi olan Arvanitou ve arkadaşlarının [3] çalışması esas alınıp, bilimsel yazılım geliştirmede yazılım mühendisliği pratikleri, iyi uygulamaları, eksik ve zorluklarına yönelik bir çekirdek liste çıkarılmıştır. Arvanitou ve arkadaşları çalışmasında taranan makaleler son olarak 2019 yılındadır. Daha yeni tarihli olup aynı çalışmanın diğer tarama kriterlerine uyan makalelerle liste genişletilerek bilimsel yazılım geliştirme yazılım mühendisliği yetkinlik veya pratik gereksinim listesi oluşturulmuştur. Gereksinim listesi oluşturulması ve yetkinlik envanteriyle, yetkinlik veya pratik gereksinim listesinin eşleştirme çalışması Bölüm 2.2’de verilmiştir. Yapılan eşleştirmenin Yazılım Mühendisliği alanında çalışan iki akademisyen kodlayıcı tarafından değerlendirilmesinden ortaya çıkarılan geçerlik çalışması ise Bölüm 2.3’te aktarılmaktadır. Bu iki sürecin nasıl ilerlediği görsel olarak da bir iş süreçleri şemasıyla Şekil-1’de gösterilmiştir. Tüm çalışmanın ham verilerine (yazılım mühendisliği yetkinlikler envanteri; bilimsel yazılım geliştirme gereksinimleri listesi, eşleştirme tabloları ve hakem değerlendirmelerinin tümüne) çevrimiçi olarak ulaşılabilir [16].



Şekil-1: Yöntem akış şeması

## 2.1 Yazılım Mühendisliği Yetkinlikleri

Yazılım mühendisliği organizasyonlarının (özellikle, Association for Computing Machinery (ACM) ve Institute of Electrical and Electronics Engineers (IEEE) Computer Society (IEEE-CS)) üniversite ve endüstri temsilcileriyle işbirliği içinde oluşturdukları görev güçlerince üretilen birkaç yazılım mühendisliği bilgi dağarcığı ve yetkinlik raporu vardır. Bilişim Alanları Müfredatı (The Computing Curriculum 2020- CC2020) bilişim alanlarında lisans düzeyinde müfredatlar için geliştirilmiş yetkinlik bazlı bir rehberdir ve yazılım mühendisliği alanında da taslak halinde bir alt rehber içermektedir [17]. CC2020 rehberinde yetkinlik cümleleri, bilgi bileşenleri ile beceri düzeylerini eşleştirmekte; teknik becerilere inisiyatif almak gibi sosyol duygusal beceriler de eşlik etmektedir. SWECOM ise yazılım mühendisliği tabanlı iş tanımlarının farklı düzeyleri için beceri setleri ve ilgili aktiviteleri formüle eden bir yetkinlik modelidir [7]. Yazılım Mühendisliği Bilgi Dağarcığı Rehberi (Software Engineering Body of Knowledge - SWEBOK) ise yazılım mühendisliği bilgi birikimini hiyerarşik bir taksonomi olarak sınıflandırmaktadır [18].

CC2020 rehberi en yakın tarihli rehber olduğu ve çalışmanın yapıldığı tarihte SWEBOK ve SWECOM'un yeni güncellemeleri üzerinde çalışılmakta olmasına rağmen kamuya açılmış güncellemeleri olmamasından dolayı yazılım mühendisliği yetkinlik envanteri çalışması için ilk aday olarak incelenmiş; ancak genelleştirilmiş yetkinlik değil, örnek olay tabanlı olması ve konfigürasyon yönetimi gibi birçok alanda yazılım mühendisliği taslak raporunun boşluklar içermesinden dolayı ana kaynak olarak tercih edilmemiştir. SWEBOK hiyerarşisi ise yetkinlikten çok statik bilgi tabanlı olduğu için bilimsel yazılım geliştirme yayınlarında belirlenen pratiklerle eşleştirilmesinin daha zor olacağı değerlendirilmiştir. Dolayısıyla yazılım mühendisliği yetkinlik envanteri kapsamında tüm SWECOM yetkinlikleri ve buna ek olarak SWECOM'da karşılığı olmadığı için CC2020'den proje yönetimi ve bireysel ve takım çalışması sosyal yetkinlikleriyle ilgili davranışsal öznitelik başlıkları altında olan yetkinlikler kullanılmıştır. Envanter, üç düzeyli kırılımla ana yetkinlik-yetkinlik-alt yetkinlik şeklinde bir organizasyonla toplam 254 alt düzey yetkinlikten oluşmaktadır ve çalışmanın veri tablolarında İngilizce asıl ifadeleriyle mevcuttur [16].

