

## Öznitelik Belirleme Yöntemleriyle Farklı Türde Hata Ayıklamada Önemli Göz Hareket Metriklerinin Belirlenmesi\* \*\*

### Determining Important Eye Movement Metrics in Different Types of Debugging with Feature Selection Methods

Arif AKÇAY<sup>1</sup>, Arif ALTUN<sup>2</sup>

<sup>1</sup>Kastamonu Üniversitesi, Eğitim Bilimleri Bölümü, Öğretim Teknolojileri Anabilim Dalı. e-posta: aakcay@kastamonu.edu.tr

<sup>2</sup>Hacettepe Üniversitesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı. e-posta: altunar@hacettepe.edu.tr

**Makale Türü/Article Types:** Araştırma Makalesi/ Research Article  
**Makalenin Geliş Tarihi:** 03.11.2022 **Yayına Kabul Tarihi:** 02.08.2023

#### ÖZ

Bir programlama süreci içerisinde karşılaşılan olumsuz durumlara hata; hataların fark edilmesi, aranması, bulunması, çözümlenmesi ve tekrar oluşmaması için önlem alınması işlemlerine hata ayıklama denir. Bir programlama sürecinde farklı türde hatalar (derleme zamanı hataları, çalışma zamanı hataları ve mantık hataları) ile karşılaşılmaktadır. Her hata farklı özelliklere sahip olmakla birlikte bireylerin hata giderirken sergiledikleri davranışlar araştırmalarda ilgi çekicidir. Hata ayıklarken bireylerin bilişsel performanslarına ilişkin çıkarım yapmak için göz izleme yöntemi kullanılır. Araştırmalarda kullanılan göz hareket metrikleri birbirlerinden farklıdır ve neden kullanıldığına yönelik belirsizlik vardır. Bu çalışmanın amacı farklı türde hata ayıklamada önemli göz hareketlerinin belirlenmesidir. Araştırmaya mesleki ve teknik liselerin Bilişim Teknolojileri Bölümlerinde okuyan 51 öğrenci katılmıştır. Öğrencilerin hata ayıklama performanslarını belirlemek için Hata Ayıklama Performansı Testi kullanılmıştır. Göz hareketlerini belirlemek için Gazepoint GP3 göz izleme aracı kullanılmıştır. Araştırmanın

---

\***Alıntılama:** Akçay, A. ve Altun, A. (2023). Öznitelik belirleme yöntemleriyle farklı türde hata ayıklamada önemli göz hareket metriklerinin belirlenmesi. *Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, 43(2), 757-784.

\*\*Bu makale ikinci yazar danışmanlığında birinci yazarın doktora tez çalışmasının bir kısmından oluşmaktadır.

analizlerinde ilgi alanları belirlenmiş ve göz hareketleri, makine öğrenmesinde kullanılan öznitelik belirleme yöntemlerinden olan Bilgi Kazancı ve Gini Katsayısı ile incelenmiştir. Araştırma sonucunda öğrencilerin farklı türde hata ayıklarken sergiledikleri önemli göz hareketlerinin hata türlerinin özelliklerini yansıttığı belirlenmiştir. Son olarak, bu sonuca göre önerilerde bulunulmuştur.

**Anahtar Sözcükler:** Hata, hata ayıklama, hata türleri, göz izleme, göz hareketleri

### ABSTRACT

The negative situations encountered in a programming process are called errors; the processes of recognizing, searching, finding, solving and taking measures to prevent recurrence of errors are called debugging. While programming, different types of errors (compile-time bugs, run-time bugs, and logical errors) are encountered. Each type of bug has different characteristics and the behavior of individuals while debugging is interesting in research. The eye tracking method is used to make inferences about the cognitive performance of individuals while debugging. Eye tracking metrics used in research are different and there is uncertainty as to why they are used. The aim of this study is to determine the important eye movements in different types of debugging. 51 students studying at the Information Technologies Departments of vocational and technical high schools participated in the research. The Debugging Performance Test was used to determine the debugging performance of the students. Also, the Gazepoint GP3 Eye Tracker tool was used to detect eye movements. In the analysis of the research, areas of interest were determined and eye movements were examined with Information Gain and Gini Index. The results indicated that eye movements while debugging change according to the characteristics of the bug types. Thus, certain metrics should be utilized depending on the debugging types. Finally, suggestions were made according to this result.

**Keywords:** Bug, debugging, types of bugs, eye-tracking, eye movements

## GİRİŞ

Hata, programlama süreci içerisinde karşılaşılan olumsuz durumlar olarak ifade edilirken hata ayıklama ise bu olumsuz durumların giderilmesini ifade etmektedir (Downey ve Mayfield, 2016). Sıklıkla karşılaşılan hataların, hangi kod satırında, nasıl veya ne zaman karşılaşılabileceği belirsiz olmakla birlikte programın çalışmasını engellemekte, kalitesini düşürmektedir (Duraes ve diğerleri, 2016). Dolayısıyla bu hataların giderilmesi, programın sağlıklı ve işlevsel çalışması adına önemlidir. Hataların giderilmesi her ne kadar olumsuz, can sıkıcı ve uğraştırıcı bir durum gibi gözükse de programlama süreci içerisinde farklı bir yeri vardır. Çünkü hata mesajlarındaki ipuçlarından yola çıkarak programın işlevselliğini olumsuz etkileyen durumun tespitinin yapılması, çözüm yollarının aranması, deneme-yanılma yapılması, düzenlemeler

gerçekleştirilmesi ve en sonunda düzeltilmesi onu farklı kılmaktadır (Downey ve Mayfield, 2016).

Programlama süreci içerisinde hatalar farklı türlerde meydana gelebilmektedir. Yıllar içerisinde alanyazında yapılan farklı tanımlamalar söz konusu olsa da (Ahmadzadeh ve diğerleri, 2005; Downey ve Mayfield, 2016; Hristova ve diğerleri, 2003; Institute of Electrical and Electronics Engineers, 2010; Johnson ve diğerleri, 1983) tanımlamalar üç tür hatanın olduğunda yoğunlaşmaktadır. Bunlar derleme zamanı hatası, çalışma zamanı hatası ve mantık hatasıdır. Derleme zamanı hatası, programlama dilinin dilbilgisinde yapılan hataları ifade etmektedir (ör. “(1+2)” uygun iken “8)” uygun değildir). Çalışma zamanı hatası, program derleme sürecinden geçtikten sonra çalışırken bir şeylerin ters gitmesiyle oluşan hatalardır (ör. sonsuz döngü). Mantık hatası ise, program kodlarının derlenip çalışmasında bir problem olmayıp doğru çıktılar vermediğinde ortaya çıkan hatalardır (ör. Girilen Yaş Verisi: 10, Sonuç: Askere gidebilir). Hataların bu şekilde sınıflandırılması ve ayrı ayrı tanımlamalarının yapılmasının sebebi, programlama sürecinde hataları erken tespit etmek, kolay bir çözüm üretmektir (Institute of Electrical and Electronics Engineers, 2010; Robins, Haden ve Garner, 2006). Dolayısıyla bunu gerçekleştirmek için her farklı hatada, hataya özel performanslar ve davranışlar gösterilmektedir (Chen ve Lim, 2013).

Bireylerin hata ayıklarken sergilemiş oldukları davranışlar ve bilişsel süreçler araştırmacılar tarafından ilgi çekici olmuştur. Çünkü hata ayıklama, programlama yapan biri için önemli bir beceridir (Sharma ve diğerleri, 2018). Bu becerinin programlama öğrencilerine kazandırılması için ders içerikleri geliştirilmelidir (Sharma ve diğerleri, 2018; Yen, Wu ve Lin, 2012). Bununda ötesinde öğrencilere hata ayıklama etkinliklerinin yoğun şekilde yaptırıldığı kodların gözden geçirilmesi teknikleri dersinin bilgisayar bilimleri müfredatlarında olması gereklidir (Ciolkowski ve diğerleri, 2002). Fakat burada hata ayıklamanın daha iyi nasıl yapacağına yönelik pedagojik bilgi eksikliği bulunmaktadır (Fitzgerald ve diğerleri, 2008). Bu eksikliği gidermek adına öğrencilerin performansını anlamak üzere tasarlanan çalışmalarda anket, görüşme ya da gözlem gibi geleneksel (van Gog ve diğerleri, 2009) ya da sesli düşünme gibi içsel

yansıma yapılan yöntemleri kullanılmıştır (Bednarik ve Tukiainen, 2004, 2005). Bu tür yöntemlerin öznel olması ve kullanıcıya bilişsel yük getirmesi ise eleştirilen bir durumdur (van Gog ve diğerleri, 2009).

