

# Makine Öğrenmesi ile Mobil Uygulama Sınıflandırılması ve Otomatik Keşif Testi

Mehmet Çağrı Çalpur, Sevgi Arca, Tansu Çağla Çalpur ve Cemal Yılmaz

Mühendislik ve Doğa Bilimleri Fakültesi  
Sabancı Üniversitesi, İstanbul, Türkiye  
{mehmetcagri, cyilmaz}@sabanciuniv.edu

**Özet.** Uygulamaların kullanıldığı iş alanı hakkındaki bilgi yazılım testlerinin gerçekleştirilebilmesi için önemli bir kaynaktır. Bu nedenle bir yazılımın iş alanının tanımlanabilmesi ve bu iş alanında yerine getirilen görevlerin bulunması, otomasyonun yapılabilmesinin temelindedir. Bu bilgiler ile testlerde kullanılacak veri seti de otomatik olarak oluşturulabilir ve tüm test sürecinin otomatize edilmesi mümkün olabilir. Bu çalışmada uygulamaların benzerlik derecelerinin bulunması için makine öğrenmesi tekniklerini kullanıyoruz. Aynı iş alanı için geliştirilen uygulamaların benzerlik oranının yüksek olması gerektiği varsayımından yola çıkarak, aynı iş alanındaki uygulamaların, o alanın içindeki benzer görevleri yerine getirmekle yükümlü olacağı öngörülebilir. Bu hipotez doğrultusunda, belli bir iş alanı için genelleştirilmiş bir sonlu durum makinesi modeli oluşturulabilmesi durumunda; aynı iş alanındaki uygulamaların yerine getirdiği görevlerin bu modeldeki belirli durum ve geçiş kümeleri ile eşleştirilmesi mümkün olabilir. Bir iş alanı ile ilgili uygulamaların genel modelindeki durum ve geçiş kümeleri eşleşen uygulamaların test edilmesi için, daha önceden bu kümeyle eşleştirilmiş uygulamalar için hazırlanmış olan test betiklerinin ve veri setinin yeniden kullanılması mümkün olabilecektir. Bu çalışmada iki yeni kavramı öne sürüyoruz. *Model Giydirme* ve *Otomatik Keşif Testi*. Model giydirme kavramı bir uygulamayı, o uygulamanın iş alanı için üretilmiş bir modele eşleştirme işlemini tanımlamaktadır. Otomatik Keşif Testi ise bir iş alanının bilgisi ve bu alandaki uygulamaların test edilmesi ile elde edilen tecrübenin uygulama testinde yeniden kullanılmasıdır.

**Anahtar Kelimeler:** Keşif Testi, Model Bazlı Test, Yazılım İş Alanı Analizi, Uygulama Metin Madenciliği, Otomatik Test, İş Alanı Sınıflandırması

# Mobile Application Classification using Machine Learning and Automated Exploratory Testing

Mehmet Cagri Calpur, Sevgi Arca, Tansu Cagla Calpur and Cemal Yilmaz

Faculty of Engineering and Natural Sciences  
Sabanci University, İstanbul, Turkey  
{mehmetcagri,cyilmaz}@sabanciuniv.edu

**Abstract.** The knowledge of the business domain of a Software-Under-Test (SUT) is crucial for testing. Therefore identification of business domain and the underlying business processes is the basis for automated testing. Test cases and test input set can be automatically generated depending on the domain and process information. In this research, we apply machine learning techniques to determine the similarity of applications. Applications in the same domain should be highly similar and we can say that, same business processes are implemented in the applications of a business domain. Our hypothesis argues that assuming we can create a generalized Finite State Machine (FSM) model of a business domain, the states and transitions of the FSM could be matched to the business processes of a business domain. Previously created test cases and test input could be used for testing an application that is coherent with the states and transitions of the formal model. In this research we coin two novel terms, *Model Dressing* and *Automated Exploratory Testing*. Model dressing is matching an application to the generalized model of a business domain. Automated exploratory testing is using the previously gathered business domain knowledge to test new applications and gradually merging outcome to the previous know-how to improve testing process.

