Haziran'22

Sayı/Number: 23/Volume: 12
Yıl/Year: 2022

Yayın Sahibi
TMMOB
Elektrik Mühendisleri Odası Adına
Mahir ULUTAŞ

Sorumlu Yazı İşleri Müdürü
Eylem ÖLMEZOĞLU

Yayın İdare Merkezi
Ihlamur Sokak No:10 Kat:3
Kızılay - Ankara
Tel: (312) 425 32 72
Faks: (312) 417 38 18
http://bilimseldergi.emo.org.tr
bilimseldergi@emo.org.tr
EMO Üyelerine parasız dağıtılır

Dergi Koordinatörü
Sevgi Kınacı
sevgi.kinaci@emo.org.tr

Sayfa Düzeni
TMMOB
Elektrik Mühendisleri Odası

Yayın Türü
Yerel Süreli Yayın
6 ayda bir yayınlanır

Basım Adedi
500

Basım Tarihi
Haziran 2022

EMO BİLİMSEL DERGİ

Elektrik, Elektronik, Bilgisayar, Biyomedikal Mühendisliği Bilimsel Dergisi

The Journal of Electrical, Electronics, Computer and Biomedical Engineering

1954

TMMOB
Elektrik Mühendisleri Odası
UCTEA/Chamber of Electrical Engineers

EMO BİLİMSEL DERGİ
DANIŞMA KURULU

EMO Bilimsel Dergi Danışma Kurulu

Prof. Dr. A. Hamit Serbest	Çukurova Üniversitesi
Prof. Dr. Altay Güvenir	Bilken Üniversitesi
Prof. Dr. Cengizhan Öztürk	Boğaziçi Üniversitesi
Prof. Dr. Erhan Akın	Fırat Üniversitesi
Prof. Dr. Güven Önbilgin	Ondokuz Mayıs Üniversitesi
Prof. Dr. Murat Eyüboğlu	Orta Doğu Teknik Üniversitesi
Prof. Dr. Timur Aydemir	Gazi Üniversitesi
Prof. Dr. Ali Hikmet Doğru	Orta Doğu Teknik Üniversitesi
Prof. Dr. Arif Nacaroğlu	Gaziantep Üniversitesi
Prof. Dr. Atilla Bir	İstanbul Teknik Üniversitesi
Prof. Dr. Aydın Köksal	Bilişim A.Ş.
Prof. Dr. Aydoğan Özdemir	İstanbul Teknik Üniversitesi
Prof. Dr. Ayşın Baytan Ertüzün	Boğaziçi Üniversitesi
Prof. Dr. Belgin Türkay	İstanbul Teknik Üniversitesi
Prof. Dr. Bülent Sankur	Boğaziçi Üniversitesi
Prof. Dr. Cüneyt Güzeliş	Yaşar Üniversitesi
Prof. Dr. Erdal Panayırcı	Kadir Has Üniversitesi
Prof. Dr. Erkan Afacan	Gazi Üniversitesi
Prof. Dr. Ferit Acar Savacı	İzmir İleri Teknoloji Enstitüsü
Prof. Dr. H. Bülent Ertan	Atılım Üniversitesi
Prof. Dr. Haldun Karaca	Dokuz Eylül Üniversitesi
Prof. Dr. İbrahim Eksin	İstanbul Teknik Üniversitesi
Prof. Dr. İnci Çilesiz	İstanbul Teknik Üniversitesi
Prof. Dr. İrfan Karagöz	Gazi Üniversitesi
Prof. Dr. İsmail Hakkı Altaş	Karadeniz Teknik Üniversitesi
Prof. Dr. İsmail Hakkı Çavdar	Karadeniz Teknik Üniversitesi
Prof. Dr. Kemal Leblebicioğlu	Orta Doğu Teknik Üniversitesi
Prof. Dr. Lale Tükenmez Ergene	İstanbul Teknik Üniversitesi
Prof. Dr. Mithat İdemen	İstanbul Teknik Üniversitesi
Prof. Dr. Muhittin Gökmen	MEF Üniversitesi
Prof. Dr. Murat Aşkar	İzmir Ekonomi Üniversitesi
Prof. Dr. Müjde Güzelkaya	İstanbul Teknik Üniversitesi
Prof. Dr. Osman Eroğul	TOBB Ekonomi ve Teknoloji Üniversitesi
Prof. Dr. Oya Kalıpsız	Yıldız Teknik Üniversitesi
Prof. Dr. Özlem Özgün	Hacettepe Üniversitesi
Prof. Dr. Sermin Onaygil	İstanbul Teknik Üniversitesi
Prof. Dr. Tayfun Akgül	İstanbul Teknik Üniversitesi
Doç. Dr. Ahmet Koltuksuz	Yaşar Üniversitesi
Doç. Dr. Osman Abul	TOBB Ekonomi ve Teknoloji Üniversitesi
Bora Güngören	Portakal Teknoloji
Fikret Küçükdeveci	TEPA
Hasan Ali Pazar	Siemens