Bireylerin hata ayıklama performanslarını anlamak için bilişsel paradigmaya dayalı çalışmalar yapan araştırmacıların son yıllarda kullandığı yöntemlerden biri göz izlemedir. Bu yöntem sayesinde hata ile karşılaşan bireylerin kodlar üzerindeki sergiledikleri göz hareketleri eş zamanlı olarak belirlenebilmekte ve bunlara yönelik çıkarımlar yapılabilmektedir (Bednarik ve Tukiainen, 2007; Lin ve diğerleri, 2016). Bu yöntemle programcıların hata ayıklarken yaşadıkları bilişsel süreçler hakkında bilgi toplanabilir, programlama eğitiminde kullanılacak bilgiler edinilebilir ve öğretim sürecinde iyileştirmeler yapılabilir (Sharma ve diğerleri, 2018).

Göz izleme araçları sayesinde ilgi çekici kodlara odaklanma, kodlar arasındaki hızlı sıçramalar, belirli bir kod üzerinde yapılan odaklanma süresi, belirli bir koddan diğerine geçme sayısı gibi çok sayıda metrik üzerinden veri elde edilebilmektedir. Bireylerin bu göz hareketlerine yönelik alan yazında belirtilen yorumlar ve çıkarımlar aşağıda verilmiştir (Tablo 1).

**Tablo 1.** Göz İzleme Metrikleri ve Yorumlamaları

Göz İzleme Metriği	Yorumlanması
İlk Odaklanmaya Kadar Geçen Süre (Time to First Fixation)	İlgili kodun seçilmesi ve organize edilmesi, ilgili koda ulaşma süresi (Scheiter ve Eitel, 2015, 2017; van Meeuwen ve diğerleri, 2014)
Odaklanma Sayısı (Fixation Count)	İlgili alanda gösterilen bilişsel çaba, gayret durumu (Sharafi ve diğerleri, 2012, 2013; Turner ve diğerleri, 2014)
Odaklanma Süresi (OS) (Fixation Duration)/ Toplam OS (Total Fixation Duration)	İlgili alanda gösterilen bilişsel çaba, gayret durumu ve işlemenin derinliği (Jacob ve Karn, 2003; Jian, Wu ve Su, 2014; Sharafi ve diğerleri, 2013; Shojaeizadeh, Djamasbi ve Trapp, 2016; Turner ve diğerleri, 2014)
Odaklanma Sırası (Fixation Sequences)	Sunulan ilgili alanlar içerisinde sıralı olarak gösterilen görsel dikkat örüntüleri (Jian, Wu ve Su, 2014; Lin ve diğerleri, 2016)
Göz Kırpma (Blink Rate)	Sunulan içeriğin bireyde oluşturduğu bilişsel yükü (Holland ve Tarlow, 1972)
Geçiş Sayısı (Switching) / Yeniden Ziyaret Sayısı (Revisit)	İki veya daha fazla kod alanları arasındaki bağlantı ve ilişki düzeyi (Obaidellah, al Haek ve Cheng, 2018)
Tarama Yolu (Scanpath)	Programlama kodları ve çeşitli öğrenme ortamlarında öğrencinin sunulan bilgiyi işleme mantığı (Obaidellah, al Haek ve Cheng, 2018)

Bu yorumlamalar ve çıkarımların dışında Obaidellah, al Haek ve Cheng (2018) odaklanma sayısı arttıkça koda verilen önemin arttığını, odaklanma süresinin arttıkça kodu zor anladıklarını ve karmaşık bulduklarını ifade etmişlerdir. Tüm bu yorum ve çıkarımlar bireylerin hata ayıklarken sergiledikleri bilişsel performanslarını anlamak için kullanılabilir. Göz izleme araçlarının bu bakımdan araştırmacılara yardımcı olabileceği, hata ayıklama davranışlarının anlaşılabilirliği ve bu davranışlardan yola çıkarak pedagojik bilgiler edinebilecekleri söylenebilir.

Hata ayıklama performanslarının geliştirilmesi için yapılan çalışmalar incelendiğinde araştırmacıların çalışmalarında farklı amaçlarla farklı göz hareket metrikleri kullandıkları dikkat çekmektedir (örn., Chandrika ve Amudha, 2017; Chen ve Lim, 2013; Kovari, Katona ve Costescu, 2020; Peng ve diğerleri, 2016; Sharif ve diğerleri, 2012). Çalışmalarda, farklı hata türlerine ise odaklanılmadığı (Bednarik ve Tukiainen, 2004; Chandrika ve Amudha, 2017; Kovari, Katona ve Costescu, 2020; Mangaroska ve diğerleri, 2018) ya da sadece belli bir hata türüne odaklandıkları gözlemlenmektedir (Sharif ve diğerleri, 2012). Ayrıca, çalışmalarda incelenen göz hareket metriklerinin neden seçildiğine yönelik yeterli açıklamanın da yapılmadığı görülmektedir (Bednarik ve Tukiainen, 2004; Chandrika ve Amudha, 2017; Chen ve Lim, 2013; Kovari, Katona ve Costescu, 2020; Peng ve diğerleri, 2016; Sharif ve diğerleri, 2012; Turner ve diğerleri, 2014). Dolayısıyla, bu durumlar seçilen göz hareket metrikleri ile belirlenen hata ayıklama performanslarında bireylerin davranışlarını anlamak ve buna yönelik pedagojik çıkarımlar yapmak bakımından sınırlılık oluşturmaktadır.

Hata ayıklama performansları üzerinde etkili olan önemli göz hareketlerinin belirlenmesi, farklı türdeki hatalarda nasıl bir görsel dikkat sergilenmesi gerektiğini açıklamada yarar sağlayacaktır. Bu çalışmada, farklı türde hata ayıklarken sergilenen önemli göz hareketlerinin belirlenmesi amaçlanmıştır. Bu amaç doğrultusunda aşağıda verilen üç araştırma sorusuna cevap aranmaktadır.

- a) Derleme zamanı hataların ayıklanmasında hangi göz hareket metrikleri önemlidir?
- b) Çalışma zamanı hataların ayıklanmasında hangi göz hareket metrikleri önemlidir?
- c) Mantık hataların ayıklanmasında hangi göz hareket metrikleri önemlidir?

## **YÖNTEM**

Araştırmanın deseni, makine öğrenmede kullanılan öznitelik belirleme yöntemidir. Bu yöntem, araştırmalarda sınıflama yaparken önemli olan değişkenlerin belirlenmesinde kullanılmaktadır (Breiman, 2001; Elith, Leathwick ve Hastie, 2008). Diğer bir tanımda

ise bağımlı değişkeni tahmin etmek için nitelikli değişken kümesinin belirlenmesidir (Ladha ve Deepa, 2011).

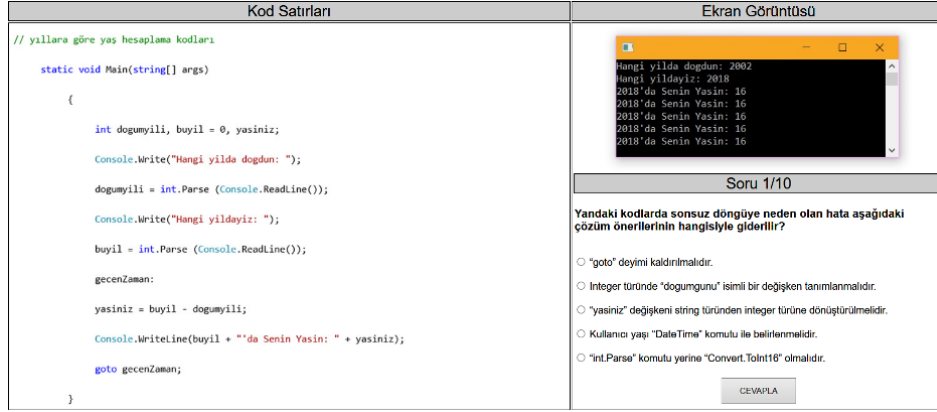
### **Katılımcılar**

Araştırmaya mesleki ve teknik liselerin Bilişim Teknolojileri Bölümünde okuyan 51 öğrenci katılmıştır. Öğrencilerin 25'i (%49) erkek, 26'sı (%51) kadın öğrencidir. 24'ü (%47) 10. sınıf, 27'si (%53) 11. sınıf öğrencisidir. Öğrencilerin hiçbiri liseden önce programlama eğitimi almamıştır. Öğrencilerin hepsi C# programlama dilini bilirken, 27'si (%53) bu dile ek olarak HTML dilini bildiklerini 4'ü (8%) ise diğer dilleri bildiklerini belirtmişlerdir.

Araştırmanın yapılması, Hacettepe Üniversitesi Etik Komisyonununun 35853172-300 sayılı 4 Eylül 2018 tarihli toplantısında etik açıdan uygun görülmüştür. Araştırmaya başlamadan önce araştırmaya dair bilgilendirmeler gerçekleştirilmiştir. Öğrenci ve velilerinden gönüllülük formlarını doldurmalarının ardından araştırmaya dâhil edilmiştir.

