

# Geçmiş Dönük Hata Tespit Oranlarının Zamanla Değişen Etkisine Göre Regresyon Testlerinin Önceliklendirilmesi

Deniz Kurt<sup>1</sup>, Hasan Sözer<sup>2</sup>

<sup>1</sup> VESTEK AR-GE, İstanbul, Türkiye  
deniz.kurt@vestel.com.tr

<sup>2</sup> Özyeğin Üniversitesi, İstanbul, Türkiye  
hasan.sozer@ozyegin.edu.tr

**Özet.** Büyük ölçekli yazılım sistemlerinin regresyon testleri masraflıdır. Kaynak kısıtları sebebiyle genellikle tüm test durumlarını değerlendirmek mümkün olmamaktadır. Dolayısıyla, hataları mümkün olan en kısa sürede tespit edecek şekilde test durumlarını önceliklendirmek gerekmektedir. Bu çalışmada, test durumlarını önceliklendirmek için geçmişte tespit edilen hata miktarlarının analizine dayanan bir yöntem öneriyoruz. Bu analizin ne derecede yakın geçmişe veya uzak geçmişe odaklanacağı, yöntemin bir parametresi olarak uyarlanabilmektedir. Bir dijital TV yazılımı için regresyon testlerini çeşitli parametre değerleri ile önceliklendirerek, elde edilen sıralamaların etkinliklerini karşılaştırdık. Yöntemimizin mevcut sıralamaya göre hata tespit oranı açısından daha etkili sıralamalar elde etmemizi sağladığını gördük.

## 1 Giriş

Yazılım sistemleri değişen gereksinimler neticesinde idame süresince değişikliklere tabidir. Yapılan değişiklikler sonrasında sistemde oluşabilecek yeni hataları tespit etmek amacıyla regresyon testleri gerçekleştirilmektedir. Büyük ölçekli yazılım sistemlerinde test edilmesi gereken çok fazla durum bulunmaktadır. Kaynaklar kısıtlı olduğundan tüm test durumlarını değerlendirmek her zaman mümkün olmamaktadır. Bu sebeple test durumlarını önceliklendirmek ve önceliklerine göre sıralamak gerekmektedir. Böylece, test durumları elde edilen sıralamaya göre kaynaklar yeterli olduğu sürece değerlendirilebilirler.

Bu çalışmada, test durumlarını önceliklendirmek için uyarlanabilir bir yöntem öneriyoruz. Yöntemin amacı, hataları mümkün olan en kısa sürede tespit edecek şekilde test durumlarını sıralamaktır. Yöntemimizde her test durumu, geçmişte tespit edebildiği hata miktarına göre öncelik kazanmaktadır. Önceliğin belirlenmesi için ne derecede yakın geçmişe veya uzak geçmişe odaklanılacağı bir parametre ile uyarlanmaktadır.

Yöntemimizi değerlendirmek üzere büyük ölçekli gerçek bir yazılım sistemi ile vaka analizi gerçekleştirdik. Bir dijital TV (DTV) sisteminin yazılımının regresyon testleri için oluşturulmuş test durumlarını çeşitli parametre değerleri ile

önceliklendirdik ve sıraladık. Elde edilen sıralamaların etkinliklerini iki farklı sıralama ile karşılaştırdık: birincisi, yöntemi uygulamadan önce kullanılmakta olan mevcut sıralama; ikincisi, oluşturduğumuz bir kahine göre belirlenmiş olan en iyi sıralama. Hata tespit oranı açısından yöntemimizin mevcut sıralamaya göre daha etkili olduğunu gösterdik. Mümkün olabilecek en iyi sıralama ile karşılaştırıldığında ise %10 ile %26 arasında bir etkinlik farkı gözlemledik.

Literatürdeki diğer çalışmalar ile kıyaslandığında, bu bildirinin iki temel katkısı bulunmaktadır:

- Testleri önceliklendirmek için ne derecede yakın geçmişe veya uzak geçmişe odaklanılacağına bir parametre ile uyarlanabilir kılındığı özgün bir yöntem önerilmektedir.
- Gerçek bir endüstriyel sistem üzerinde vaka analizi yapılarak yöntemin başarısı değerlendirilmekte ve farklı parametre değerlerinin etkileri analiz edilmektedir.

Bildirinin organizasyonu: Bir sonraki bölümde, test durumlarının önceliklendirilmesi sonucunda ortaya çıkan sıralamanın etkinliğini değerlendirmek üzere kullanılan kriterleri özetliyoruz. Ayrıca, bu çalışmada yöntemimizi değerlendirmek üzere kullandığımız bir metriği tanıtlıyoruz. 3. Bölüm'de yöntemimizi açıklıyoruz. 4. Bölüm'de deney sonuçlarını paylaşarak yöntemimizi değerlendiriyoruz. 5. Bölüm'de literatürdeki ilgili çalışmaları özetliyoruz. Son olarak, 6. Bölüm'de temel çıkarımları özetleyerek ileriye dönük yapılacak çalışmaları listeliyoruz.