**Keywords:** Exploratory Testings, Model-based Testing, Business Domain Analysis, Application Text Mining, Automated Software Testing, Domain Classification

## 1 Giriş

Yazılım testlerinin otomasyonunun doğru yapılabilmesi ve girilen veri seti ile elde edilen sonuçların değerlendirilebilmesi için test edilmekte olan yazılımın iş alanı ile ilgili bilgiye sahip olunması gerekmektedir. Tamamen otomatikleştirilmiş bir yazılım test platformu tasarlanabilmesi için yazılımların iş alanlarına göre sınıflandırılması ilk adım olarak görülebilir. Benzer uygulamaların benzer görevleri yerine getirdikleri varsayımından yola çıkılarak önceden iş alanı bilinen bir yazılım için hazırlanmış veri seti ve test durumlarının aynı iş alanına sınıflandırılmış bir uygulamanın test edilmesinde kullanılabilir [1]. Araştırmamızda yazılımların iş alanlarına göre sınıflandırılması için uygulamaların içindeki metinleri kullanıyoruz. Uygulamaların içindeki metinler geliştirme aşamasında, veya uygulama yayınlanmasından sonra elde edilebilmektedir. Uygulamanın yayınlanmasından sonra da metin bilgilerinin elde edilebilmesi, iş alanı sınıflandırmasının dinamik uygulama test sürecine entegre edilmesini de sağlar.

Grafik arayüzü temelli olmaları ve hızlı gelişen bir ekosisteme sahip olması nedeniyle araştırmamızı mobil uygulamalar üzerinde gerçekleştirdik. Mobil uygulamalar genel olarak ince istemci mimarisine sahip uygulamalar olmaktadır. Mobil uygulamalar aktivite adı verilen ve birbirinden bağımsız olan program bileşenlerinden oluşmaktadır. Her bir aktivite veya ardışık aktiviteler grubu bir uygulama iş alanındaki iş görevlerinin hayata geçirilmesinden sorumlu olmaktadır. Aktiviteler arası geçişler veri girişi, grafik arayüz elemanlarının tıklanması veya jestlerle yapılabilir. Kullanıcı etkileşimi mobil telefon ekranı ve düğmeler aracılığıyla fiziksel olarak gerçekleşmektedir. Bu nedenle de mobil uygulamaların sistematik test edilmesi için normal bir kullanıcının etkileşiminin simüle edilmesi gerekmektedir. Standart bir kullanıcı bilinci yardımıyla bir uygulama ile olan etkileşimini ve girmesi gereken verinin belirlenmesini sezgisel olarak yerine getirebilmektedir. Araştırmamızda böyle bir bilginin otomatik olarak keşfedilip edilmeyeceği sorusunu soruyoruz.

Mobil uygulama marketlerindeki kategorizasyon, araştırmamızdaki iş alanı sınıflandırması için gereken kategorilerin temelini oluşturdu. Mobil uygulamaların grafik arayüzleri, uygulama kaynak kodu ve konfigürasyon dosyaları zengin metin içeriğine sahiptir ve bu metinler sınıflandırma işleminin veri kaynağını oluşturmaktadır. Araştırmamızın ilk aşaması metin verileri üzerinde makine öğrenmesi teknikleri kullanılarak uygulama benzerliklerinin bulunması oldu. İkinci aşamada aynı iş alanında sınıflandırılmış olan uygulamaların gerçekleştirdiği görevlerin kendi içlerinde sınıflandırılması yapılmıştır. Bu sınıflandırma deneyi aynı iş alanındaki uygulamaların yerine getirdiği temel görevler için ortak çözümler kullanması ve bu ortak çözümlerin test edilmesinde daha önce benzer uygulamalarda kullanılan test durumlarının yeniden kullanılabilmesi varsayımına dayanmaktadır. Üçüncü aşama mobil uygulamaların arayüzünde bulunan bir arayüz elemanın uygulamanın hangi görevi yerine getiren aktivitesinde olduğunun tahmin edilmesinden oluşmaktadır. Araştırmamız için çok çeşide sahip bir örnek mobil uygulama havuzu oluşturduk. Bu havuzda 10 farklı iş alanı ve her iş alanında da 10 mobil uygulama bulunmaktadır.