### **Veri Toplama Araçları**

*Hata Ayıklama Performansı Testi:* Akçay ve Altun (2021) tarafından geliştirilen Hata Ayıklama Performans Testi, meslek lisesi BT bölümünde okuyan öğrencilerin C# programlama dilinde hata ayıklama performanslarını belirlemeyi sağlamaktadır. Testte 10 tane kod senaryosu bulunmaktadır. 3 senaryoda derleme zamanı hatası, 3 senaryoda çalışma zamanı hatası ve 4 senaryoda mantık hatası bulunmaktadır. Test, bilgisayar ortamında yapılmaktadır (Şekil 1). Testten en yüksek 10, en düşük 0 puan alınmaktadır. Test ortamında programlama gerçekleştirilen ortamın bağlamına uygun olarak derleme zamanı hatalarında hatalı satırının vurgulanmış ve hata mesajının ekran çıktısının sunulmuştur. Ayrıca çalışma zamanı ve mantık hatalarında yine oluşan ekran çıktısının sunulması söz konusudur.

**Şekil 1.** Hata ayıklama performansı testi ekran görüntüsü (Akçay ve Altun, 2021)

**Göz İzleme Aracı:** Çalışmada göz hareketlerine ilişkin veriler Hacettepe Üniversitesi Eğitsel Ontoloji ve Biliş Laboratuvarı (ONTOLAB) bünyesinde bulunan Gazepoint GP3 Eye Tracker cihazı ile elde edilmiştir. Bu cihaz 20 inç 1600x900 çözünürlüğüne sahip bir monitör ile kullanılmıştır. Taşınabilir olan cihaz, 60 hertz veri toplama frekansına sahiptir.

### Veri Toplama Süreci

Veri toplama süreci, çalışmaya katılan öğrencilerin okullarında gerçekleştirilmiştir. Araştırmaya katılabilecek muhtemel öğrencilere bilgilendirmeler yapılmış ve velileri ve kendileri gönüllü olan öğrenciler deney ortamına davet edilmiştir. Deney ortamları için okullarda sessiz ve dikkat çekici ya da dağıtıcı nesne ya da kişinin olmadığı ortamlar tercih edilmiştir. Ayrıca deneyler gündüz saatlerinde, gün ışığının bulunduğu zamanlarda gerçekleştirilmiştir.

Deney ortamında gereken bilgilendirmeler yapılmış ve varsa soruları cevaplandırılmıştır. Hata Ayıklama Performans Testi'nin her bir sorusundan önce ekranda 9 noktalı kalibrasyon gerçekleştirilmiştir. Öğrencilerin her bir soru için verdikleri cevaplar veri tabanına kaydedilmiştir. Öğrencilerin her bir sorudaki göz hareketleri Gazepoint Analysis Professional yazılımı ile elde edilmiştir. Bu yazılım, yukarıda belirtilen göz izleme cihazı ile entegre bir şekilde çalışan verilerin



toplanmasında ve ilgili metriklerle ait verileri elde etmek amacı ile ilgi alanlarının (Area of Interests) belirlenmesinde kullanılmaktadır. Bu süreç çalışma boyunca ortalama 20 dakika sürmüştür. Uygulama sonrasında katılımcılara teşekkür edilmiş ve çalışma oturumu sonlandırılmıştır.

### Verilerin Analizi

Öğrencilerin program senaryoları içerisinde bulunan hataları ayıklarken sergiledikleri göz hareketlerine yönelik verileri belirlemek için ilgi alanları (Area of Interests) belirlenmiştir. Bu ilgi alanları hatanın olduğu kod satırı (Hata Satırı [HATA]), hata haricindeki diğer kodlar (Diğer Kodlar [DİGER]) ve hatanın ekran gösterimidir (Ekran Çıktısı [EKRAN]) (Şekil 2). İlgi alanlarını belirlerken bir programlama ortamı arayüzünde var olan bileşenler ve çalışmanın amacına uygun kod satırının seçilmesine dikkat edilmiştir.

Şekil 2. İlgi alanları



Şekil 2'de görüldüğü üzere ilgi alanları belirlenmiş ve bu bölgelerde sergilenen göz hareketlerine ilişkin veriler elde edilmiştir. Bu veriler, test ekranının açılmasından

itibaren ilgi alanına yapılan ilk odaklanmaya kadar geçen süreyi ifade eden “İlk Odaklanmaya Kadar Geçen Süre (IOKGS)”, ilgi alanında yapılan toplam odaklanma süresini ifade eden “Toplam Odaklanma Süresi (TOS)”, ilgi alanında yapılan toplam odaklanma sayısını ifade eden “Odaklanma Sayısı (OSA)”, ilgi alanında sergilenen odaklanma süresinin tüm odaklanma süresine oranını ifade eden “Odaklanma Süresi Yüzdesi (OSY)”, ilgi alanındaki toplam odaklanma süresinin odaklanma sayısına oranını ifade eden “Ortalama Odaklanma Süresi (OOS)” ve ilgi alanına yapılan başvuru sayısını ifade eden “Ziyaret Sayısı (ZS)”dır. Her bir hatalı kod senaryosu için üç ilgi alanında yapılan altışar metrik ile toplam 18 göz izleme verisi incelenecek veri setini oluşturmaktadır. Verilerin geçerliğinin sağlanması için göz izleme cihazında 9 kalibrasyon noktası kullanılırken katılımcıların göz hareketlerinin edinme oranının %80 ve üzerinde olduğu görülmüştür. Elde edilen verilerde aykırı ve kayıp değer incelemeleri yapılmış, aykırı veya kayıp değer olmadığı görülmüştür.

Göz izleme aracı ile elde edilen veri seti içerisinde önemli olan göz izleme verilerini belirlemek için “Bilgi Kazancı” ve “Gini Katsayısı” olmak üzere iki farklı öznitelik belirleme yöntemi kullanılmıştır. Bu yöntemlerin tercih edilmesinde verilerle ilgili belirsizlik ölçüsü entropiye dayalı olan ve olmayan iki yöntem olmasına dikkat edilmiştir. Bilgi Kazancı (Information Gain), hata ayıklama performansını temsil edecek nitelik kümesini belirlemek amacıyla kullanılan, ilgili değişkende davranış görülmesi durumuna göre frekansına bakılarak ölçülen öznitelik belirleme yöntemidir (Baig, Shaheen ve AbdelAal, 2012; Kaliyeva, 2013; Porkodi, 2014). Çalışmada kullanılan diğer öznitelik belirleme yöntemi Gini Katsayısı’dır (Gini Index). Gini Katsayısı ise Bilgi Kazancı’ndan farklı olarak formülünde entropi değeri kullanılmamakta olup, sınıflama için çok değişkenli öznitelik belirleme yöntemleridir (Kaynar ve diğerleri, 2018; Porkodi, 2014). Araştırma amacı doğrultusunda önemli göz hareket metriklerini belirlemek için açık kaynaklı veri görselleştirme, makine öğrenmesi ve veri madenciliği araç seti olan Orange 3.24.1 ile Microsoft Excel 2016 programı kullanılmıştır. Çalışmada farklı türde hata ayıklama performanslarını açıklanmasında önem düzeyine göre ilk beş göz izleme metriği belirlenmiştir.

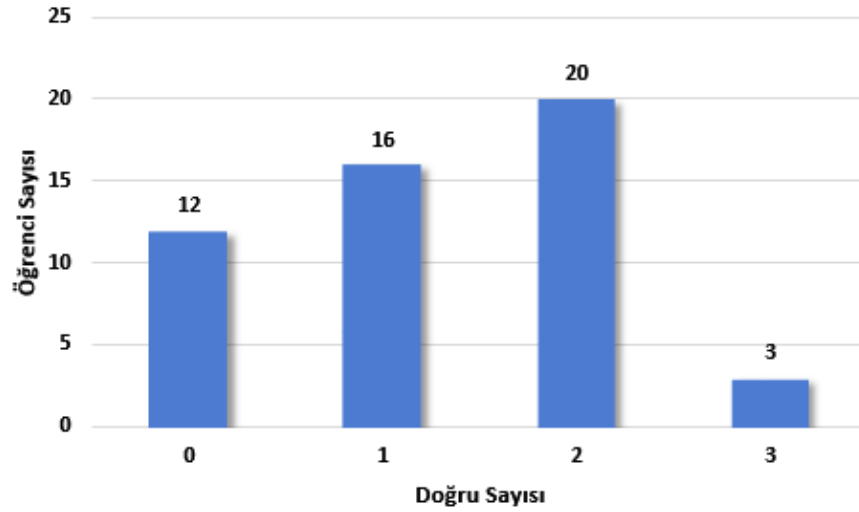
## BULGULAR

Bu çalışmada farklı türlerde hata ayıklama sırasında önemli göz hareketlerinin belirlenmesi amaçlanmıştır. Buna göre analiz sonucunda elde edilen derleme zamanı hatası, çalışma zamanı hatası ve mantık hatası ayıklama performanslarında önemli olan göz hareketleri başlıklar halinde aşağıda verilmiştir.