İÇİNDEKİLER / CONTENTS

Makale Türü: Araştırma Makalesi
Makale Gönderim Tarihi:08.03.2021
Makale Kabul Tarihi: 10.02.2022

Grafiksel Kullanıcı Arayüzü Testi İçin Bir Uçtan Uca Model Tabanlı Yaklaşım
An End-to-End Model-Based Approach for Graphical User Interface Testing

Alper Silistre[1], *Onur Kılınççeker*[2,3,5], *Fevzi Belli*[2,4], *Moharram Challenger*[5], *Geylani Kardaş*[1]

[1] Uluslararası Bilgisayar Enstitüsü
Ege Üniversitesi, Türkiye
alpersilistre@gmail.com, geylani.kardas@ege.edu.tr

[2] Faculty of Computer Science
Paderborn University, Germany
okilinc@mail.upb.de, belli@upb.de

[3] Bilgisayar Mühendisliği Bölümü
Muğla Sıtkı Koçman Üniversitesi, Türkiye
kilincceker@mu.edu.tr

[4] Bilgisayar Mühendisliği Bölümü
İzmir Yüksek Teknoloji Enstitüsü, Türkiye
belli@upb.de

[5] Department of Computer Science
University of Antwerp and Flanders Make, Belgium
moharram.challenger@uantwerpen.be

Özet

'odel tabanlı Grafiksel Kullanıcı Arayüzü (GUI) testi, yazılım UI testi içerisinde önemli bir yer tutmaktadır. Manuel test, 'man alıcı bir iştir ve büyük ölçüde hataya açıktır. Yazılım st topluluğunun uzun yıllardır üzerinde çalıştığı ve genel ıllanımda olan birkaç test modeli vardır. Bu makale, model banlı GUI testinde kullanılan farklı modelleri celemektedir. Test senaryoları oluşturmak ve bunları tek bir odelde birleştirmek amacıyla, kabul gören birkaç modelin lay Sıra Çizgelerine (ESG) nasıl dönüştürüleceğine ilişkin r yöntem önerilmiştir ve bunun kullanımını örnekleyen bir ıka çalışması sunulmuştur. Ayrıca bu makalede diğer odellerden dönüşümle elde edilen ESG modelinden test imelerinin üretilmesi ve çalıştırılmasını içeren bir yaklaşım nıtılmıştır. Deneysel çalışmalar öne sürülen bu yaklaşımın gun ve etkili olduğunu göstermiştir. Bu kapsamda ESG'den de edilen 20 mutant için öne sürülen yaklaşım en yüksek viyede mutasyon skoru sonucunu vermiştir. Ayrıca gerçek r sistem için gerçekleştirilen deneysel çalışmalar, ESG odelinden elde edilen test kümelerinin daha kompakt ve hata ıkalamada daha başarılı olduğunu göstermektedir.

nahtar kelimeler: GUI Testi, Model-Tabanlı Test, Sonlu urum Makinesi, Olay Sıra Çizgesi, Olay Akış Çizgesi, üzenli İfade

Abstract

Model-based Graphical User Interface (GUI) testing keeps its importance in software GUI testing. Manual testing is time-consuming and highly error prone. There are several test models in general use that the software testing community has been working on for many years. This article examines the different models used in model-based GUI testing. To create test cases and combine them into a single model, a method for how to convert several accepted models into Event Sequence Graphs (ESG) has been proposed, and a case study illustrating its use is presented. In addition, this article introduces an approach involving generating and running test sets over an ESG model obtained by the transformation from other models. Experimental studies have shown that this proposed approach is appropriate and effective. In this context, the application of the proposed approach enabled to receive the highest mutation score for 20 mutants obtained from the ESG. In addition, experimental studies for a real system show that the test sets obtained from the ESG model are more compact and more successful in detecting faults.