## 2 Test Durumu Önceliklendirme Kriterleri ve Hata Tespit Oranı

Test durumlarını önceliklendirmek üzere geliştirilen teknikler, regresyon testlerinin etkinliklerini çeşitli amaçlar doğrultusunda geliştirmeyi hedeflemektedirler [2]. Yaygın olarak benimsenen bazı amaçlar aşağıda listelenmiştir:

- Regresyon testleri sırasında hataların erken tespit edilme olasılığını artırmak.
- Kritik olan hataların tümünün tespit edilmesini sağlamak.
- Belirli kod değişikliklerine ilişkin hataları ortaya çıkarmak.
- Regresyon testleri sırasında kod kapsamını daha hızlı artırmak.
- Test edilen sistemin güvenilirliğini hızlı bir şekilde sınamak.

Biz bu çalışmada, yukarıda listelenen ilk amaca odaklanıyoruz: hataların erken tespit edilme olasılığını artırmak. Böylece, regresyon testleri sırasında erken bir geribesleme almak mümkün olacaktır. Bu amaç kapsamında bir tekniğin ne derecede etkin olduğunu ölçebilmek için çeşitli metrikler geliştirilmiştir. Biz çalışmamızda hata tespit oranını ölçen APFD (Average of the Percentage of Faults Detected) metriğini [1] kullandık. Bu metrik, test durumları tarafından tespit edilen hataların sayısının ağırlıklı ortalamasını hesaplamaktadır. Bir test durumuna atanan ağırlık değeri, bu test durumundan önce tespit edilen hataların oranı ile doğru orantılı olarak artmaktadır. APFD değeri 0 ile 100 arasında olup, yüksek değerler yüksek hata tespit oranına

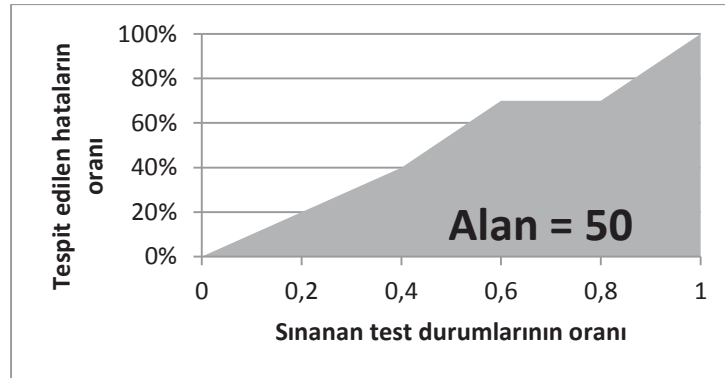
işaret ederler. Aşağıda APFD değerinin hesaplanmasına ilişkin bir örneğe [1] yer verilmiştir.

Bir yazılım sisteminde 10 adet hata olduğunu ve test kümesinde 5 adet test durumu olduğunu varsayalım. Bu test durumlarını A, B, C, D ve E olarak adlandıralım ve bu test durumlarının hepsinin birlikte tüm hataları tespit edebildiğini varsayalım. Tablo 1 her test durumu tarafından tespit edilebilen hataları göstermektedir. Örneğin, test durumu A sadece birinci ve beşinci hataları tespit edebilmektedir. Diğer yandan, test durumu C ilk 7 hatayı tek başına tespit edebilmektedir.

**Tablo 1. Test durumları tarafından tespit edilen hatalar.**

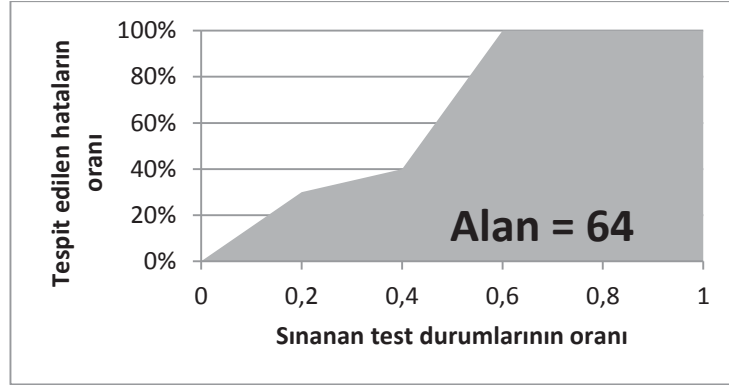
|                |   | Hatalar |   |   |   |   |   |   |   |   |    |   |  |
|----------------|---|---------|---|---|---|---|---|---|---|---|----|---|--|
|                |   | 1       | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |   |  |
| Test Durumları | A | ✓       |   |   |   | ✓ |   |   |   |   |    |   |  |
|                | B | ✓       |   |   |   | ✓ | ✓ | ✓ |   |   |    |   |  |
|                | C | ✓       | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |   |   |    |   |  |
|                | D |         |   |   |   | ✓ |   |   |   |   |    |   |  |
|                | E |         |   |   |   |   |   |   |   | ✓ | ✓  | ✓ |  |

Bu şartlar altında, eğer test durumlarını A-B-C-D-E şeklinde sıralarsak, hata tespit oranı Şekil 1’de gösterildiği şekilde artacaktır. Şekil 1’de x eksenini sınanmış olan test durumlarının sayısının toplam test durumu sayısına oranını, y eksenini ise tespit edilen hataların sayısının toplam hata sayısına oranını göstermektedir. Eğri altında kalan alan ise tüm test durumu kümesi için tespit edilen hataların ağırlıklı yüzdesini temsil etmektedir. Bu alan APFD değerine eşittir ve örnekte söz konusu sıralama için değeri 50 olarak hesaplanmıştır.