### Derleme Zamanı Hatalarını Ayıklama Performansında Önemli Göz Hareket Metrikleri

Çalışmada öğrenciler derleme zamanı hatası içeren kodlarda hata ayıklama performanslarını sergilemişlerdir. Öğrencilerin hata ayıklama durumları aşağıda belirtilmiştir (Şekil 3).

Şekil 3. Öğrencilerin derleme zamanı hataları ayıklama durumu



Şekil 3'te görüldüğü üzere öğrencilerin 3'ü bütün derleme zamanı hatası ayıklama sorularına doğru yanıt vermiştir. 20'si iki doğru yanıt verirken, 16'sı sadece bir doğru

yanıt vermiştir. Öğrencilerin 12'si ise hiçbir derleme zamanı hatası ayıklama sorusunu doğru yanıtlamamıştır. Öğrencilerin sergiledikleri derleme zamanı hatası ayıklama performansını belirlemede önemli bulunan göz hareket metrikleri aşağıda belirtilmiştir (Tablo 1).

**Tablo 1.** Derleme Zamanı Hatalarını Ayıklama Performansında Önemli Göz Hareket Metrikleri

Bilgi Kazancı			Gini Katsayısı		
Değişken	Önem Düzeyi	Durum	Değişken	Önem Düzeyi	Durum
HATA TOS	.27	+	HATA TOS	.10	+
DIGER IOKGS	.25	-	HATA OSY	.08	+
HATA OSY	.23	+	EKRAN TOS	.07	+
EKRAN OSA	.20	+	EKRAN ZS	.07	+
EKRAN TOS	.19	+	DIGER IOKGS	.07	-

Tablo 1'de analizi yapılan Bilgi Kazancı yöntemine göre en önemli değişkenin hata ayıklama satırına yapılan toplam odaklanma süresinin olduğu görülmüştür (Önem Düzeyi: .27). Bu sürenin uzun olması hatanın ayıklanmasında yarar sağlayacaktır. Bundan sonraki en önemli değişken diğer kodlara yapılan ilk odaklanmaya kadar geçen süredir (Önem Düzeyi: .25). Bu sürenin kısa olması derleme zamanı hatalarının ayıklanmasında önemlidir. Hata satırına yapılan odaklanma süresinin yüzdesi (Önem Düzeyi: .23) ve ekran görüntüsüne yapılan odaklanma sayısı (Önem Düzeyi: .20) önemli olan diğer değişkenlerdir. Bu değişkenlerinde yüksek olması derleme zamanı hatalarını ayıklama performansını artırıcı değişkenlerdir. Analiz sonucunda derleme zamanı hatasını ayıklamada önemli son değişkende ekran çıktısına yapılan toplam odaklanma süresidir (Önem Düzeyi: .19). Bu sürenin uzun olması da derleme zamanı hatalarının ayıklanmasında önemlidir.

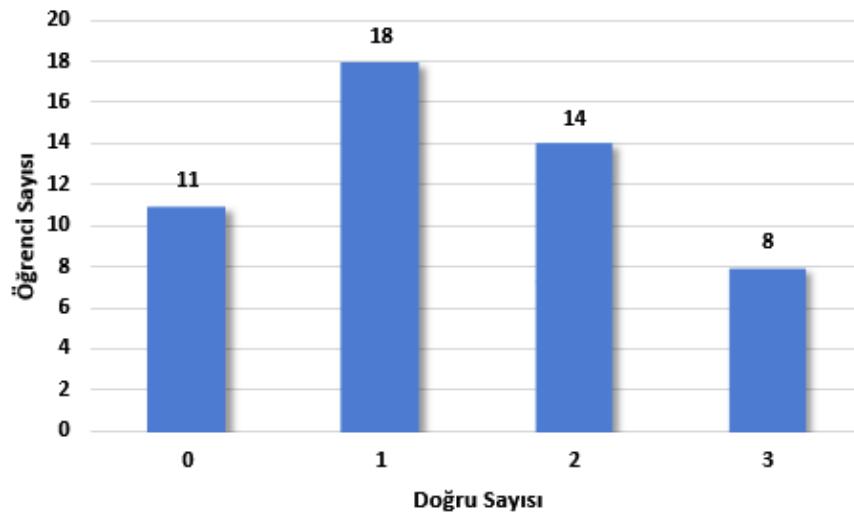
Gini Katsayısı ile belirlenen derleme zamanı hata ayıklama performansında önemli olan değişkenlerin dördünün Bilgi Kazancı ile aynı olduğu belirlenmiştir (Tablo 1). Buna göre en önemli değişkenlerin hata satırına yapılan toplam odaklanma süresi (Önem

Düzeyi: .10) ve bunun tüm odaklanma sürelerine olan yüzdesi (Önem Düzeyi: .08) olduğu görülmüştür. Ekran çıktısına olan toplam odaklanma sayısı (Önem Düzeyi: .07) ve ziyaret sayısı (Önem Düzeyi: .07) derleme zamanı hatası ayıklama performansının değerlendirilmesinde önemli diğer değişkenlerdir. Tüm bu değişkenlerin yüksek olması derleme zamanı hatası ayıklanmasında önemlidir. Gini Katsayısı'na göre son önemli değişken ise diğer kodlara yapılan ilk odaklanmaya kadar geçen süredir (Önem Düzeyi: .07). Bu süre kısa olması derleme zamanı hatasının ayıklanma performansında önemlidir.

### Çalışma Zamanı Hatalarını Ayıklama Performansında Önemli Göz Hareket Metrikleri

Kodlar içerisinde bulunan çalışma zamanı hatalarının ayıklanma performansında önemli olan değişkenler belirlenmiştir. Bunun için öncesinde öğrencilerin çalışma zamanı hatalarını ayıklama durumları aşağıda belirtilmiştir (Şekil 4).

Şekil 4. Öğrencilerin derleme zamanı hataları ayıklama durumu



Şekil 4'te görüldüğü üzere öğrencilerin 8'i üç soruyu doğru cevaplamıştır. 14'ü iki soruya doğru cevap verirken bir soruya doğru cevap verenlerin sayısı 18'dir. Soruların

hiçbirine doğru cevap veremeyen öğrenci sayısı ise 11'dir. Öğrencilerin çalışma zamanı hatalarını ayıklarken sergiledikleri performansta belirleyici ve önemli bulunan göz hareket metriklerine ilişkin elde edilen bulgular aşağıda verilmiştir (Tablo 2).

**Tablo 2.** Çalışma Zamanı Hatalarını Ayıklama Performansında Önemli Göz Hareket Metrikleri

Bilgi Kazancı			Gini Katsayısı		
Değişken	Önem Düzeyi	Durum	Değişken	Önem Düzeyi	Durum
EKRAN OOS	.21	+	DIGER OOS	.10	+
DIGER IOKGS	.20	-	HATA OSY	.08	+
EKRAN OSY	.19	+	DIGER IOKGS	.07	-
HATA OSY	.18	+	EKRAN OOS	.07	+
DIGER OOS	.18	+	EKRAN IOKGS	.07	-

Tablo 2 incelendiğinde Bilgi Kazancı yöntemine göre belirlenen değişkenlerden en önemli olanı ekran çıkışına yapılan ortalama odaklanma süresidir (Önem Düzeyi: .21). Bu sürenin uzun olması çalışma zamanı hatalarının ayıklanması için önemlidir. Sonraki önemli değişken ise diğer kodlara yapılan ilk odaklanmaya kadar geçen süredir (Önem Düzeyi: .20). Bu sürenin kısa olması çalışma zamanı hatalarının ayıklanma performansında önemlidir. Sonraki önemli değişkenler ekrana yapılan odaklanma süresi yüzdesi (Önem Düzeyi: .19) ve hata satırına yapılan odaklanma süresi yüzdesidir (Önem Düzeyi: .18). Bu yüzdelerin yüksek olması çalışma zamanı hatalarının ayıklanma performansını belirlemede önemlidir. Son olarak diğer kodlara yapılan ortalama odaklanma süresi önemli son değişkendir (Önem Düzeyi: .18). Bu sürenin uzun olması çalışma zamanı hatalarını ayıklama performansında önemlidir.