## 2.2 Bilimsel Yazılım Geliştirme Gereksinimlerinin Oluşturulması ve Eşleştirilmesi

Bilimsel Yazılım Geliştirme yetkinlikleri gereksinim listesinin oluşturulması için iki aşamalı bir yöntem kullanılmıştır. İlk olarak Arvanitou ve arkadaşlarının [3] bilimsel yazılım geliştirme süreçlerinde yer aldığı veya eksikliğini belirlediği, sistematik taramayla belirlenen araştırmalardan çıkarılan yazılım geliştirme pratikleri ve yetkinlikleri kataloğu, gereksinim listesinin temeli olarak alınmıştır. Arvanitou ve arkadaşları çalışmalarında sistematik bir şekilde IEEE Explore, ACM, Science Direct ve Springer sayısal kütüphanelerinde yer alan ve 2019 yılının sonuna kadar olan sürede yayınlanmış olan 359 ana çalışmayı kullanmışlardır. Arvanitou ve arkadaşları bu ana çalışmalarda toplamda en az 9 defa (en fazlası 41 defa) yer alan pratikleri yeterince sık kabul edip tablolaştırarak bilimsel yazılım geliştirmeyle en ilgili 20 yazılım mühendisliği pratiğinin listesini oluşturmuşlardır (bkz [3]'te, bu çalışmanın veri tablolarında da tekrarlanan Çizelge-10). Arvanitou ve arkadaşlarının listesinde olmayan daha yeni veya farklı dijital dizinlerde olup diğer ölçütleri sağlayan bazı ana makalelerde [10–11,19] belirtilen gereksinimler de eğer Arvanitou ve ark. listesinde bir maddeyle birebir eşleşmiyor veya daha üst bir kategori tarafından içerilmiyorsa, listeye eklenmiştir; araştırmacı-geliştiricinin "yeterince bireysel zamanı olmaması" gibi genel proje yönetimiyle ilgili olabilecek ancak yazılım mühendisliği veya yazılım proje yönetimiyle ilişkilendirilmesi doğrudan mümkün olmayan genel bulgular kapsam dışı tutulmuştur. Böylece Çizelge-1'de alıntı yapılan araştırmalar bazında alfabetik sırayla görülebileceği gibi toplam 32 maddeden oluşan bilimsel yazılım geliştirirken özellikle yazılım mühendisliği kapsamında ihtiyaç duyulan bir yazılım geliştirme pratikleri gereksinim listesi oluşturulmuştur. Listenin orijinal İngilizce dökümlerine çalışmanın veri tablolarından ulaşılabilir [16].

Yazılım mühendisliği yetkinlikleri envanteriyle, bilimsel yazılım geliştirme süreçleri yetkinlik gereksinim listesi eşleştirmesi bu paragrafta aktarılabilecek olan yöntemle yapılmıştır. Her bilimsel yazılım geliştirme yetkinlik gereksinimi (yani 32 maddeden oluşan listedeki her bir madde) için, SWECOM-CC2020 kaynaklı yazılım mühendisliği yetkinlikleri envanterinin tüm maddeleri taranmış; en ilgili en fazla üç madde olmak üzere (daha az maddeyle eşleşiyorsa en az bir, en fazla üç olmak üzere eşleştiği kadar) kapsamı en çok örtüşen maddeler Çizelge-2'deki Türkçeleştirilmiş örneklerde görüldüğü üzere eşleştirme tablosuna eklenmiştir. Yazılım mühendisliği yetkinlik envanteri (SWECOM) üç düzeyli hiyerarşik bir organizasyona sahiptir; dolayısıyla eşleştirmelerin detay düzeyinin yüksek olması için örtüşme düzeyinin en çok olduğu mümkün olan en alt, yani en detaylı kırılım kullanılmıştır. İstisna olarak, eğer ikiden fazla en alt düzey kırılım aynı anda bir bilimsel yazılım gereksinimi yetkinliği ile eşleşiyorsa, bir üst düzey başlık üç eşleştirme hanesinden birine yazılmıştır. Bu durumda daha üst düzey bir yetkinliğin tümüne gereksinim duyulduğu varsayılmıştır. Çizelge-2'de örnekleri olan eşleştirmelerin tümünün orijinal İngilizce dökümüne yine çalışmanın veri tablolarından ulaşılabilir [16].