Keywords: GUI Testing, Model-Based Testing, Finite State Machine, Event Sequence Graph, Event Flow Graph, Regular Expression

1. Giriş

Grafiksel Kullanıcı Arayüzü (Graphical User Interface; GUI), web, mobil veya masaüstü tüm bilgisayar uygulamalarının önemli bir parçasıdır. Uygulamalar içerisinde dolaşabilmek ve uygulamanın bize sunduğu özelliklerini kullanabilmek için her türlü GUI elemanı ile etkileşime giriyoruz. GUI esas olarak yazılımlar ile iletişim kurabilmemiz için bir arayüzdür. Kullanıcı, uygulamayla etkileşim kurmak için bir düğmeyi tıklayarak veya bir giriş alanına metin yazarak bir eylem gerçekleştirebilir. GUI Testi, GUI test eden kişiler tarafından karar verilen ön koşullara dayalı olarak GUI elemanlarının davranışlarını ve durumlarını kontrol etme ve doğrulama işlemidir [1]. Mevcut yazılım ekosisteminde, GUI elemanlarının arkasındaki iş mantığını doğrulamak büyük öneme sahiptir. Her ne kadar GUI testi bir uygulama için önem arz etse de uygulamada az sayıda GUI öğesi olsa bile test edilmesi gereken çok sayıda olası test senaryosu nedeniyle uygulama geliştiricileri tarafından genellikle ihmal edilir. Aynı eylem, programın durumuna bağlı olarak programı bir hata durumuna sokabilir. Bunu manuel olarak test etmek zordur ve uygulamaların içinde hatalar varken işletilmesine neden olur. Bu nedenle, bir uygulamanın GUI'sini düzgün bir şekilde test etmek ve doğrulamak, hataları ve kusurları ortaya çıkarabilir. Bu, temelde yatan iş mantığını (business logic) test etmek kadar önemlidir. Ayrıca, modern yazılım dünyasında, özellikle akıllı telefonlardaki mobil uygulamalar gibi tüketici hedefli uygulamalar için uygulamanın kullanılabilirliği ayırt edici bir faktördür [2]. GUI testi bu gibi alanlarda önemli yer kaplamaktadır.

Model tabanlı test, yazılımda Kara Kutu Testi (Black-Box Testing) için popüler bir yöntemdir [12]. Sistem modelini daha yüksek bir soyutlama katmanında oluşturmak, bu modele dayalı olarak test senaryolarını oluşturmamıza olanak sağlar. Literatürde, Sonlu Durum Makinesi (Finite-State Machine; FSM) [3], Olay Akış Çizgesi (Event-Flow Graph; EFG) [7], Olay Sıra Çizgesi (Event Sequence Graph; ESG) [6][12] ve Düzenli İfade (Regular Expression; RE) [25][26] gibi farklı modeller mevcuttur. Model tabanlı test, test edilen sistemin (System Under Test; SUT) modeline (soyutlamaya) dayalı test senaryoları oluşturmamıza ve ardından bu testleri, anımlanmış bir test kâhinine (test oracle) göre çalıştırmamıza olanak tanır. Bu konu etrafında incelenen ve geliştirilen otomatik araçlar ve süreçler mevcuttur. Kod tabanlı yöntemler yerine model tabanlı yöntemleri kullanmak, bu testleri kodla yürütmekten daha verimli bir şekilde test dizileri oluşturmamıza ve çalıştırmamıza olanak sağlar.