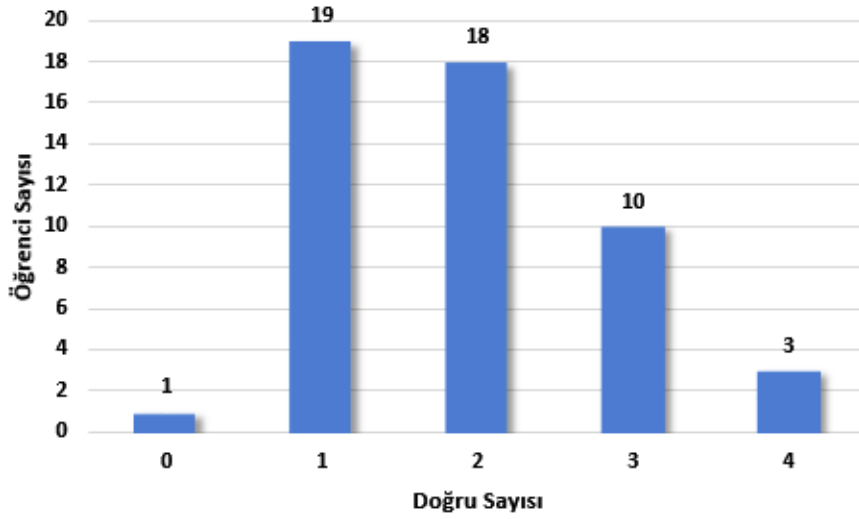
Çalışma zamanı hatalarının ayıklanmasında en çok bilgi veren değişkenlerin incelendiği Gini Katsayısı'na göre en önemli değişkenin diğer kodlara yapılan ortalama odaklanma süresinin olduğu belirlenmiştir (Önem Düzeyi: .10) (Tablo 2). Bu sürenin uzun olması çalışma zamanı hatalarının ayıklanması bakımından önemlidir. Sonraki önemli değişken ise hata satırına yapılan odaklanma süresinin yüzdesidir (Önem Düzeyi: .08). Bu

yüzdenin yüksek olması çalışma zamanı hatalarının ayıklanması bakımından önemlidir. Diğer önemli değişkenlerde sırasıyla diğer kodlara yapılan ilk odaklanmaya kadar geçen süre (Önem Düzeyi: .07), ekran çıktısına yapılan ortalama odaklanma süresi (Önem Düzeyi: .07) ve ilk odaklanmaya kadar geçen süredir (Önem Düzeyi: .07). Bu sürelerden ilk odaklanmaya kadar geçen sürelerin kısa olması ortalama odaklanma süresinin uzun olması çalışma zamanı hatalarının ayıklanma performansı bakımından önemlidir.

### Mantık Hatalarını Ayıklama Performansında Önemli Göz Hareket Metrikleri

Öğrencilerin mantık hatalarını ayıklama performansında önemli olan değişkenler belirlenmiştir. Öğrencilerin mantık hatalarını ayıklama durumuna aşağıda gösterilmiştir (Şekil 5).

Şekil 5. Mantık hataları ayıklama durumuna göre öğrenci dağılımı



Şekil 5'te görüldüğü üzere öğrencilerin 3'ünün bütün sorulara doğru cevap verdikleri görülmektedir. 10'u üç soruya, 18'i iki soruya doğru cevap vermiştir. Öğrencilerin 19'u yalnızca bir soruya doğru yanıt verirken, 1 öğrenci hiçbir soruya doğru yanıt

verememiştir. Öğrencilerin sergiledikleri bu performansları hakkında en çok bilgi veren önemli değişkenleri belirlemek için göz hareketleri incelenmiş ve bulgular aşağıda verilmiştir (Tablo 3).

**Tablo 3.** Mantık Hatalarını Ayıklama Performansında Önemli Göz Hareket Metrikleri

Bilgi Kazancı			Gini Katsayısı		
Değişken	Önem Düzeyi	Durum	Değişken	Önem Düzeyi	Durum
EKRAN TOS	.43	+	EKRAN TOS	.10	+
HATA IOKGS	.32	+	DİGER TOS	.09	+
DİGER TOS	.28	+	HATA IOKGS	.08	+
HATA ZS	.24	-	HATA ZS	.06	-
EKRAN ZS	.23	+	EKRAN ZS	.06	+

Mantık hatalarının ayıklanmasında en çok bilgi veren önemli göz hareketlerinin belirlenmesi için yapılan analizlerde Bilgi Kazancı yöntemi kullanılmıştır (Tablo 3). Analizde ulaşılan en önemli değişken, ekran çıktısına yapılan toplam odaklanma süresidir (Önem Düzeyi: .43). Bu sürenin artması mantık hatasının ayıklanması performansında önemlidir. Bilgi Kazancı'na göre sıradaki en önemli değişkenler hata satırına yapılan ilk odaklanmaya kadar geçen süre (Önem Düzeyi: .32) ile diğer kodlara yapılan toplam odaklanma süreleridir (Önem Düzeyi: .28). Bu sürelerin uzun olması mantık hatalarının ayıklanması performansı açısından önemlidir. Sonraki önemli değişkenler ise hata satırına (Önem Düzeyi: .24) ve ekran çıktısına (Önem Düzeyi: .23) yapılan ziyaret sayılarıdır. Hata satırına yapılan ziyaret sayılarının az olması ekran çıktısına ziyaret sayısının çok olması mantık hatalarının ayıklanması bakımından önemlidir.

Mantık hatalarının ayıklanmasında en çok bilgi veren değişkenlerin belirlenmesi için Gini Katsayısı incelenmiştir (Tablo 3). Bilgi Kazancı yöntemine benzer olarak en çok bilgi veren değişken ekran çıktısına yapılan toplam odaklanma süresidir (Önem Düzeyi: .10). Bu sürenin uzun olması mantık hatalarının ayıklanmasında önemlidir. Daha sonra sırasıyla diğer kodlara yapılan toplam odaklanma süresi (Önem Düzeyi: .09) ve hata



satırına yapılan ilk odaklanmaya kadar geçen süre (Önem Düzeyi: .08) mantık hatası ayıklamada önemlidir. Bu sürelerin uzun olması mantık hatası ayıklama performansı açısından önemlidir. Mantık hatalarının ayıklanmasında diğer önemli değişkenler hata satırına (Önem Düzeyi: .06) ve ekran çıktısına (Önem Düzeyi: .06) yapılan ziyaret sayısıdır. Hata satırına ziyaret sayısının az olması ve ekran çıktısına yapılan ziyaret sayısının çok olması mantık hatasının ayıklanma performansının açısından önemlidir.

## TARTIŞMA

Bu çalışmada öznitelik belirleme yöntemleri kullanılarak farklı türde hata ayıklamada önemli göz hareketlerinin belirlenmesi amaçlanmıştır. Çalışmada uygulanan iki farklı yöntemden elde edilen bulgular temel alınarak tartışma yapılmıştır.

Öğrencilerin derleme zamanı hatalarını ayıklarken hata satırı üzerinde yapmış oldukları toplam odaklanma süresi ve odaklanma süresi yüzdesinin en çok bilgi veren önemli metrikler olduğu belirlenmiştir. Çünkü derleme zamanı hataları hatanın olduğu satırda yapılan bir yanlış sözdiziminden kaynaklanmaktadır (Ahmadzadeh, Elliman ve Higgins, 2005; Downey ve Mayfield, 2016; Hristova ve diğerleri, 2003). Hatanın düzeltilmesi için çoğu zaman bu satırın incelenmesi, üzerinde işlem yapılması gereklidir. Bunun için öğrencilerin hata satırı üzerinde bilişsel bir çaba gösterdiği, ilgili kodları işlemeye çalıştığı söylenebilir (Jacob ve Karn, 2003; Jian, Wu ve Su, 2014; Obaidellah, al Haek ve Cheng, 2018; Schwonke, Berthold ve Renkl, 2009; Sharafi ve diğerleri, 2013; Shojaeizadeh, Djamasi ve Trapp, 2016). Öte yandan çalışmada yüksek düzeyde hata ayıklama performansı gösteren bireylerin hata olması olası kodlara bakmak yerine programlama mantığını takip ettikleri araştırmalarda vardır (ör. Lin ve diğerleri, 2016). Fakat bu araştırmalar incelendiğinde aynı kod senaryosu içerisinde birden fazla hata bulunmaktadır. Ayrıca, programlama yazılımlarına özgü olarak derleme zamanı hatalarının olduğu kod satırına yapılan vurgulamanın yapılmadığı görülmüştür.

Derleme zamanı hatalarının ayıklanmasında önemli diğer göz hareketleri, hataya ve meydana geldiği satıra yönelik bilgi içeren metin mesajının ekran çıktısına yapılan

toplam odaklanma süresi, odaklanma sayısı ve ziyaret sayılarıdır. Öğrencilerin sergiledikleri bu göz hareketleri, hata ayıklama sırasında muhtemel bir karışıklık yaşadıklarını (Obaidellah ve diğerleri, 2018), bu karışıklığı çözmek adına bilişsel çaba gösterdiklerini (Jacob ve Karn, 2003; Jian, Wu ve Su, 2014; Obaidellah, al Haek ve Cheng, 2018) ve hata mesajı ile kodlar arasındaki bağlantıyı kurma çabalarını (Obaidellah, al Haek ve Cheng, 2018) göstermektedir. Derleme zamanı hatalarının ayıklanmasında önemli diğer değişken ise diğer kodlara yapılan ilk odaklanmaya kadar geçen sürenin kısa olmasıdır. Bu durum öğrencilerin hata ile karşılaştıklarında ayıklamaya başlamadan önce kodlar arasında tarama yapmaları, bir arayış içerisinde davranmalarıyla açıklanabilir (Sharif, Falcone ve Maletic, 2012).