**Çizelge-1: Bilimsel Yazılım Geliştirme Süreçlerinde Yazılım Mühendisliği Pratikleri Gereksinimleri**

Pratik Adı	Kaynak	Pratik Adı	Kaynak	
Alan Yönelik Diller	[3]	Tasarım ve Mimari Modelleme	[3]	
Bilimsel Test Yöntemleri		Yazılım Mimariği		
Bileşen-Tabanlı Yazılım Geliştirme		Yazılım sınavı (bağlanımlı, otomatik, kahinli/kahinsiz)		
Birleştirilmiş Geliştirme Ortamları (IDE)		Yazılım Süreç İyileştirme ve Döngü Yönetimi		
Çevik Yöntemler		Bağımlılıklar Hakkında Bildirim		[10]
Geliştirme Çerçevesi Önerme		Sürüm Kontrol Sistemleri Kullanımı		[10]
Gereksinim Belirtimi		Veri Yönetimi		[11]
İnsan Yönetimi ve İletişim		Yazılım Gereksinimleri Yönetimi		
Kalite Optimizasyonu		Yazılım Performansı Optimizasyonu		[11]
Kod Kütüphaneleri ve Arayüzleri (API) Kullanımı		Yazılım Sürüm Oluşturma Mühendisliği		[19]
Kod Üretme		Destek Araçları Kullanımı		
Ölçümlü veya Nitel Kalite Yönetimi		Eleman Bulmak ve Geliştirici Devrini Yönetme		
Paralel veya Dağıtık Programlama		En İyi yöntemler ve Hızlı İş Yetiştirme Arasındaki Denge		
Programlama Dili Edinimi ve Kullanımı		Gereksinim Toplama ve Dokümantasyonu Araçları Kullanımı		
Programlama Teknikleri		Modern Yazılım Mühendisliği Araçları Kullanabilme		
Proje Yönetimi		Yazılım Mimarisini ve Tasarım Araç Kullanımı		

**Çizelge-2: Bilimsel Yazılım Geliştirme Gereksinimleri ile SWECOM/CC2020 Yetkinlikleri Eşleştirme Örnekleri**

Gereksinimler	Eşleştirilen Yetkinlikler
Kod Kütüphaneleri ve Arayüzleri (API) Kullanımı	Yazılım Yapımı Planlama (SCP)*
Eleman Bulmak ve Geliştirici Devrini Yönetme	Proje Yönetimi (PM) Davranışsal Yetkinlikler (BA)
Veri Yönetimi	Yazılım Tasarımı Temel Yetkinlikleri (SDF)
	Yazılım Konfigurasyon Yönetimi Planlaması (PSCM)
	Yazılım Kalitesi İstatistiksel Kontrol (SQSC)

\*Parantez içindeki kodlar orijinal SWECOM/CC2020 yetkinliklerini veri setinde kodlamak için kullanılmıştır.