Bu çalışmada, bütünsel bir test üretme süreci oluşturmak amacıyla diğer modelleri ESG modeline dönüştürmeyi içeren ve elde edilen birleşik modeli kullanarak model-bazlı test yürütme işlemini sağlayan bir yöntem öneriyoruz. ESG modelinin, yararlanmak istediğimiz diğer modellere göre basitlik, genellik ve ölçeklenebilirlik gibi çeşitli avantajları vardır. Bu yöntem ile test dizileri otomatik olarak oluşturulur ve model tabanlı test oluşturma ve çalıştırma süreçlerini birleştirmek için bunlar ESG modelinde yürütülür. Mevcut modelleri ESG'ye dönüştürmenin ana nedeni, farklı modellerin üçtan uca model tabanlı testlerini uygulamak için farklı süreçlere ve uygulamalara ihtiyaç duymasıdır. Çalışmamızla, bu çabaları, test dizilerini oluşturmak ve çalıştırmak için verimli olan tek bir modelde birleştirmeyi amaçladık.

Deneyimlerimize ve literatür incelemesinden elde ettiğimiz sonuçlara dayanarak, test oluşturma ve çalıştırma adımları için ESG modelini kullanmaya karar verdik. Önerdiğimiz yöntemin kullanımını örneklemek için ayrıca bir durum çalışması da yine bu makalede yer almaktadır. Çalışmamızda ESG modelinin diğer modellere göre avantajları ortaya konulmuş ve elde edilen sonuçlar değerlendirilmiştir.

Bu makale, daha önceki konferans bildirimizin [31] genişletilmiş bir versiyonudur. [31]'de tanıtılan çalışma bu yeni çalışmada önerdiğimiz yöntemin öncül bir tasarımını sunmuştur. Bu makalede ise daha önce verilen bu tasarım genişletilmiş ve örnek bir sistem üzerinde uygulanmıştır. Yeni çalışmanın öncekini nasıl genişlettiği ve iyileştirdiği aşağıdaki maddelerde listelenmektedir:

- Kullanılan modellerin formel tanımlarına bağlı olarak, bu modellerden ESG modeline dönüşüm algoritmaları ve karmaşıklık analizleri verilmiştir.
- Verilen dönüşüm algoritmaları kullanılarak çalışmada adı geçen diğer modellerin ESG'ye dönüşüm işlemleri ve bunların örneklendirilmesi yerine getirilmiştir.
- Önerilen yöntem kapsamında gereken test kümesi üretim ve test koşumu aşamaları detaylandırılmıştır ve uygulamaları örnek çalışma üzerinde gösterilerek sonuçları verilmiştir.

Makalenin geri kalanı şu şekilde düzenlenmiştir: Bölüm 2 ilgili çalışmaları anlatmaktadır. Bölüm 3 önerilen yeni yöntemi tanıtmaktadır Bölüm 4 uygulama sonuçlarını ve mevcut çalışmanın geçerliliğine yönelik olası tehditleri aktarmaktadır. Son olarak, Bölüm 5'te sonuç ve geleceğe yönelik çalışmalar verilmiştir.

2. İlgili Çalışmalar

Bu bölüm, GUI testinde halihazırda var olan modellerle ilgili çalışmaları tanıtmaktadır.

Memon vd. [14] GUI'lerin kapsam kriterlerine odaklanır ve test edilecek önemli test dizilerini belirlemek ve GUI'yi bir hiyerarşide yapılandırmak için GUI bileşeni (component) terimini tanımlar. Bir GUI bileşeninde bulunan GUI elemanları arasındaki etkileşimi tanımlayan EFG'yi kullanarak GUI bileşenini temsil eder ve açıklar.

Memon [15], GUI tabanlı yazılım uygulamalarını test etmek için geleneksel yazılım tekniklerinin ve araçlarının neden GUI testi için uygun olmadığını açıklamaktadır. Ona göre GUI'ler soyutlama seviyeleri açısından uygulama kodlarından farklıdır. GUI testi sürecini ve GUI testini yapanların bu sürece nasıl yaklaşması gerektiğini anlatır. Makalede verilen örnekler yazım zamanı olan 2002'yi yansıtsa da karşılaşılabilecek zorluklar ve süreç günümüzde hala kabul edilir ve uygulanabilir durumdadır.