Çalışmada ele alınan bir diğer hata türü olan çalışma zamanı hatalarının ayıklanmasına yönelik en çok bilgi alınan önemli göz hareketleri belirlenmiştir. Buna göre öğrencilerin ekran çıktısına yaptıkları odaklanma süreleri, yüzdeleri ve diğer kodlara yaptıkları odaklanma sürelerinin önemli olduğu belirlenmiştir. İlgili hatalar, çalışırken ekran çıktısının kapanması ve birden durmasıyla fark edilir (Downey ve Mayfield, 2016; Hristova ve diğerleri, 2003). Ekran çıktısı ve diğer kodlar üzerinde bilişsel bir çaba gösterilmesi (Jacob ve Karn, 2003; Jian, Wu ve Su, 2014; Obaidellah, al Haek ve Cheng, 2018) araştırmada beklendiği bir sonuç olmuştur. Aynı zamanda ekran çıktısına ve diğer kodlara yapılan ilk odaklanmaya kadar geçen sürenin önemli olduğu belirlenmiştir. Bu göz hareketlerinin önemli olması çalışma zamanı hatalarının ayıklanırken bir şeylerin ters gittiği kodun anlaşılması (Downey ve Mayfield, 2016), ilgili kodun seçilmesi (Scheiter ve Eitel, 2017) ve hatanın meydana geldiği kodun bulunması ile ilgili olduğu söylenebilir. Çalışma zamanı hatalarının ayıklanmasında önemli diğer bir göz hareketi hata satırına yapılan odaklanma süresi yüzdesidir. Bu durum öğrencilerin hata satırı üzerinde bilişsel bir çaba sergilemelerinin, hatayı anlamaya çalışmalarının hatayı ayıklamaları için önemli olduğunu göstermektedir (Jian, Wu ve Su, 2014; Majooni Masood ve Akhavan, 2016; Obaidellah, al Haek ve Cheng, 2018; Schwonke, Berthold ve Renkl, 2009).

Öğrencilerin mantık hatalarını ayıklama performanslarını açıklamada önemli göz hareketleri belirlenmiştir. Bunlardan ilki ekran çıktısına ve diğer kodlara yapılan toplam odaklanma süreleridir. Mantık hataları, hatanın meydana geldiği satırdan ziyade algoritmadan kaynaklandığından dolayı (Downey ve Mayfield, 2016) öğrencinin, ekran çıktısına bakıp algoritmanın hangi kısmında hatanın oluştuğunu anlaması, üzerine bilişsel çaba sergilemesi (Jacob ve Karn, 2003; Jian, Wu ve Su, 2014; Obaidellah, al Haek ve Cheng, 2018; Schwonke, Berthold ve Renkl, 2009) beklendik bir sonuçtur. Hata satırına yapılan ilk odaklanmaya kadar geçen sürenin uzun olması mantık hatalarının ayıklanmasında önemlidir. Ayrıca hata satırına başvurunun düşük olması ve hatanın ayıklanmasında önemli ekran görüntüsüne başvurunun yüksek olması mantık hatalarının ayıklanmasında önemlidir. Bu değişkenlerin önemi bu hata türünün algoritmaya dayalı olması ile ilgili satırın hemen seçilmesinin (Scheiter ve Eitel, 2017) hata ayıklamada gerekli olmaması, diğer kodlarla yeterli bağının olmaması (Obaidellah, al Haek ve Cheng, 2018; Scheiter ve Eitel, 2017) ve diğer kodlar üzerinde odaklanmasının gerekliliği ile açıklanabilir.

## SONUÇLAR

Yapılan çalışma çerçevesinde incelenen alanyazında farklı türde hata ayıklama sürecinde katılımcıların göz hareketlerini izleyerek yapılan çalışmalar olmasına rağmen, göz hareket metriklerinin önem durumunun belirlenmesine yönelik bir çalışma ile karşılaşılmamıştır. Bu çalışmada farklı türde hata ayıklama ortamlarında öğrencilerin bilişsel süreçlerine dair bilgi veren önemli göz hareket metrikleri ortaya konulmuştur. Çalışmada, farklı türde hata ayıklama süreçlerinde farklı göz hareketlerinin ön plana çıktığı, hata ayıklama performansları üzerinde etkili olabilecek farklı göz hareketleri sergiledikleri belirlenmiştir. Öğrencilerin sergiledikleri göz hareketleri incelendiğinde hata türlerinin genel özelliklerine göre davrandıkları, göz hareketlerinin bu özellikleri yansıttıkları belirlenmiştir. Dolayısıyla, hata ayıklama süreçlerinde göz hareketlerinin incelendiği çalışmalarda hataların özelliklerinin göz hareket metriklerinin seçilmesinde göz önünde bulundurulması tavsiye edilmektedir. Bu süreçte elde edilen bulgulara

dayalı olarak programlamada karşılaşılan hataların ayıklanmasında öğrencilere ipuçları sağlanabilir. Öğrencilere hataların özellikleri ve bu özelliklere dayalı olarak nasıl bir arayış gerçekleştirmeleri, çözümlenme yapmaları gerektiğine ilişkin bilgiler verilebilir, strateji geliştirmeleri sağlanabilir.

### **Sınırlılıklar**

Araştırmada elde edilen bulguların sınırlılıkları bulunmaktadır. Bu çalışmada kullanılan hata ayıklama ortamı deneysel bir ortam olup gerçek zamanlı bir programlama sürecinde karşılaşılabilecek hataların ayıklanması üzerine bir çalışma gerçekleştirilebilir. Ayrıca, Millî Eğitim Bakanlığı (2011) tarafından “Programlama Temelleri” dersi kapsamında C# dili ile eğitim yapılmaktadır. Benzer çalışmaların farklı dillerin öğretim süreçlerinde de gerçekleştirilecek çalışmalara gereksinim bulunmaktadır. Ayrıca, araştırmada sergilenen kod senaryoları Millî Eğitim Bakanlığı tarafından belirlenen konular ile sınırlıdır. Araştırmada incelenen göz hareketleri metrikleri kullanılan göz izleme aracının ve yazılımının sunmuş olduğu metrikler ile sınırlıdır. Farklı göz izleme metriklerine ait verilerin elde edildiği araç ve yazılımlar ile araştırmanın tekrarlanması önerilmektedir. Araştırma sonuçlarının genelleştirilmesi adına daha fazla katılımcı sayısı ile araştırmanın derinleştirilmesi önerilmektedir. Ayrıca farklı bilgi düzeylerine göre değişkenlik gösterme durumunun da incelendiği çalışmalar yapılabilir.

**KAYNAKLAR**

- Ahmadzadeh, M., Elliman, D., and Higgins, C. (2005). An analysis of patterns of debugging among novice computer science students. *ACM SIGCSE Bulletin*, 37(3), 84–88. <https://doi.org/10.1145/1151954.1067472>
- Akçay, A., ve Altun, A. (2021). Hata ayıklama performansı testi geliştirme: Geçerlik ve güvenilirlik çalışması. *Erzincan University Journal of Education Faculty*, 23(3), 667-685. <https://doi.org/10.17556/erziefd.815922>
- Baig, Z. A., Shaheen, A. S., and AbdelAal, R. (2012). One-dependence estimators for accurate detection of anomalous network traffic. *International Journal for Information Security Research*, 2(4), 202–210. <https://doi.org/10.20533/ijisr.2042.4639.2012.0025>
- Bednarik, R., and Tukiainen, M. (2004). Visual attention and representation switching in Java program debugging: A study using eye movement tracking. In E. Dunican ve T. R. G. Green (Eds.), *Proceedings of 16th Workshop of Psychology of Programming Interest Group* (ss. 159–169). Berlin, Germany: Springer. <https://www.ppig.org/papers/2004-ppig-16th-bednarik/>
- Bednarik, R., and Tukiainen, M. (2005). Effects of display blurring on the behavior of novices and experts during program debugging. In G. van der Veer (Ed.), *CHI 2005 Conference on Human Factors in Computing Systems* (ss. 1204–1207). New York, USA: Association for Computing Machinery. <https://doi.org/10.1145/1056808.1056877>
- Bednarik, R., and Tukiainen, M. (2007). Validating the restricted focus viewer: A study using eye-movement tracking. *Behavior Research Methods*, 39(2), 274–282. <https://doi.org/10.3758/BF03193158>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5-32. <https://doi.org/10.1023/A:1010933404324>
- Chandrika, K. R., and Amudha, J. (2017). An eye tracking study to understand the visual perception behavior while source code comprehension. *International Journal of Control Theory and Applications*, 10(19), 169–175. [https://serialsjournals.com/abstract/11377\\_20-chandrika\\_k.r..pdf](https://serialsjournals.com/abstract/11377_20-chandrika_k.r..pdf)
- Chen, M., and Lim, V. (2013). Eye gaze and mouse cursor relationship in a debugging task. In Stephanidis C. (Ed.), *HCI International 2013 - Posters' Extended Abstracts* (ss. 468–472). Berlin, Germany: Springer. [https://doi.org/10.1007/978-3-642-39473-7\\_93](https://doi.org/10.1007/978-3-642-39473-7_93)
- Ciolkowski, M., Laitenberger, O., Rombach, D. H., Shull, F., and Perry, D. (2002). Software inspections, reviews and walkthroughs. In W. Tracz (Ed.), *Proceedings of the 24th International Conference on Software Engineering* (ss. 641–642).