## 2.3 Eşleştirme Geçerlik Kontrolü

Çalışmanın yazılım mühendisliği uzmanlığı olan tek bir araştırmacı tarafından yapılmış olması geçerlik çalışması ihtiyacını doğurmaktadır. Geçerlik kontrolü için uzmanlığı yazılım mühendisliği olan iki değerlendirici veya kodlayıcı, yapılan eşleştirme çalışmasını, 32 maddeden her biri için 1 ile 3 üzerinden puanlandırmışlardır. Değerlendirici veya kodlayıcılar, puanlamaları birbirlerinden bağımsız ve birbirleriyle iletişim kurmadan yapmış, aradaki farklılıklar daha sonra araştırmacı tarafından ele alınmıştır. Yapılan eşleştirmenin uygunluğuna dair (bir pratik birden fazla yetkinlikle eşleştirilmişse hepsini toplu değerlendirerek) aşağıdaki değerlere göre bir kodlama yapmışlardır: 1-Uygun değil; 2-Nötr (Emin olamama hali dahil) 3-Uygun. Üçlü basit bir skala seçimi ile değerlendiricilerin tereddütte kalmadan araştırmacının yaptığı eşleştirmeleri yorumlayabilmeleri hedeflenmiştir. Değerlendirici veya kodlayıcının, “Uygun” kararını, araştırmacı tarafından yapılan eşleştirmenin, değerlendiricinin kişisel yazılım mühendisliği bilgisi ve SWECOM/CC2020 tabanlı yazılım mühendisliği yetkinlik envanterinin ilgili maddeleriyle uyumlu olduğu değerlendirilmesi üzerine vermesi istenmiş; her değerlendirici veya kodlayıcıya birebir bilgi ve örnek aktarımı yapılmıştır. Ayrıca kodlayıcılar, uygun bulmadıkları ve emin olmadıkları eşleştirmeler için isteğe bağlı bir yorum sütununa, yorum ve önerilerini girmişlerdir. Yazılım mühendisliği yetkinlik envanteri listesini kolay taramaları ve ayrıntılı referanslara bakabilmeleri için gerekli yönergeler kendilerine verilmiştir [16]. Bu yorumların değerlendirmesi Bölüm 3.1’de yapılmıştır ve sonucunda mevcut çalışmanın yorumlara göre revize hali de ayrı bir tabloyla oluşturulmuştur. İki kodlayıcının her madde için yaptığı değerlendirmeler ham haliyle [16] da verilen veri tablosunun orijinal pratik yetkinlik eşleştirme yapığında görülebilir.

İki uzman kodlayıcının, yanıt özet değerleri ve kodlayıcılar arasındaki uyum Çizelge-3’de görülebilir. İki kodlayıcı arasında uyum %69 olarak gerçekleşmiş ve orta düzey olarak kabul edilen şansa göre düzenlenmiş kodlayıcı uyum katsayısı 0,22 (Cohen Kappa değeri) ve 0,61 (Gwet AC1 değeri) olarak hesaplanmıştır [20–21]. İki kodlayıcının da uygun bulduğu (yani 3 ile kodladığı) eşleştirme sayısı 32 üzerinden 20; ve birinin uygun (3 ile kodladığı), diğerinin nötr değerlendirdiği (2 ile kodladığı) eşleştirme sayısı 8; ikisinin de nötr değerlendirdiği (2 ile kodladığı) eşleştirme sayısı 2; ve bir kodlayıcının nötr (2 ile kodladığı), diğer kodlayıcının uygun değil şeklinde değerlendirdiği (1 ile kodladığı) eşleştirme sayısı 2’dir. Bu veriler çalışmanın geçerliği açısından yeterli olarak değerlendirilmiştir.

**Çizelge-3: Ek kodlayıcı Değerlendirmeleri**

Kodlayıcı_1		Kodlayıcı_2		Ortak Kodlar	
Değer*	Sıklık	Değer	Sıklık	Değer	Sıklık
3	26	3	22	İkisi de 3	20
2	4	2	10	İkisi de 2	2
1	2	1	0	Biri 3, biri 2	8
	32		32	Biri 2, biri 1	2

\* 1-Uygun değil; 2-Nötr (Emin olamama hali dahil) 3-Uygun.

### 3. Bulgu ve Yorumlar

Bu bölümde eşleştirme çalışması sonucu bilimsel yazılım geliştirme gereksinimleri hakkında elde edilen ana bulgular, geçerlik kontrolü ile yapılan genişletmeler ve bulguların doğruluğu sonuçlar ele alınarak, geçerlik tehditleri açısından mevcut çalışmanın risklerinin nasıl azaltıldığı ve bu tehditlere bağlayarak gelecekte daha detaylı nasıl bir çalışma yapılabileceği aktarılmaktadır.