**Tablo 1.** İş Alanları ve Örnek Uygulamalar

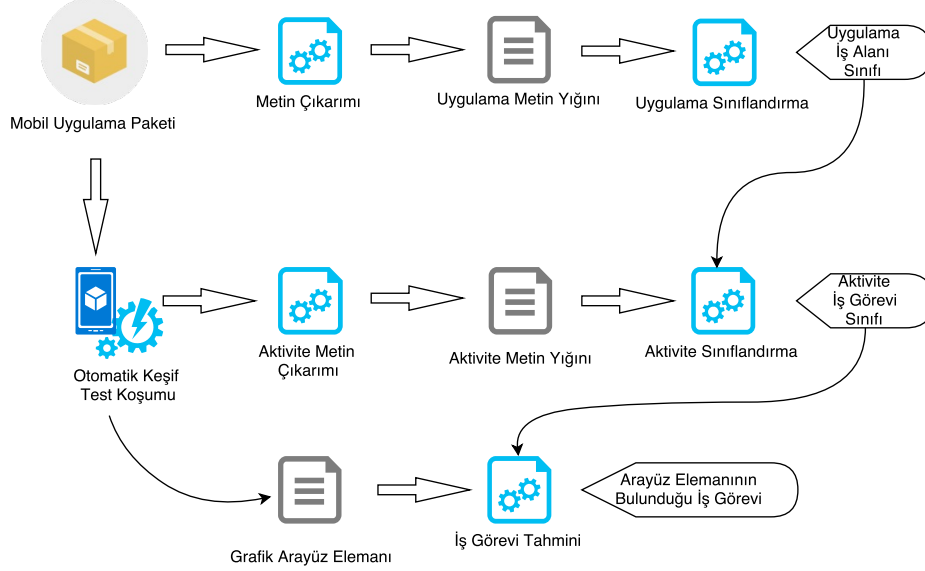
Sanat ve Tasarım	Mobil Bankacılık	Takvim	E-Ticaret	Eğitim
Adobe Illustrator Draw	American Express	aCalendar	Alibaba	Coursera
Adobe Photoshop Express	Bank of America	Business Calendar	Amazon	C Programming
Adobe Photoshop Lightroom	Bank of China	CAL	Ebay	Duolingo
Adobe Photoshop Sketch	BNP Paribas	Month: Calendar Widget	Etsy	Elevate - Brain Training
ArtFlow	Citi Mobile	Google Calendar	Flipkart	Khan Academy
ArtRage Oil Painter	Goldman Sachs	Jorte Calendar	Geek	Lynda
Google Arts & Culture	HSBC	Sol Calendar	LetGo	Memrise
Infinite Painter	JP Morgan Chase	Today Calendar	Myntra	SoloLearn
SketchBook	Wells Fargo Mobile	Wave	Snapdeal	Udacity
Tayasui Sketches	Yapi Kredi Bank	Xperia Calendar	Wish	Udemy
Sağlık ve Formda Kalma	Havayolu Rezervasyon	Harita ve Navigasyon	Spor	Hava Durumu
Calorie Counter	American Airlines	City Mapper	Bleacher Report Team Stream	AccuWeather
Couch to 5K	British Airways	Google Maps	CBS Sports	Arcus
Endomondo	The Emirates	HERE We Go Maps	ESPN Fantasy Sports	Dark Sky
Google Fit	Air France	MAPSME	ESPN	Morecast
HumanActivity	Alitalia	MapFactor	FOX Sports Mobile	YoWindow
JEFIT Workout	Delta	Map Quest	NBA	The Weather Channel
Nike Run Club	KLM	Polaris GPS Navigation	NFL Mobile	Weather Bug
Pocket Yoga	Pegasus	Sygyic Maps	Sky Sports Live Football	Weather Live
Runkeeper	Swissair	Tomtom GPS Nav	Yahoo Sports	Weather Underground
Workout Trainer	Turkish Airlines	Waze GPS Maps	The Score	Yahoo Weather

## 2 Uygulama İş Alanları ve Sınıflandırılması

Otomatik keşif testi için ilk adım uygulamanın içinde bulunduğu veya benzediği iş alanına sınıflandırılması işlemidir. İş alanının belirlenmesi test otomatının uygulamanın bağlamından haberdar olması nedeniyle veri ve test uzayının daraltılmasını sağlayacaktır. İş alanı içindeki belirli iş görevleri mobil uygulamadaki aktivitelere veya aktivite gruplarına karşılık gelmektedir. Bu aktiviteler ile de uygulama sonlu durum modelindeki durumlar eşleştirilebildiğinde, aynı alandaki uygulamaların test edilmesinde kullanılacak temel bir test durumu deposu oluşturulabilir. Test otomatı uygulama içinde bir grafik arayüz elemanı ile karşılaştığında, bu elemandan elde edeceği metin bilgisi kullanılarak önceden öğrenilmiş uygulama modelinde bir aktiviteye eşleşmesi sağlanabilir. Bu sayede test otomatı hangi iş görevinin yerine getirildiği aktivitede bulunduğunu tanımlayabilir ve aktivite ile ilgili testleri başlatabilir. Sınıflandırma süreçleri görsel şekil 1’de incelenebilir.