- New York, USA: Association for Computing Machinery.  
<https://doi.org/10.1145/581339.581422>
- Downey, A. B., and Mayfield, C. (2016). *Think Java: How to think like a computer scientist*. Green Tea Press. <https://open.umn.edu/opentextbooks/textbooks/think-java-how-to-think-like-a-computer-scientist>
- Duraes, J., Madeira, H., Castelhana, J., Duarte, C., and Branco, M. C. (2016). WAP: Understanding the brain at software debugging. *Proceedings - International Symposium on Software Reliability Engineering, ISSRE*, 87–92.  
<https://doi.org/10.1109/ISSRE.2016.53>
- Elith, J., Leathwick, J. R., and Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology*, 77, 802–813. <https://doi.org/10.1111/j.1365-2656.2008.01390.x>
- Fitzgerald, S., Lewandowski, G., McCauley, R., Murphy, L., Simon, B., Thomas, L., and Zander, C. (2008). Debugging: Finding, fixing and flailing, a multi-institutional study of novice debuggers. *Computer Science Education*, 18(2), 93–116. <https://doi.org/10.1080/08993400802114508>
- Holland, M. K., and Tarlow, G. (1972). Blinking and mental load. *Psychological Reports*, 31(1), 119–127. <https://doi.org/10.2466/pr0.1972.31.1.119>
- Hristova, M., Misra, A., Rutter, M., and Mercuri, R. (2003). Identifying and correcting Java programming errors for introductory computer science students. In S. Grissom (Ed.), *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education* (ss. 153–156). New York, USA: Association for Computing Machinery. <https://doi.org/10.1145/611892.611956>
- Institute of Electrical and Electronics Engineers. (2010). *IEEE standard classification for software anomalies*. <https://ieeexplore.ieee.org/document/5399061>
- Jacob, R. J. K., and Karn, K. S. (2003). Eye tracking in human-computer-interaction and usability in research: Ready to deliver the promises. In R. Radach, J. Hyona, ve H. Deubel (Eds.), *The Minds Eyes: Cognitive and Applied Aspects of Eye Movements* (ss. 573–605). Amsterdam, Holland: Elsevier.  
<https://doi.org/10.1016/B978-044451020-4/50031-1>
- Jian, Y. C., Wu, C. J., and Su, J. H. (2014). Learners' eye movements during construction of mechanical kinematic representations from static diagrams. *Learning and Instruction*, 32, 51–62.  
<https://doi.org/10.1016/j.learninstruc.2014.01.005>
- Johnson, W. L., Soloway, E., Cutler, B., and Draper, S. (1983). *Bug catalogue: I*. <https://cpsc.yale.edu/research/technical-reports/1983-technical-reports>
- Kaliyeva, S. (2013). *Bilimsel makalelerin metin işleme yöntemleri ile sınıflandırılması* (Yüksek Lisans Tezi). Fen Bilimleri Enstitüsü, Ankara.

- Kaynar, O., Arslan, H., Görmez, Y., ve Işık, Y. E. (2018). Makine öğrenmesi ve öznitelik seçim yöntemleriyle saldırı tespiti. *Bilişim Teknolojileri Dergisi*, 11(2), 175–185. <https://doi.org/10.17671/gazibtd.368583>
- Kovari, A., Katona, J., and Costescu, C. (2020). Evaluation of eye-movement metrics in a software debugging task using GP3 eye tracker. *Acta Polytechnica Hungarica*, 17(2), 57–76. <https://doi.org/10.12700/APH.17.2.2020.2.4>
- Ladha, L., and Deepa, T. (2011). Feature selection methods and algorithms. *International Journal on Computer Science and Engineering*, 3(5), 1787–1797. <http://www.enggjournals.com/ijcse/doc/IJCSE11-03-05-051.pdf>
- Lin, Y. T., Wu, C. C., Hou, T. Y., Lin, Y. C., Yang, F. Y., and Chang, C. H. (2016). Tracking students' cognitive processes during program debugging-An eye-movement approach. *IEEE Transactions on Education*, 59(3), 175–186. <https://doi.org/10.1109/TE.2015.2487341>
- Majooni, A., Masood, M., and Akhavan, A. (2016). An eye tracking experiment on strategies to minimize the redundancy and split attention effects in scientific graphs and diagrams. In G. di Bucchianico ve P. Kercher (Eds.), *Advances in Design for Inclusion* (ss. 529–539). Berlin, Germany: Springer. [https://doi.org/10.1007/978-3-319-41962-6\\_47](https://doi.org/10.1007/978-3-319-41962-6_47)
- Mangaroska, K., Sharma, K., Giannakos, M., Traeteberga, H., and Dillenbourg, P. (2018). Gaze-driven design insights to amplify debugging skills: A learner-centred analysis approach. *Journal of Learning Analytics*, 5(3), 98–119. <https://doi.org/10.18608/jla.2018.53.7>
- Millî Eğitim Bakanlığı (MEB). (2011). *Bilişim teknolojileri alanı çerçeve öğretim programı*. [http://www.megep.meb.gov.tr/dokumanlar/Çerçeve Öğretim Programı/BİLİŞİM TEKNOLOJİLERİ\\_ÇÖP\\_10.pdf](http://www.megep.meb.gov.tr/dokumanlar/Çerçeve Öğretim Programı/BİLİŞİM TEKNOLOJİLERİ_ÇÖP_10.pdf) adresinden erişilmiştir.
- Obaidallah, U., al Haek, M., and Cheng, P. C.-H. (2018). A survey on the usage of eye-tracking in computer programming. *ACM Computing Surveys*, 51(1), 1–58. <https://doi.org/10.1145/3145904>
- Peng, F., Li, C., Song, X., Hu, W., and Feng, G. (2016). An eye tracking research on debugging strategies towards different types of bugs. In S. I. Ahamed, C. K. Chang, W. Chu, I. Crnkovic, P.-A. Hsiung, G. Huang, ve J. Yang (Eds.), *2016 IEEE 40th Annual Computer Software and Applications Conference* (ss. 130–134). New York, USA: IEEE. <https://doi.org/10.1109/COMPSAC.2016.57>
- Porkodi, R. (2014). Comparison of filter based feature selection algorithms: An overview. *International Journal of Innovative Research in Technology ve Science*, 2(2), 108–113. <http://ijirts.org/volume2issue2/IJIRTSV2I2034.pdf>
- Robins, A., Haden, P., and Garner, S. (2006). Problem distributions in a CS1 course. In D. Tolhurst ve S. Mann (Eds.), *Conferences in Research and Practice in*