#### 3.1 Bulgular

Bölüm 2.2’de yöntemsel olarak anlatıldığı ve Çizelge-2’de örneklendiği üzere yapılan eşleştirme çalışmasında 32 maddeden oluşan bilimsel yazılım geliştirme yetkinlik gereksinim listesine, SWECOM’un tümü ve CC2020 den yapılan eklerle oluşturulan 254 maddelik yetkinlik envanterinden toplam 56 eşleştirme yapılmıştır. Bu eşleştirmeler 32 bilimsel yazılım geliştirme gereksiniminden 5’i için maksimum eşleştirilebilecek yetkinlik sayısı olan 3’er madde; 14 gereksinim için 2’şer madde ve 13 gereksinim için birer madde olarak gerçekleşmiştir. Üç ve üstü sıklıkla eşleştirilen yazılım mühendisliği yetkinlik kategorilerinin yer aldığı Çizelge-4’te görüleceği üzere en çok eşleştirme gerçekleştirilen (toplam 13 eşleştirme) yetkinlikler, Yazılım Yapımı alanındadır. Özellikle Yazılım Yapımı Planlama (6 eşleştirme), Ayrıntılı Tasarım ve Kodlama (3 eşleştirme) en çok ihtiyaç duyulan alt yetkinlikler olarak ortaya çıkmakta ve alt kırılımlara girmeyip en üst kırılımında Yazılım Yapımı (4 eşleştirme) altında yapılan ek eşleştirmeler ile toplamda bu alanda 13 eşleştirme yer almaktadır. İkinci sırada ise toplamda 10 eşleştirme ile Yazılım Tasarımı üst yetkinliğinin ikinci düzey kırılımları olan Yazılım Tasarımı Temelleri (3 eşleştirme), Yazılım Mimarisi Tasarımı (3 eşleştirme), Yazılım Tasarımı Stratejileri ve Yöntemleri (1 eşleştirme) ve Yazılım Tasarımı Kalite Analizi ve Değerlendirmesi (1 eşleştirme), ve alt kırılımlara bölünemeyen iki ek Yazılım Tasarımı üst yetkinliğiyle beraber yer almaktadır. Bu yetkinlikleri, Yazılım Konfigurasyon Yönetimi (5 eşleştirme) ve onu takiben Proje Yönetimi (4 eşleştirme) ve Yazılım Kalite Aktiviteleri (4 eşleştirme) izlemektedir. Üçer eşleştirme sıklığı ile gereksinim duyulan diğer yetkinlikler olarak ise Yazılım Gereksinim Yetkinlikleri, Yazılım Süreç ve Yaşam Döngüsü Aktiviteleri ve Yazılım Sınama Aktiviteleri yer almaktadır. Kalan 12 eşleştirme diğer yetkinlik başlıkları altında dağılmıştır. Eşleştirmelerin tümünün sıklık-sıralı orijinal İngilizce dökümüne yine çalışmanın veri tablolarından ulaşılabilir.

Bu sıklık analizinden çıkarılabilecek bir yorum, literatürde bilimsel yazılım geliştirme alanında yapılan çalışmalarda en çok ihtiyaç duyulan yetkinliklerin yazılım tasarımı temel teknikleri (soyutlama, koşut zamanlılık, yeniden yapılandırma vb); yazılım mimarisi seçimi ve uygulaması ve tasarım kalitesinin ve stratejisinin yönetilmesi üzerinde olduğudur. Tasarım aşamasıyla beraber yazılım yaşam döngüsünde sadece klasik anlamda kodlama değil; probleme uygun çerçeve, platform, programlama dilleri, kütüphaneleri ve geliştirme ortamlarının seçimi, tasarım örüntüleri kullanımı, güvenli kodlama, kod standartlarının geliştirilmesi, kod generasyonu gibi öğeleri de içeren detaylı tasarım tekniklerinin yetkinliklerinin de öne çıktığı görülmektedir. Bu

iki başlık, konfigürasyon ve proje yönetimi teknikleriyle beraber bilimsel yazılım geliştiricilerin yazılım mühendisliği bilgi ve becerileri açısından en ihtiyaç duydukları yetkinlikler olarak ortaya çıkmıştır. Sınama, konfigürasyon yönetimi becerilerine olan gereksinimler daha çok dile getirilmiş olsa bile tasarım ve yapım aşamasındaki modern yazılım mühendisliği tekniklerine duyulan gereksinim, mevcut literatürde bu şekilde bir envanter eşleştirilmesi yapılmış olmadığı için yeterince belirgin değildir [4,11]; bu sıklık çalışmasının getirdiği öncelik mevcut çalışmanın özgün bir katkısıdır.