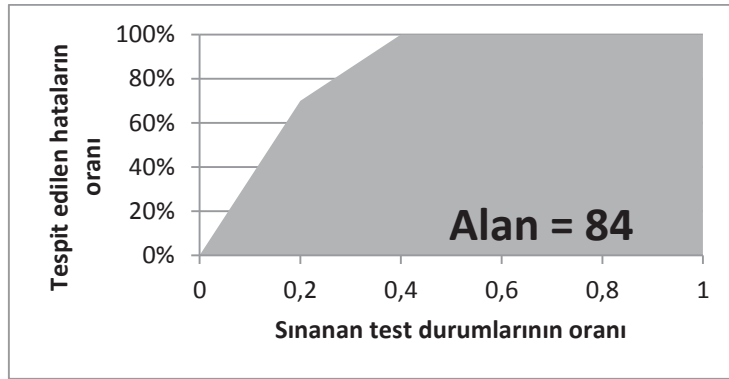
**Şekil 1.** A-B-C-D-E test durumu sıralaması için APFD değeri.

Eğer test durumlarının sıralamasını E-D-C-B-A şeklinde değiştirirsek, hata tespit oranındaki artış Şekil 2’de gösterildiği gibi olacaktır. Bu sıralama ile, test durumlarının %60 kadarı değerlendirildikten sonra tespit edilemeyen hata kalmamaktadır. Bu yüksek hata tespit oranı APFD değerine de yansımaktadır. Bir önceki sıralama için APFD değeri 50 iken, bu sıralama için 64 olmuştur.



Şekil 2. E-D-C-B-A test durumu sıralaması için APFD değeri.

Son olarak üçüncü bir sıralama alternatifini değerlendirelim: C-E-B-A-D. Bu alternatife ilişkin hata tespit oranları Şekil 3’de gösterilmiştir. Esasen bu sıralama, hata tespit oranı açısından mümkün olan en iyi sıralamadır. APFD değeri 84 olarak hesaplanmaktadır.



Şekil 3. C-E-B-A-D test durumu sıralaması için APFD değeri.

Test edilen sistemin özelliklerine göre her zaman en iyi sıralamayı tayin etmek mümkün olmamaktadır. Dolayısıyla test durumlarını önceliklendirmek için geliştirilen teknikler genelde sezgisel yöntemler kullanmaktadır [2]. Biz de bu çalışmada sezgisel bir yöntem öneriyoruz. Bir sonraki bölümde bu yöntemi ve daha sonra da bu yöntem ile elde ettiğimiz sonuçları açıklıyoruz.

### 3 Yöntem

Yöntemimizde test durumlarını önceliklendirmek için geçmişte gözlemlenen hata tespit oranları değerlendirilmektedir. Her bir test durumu için hata tespit oranı aşağıdaki gibi hesaplanmaktadır.

$$hto(i, t) = \frac{t \text{ zamanında, test durumu } i \text{ ile tespit edilen hata sayısı}}{\text{test durumu } i \text{ büyüklüğü}}$$

Burada regresyon testlerinin farklı  $t$  ( $t > 0$ ) zamanlarında gerçekleştirildiği varsayılmaktadır. Büyük ölçekli test kümelerinde, test durumları genelde hiyerarşik bir yapıda gruplanmaktadır. Dolayısıyla, tanımladığımız her test durumu  $i$  ( $i > 0$ ), esasen belirli bir işleyle ilgili olan bir test durumu grubunu ifade etmekte ve birçok farklı test durumu içermektedir. Bir test durumunun büyüklüğü, içerdiği farklı test durumlarının sayısı olarak ölçülmektedir.

Hipotez olarak, yakın geçmişte hata oranı yüksek olan test durumlarına öncelik verilmesinin, hata tespit oranını artıracakları öngörülmektedir. Bu hipotezin aksine, bir önceki regresyon testi döngüsünde hata tespit oranları yüksek olan test durumlarının (bu hataların ayıklanmasına yönelik kod güncellemeleri nedeniyle) bir sonraki regresyon testi döngüsünde düşük hata tespit oranlarına sahip olmaları beklenebilir. Fakat söz konusu test durumları, aslında aynı işlev ile ilgili birer test durumu grubu olup, birden fazla test durumu içermektedir. Yüksek hata oranları, ilgili işlevin güvenilirlik açısından henüz olgunlaşmadığının işaretidir. Ayrıca, hataların düzeltilmesi amacıyla yapılan değişiklikler genelde aynı işlev kapsamında başka hataların ortaya çıkmasına sebep olmaktadır. Sezgisel yöntemimiz bu varsayımlara dayanmaktadır.