- Information Technology Series* (ss. 165–173). New York, USA: Association for Computing Machinery. <https://dl.acm.org/doi/10.5555/1151869.1151891>
- Scheiter, K., and Eitel, A. (2015). Signals foster multimedia learning by supporting integration of highlighted text and diagram elements. *Learning and Instruction*, 36, 11–26. <https://doi.org/10.1016/j.learninstruc.2014.11.002>
- Scheiter, K., and Eitel, A. (2017). The use of eye tracking as a research and instructional tool in multimedia learning. In C. Was, F. Sansosti, ve B. Morris (Eds.), *Eye-Tracking Technology Applications in Educational Research* (ss. 143–164). Hershey, PA, USA: IGI Global. <https://doi.org/10.4018/978-1-5225-1005-5.ch008>
- Schwonke, R., Berthold, K., and Renkl, A. (2009). How multiple external representations are used and how they can be made more useful. *Applied Cognitive Psychology*, 23, 1227–1243. <https://doi.org/10.1002/acp.1526>
- Sharafi, Z., Marchetto, A., Susi, A., Antoniol, G., and Guéhéneuc, Y.-G. (2013). An empirical study on the efficiency of graphical vs. textual representations in requirements comprehension. In H. Kagdi, D. Poshyvanyk, ve M. di Penta (Eds.), *2013 21st International Conference on Program Comprehension (ICPC)* (ss. 33–42). New York, USA: IEEE. <https://doi.org/10.1109/ICPC.2013.6613831>
- Sharafi, Z., Soh, Z., Guéhéneuc, Y. G., and Antoniol, G. (2012). Women and men - Different but equal: On the impact of identifier style on source code reading. *2012 20th IEEE International Conference on Program Comprehension (ICPC)*, 27–36. <https://doi.org/10.1109/ICPC.2012.6240505>
- Sharif, B., Falcone, M., and Maletic, J. I. (2012). An eye-tracking study on the role of scan time in finding source code defects. In S. N. Spencer (Ed.), *Proceedings of the Symposium on Eye Tracking Research and Applications*, (ss. 381–384). New York, USA: Association for Computing Machinery. <https://doi.org/https://doi.org/10.1145/2168556.2168642>
- Sharma, K., Mangaroska, K., Giannakos, M., and Dillenbourg, P. (2018). Interlacing gaze and actions to explain the debugging process. In J. Kay, ve R. Luckin (Ed.) *13th International Conference of the Learning Sciences* (ss. 640–647). London, UK: International Society of the Learning Sciences. <http://hdl.handle.net/11250/2582513>
- Shojaeizadeh, M., Djamasbi, S., and Trapp, A. C. (2016). Density of gaze point within a fixation and information processing behavior. In G. Goos, J. Hartmanis, ve J. van Leeuwen (Eds.), *Universal Access in Human-Computer Interaction: Methods, Techniques, and Best Practices* (ss. 465–471). Berlin, Germany: Springer. [https://doi.org/10.1007/978-3-319-40250-5\\_44](https://doi.org/10.1007/978-3-319-40250-5_44)
- Turner, R., Falcone, M., Sharif, B., and Lazar, A. (2014). An eye-tracking study assessing the comprehension of C++ and Python source code. In P. Qvarfordt ve



- D. W. Hansen (Eds.), *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14* (ss. 231–234). New York, USA: Association for Computing Machinery. <https://doi.org/10.1145/2578153.2578218>
- van Gog, T., Kester, L., Nievelstein, F., Giesbers, B., ve Paas, F. (2009). Uncovering cognitive processes: Different techniques that can contribute to cognitive load research and instruction. *Computers in Human Behavior*, 25(2), 325–331. <https://doi.org/10.1016/j.chb.2008.12.021>
- van Meeuwen, L. W., van Merriënboer, J. J. G., Jarodzka, H., Brand-Gruwel, S., Kirschner, P. A., ve de Bock, J. J. P. R. (2014). Identification of effective visual problem solving strategies in a complex visual domain. *Learning and Instruction*, 32, 10–21. <https://doi.org/10.1016/j.learninstruc.2014.01.004>
- Yen, C. Z., Wu, P. H., and Lin, C. F. (2012). Analysis of experts' and novices' thinking process in program debugging. In K. C. Li, F. L. Wang, K. S. Yuen, S. K. S. Cheung, ve R. Kwan (Eds.), *Engaging Learners Through Emerging Technologies* (ss. 122–134). Berlin, Germany: Springer. [https://doi.org/10.1007/978-3-642-31398-1\\_12](https://doi.org/10.1007/978-3-642-31398-1_12)

## SUMMARY

*Bugs are negative situations encountered during the programming process. Its prevent the program from running and reduce program quality. Therefore, the elimination of these bugs is important for the functional operation of the program.*

*There are three types of bugs. The first of these, compile-time bug, refers to errors made in the grammar of the programming language. The other type of bug, runtime bug occurs when something goes wrong while the program is running after it has gone through the compilation process. The last type of bug, logical bugs, occurs when there is no problem in compiling and running the program codes and does not give correct outputs. Programmers show different performances and behavior for debugging each type of bug.*

*One of the methods used in recent years by researchers working on the cognitive paradigm to understand the debugging performance of programmers is eye tracking. Thanks to this method, the eye movements of the programmers on the codes can be determined simultaneously and inferences can be made about the cognitive processes of the programmers.*

*The aim of this study is to identify the important eye movements exhibited while debugging different types. For this purpose, answers to the following three research questions were sought.*

- a) Which eye movement metrics are important in compile-time debugging?*
- b) Which eye movement metrics are important in runtime debugging?*
- c) Which eye movement metrics are important in logical debugging?*

*In this research, feature determination methods used in data mining were used. 51 students studying in the Information Technologies Department of vocational and technical high schools participated in the research. "Debugging Performance Test" were used to determine the debugging performance and the Gazepoint GP3 Eye Tracker for obtaining data on the eye movements of the students in the IT department of vocational high schools in the C # programming language. To analyze eye movements, areas of interest on the bug line, other lines of code, and screenshots of the output were identified. The metrics "Time to First Fixation", "Total Fixation Duration", "Fixation Count", "Percentage of Fixation Duration", "Average Fixation Duration" and "Revisit Count" were analyzed for these areas of interest. To examine the significance of these metrics for debugging, the Information Gain and Gini Index were used.*


*Important eye movement metrics in debugging compile-time errors were identified. According to the findings, the total fixation duration and the percentage of fixation duration on the error line, the fixation count, the total fixation duration and the revisit count on screenshots of the output, and the time to first fixation on other lines of code are important in compile-time debugging.*


*In the study, eye movement metrics that are important in runtime debugging were identified. According to the findings, the average fixation duration, the percentage of fixation duration and the time to first fixation on the screenshot, the time to first fixation and the average fixation duration on other lines of code, and the percentage of fixation duration on the bug line are important in runtime debugging.*

*Finally, the important eye movement metrics in the logical debug were analyzed. According to the findings obtained as a result of the analysis, the total fixation duration and the revisit count on the screenshot of the output, the time to first fixation and the revisit count on the error line, and the total fixation duration on other codes are important for logical debugging.*

*As a result of the study, different eye movements were important while debugging different types, and the students exhibited different eye movements. The eye movements of the students reflected the general characteristics of the bug types. Therefore, the characteristics of the bugs should be taken into account when selecting eye movement metrics in studies that examine eye movements in debugging.*

## ORCID

Arif AKÇAY  0000-0001-9103-9469

Arif ALTUN  0000-0003-4060-6157

## Araştırmacıların Katkı Oranı Beyanı

Bu çalışmanın planlanması, yürütülmesi ve yazılı hale getirilmesinde araştırmacılar eşit oranda katkı sağlamıştır.

**Destek ve Teşekkür Beyanı**

Bu çalışma hazırlanması sürecinde TÜBİTAK tarafından “2211-A Genel Yurt İçi Doktora Burs Programı” kapsamında verilen destek için teşekkür ederiz.

**Çatışma Beyanı**

Araştırmacıların, araştırma ile ilgili diğer kişi ve kurumlarla herhangi bir kişisel ve finansal çıkar çatışması yoktur.

**Etik Kurul Beyanı**

Araştırma, Hacettepe Üniversitesi Etik Komisyonunun 10.09.2018 tarih ve 35853172-300 sayılı onayı ile yürütülmüştür. Etik Komisyon Onay Belgesi Ek 1’de sunulmuştur.

**Ek 1. Etik Komisyon Onay Belgesi**



T.C.  
HACETTEPE ÜNİVERSİTESİ  
Rektörlük



Sayı : 35853172-300  
Konu : Arif AKÇAY Hk.(Etik Komisyon)

**EĞİTİM BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE**

İlgi : 08.08.2018 tarihli ve 51944218-300/00000186064 sayılı yazınız

Enstitünüz Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı Doktora programı öğrencilerinden **Arif AKÇAY**'ın, **Prof. Dr. Arif ALTUN** danışmanlığında yürüttüğü "**Dikkat Ağlarının ve Görsel Dikkatlerin Hata Ayıklama Performansları Üzerine Etkisinin İncelenmesi**" başlıklı tez çalışması, Üniversitemiz Senatosu Etik Komisyonunun **4 Eylül 2018** tarihinde yapmış olduğu toplantıda incelenmiş olup,etik açıdan uygun görülmüştür.

Bilgilerinizi ve gereğini saygılarımla rica ederim.

e-İmzalıdır  
Prof. Dr. A. Haluk ÖZEN  
Rektör

Evrakın elektronik imzalı suretine <https://belgedogrulama.hacettepe.edu.tr> adresinden ae85b809-544e-4bc9-a271-5ff619f046c9 kodu ile erişebilirsiniz.  
Bu belge 5070 sayılı Elektronik İmza Kanunu'na uygun olarak Güvenli Elektronik İmza ile imzalanmıştır.

Hacettepe Üniversitesi Rektörlük 06100 Sıhhiye-Ankara  
Telefon:0 (312) 305 3001-3002 Faks:0 (312) 311 9992 E-posta:yazimd@hacettepe.edu.tr İnternet  
Adresi: www.hacettepe.edu.tr

Duygu Didem İLFPİ