**Çizelge-4: Bilimsel Yazılım Geliştirme Yetkinlikler Envanteri Sıklık Çizelgesi**

Yetkinlikler	Eşleştirme Sıklığı
Yazılım Yapımı (SC)*	13
Yazılım Yapımı Planlama (SCP)	6
Detaylı Tasarım ve Kodlama	3
Yazılım Yapımı -Ayrıştırılmamış (SC)	4
Yazılım Tasarımı (SD)	10
Yazılım Tasarımı Temelleri (SDF)	3
Yazılım Mimarisi Tasarımı (SAD)	3
Yazılım Tasarımı Stratejileri ve Yöntemleri (SDSM)	1
Yazılım Tasarımı Kalite Analizi ve Değerlendirmesi (SDQAE)	1
Yazılım Tasarımı-Ayrıştırılmamış (SD)	2
Yazılım Konfigurasyon Yönetimi (SCM)	5
Proje Yönetimi (PM)	4
Yazılım Kalite Aktiviteleri (SQ)	4
Yazılım Gereksinim Yetkinlikleri (SR)	3
Yazılım Süreç ve Yaşam Döngüsü Aktiviteleri (SPLC)	3
Yazılım Sınama Aktiviteleri (ST)	3
*Sadece üst kırılım bazında üçten fazla sıklıkla tekrarlanan yetkinlikler verilmiştir. Tüm liste veri setinde mevcuttur. Parantez içindeki kodlar orijinal SWECOM/CC2020 yetkinliklerini veri setinde kodlamak için kullanılmıştır.	

Geçerlik kontrolü çalışması sonucu kodlayıcı uzmanlardan iki türlü öneri gelmiştir: Mevcut 5 eşleştirmenin envanterde kalıp kalmaması konusunda emin olmadıklarını belirten “çıkarma” önerileri ve ek olarak mevcut envanterde olmayan 2’si en alt kırılım düzeyinde alt yetkinlik, 4’ü bir üst kırılım düzeyinde yetkinlik olmak üzere 6 “ekleme” önerisi. İki listenin de arka planını oluşturan referanslardan yapılan kontroller sonucu, çıkarılması açısından tereddüt oluşturan öğelerin, eşleştirme açısından doğru olduğu teyit edilmiş ve çıkarma yapılmamıştır [3,7]. Mevcut envanterde olmayan ekleme önerileri ise aynı şekilde kontrolden sonra uygun bulunarak eklenmiştir ve çalışmanın revize veri tablolarından ulaşılabilir. Yapılan eklemeler, bu bölümde yer alan sıklık yorumlarında bir farklılık yaratmamıştır.

#### 3.2 Geçerlik Tehditleri ve Yapılabilecek Çalışmalar

Yapılan çalışma, büyük ölçüde, Arvenitou ve arkadaşlarının [3] sistematik tarama çalışmasındaki bulgulara dayanmaktadır. Arvanitou ve arkadaşları çalışmalarını aslında yazılım kalitesi bileşenleri ile yetkinlik ve süreç gereksinimlerini eşleştirme amaçlı gerçekleştirmiştir. Yapılan taramanın sistematigi sağlam olmakla beraber geçerlik endişeleri açısından birinci

elden yani bilimsel yazılım geliştiriciler topluluğundan doğrudan veri elde etmenin, daha somut ve sağlam bir envanter çıkarabilmek açısından yararı olacaktır. SWECOM, SWEBOOK ve CC2020 gibi yetkinlik/bilgi envanterlerinin üzerinde çalışılan yeni sürümleri kamuya paylaşıldıktan sonra, bu envanterler üzerinden detaylı bir anket çalışmasıyla bilimsel amaçlı yazılım geliştiren araştırmacılara uluslararası düzeyde ulaşıp daha detaylı ve birinci elden veri toplayarak geçerliliği yüksek bir tekrar çalışması yürütülebilir.