Her test durumu için farklı zamanlarda gerçekleştirilen regresyon testleri sırasında farklı hata oranları elde edilebilir. Dolayısıyla bir test durumunun önceliği zaman içerisinde değişebilir. Herhangi bir  $t$  zamanı için bir test durumunun önceliği aşağıdaki gibi hesaplanmaktadır.

$$r(i, t) = \begin{cases} 0, & t = 1 \\ hto(i, t - 1), & t = 2 \\ \alpha \times r(i, t - 1) + (1 - \alpha) \times hto(i, t - 1), & t > 2 \wedge 0 \leq \alpha \leq 1 \end{cases}$$

İlk regresyon testi sırasında ( $t = 1$ ) daha önceden hiç bir test durumu sınanmamış olduğundan hata tespit oranları bilinmemektedir. Dolayısıyla her test durumu  $i$  için öncelik 0 olarak belirlenmiştir. İkinci regresyon testi sırasında ( $t = 2$ ) ilk regresyon testleri sırasında ölçülen hata tespit oranı, ilgili test durumu için öncelik olarak belirlenir. Daha sonraki regresyon testlerinde ( $t > 2$ ) ise bir önceki testler sırasında ölçülen hata oranı (yani,  $hto(i, t - 1)$ ) ile daha önceden belirlenmiş olan öncelik değerinin (yani,  $r(i, t - 1)$ ) lineer bir kombinasyonu öncelik değeri olarak belirlenir. Burada, 0 ile 1 arasında değerler alan  $\alpha$  parametresi, daha önceden belirlenmiş öncelik değeri ile en son hesaplanmış hata tespit oranı için bir ağırlık tayin eder. Bu değer ne

kadar küçük olursa, önceden hesaplanmış öncelik değerinin etkisi o kadar küçük olur. Yüksek  $\alpha$  değeri ise önceki hata oranı hesaplamalarına ilişkin daha kalıcı bir hafıza göz önünde bulundurulmasını sağlar. Bu değer 0 olduğunda, bir test durumunun önceliği her zaman bir önceki testlerde hesaplanan hata tespit oranına eşit olmaktadır. Daha yüksek  $\alpha$  değerlerinde, sadece bir önceki değil, daha önceden hesaplanan hata tespit oranları da öncelik hesaplamasına etki etmektedir. Bu etki zamanla azalmaktadır.

Bir sonraki bölümde yöntemimizi değerlendirmek üzere gerçekleştirdiğimiz vaka analizini açıklıyor ve elde ettiğimiz sonuçları paylaşıyoruz.

## 4 Değerlendirme

Yöntemimizi değerlendirmek üzere Vestel<sup>1</sup> tarafından geliştirilen DTV ürünlerinde kullanılan bir yazılım birimi üzerinde vaka analizi gerçekleştirdik. Aşağıda ilk olarak vaka analizimiz ile detayları paylaşıyoruz. Daha sonra elde ettiğimiz sonuçları sunuyor ve tartışıyoruz.

### 4.1 Vaka Analizi: DTV Regresyon Testleri

Vestel regresyon testlerini otomatik olarak çalıştırmak üzere Python tabanlı bir otomasyon sistemi kullanmaktadır. Bu sistem DTV sistemine uzaktan kumanda sinyalleri göndererek bir kullanıcı gibi davranmakta, bu sırada DTV ekran görüntülerini referans görüntüler ile karşılaştırmaktadır. Karşılaştırma sırasında tespit edilen tutarsızlıklar birer hata olarak raporlanmaktadır.

Test otomasyonu Buildbot [8] çerçevesini kullanmaktadır. Otomatik olarak sınanan test durum kümesi binlerce test durumu içermektedir. Vaka analizi için kullandığımız test durumları 94 farklı test durum grubu altında toplanmıştır ( $1 \leq i \leq 94$ ). Geceleri ve/veya haftasonları gerçekleştirilen testlerin tamamlanması birkaç saat veya birkaç gün sürebilmektedir. Bu çalışma kapsamındaki değerlendirmemizde, ardışık olarak gerçekleştirilmiş 5 regresyon testi sonuçlarını kullandık ( $1 \leq t \leq 5$ ).