Deney platformu olarak Android Platformunu kullanıyoruz ancak önerdiğimiz yöntem donanım ve yazılım platformlarından bağımsız prensiplere dayanmaktadır. Bir android uygulaması aktivite denilen programlardan oluşmaktadır. Aktiviteler başka aktivitelerden gelecek bilgi ile çalışması gerekse bile, yaşam süreçleri uygulamadaki diğer aktivitelerden bağımsızdır. Uygulama paketleri her aktivite için gerekli bilgi ve kaynak dosyalarını barındırmaktadır; aktiviteler ve aralarındaki ilişkiler de bu paketlerdeki veri ile oluşturulmaktadır. Aktivitelerin birbirinden bağımsız olmaları onları sonlu durum makinasının durumları olarak tanımlamak için uygun bir temel oluşturmaktadır. Aktivitenin konfigürasyonu değişse bile temel yerine getirdiği iş görevi aynı kalacaktır, bu nedenle aktiviteleri "Meta-Durum" olarak da tanımlayabiliriz ve konfigürasyonlarını da

Şekil 1. Otomatik Keşif Testi ve Sınıflandırma Süreç Şeması



alt/iç-durum olarak adlandırabiliriz. *Geçiş*, veri girişi ve aksiyon kombinasyonları sonrası bir durumdan başka bir duruma ulaşmak olarak tanımlanabilir. Bir geçiş kendine döngüye neden olabilir, yani belirli bir geçiş aksiyonu sonrası aynı durumda kalma/ulaşma ile sonlanabilir. Mobil uygulamanın değerlendirmede geçersiz işlemler yapılması nedeniyle uygulamanın geçerli işlemleri bekleyerek aynı durumda bulunması ile tanımlanan geçiş kendine döngü olur. İkinci kendine döngü ise alt-durumlar arası geçiş nedeniyle Meta-Durumda kendine döngü olan geçiş tanımlanmasıdır. Bu durumda Meta-Durum değişmez ancak aktivite konfigürasyonu değişmiştir, bu nedenle alt-durumlar arası geçiş olmuştur. Bu durumda etkileşim sağlanan parçalar değişmiş olabilir bu da geçiş tanım kümesi fonksiyonunu değiştirir. Kısaca özetlemek gerekirse, önerdiğimiz yöntem bir aktiviteyi model durumuna, bir veri giriş alanını durum parametresine ve kullanıcı aksiyonunu (kaydırma, tıklama, v.b.) da geçişe eşleştirmektedir.

## 2.1 Mobil Uygulamaların Sınıflandırılması

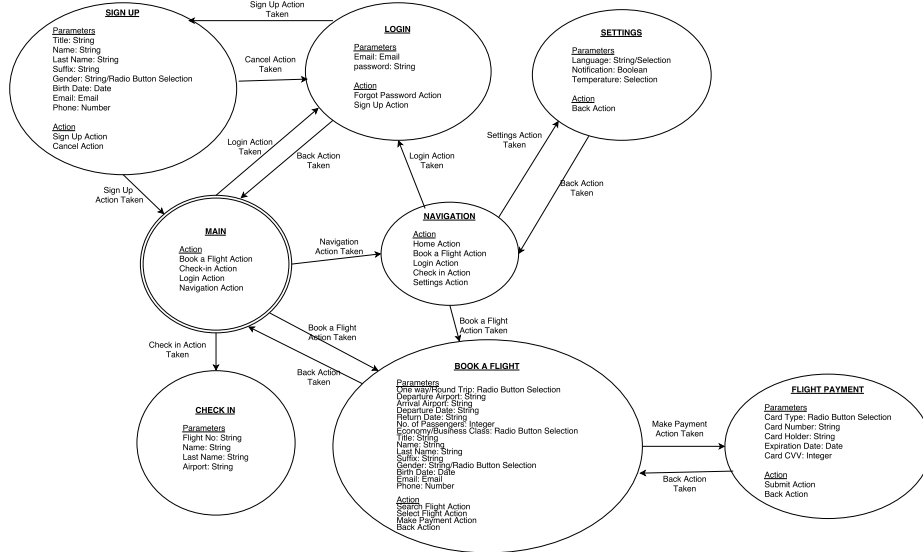
Uygulama fonksiyonel gereksinimlerinin test edilmesi için uygulanacak kara kutu test yönteminin doğru yapılabilmesi için uygulamanın iş alanı bilgisi gerekmektedir. Yazılım testlerinin çok çeşitli iş alanlarına giren çok sayıda uygulamanın test edildiği geniş çaplı bir test servisi öngörüyoruz. Bu yoğunlukta bir servis devasa boyutlarda uygulama verisinin ve test yapının (test durumu, girdi, v.s.) sağlayabilir. Bu veri birikimi uygulamaların iş alanlarını sınıflandırma yöntemi ile otomatik belirlenmesinde kullanılabilir. Araştırmamızdaki öngörümüz, mobil uygulamaların kaynak dosyalarından metin madenciliği ile toplanan verilerin