Benzer şekilde, SWECOM [7] yetkinlik envanterinde, yetkinlikler, teknisyenden, deneyimli yazılım mühendisine kadar değişken bir yetkinlik yelpazesini gösteren beşli bir ölçekte iş tanımlarının farklı aktivitelere göre eşleştirmesiyle (“Takip eder”; “Yardım eder”; “Katılır”; “Liderlik eder” gibi) detaylandırılmışlardır. Mevcut çalışmada bu detaya inebilmek mümkün olmamıştır; çünkü bilimsel yazılım geliştirme süreçlerindeki gereksinimlerin farklı iş tanımları ve aktivitelere göre detaylandırıldığı bir çalışma henüz literatürde görülmemektedir. Yukarıdaki paragrafta önerilen çalışma, bu tip bir çeşitlemeye de zemin sağlayabilir.

Eşleştirme çalışmasının tek uzmanla yürütülmesinin sakıncaları, Bölüm 2.3’te aktarıldığı gibi, iki farklı uzmanca yapılan kodlamalarla sağlama ve yapılan önerileri kontrol edip, envantere katarak ulaşılan genişletme çalışmasıyla hafifletilmiştir. Birinci elden elde edilen veriyle yapılacak bir çalışmada, eş güdümlü çalışan bir grup uzmanın eş zamanlı, istatistiki kontrolle eşleştirme yapması çalışmanın iç geçerliliğine katkı sağlayacaktır.

#### 4. Sonuç

Bu çalışmada, bilimsel yazılım geliştirme sürecindeki yetkinlik gereksinimleri, yazılım mühendisliği yetkinlik envanterleriyle (SWECOM-CC2020) eşleştirilerek, hem ihtiyaç duyulan yetkinliklerin standart terminolojiyle kullanılması, hem de farklı bilim dallarından bilimsel yazılım geliştiricilerin hangi yazılım mühendisliği pratiklerini öğrenmeye ve uygulamaya en çok ihtiyaçları olduğunun alan rehberleri yetkinliklerine odaklanarak belirlenmesi hedeflenmiştir. Yayın özetinde belirtildiği gibi çeşitli çalışmalarda anket gibi yöntemlerle elde edilmiş, bilimsel yazılımların geliştirilmesinde gereksinim duyulan yetkinlik listeleri mevcuttur [3–5,19]. Bu çalışmanın, alan yayınlarında bahsedilen çalışmalara ek olarak getirdiği katkılar:

1. Kapsamlı bir yetkinlik gereksinim listesi oluşturulması,
2. IEEE ve ACM’in yazılım mühendisliği ölçünlü yetkinlik envanterleriyle bir eşleştirme sağlayarak, terminolojinin birleştirilmesi,
3. Yetkinlik envanterlerinin tümü ele alındığında hangi alanların kapsandığının veya kapsamadığının daha iyi değerlendirilebilmesi olmuştur.

Bu çalışmanın ışığında, özellikle tasarım ve yapım aşamasında bilimsel yazılım geliştirme için gerekli yetkinliklerin mevcut bilimsel yazılım geliştirme eğitimleri girişimlerinde yeterli düzeyde kapsamadığı gözlenmiştir [14–15]. Ortaya çıkan eşleştirme listesi, Yazılım Marangozluğu [14] ve Kod Rafinerisi [15] gibi mevcut uluslararası girişimlerin müfredatlarının

yenilenmesine temel oluşturabileceği gibi; ulusal düzeyde de Ar-Ge politikaları geliştirirken, üniversite araştırma merkezleri ve yazılım mühendisliği dışındaki bilim alanlarından lisansüstü eğitim öğrencilerinin ve diğer araştırmacıların ulaşabileceği yazılım mühendisliği teknikleri eğitim programları tasarlanmasında da kullanılabilir.

#### Teşekkür

Atilım Üniversitesi Yazılım Mühendisliği Bölümü öğretim üyeleri Prof. Dr. Ali Yazıcı, Doç. Dr. Çiğdem Turhan ve Bilişim Sistemleri Mühendisliği Bölümü öğretim üyesi Dr. Öğr. Üyesi Tuna Hacaloğlu’na çalışmaya yaptıkları katkı için teşekkür ederim.