Yöntemimizi farklı  $\alpha$  değerleri ile uygulayarak elde edilen APFD değerlerini karşılaştırdık. Ayrıca bu değerleri aşağıdaki iki farklı sıralama ile elde edilen APFD değerleri ile karşılaştırdık:

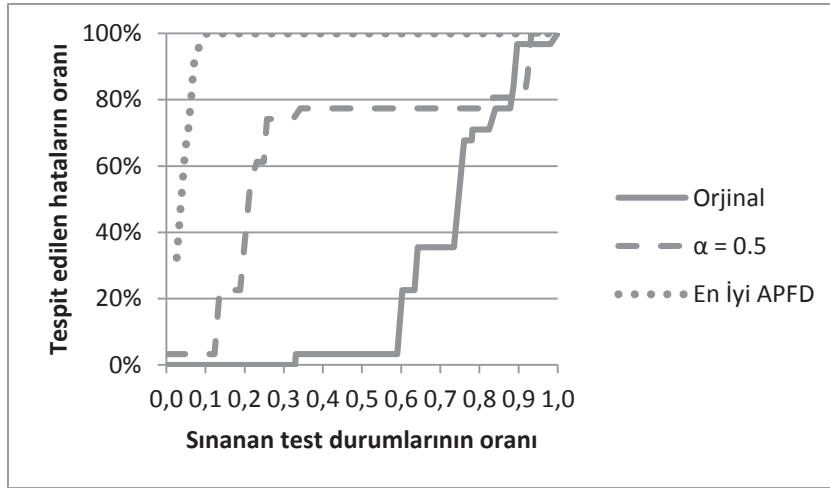
- Önceliklendirme yapılmaksızın, her regresyon testinde olduğu gibi kullanılan orjinal sıralama.
- Regresyon testlerinin gerçek sonuçları temel alınarak oluşturulan kahine göre belirlenmiş en iyi sıralama (En yüksek APFD değerine sahip sıralama).

---

<sup>1</sup> <http://www.vestel.com.tr>

## 4.2 Sonular

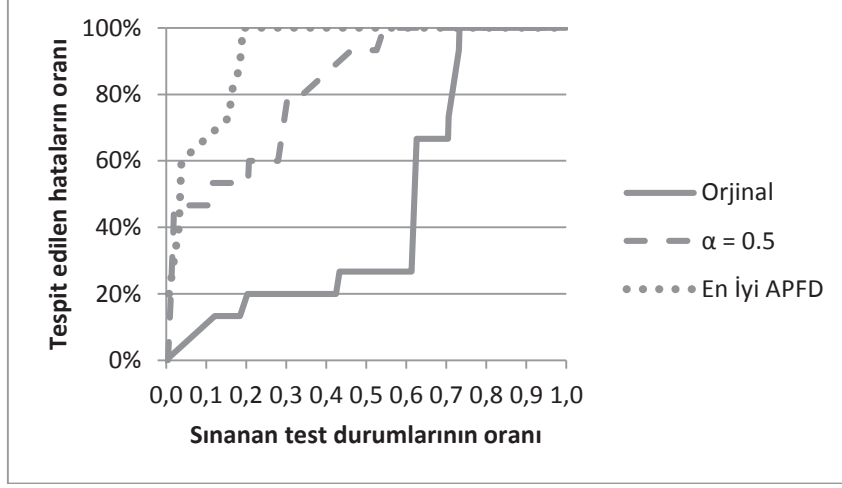
Őekil 4, gerekleŐtirilen ikinci regresyon testi iin ( $t = 2$ ) u farklı sıralamanın hata tespit oranı aısından etkinliĐini gstermektedir. Düz izgiler orjinal sıralamayı temsil etmektedir. Bu sıralama iin test durumlarının %60 kadarı deĐerlendirilmesine karŐın tespit edilen hata oranının %5 altında olduĐu grlmektedir. Őekildeki kesikli izgiler yntemimiz ile elde edilen sonuları gstermektedir. Burada  $\alpha$  deĐeri 0,5 olarak belirlenmiŐ olsa da,  $t = 2$  olduĐunda bu parametrenin sonulara bir etkisi yoktur. Yntemin sıralamayı geliŐtirdiĐi gze arpmaktadır. Test durumlarının %30 kadarı deĐerlendirildikten sonra tespit edilen hata oranı %80 deĐerine ulaŐmıŐtır. Noktalı izgiler en iyi sıralamayı gstermektedir. Bu sıralama iin APFD deĐeri 96 olarak hesaplanmıŐtır.



Őekil 4. İkinci regresyon testi ( $t = 2$ ) iin hata tespit oranı.