böyle bir sınıflandırma için gerekli ve yeterli olabileceğidir. Sınıflandırma için elde edilen metin verisinin toplu halde kullanılması mümkündür, ancak önceki bölümde bahsedilen iş görevi ve grafik arayüz elemanı sınıflandırma işlemleri için veri üzerinden ek işlemler gerekecektir.

## 2.2 Genelleştirilmiş Sonlu Durum Makinesi

İş alanının otomatik olarak tanımlanmasından sonra, iş alanları için biriken bilgi ile bir iş alanı modeli oluşturulabilir. Bu model hem iş görevlerini hem de kullanıcıların uygulamayı kullanırken gerçekleştirebileceği aksiyonları tanımlar. Bu modelin oluşturulması için sonlu durum makinası konsepti kullanılabilir. Uygulamayı oluşturan alt üniteler(aktiviteler) de bu sonlu durum makinasında gösterilebilir. Şekil 2 havayolu rezervasyon sistemi iş alanı için oluşturulabilen genelleştirilmiş sonlu durum makinasına örnektir.

**Şekil 2.** Havayolu rezervasyon iş alanının kısmi sonlu durum modeli ve durum parametreleri. Örnek uygulamalardaki metinler İngilizcedir, model içindeki durum isimleri ve parametreler de aslına uygun olarak İngilizce isimleri ile eklenmiştir.



Elimizde bir iş alanı modeli ve test edilecek bir uygulama olduğunda, hedefimiz bu uygulamayı model ile eşleştirebilmektir. Bu sayede test edilecek program için test durumları bu model üzerinden üretilebilir. Uygulama - model eşleştirme işlemini *model giydirme*, modelde birikmiş bilgiyi kullanarak testleri yürütmeyi ise *otomatik keşif testi* olarak adlandırıyoruz. Model giydirmenin amacı, bir test otomasyon ajanının önceden öğrenilmiş bir uygulamanın test durumlarını, yeni test edilecek uygulamaya uyarlamasını sağlamaktır. Önerilen yöntemin

pratik faydası olduğuna inanıyoruz, çünkü bulut tabanlı test servisleri sayıca artmakta ve bu servislerin test ettikleri her uygulamadan test ile ilgili yeni bir şey öğrenmesi ve öğrendiği bu test bilgisi yığınına gelecek test işlerini geliştirmek için kullanması mümkündür.

### 2.3 Uygulama İş Görevlerinin Sınıflandırılması

Aynı iş alanı içinde sınıflandırılan uygulamaların bilgilerinin kendi içinde sınıflandırılması ile uygulama iş görevlerini ayırtmak mümkün olabilecektir. Aktivitelerden elde edilen metin yığınlarını aktivite adları ile etiketleyerek benzerlik sınıflandırması yaptığımızda aynı iş görevi ile ilgili görevi yerine getiren aktivitelerin aynı sınıfa düşmesi ile görevleri sınıflandırmamız mümkün olacaktır. Bu sınıflandırma sonrasında da uygulamaların sahip olduğu aktiviteler geliştirilmiş sonlu durum makinesindeki durumlara eşleştirerek model giydirme işlemi yerine getirilir.