#### Kaynakça

- [1] Johanson, A., & Hasselbring, W. (2018). Software Engineering for Computational Science: Past, Present, Future. *Computing in Science and Engineering*. doi:10.1109/MCSE.2018.108162940
- [2] Taatgen, N. A., Vugt, M. K. van, Borst, J. P., & Mehlhorn, K. (2016). Cognitive modeling at ICCM: state of the art and future directions. *Topics in Cognitive Science*. 8(1), 259-263. doi:10.1111/tops.12185.
- [3] Arvanitou, E. M., Ampatzoglou, A., Chatzigeorgiou, A., & Carver, J. C. (2021). Software engineering practices for scientific software development: A systematic mapping study. *Journal of Systems and Software*, 172. doi:10.1016/j.jss.2020.110848
- [4] Heaton D., & Carver, J. (2015). Claims about the use of software engineering practices in science: A systematic literature review. *Information and Software Technology*, 67, 207-219. doi:10.1016/j.infsof.2015.07.011.
- [5] Storer, T. (2017). Bridging the chasm: a survey of software engineering practice in scientific programming. *ACM Computing Surveys*, 50(4), 1-32. doi:10.1145/3084225.
- [6] Segal, J., & Morris, C. (2008). Developing scientific software. *IEEE software*, 25(4), 18-20. doi:10.1109/MS.2008.85
- [7] IEEE. (2014). *A Software Engineering Competency Model (SWECOM)*. Version 1.0. IEEE Computer Society Press.
- [8] Kurtaran, F. (2018). “An Evaluation of the Use of Software Engineering Practices by Cognitive Modeling Researchers.” MSc Thesis. Department of Computer Engineering. Atilim University, Turkey. <https://tez.yok.gov.tr/UlusalTezMerkezi/>
- [9] Wagner S., Pflüger, D., & Mehl, M. (2015). Simulation software engineering: experiences and challenges. *Proceedings of the 3rd International Workshop on Software Engineering for High Performance Computing in Computational Science and Engineering - SE-HPCCSE '15*, 1-4. doi:10.1145/2830168.2830171.
- [10] AlNoamany Y, Borghi JA. (2018). Towards computational reproducibility: researcher perspectives on the use and sharing of software. *PeerJ Computer Science* 4:e163. doi:10.7717/peerj-cs.163
- [11] Wiese, I., Polato I. & Pinto, G. (2020). Naming the Pain in Developing Scientific Software. *IEEE Software*, 37(4), 75-82, July-Aug. 2020, doi:10.1109/ms.2019.2899838.
- [12] Hermann, S., Fehr, J. (2022). Documenting research software in engineering science. *Scientific Reports*, 12 (6567). doi:10.1038/s41598-022-10376-9
- [13] Sanders, R. & Kelly, D., 2008. Dealing with Risk in Scientific Software Development. *IEEE Software*, 25(4), pp.21–28. doi:10.1109/ms.2008.84.

- [14] Software Carpentry. <https://software-carpentry.org/>. Erişim Tarihi: Haziran, 2022.
- [15] Code Refinery. <https://coderefinery.org/> . Erişim Tarihi: Haziran, 2022.
- [16] Say, B. Bilimsel Yazılım Geliştirme Yetkinlik Envanteri Çalışması Tabloları [https://drive.google.com/drive/folders/10SP32GLOXTNXKBxstVBhme1XkfWJXwwm?usp=sharing\\_](https://drive.google.com/drive/folders/10SP32GLOXTNXKBxstVBhme1XkfWJXwwm?usp=sharing_). Erişim Tarihi: Kasım, 2022.
- [17] CC2020 Task Force. (2020). *Computing Curricula 2020: Paradigms for Global Computing Education. Computing Curricula 2020*. ACM. doi:10.1145/3467967
- [18] Bourque P. & Fairley, R. E. (Ed.) (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK), Version 3.0*. Retrieved from <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
- [19] Carver JC, Weber N, Ram K, Gesing S, Katz DS. (2022). A survey of the state of the practice for research software in the United States. *PeerJ Computer Science*, doi:10.7717/peerj-cs.963
- [20] The jamovi project (2022). jamovi. (Version 2.3) [Yazılım]. <https://www.jamovi.org>.
- [21] Balcı, S. (2022). ClinicoPath jamovi Module doi:10.5281/zenodo.3997188. [R package]. <https://github.com/sbalci/ClinicoPathJamoviModule>.