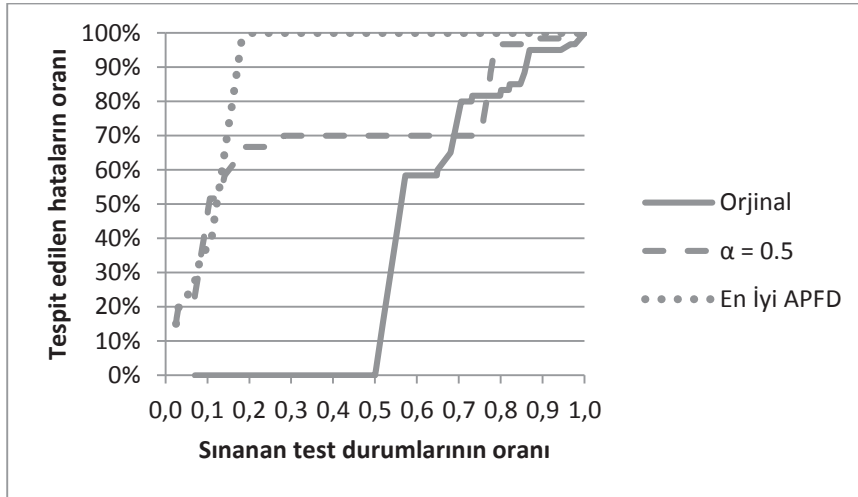
Üüncü regresyon testlerinin sonularını Őekil 5 gstermektedir. Burada orjinal sıralama ile test durumlarının %70 kadarı deĐerlendirildikten sonra tüm hataların tespit edilmiŐ olduĐu grlmektedir. Fakat yine de APFD deĐeri daha da geliŐtirilebilir. Yntemimiz,  $\alpha$  deĐeri 0,5 iken daha iyi sonu vermiŐtir; test durumlarının %55 kadarının deĐerlendirilmesi, tüm hataların tespiti iin yeterli olmuŐtur. En iyi APFD deĐerine sahip olan sıralamada ise test durumlarının %19 kadar bir blümünün deĐerlendirilmesi ile tüm hatalar tespit edilebilmektedir. Üüncü regresyon testlerinin sonucunda toplamda sadece 15 adet hata tespit edilmiŐtir. Toplamda bu kadar az sayıda hatanın tespit edilmesi, az oranda test durumu ile hızlı bir Őekilde tüm hataların tespit edilebilmesini mümkün kılmıŐtır.

Őekil 6, drdüncü regresyon test sonularını ( $t = 4$ ) gstermektedir. İkinci test sonularında da olduĐu gibi, orjinal sıralama ile test durumlarının yarısı deĐerlendirilmesine karŐın hataların hibiri tespit edilememiŐtir.



Şekil 5. Üçüncü regresyon testi ( $t = 3$ ) için hata tespit oranı.

Bu sonuçlar da, test durumlarının sabit bir şekilde sıralanmasının hata tespit oranının çok düşük olmasına sebep olduğunu göstermektedir. Hata tespit oranının düşük olması ise kaynakların verimsiz kullanılmasına yol açmaktadır. Özellikle regresyon testlerinin gerçekleştirilmesi için kaynakların kısıtlı olduğu durumlarda, tüm test durumlarını değerlendirmek mümkün olmadığından, hataların tümü tespit edilememiş olacaktır.

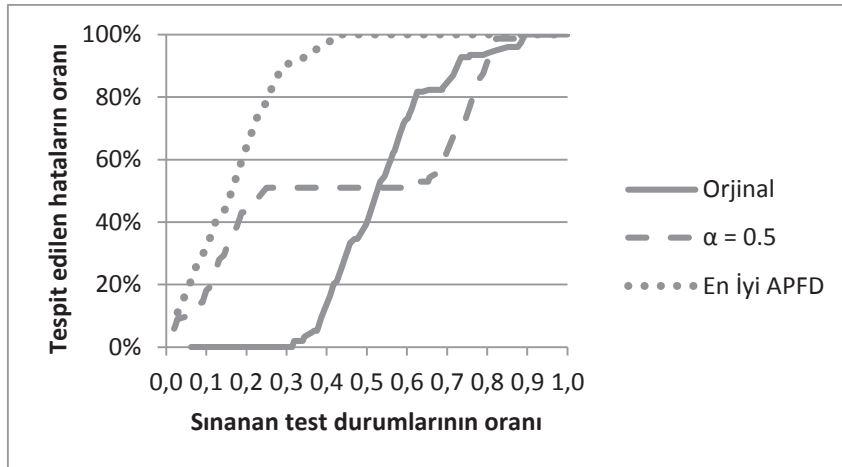


Şekil 6. Dördüncü regresyon testi ( $t = 4$ ) için hata tespit oranı.



Üçüncü ve dördüncü regresyon testlerine ilişkin sonuçların gösterildiği Şekil 5 ve Şekil 6'da, sezgisel yöntemle elde edilen sıralamanın, en iyi APFD değerine ilişkin sıralamaya göre, ilk birkaç test için daha yüksek hata tespit oranına ulaştığı gözlemlenmektedir. Bu istisnai durumun sebebi, test gruplarının içerdiği test durumu sayıları arasındaki yüksek fark ve en iyi APFD değerine ulaşmak için test sıralamasının belirlenme şeklidir. Bu sıralama, testlerin gerçekte tespit ettikleri hata sayılarının azalan sıralaması ile belirlenmiştir. En iyi APFD değeri için belirlenen sıralamadaki ilk üç test grubu incelendiğinde, ikinci test grubunun 15, diğer test gruplarının ise 4 adet test durumu içerdiği görülmektedir. İkinci test grubu ile tespit edilen hata sayısı fazla olsa da, test durumlarının sayısı da görece fazla olduğundan, hata tespit oranı düşük çıkmıştır. Diğer test grupları için tespit edilen hata sayıları ile hata tespit oranlarının tutarlı olduğu görülmüştür.