### 2.4 Grafik Arayüz Bilgilerinden İş Görevi Belirlenmesi

İş görevi sınıflandırması için kullanılan makine öğrenmesi modelleri, test otomasyon ajanlarının test edilen uygulamada buldukları yeri otomatik tespit etmeleri için kullanılabilir. Test edilmeye başlanan bir uygulamanın kullanıcı arayüzündeki elemanların bilgisi kullanılarak hangi iş görevi ile ilgili aktivitede bulunduğu tespit edilmesi ile testler aktivite bilgisi doğrultusunda devam ettirilir. Bu yer belirleme gereksinimi uygulama modelinin tüm uygulamalar için geliştirilmesi nedeniyle ortaya çıkmaktadır. Hakkında iş alanı dışında bilgi sahibi olunmayan bir uygulamanın aktif halde bulunan aktivitesinin hangi iş görevi ile ilgili olduğu bu sayede belirlenir ve otomatik keşif testine devam edilir.

## 3 Deneysel Çalışmalar ve Sonuçların Analizi

Araştırmamızın ilk aşaması olan iş alanı sınıflandırması ile aynı iş alanının bulunması için yaptığımız araştırmada, mobil uygulama paketlerinin içinden metin madenciliği ile çıkardığımız metin yığınlarını uygulama marketlerinden aldığımız iş alanı bilgisi ile etiketledik. 4 katlı çapraz doğrulama destek vektör makinesi kullanarak yaptığımız deneylerde yüksek kesinlik ve duyarlık değerlerinde sınıflandırma sonuçları elde ettik (Tablo 2 ve 3). F1-ölçütünü baz aldığımızda ortalama %92'lik başarımla ortaya çıkmaktadır.

Deney setimizdeki iş alanlarından havayolu rezervasyon ve hava durumu kategorilerini seçerek iş görevi sınıflandırma deneylerimizi gerçekleştirdik. Manuel olarak gerçekleştirdiğimiz iş görevi tanımlama işlemi sonucunda bu kategoriler için bulduğumuz iş görevleri ve kaç tane uygulamanın bu görevleri içerdiği ile ilgili bilgi aşağıdaki tablolarda (4 ve 5) incelenebilir.

Tablo 4 ve 5'te verilen iş görevleri ile etiketlenmiş aktivite bilgileri üzerinde gerçekleştirdiğimiz 4 katlı çapraz doğrulama destek vektör makinesi sınıflandırması sonuçları aşağıda görülebilir (Tablo [6,7,8,9]). Bu iki iş alanındaki sonuçlarda ortaya çıkması tahmin edilebilir bir ilişki göze çarpmaktadır. İş alanı için

**Tablo 2.** İş alanı sınıflandırması için 4 katlı çapraz doğrulama destek vektör makinesi

	Kesinlik	Duyarlık	F1-Ölçütü	Örnek Sayısı
Sanat ve Tasarım	1.000	0.900	0.947	10
Mobil Bankacılık	0.818	0.900	0.857	10
Takvim	0.833	1.000	0.909	10
E-Ticaret	0.900	0.900	0.900	10
Eğitim	0.818	0.900	0.857	10
Sağlık ve Formda Kalma	1.000	0.800	0.889	10
Havayolu Rezervasyon	1.000	1.000	1.000	10
Harita ve Navigasyon	1.000	0.900	0.947	10
Spor	1.000	1.000	1.000	10
Hava Durumu	0.900	0.900	0.900	10
ort.	0.927	0.920	0.921	100

**Tablo 3.** İş alanı sınıflandırması karışıklık matrisi

Sanat ve Tasarım	9	0	0	0	1	0	0	0	0	0
Mobil Bankacılık	0	9	1	0	0	0	0	0	0	0
Takvim	0	0	10	0	0	0	0	0	0	0
E-Ticaret	0	0	0	9	0	0	0	0	0	1
Eğitim	0	0	1	0	9	0	0	0	0	0
Sağlık ve Formda Kalma	0	1	0	0	1	8	0	0	0	0
Havayolu Rezervasyon	0	0	0	0	0	0	10	0	0	0
Harita ve Navigasyon	0	1	0	0	0	0	0	9	0	0
Spor	0	0	0	0	0	0	0	0	10	0
Hava Durumu	0	0	0	1	0	0	0	0	0	9
	Sanat	Bank.	Takvim	E-Tic.	Eğit.	Sağlık	Havayolu	Harita	Spor	H. Durumu

**Tablo 4.** Havayolu rezervasyon iş görevleri ve buldukları örnek uygulama sayısı

İş Görevi	Uygulama Sayısı
Bilet Alma	10
Uçuş Durumu	6
Menü	10
Üye Girişi	9
Ödeme	8
Check-in	6