Belli [6], "bütünsel" (holistic) yaklaşım dediği yeni bir yaklaşımı önermektedir. Bu yaklaşımda, girdiler ve gerçekleştirilen eylemler hatalı olsa bile uygulamanın hata vermeden çalışması gerektiğini göstermek için sistemlerin sadece doğru girdilerle test edilmemesi, bu hatalı girdi ve eylemleri içeren test dizileri ile de test edilmesi gerektiğinden bahsetmiştir. Bu yöntemle, uygulama davranışı açısından

ksiksiz bir test kapsamasına sahip olacağımızdan ahsetmektedir.

hehady ve Siewiorek [3], sistem tasarımını FSM ile eşdeğer utarken, bir GUI için FSM'den daha az duruma sahip Değişken Sonlu Durum Makinesi (VFSM) adlı yeni bir model unar. VFSM, test senaryoları oluşturmak için herhangi bir amanda buna karşılık gelen bir FSM'e dönüştürülebilir. Toplam durum sayısı daha az olduğu için, bir sistemi VFSM le modellemek FSM'den daha kolaydır ve bunun daha kısa ürede yapılacağından bahsetmişlerdir. White ve Almezen [5], istem üzerinde gözlemlenebilir bir etki ile sonuçlanan bir eya daha fazla GUI nesnesini içeren bir etkinliği temsil eden e sorumluluk (responsibility) adı verilen bir kavramı ullanırlar. Bu tanımlanmış sorumluluk için, bu sorumluluğu ağırabilen tüm eylemlerin ve GUI nesnelerinin bir birleşimi lan Tam Etkileşim Dizilerini (Complete Interaction equences; CIS) oluştururlar.

Memon vd. [7], EFG modelinden otomatik test üretimi için apay zekâ tabanlı bir planlama algoritması olan yeni bir eknik sunar. Tanımlanan operatörlere bağlı olarak, planlama lgoritmasını EFG modelinde uygulamak için ilk ve son dımlar oluşturulur. Algoritma, GUI olaylarını ve tkileşimlerini dikkate alarak ilk ve son durumlar arasında test izileri oluşturur.

Memon [8], olay alanı keşif stratejilerini kullanarak modele ayalı test için yeni bir yöntem sunar. Model tabanlı test için im modelleri olay akışı modeli adı verilen ölçeklenebilir tek ir modelde birleştirir. Model oluşturma adımlarının naliyetini ve çabasını azaltmak için prosedürü tomatikleştirir.

Xie ve Memon [9], EIG ve EFG üzerindeki önceki alışmalarını kullanarak GUI modelinde hatayı göstermek için ereken en kısa olay dizileri olan ve Minimal Etkili Olay Bağlamı (Minimal Effective Event Context; MEEC) adı verilen yeni bir kavram tanımlamışlardır. Bir GUI sisteminde, bir olaya verilen yanıt, bir sistemin mevcut durumuna bağlı olarak ertelenebileceğinden hatayı tespit etmek için test dizisini gereksiz yere uzun hale getirecek bir olay kombinasyonu üzerinden geçilebilir. Bunun yerine MEEC, arızayı tespit etmek için en kısa yolu gösterir.

Huang vd. [10], testin erken sonlandırılması gibi olasılıklar nedeniyle GUI testi için yararsız olan GUI test dizilerini onarmayı amaçlayan bir yöntem geliştirirler. Bu sorunlu test serilerini düzeltmek ve kapsamı artırmak için genetik bir algoritma kullanırlar.

Belli vd. [11], GUI'lerin doğruluğu hakkında deneysel bir anlayış elde etmek için GUI'lerin güvenilirliği ve insan-makine sistemlerinde bir GUI'nin güvenilirlik modelinin seçimi hakkında bir vaka çalışması sunar. GUI testi için uygun bir modelleme tekniğinin seçilmesinin değerlendirme sürecinin ve dolayısıyla yazılımın kalitesini etkilediğini belirtmişlerdir.

Banerjee vd. [1] GUI test çalışmaları hakkında anket yapıp ilgili belgeleri sistematik bir haritalama tekniği ile eşleştirmişlerdir. GUI testi hakkında 1991 ve 2011 arasında yazılan 230 makale havuzundan çalışmalar için seçim kriterlerini belirlemişlerdir. Çalışmaları sınıflandırıp daha fazla çalışma ve araştırma gerektiren mevcut yaklaşımlara ve alanlara genel bir bakış sağlamışlardır. Ayrıca, model tabanlı GUI testi için geleneksel ve modern tekniklerden örnekler sunmuşlardır.