Beşinci ve son regresyon testleri ( $t = 5$ ) sonuçları Şekil 7'de gösterilmiştir. Bu sonuçlarda görülebildiği üzere, en iyi sıralama ile test durumlarının %42 kadar bir bölümünün değerlendirilmesi, tüm hataların tespit edilebilmesini mümkün kılmaktadır. Bu oranların ikinci, üçüncü ve dördüncü regresyon testleri için sırasıyla %10, %19 ve %21 olduğu görülmüştür.



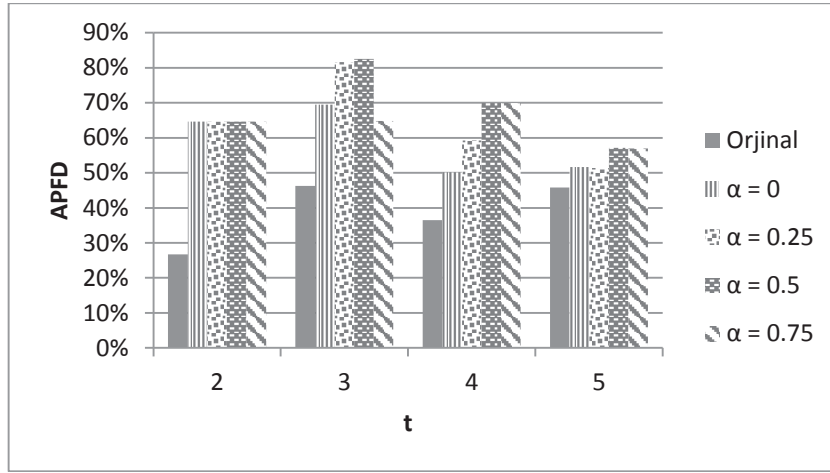
Şekil 7. Beşinci regresyon testi ( $t = 5$ ) için hata tespit oranı.

Hata miktarı arttıkça test durumlarının önceliklendirilerek sıralanması ile elde edilen fayda azalmaktadır. Örneğin, beşinci regresyon testlerinde  $\alpha$  değeri 0,5 iken elde edilen sıralama APFD değerini orjinal sıralamaya kıyasla %11 kadar artırmıştır. Bu artış, görece daha az hata tespit edilen ikinci, üçüncü ve dördüncü testler için sırasıyla %38, %37 ve %33 olarak görülmüştür. Şu da gözden kaçmamalıdır ki, hataların görece daha fazla görülmüş olduğu beşinci testlerde bile orjinal sıralama ile hataların hiçbiri, test durumlarının %31 kadar değerlendirilmiş olmasına karşın tespit edilememiştir. Sonuç olarak, her regresyon testi için sabit olarak kullanılan orjinal

sıralamanın, hataları hızlı bir şekilde tespit edebilmek için uygun bir sıralama olmadığı görülmüştür.

Şu ana kadar sunduğumuz sonuçlar  $\alpha$  değerinin sadece 0,5 olarak belirlendiği durumlarda elde ettiğimiz sonuçlardır. Aşağıda, farklı  $\alpha$  değerlerinin hata tespit oranına yaptıkları etkileri değerlendiriyoruz.

Farklı  $\alpha$  değerleri ile elde edilen sonuçları karşılaştırmak için yine APFD metriğini kullandık. Toplamda dört farklı  $\alpha$  değeri ile denemeler yaptık: 0, 0,25, 0,5 ve 0,75. Tüm regresyon testlerini ( $t = 2, 3, 4, 5$ ) bu dört farklı  $\alpha$  değeri ile elde ettiğimiz sıralamaya göre değerlendirdik. Sonuçlar Şekil 8’de sunulmaktadır. Orjinal sıralama ile elde edilen sonuçlar da burada gösterilmektedir.



Şekil 8. Çeşitli  $\alpha$  değerleri ile elde edilen APFD değerleri.

Sonuçlara göre tüm  $\alpha$  değerleri ile elde edilen sonuçlar orjinal sıralamaya göre daha iyi APFD değerleri elde edilmesini sağlamıştır. Burada ikinci regresyon testi için ( $t = 2$ )  $\alpha$  parametresi etkisiz olduğundan, farklı  $\alpha$  değerlerinin sonuçlara bir etkisi olmamıştır. Diğer testlerde ise  $\alpha$  değeri 0,5 iken en iyi sonuçlar elde edilmiştir.

Üçüncü testler için en iyi sonucun  $\alpha$  değeri 0,5 iken elde edildiği açıkça gözlemlenmektedir. Diğer testler için  $\alpha$  değeri 0,5 iken ve  $\alpha$  değeri 0,75 iken elde edilen sonuçlar, birbirlerine oldukça yakın görünmektedir. Dördüncü ve beşinci testler için  $\alpha$  değeri 0,5 iken APFD değeri sırasıyla 70 ve 57,1 olarak hesaplanmıştır. Aynı testler için  $\alpha$  değeri 0,75 iken APFD değeri sırasıyla 69,9 ve 56,9 olarak hesaplanmıştır. Şekil 8’de görüldüğü gibi, diğer  $\alpha$  değerleri ile daha düşük APFD değerleri elde edilmiştir.