**Tablo 5.** Hava durumu iş görevleri ve buldukları örnek uygulama sayısı

İş Görevi	Uygulama Sayısı
Hava Durumu	10
Lokasyon ekleme	9
Menü	8
Üye Girişi	5
Ayarlar	10



niş sayılabilecek iş görevlerinde yüksek başarımla karşı benzer metin bilgileri içeren iş görevlerinin sınıflandırmasında başarımla düşmektedir. Bu düşüş makine öğrenmesi teknikleri kullanılarak sınıflandırmanın sorun yaşayabileceği noktaları göstermekle birlikte, araştırmamızda öne sürdüğümüz otomatik keşif testinde önceden elde edilen bilgilerin yeniden kullanılması konusunda sorun teşkil etmemektedir. Benzer metinleri içeren aktivite/iş görevleri olması nedeniyle ihtiyaç duyulan veri seti ve test durumlarının benzerlik olan aktivitelerde de bulunması gerektiği düşünülebilir. Bu durumda bu veriler ve test durumları aktivite yanlış sınıflandırılabilir test kapsamını yükseltecek ve gerekli durumların test edilmesini sağlayacaktır.

**Tablo 6.** Havayolu iş alanındaki uygulamaların iş görevlerinin sınıflandırılması

	Kesinlik	Duyarlık	F1-Ölçütü	Örnek Sayısı
Bilet Alma	0.909	1.000	0.952	10
Check-in	0.800	0.667	0.727	6
Uçuş Durumu	1.000	0.833	0.909	6
Üye Girişi	0.556	0.556	0.556	9
Menü	0.667	0.600	0.632	10
Ödeme	0.750	0.750	0.750	8
Ayarlar	0.667	0.857	0.750	7
Ort.	0.754	0.750	0.748	56

**Tablo 7.** Havayolu iş görevleri karışıklık matrisi

	Bilet	Check-in	Uçuş	Üye G.	Menü	Ödeme	Ayarlar
Bilet Alma	10	0	0	0	0	0	0
Check-in	0	4	0	2	0	0	0
Uçuş Durumu	1	0	5	0	0	0	0
Üye Girişi	0	1	0	5	1	1	1
Menü	0	0	0	1	6	1	2
Ödeme	0	0	0	1	1	6	0
Ayarlar	0	0	0	0	1	0	6

## 4 İlgili Çalışmalar

İş alanları arasında benzerlik madenciliği Dong [2] tarafından ortaya atılmış yeni bir fikirdir. Bu teknik çeşitli iş alanlarındaki data setlerine uygulanarak iş alanları arasındaki analog benzerlikler ortaya çıkarılmaktadır. Yazarın ortaya attığı sav iş alanları arasındaki benzerlik madenciliğinin bir iş alanı hakkındaki bilginin başka bir iş alanına aktarılabilirliği üzerinedir. Bu sayede sınıflandırma ve kümeleme etkinliği artırılabilir anlatılmaktadır.

**Tablo 8.** Hava durumu iş alanındaki uygulamaların iş görevlerinin sınıflandırılması

	Kesinlik	Duyarlık	F1-Ölçütü	Örnek Sayısı
Lokasyon Ekleme	1.000	0.889	0.941	9
Üye Girişi	1.000	1.000	1.000	5
Menü	0.800	0.500	0.615	8
Ayarlar	0.909	1.000	0.952	10
Hava Durumu	0.692	0.900	0.783	10
Ort.	0.867	0.857	0.851	42

**Tablo 9.** Hava durumu iş görevleri karışıklık haritası

	Lokasyon	Üye Girişi	Menü	Ayarlar	Hava Durumu
Lokasyon Ekleme	8	0	0	0	1
Üye Girişi	0	5	0	0	
Menü	0	0	4	1	3
Ayarlar	0	0	0	10	0
Hava Durumu	0	0	1	0	9

Zhu vd. [4] çok büyük sayıda android uygulaması kullanarak grafik arayüz benzerliklerini ortaya çıkarmaya çalışmaktadır. Araştırma için veri uygulamaların içindeki XML kaynak dosyalarından toplanmaktadır. Resim dosyaları ve metin bilgileri uygulama benzerliği metriğinin hesaplanmasında kullanılmaktadır. Bu sayede uygulamaların telif haklarına uyulmadan kopyalanması veya zararlı içeriğin uygulamalara eklenmesinin ortaya çıkarılması hedeflenmektedir.

He vd. [7] Java grafik arayüzlerini tanımlayan XML dosyalarını işleyerek ekran benzerliklerini ortaya çıkarmaktadır. Farklı grafik arayüzler için hazırlanmış test durumlarını birleştirerek test suitinin çeşitliliğinin artırılması hedeflenmektedir.

Hallenberg ve çalışma arkadaşları çalışmalarında [3] düzgün test durumu üretilebilmesi için iş alanı bilgisine ihtiyaç olduğunu belirtmektedir. Kalite kontrol çalışanlarının test odakları ile iş süreçlerinin ihtiyaç duyduğu odaklamanın farklı olduğunu öne sürmektedirler. Grafik arayüzüne odaklanmaktadır ve uygulama modelinden gerçekleştirilebilecek test aksiyonlarını ortaya çıkarmaktalar.

Makine öğrenmesi teknikleri aynı zamanda static kaynak kod analizinde de kullanılmaktadır. Shabtai vd. [6] makine öğrenmesi araçlarının android uygulamalarının iş alanlarını öğrenmekte kullanılabileceğini söylemektedirler. Belirlenen alanın güvenlik araştırmasında kullanılabileceğini ve zararlı yazılımların ortaya çıkarılabileceğini söylemektedirler. Kaynak kodu ve XML dosyaları sınıflandırma için veri kaynağı olarak kullanılmaktadır.

Kowalczyk ve çalışma arkadaşları [5] grafik arayüz, kaynak dosyaları, uygulama marketi tanımları ve eleştirilerinden çıkartılan verilerin kullanılması ile uygulama davranışlarını incelemekteler. Makalede ortaya konan fikir bütün bu verinin toplanması ile uygulamanın ne şekilde davranacağı ile ilgili fikir sahibi olunabileceğidir.

## 5 Sonuç

Şu anda test süreçlerinde iş alanı bilgisi ve bu alandaki eksperlerin yardımı ile elde edilmekte ve manuel çalışmaya dayanmaktadır. Bu nedenle de çalışanlar için zaman kaybına ve insan hatasının test süreçlerine etki etmesine açık test süreçleri yürütülmektedir. Yürüttüğümüz araştırmanın sonuçları ile ilgili analiz bölümünde verdiğimiz başarımların değerleri otomatik keşif testinin mümkün olabileceğini göstermektedir. Test edilmekte olan bir uygulamanın iş alanının tanımlanabilmesi, iş alanı tanımı sonrası uygulama tarafından gerçekleştirilen iş görevlerinin tanımlanabilmesi, ve bu iş görevlerinin geçmiş bilgi birikimlerine bağlı olarak ortaya çıkarılabilmesinin mümkün olması sürecin otomatize edilebileceğini göstermektedir.

## Kaynaklar

1. Calpur, M.C., Arca, S., Calpur, T.C., Yilmaz, C.: Model dressing for automated exploratory testing. In: 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). pp. 577–578 (July 2017)
2. Dong, G.: Cross domain similarity mining: Research issues and potential applications including supporting research by analogy. SIGKDD Explorations 14(1), 43–47 (2012)
3. Hallenberg, N., Carlsen, P.L.: Declarative automated test. In: Proceedings of the 7th International Workshop on Automation of Software Test. pp. 96–102. AST '12, IEEE Press, Piscataway, NJ, USA (2012), <http://dl.acm.org/citation.cfm?id=2663608.2663628>
4. Jiawei Zhu, Zhengang Wu, Zhi Guan, and Zhong Chen: Appearance similarity evaluation for android applications. In: 7th International Conference on Advanced Computational Intelligence (March 27-29 2015)
5. Kowalczyk, E., Memon, A.M., Cohen, M.B.: Piecing together app behavior from multiple artifacts: A case study. In: 2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE). pp. 438–449 (Nov 2015)
6. Shabtai, A., Fledel, Y., Elovici, Y.: Automated static code analysis for classifying android applications using machine learning. In: 2010 International Conference on Computational Intelligence and Security. pp. 329–333 (Dec 2010)
7. Zhi-Wei He, Cheng-Gang Bai: Gui test case prioritization by state-coverage criterion. In: IEEE/ACM 10th International Workshop on Automation of Software Test (2015)