## 5 İlgili Çalışmalar

Test durumlarını önceliklendirmek üzere geliştirilen birçok teknik ve yöntem özetlenmiş, deneysel çalışmalarla değerlendirilmiş ve birbirleri ile karşılaştırılmışlardır [1][3]. Bu çalışmalarda, hata tespit oranlarını karşılaştırmak için metrik olarak APFD kullanılmıştır [1].

Ayrıca test durumlarını önceliklendirmek üzere odaklanılan sistemlerin özellikleri, geliştirilen tekniklerin hangi soyutlama seviyesinde (kaynak kod satırı seviyesi, mimari bileşen seviyesi, fonksiyon seviyesi, vs.) daha etkili olduklarına ilişkin değerlendirmeler de yapılmıştır [2]. Çeşitli soyutlama seviyeleri için kayıp-kazanç analizleri yapılarak test durumu önceliklendirme tekniklerinin etkinlikleri değerlendirilmiştir.

Diğer deneysel çalışmalarda [6][7], test durumlarının kaynak ihtiyacı ve tespit edilen hataların ne derecede kritik oldukları göz önünde bulundurularak en kritik hataların mümkün olan en az kaynak tüketimi ile tespit edilmesi üzerine çalışmalar yapılmıştır.

Biz bu çalışmada hata tespit oranını temel etkinlik ölçütü olarak kabul ediyoruz. Test durumlarını önceliklendirmek için geçmişte tespit edilen hataların analiz edilmesi daha önceden literatürde önerilmiştir [1][2][3][4][5]. Fakat bu analiz sırasında ne derecede yakın geçmişe veya uzak geçmişe odaklanması gerektiği uyarlanabilir bir özellik olarak tartışılmamıştır. Bu özelliğin hata tespit oranına ne derecede etki ettiği deneysel çalışmalarla değerlendirilmemiştir. Ayrıca bilgimiz dahilinde, tüketici elektroniği alanındaki gömülü yazılımlar için test durumlarını önceliklendirme konusunda deneysel bir çalışma literatürde yer almamaktadır.

## 6 Sonuç

Bu çalışmada test durumlarını önceliklendirerek sıralamak üzere uyarlanabilir bir yöntem önerdik. Geçmişte tespit edilen hataların miktarlarını kullanarak önceliklendirme yapan bu yöntemi bir dijital TV yazılımının regresyon testlerini sıralamak için değerlendirdik. Elde ettiğimiz sonuçlar neticesinde, yöntemimizin hata tespit oranını artırmak açısından etkili olduğunu, önceliklendirme yapılmaksızın kullanılan sabit bir sıralamanın kaynakların verimsiz kullanılmasına yol açtığını gördük. Özellikle de kaynakların kısıtlı olduğu, tüm test durumlarının değerlendirilemeyeceği regresyon testleri için kaynakların verimli kullanımı ve hataların hızlı tespiti önem teşkil etmektedir.

İleride bu çalışmayı farklı yönlerde genişletmeyi düşünüyoruz. Önceliklendirme ve sıralama için sadece geçmişte tespit edilmiş hatalar değil, aynı zamanda test edilen yazılım biriminin özellikleri, kullanıcıların profilleri ve test kapsamındaki fonksiyonların özellikleri de dikkate alınabilir. Bunu yanı sıra, tespit edilen hataların ne derecede kritik olduğu bilgisinin de kullanılması mümkündür.

## Referanslar

- [1] G. Rothermel, R. Untch, C. Chu and M. Harrold, "Test Case Prioritization: An Empirical Study", In Proceedings of the IEEE International Conference on Software Maintenance, pages 179-188, 1999.
- [2] G. Rothermel, R. Untch, M. Harrold, "Prioritizing Test Cases for Regression Testing," IEEE Transactions on Software Engineering, Vol. 27, No. 10, pages 929-948, 2001.
- [3] S. Elbaum, A.G. Malishevsky, G. Rothermel, "Test Case Prioritization: A Family of Empirical Studies", IEEE Transactions on Software Engineering, Vol. 28, No. 2, pages 159- 182, 2002.
- [4] J.-M. Kim, A. Porter, "A History-based Test Prioritization Technique for Regression Testing in Resource Constrained Environments", In Proceedings of the 24th International Conference on Software Engineering, pages 119-129, 2002.
- [5] C.T. Lin, C.D. Chen, C.S. Tsai, G.M. Kapfhammer, "History-based Test Case Prioritization with Software Version Awareness", In Proceedings of the 18th International Conference on Engineering of Complex Computer Systems, pages 171-172, 2013.
- [6] S. Elbaum, A.G. Malishevsky, G. Rothermel, "Incorporating Varying Test Costs and Fault Severities into Test Case Prioritization", In Proceedings of the 23rd International Conference on Software Engineering, pages 392-338, 2001.
- [7] S. Elbaum, G. Rothermel, S. Kanduri, A.G. Malishevsky, "Selecting a Cost-Effective Test Case Prioritization Technique", Software Quality Journal, Vol. 12, No. 3, pages 185-210, 2004.
- [8] Buildbot, The Continuous Integration Framework, Available [online] <http://buildbot.net>, 2014.