Belli vd. [12] modelleme ve test senaryosu oluşturma tekniklerini dikkate alarak model tabanlı GUI testine ilişkin mevcut çalışmaları ayrıntılı olarak gözden geçirmişlerdir. Bu modellerin ve kullanımlarının gerçek dünyadan örneklerini verirken bu tekniklerin optimizasyonunu incelemişlerdir.

Tablo 1: GUI Testi için Kullanılan Modellerin Avantajları ve Dezavantajları

Model	Avantajları	Dezavantajları
ESG [6]	+Basit bir modelleme mekanizması sunar +Ölçeklenebilirlik sorununa uygun bir çözüm önerir +Test oluşturma için basit ve doğru bir yol sağlar	-ESG modeli, içindeki düğümlerde (node) GUI olaylarını tutması nedeniyle FSM gibi diğer bilinen modellerle karşılaştırıldığında bağlam bilgisi gerektirir.
EFG [14]	+Daha yüksek ifade gücü sayesinde farklı GUI bileşenleri için çeşitli yollarda modellemeleri kolaylaştırır. +Test üretimi için uygulanabilir bir çözüm sunar	-Ölçeklenebilirlik sorunu ile baş edemez -EFG modeli, içindeki düğümlerde GUI olaylarını tutması nedeniyle FSM gibi diğer bilinen modellerle karşılaştırıldığında bağlam bilgisi gerektirir
FSM [3]	+Basit bir modelleme mekanizması sunar	-Ölçeklenebilirlik sorununa çözüm getirmez -Test üretimi için karmaşık bir formalizasyon gerektirir
VFSM [3]	+Ölçeklenebilirlik sorununa uygun bir çözüm önerir	-FSM'den dönüşüm nedeniyle ek maliyete neden olur -Test üretimi için karmaşık bir çözüm gerektirir
RE [22]	+Modelleme için kompakt bir çözüm sunar ve test üretimini basitleştirir	-Ölçeklenebilirliği yönetmez -FSM'den dönüşüm nedeniyle ek maliyete neden olur

Belli vd. [13], test edilen sistemin çok büyük olması durumunda katmanlı merkezli test yöntemini ve ilgili test luşturma sistemini önererek test senaryolarının sayısını ve naliyetini azaltan bir çalışma gerçekleştirmiştir. Bu netodolojiyi kullanarak, az sayıda test senaryosunda bile irçok hatalı durumun bulunabileceğini göstermiştir.

Kilincceker vd. [22], GUI'yi modellemek ve test etmek için düzenli ifadeyi tanıtıp aynı çalışmada kullanmışlardır. Ayrıca düzenli ifadeden rastgele test dizileri oluşturmuşlardır ve rastgele test üretme algoritmalarını bir vaka çalışması üzerinde değerlendirmişlerdir.

Mercan vd. [23], bir mobil uygulamanın GUI'sini modellemek ve test etmek için sonlu durum makinesini sunmuştur. Ayrıca,

onlu durum makine modeline göre hataların varlığını ve okluğunu test etmek için bir metodoloji önermiştir.

ilincceker ve Belli [24], düzenli ifadeye dayalı bir analiz acılığıyla GUI testi için dört yeni kapsam kriterini nermektedir. Normal ifadelerin analizinden sonra, kapsam riterlerini sundukları bağlamsal tablolar elde ederler. Bu apsam kriterleri, mutasyon testine dayalı kalite eğerlendirmesi dahil olmak üzere test oluşturma ve test için 26]'da kullanılmıştır.

ilincceker ve Belli [27], [43], hem donanım tasarımı hem de azılım GUI testi için birleşik bir modelleme yöntemi ınmaktadır. Ayrıca, mutasyon testiyle birleştirilmiş bütünsel ir test yöntemi kullanırlar. Modelleme ve test yöntemlerini onanım tasarımı ve yazılım GUI alanından aldıkları iki örnek lay incelemesinde değerlendirirler.

ilincceker vd. [33], GUI tabanlı sistemler için model-tabanlı ıutasyon testini kullanan ve model-tabanlı ideal test olarak llandırılan bir yöntem öne sürmüşlerdir. Öne sürülen bu aklaşım mutasyon testi ve Belli [6] tarafından ortaya atılan ütünsel testi (holistic testing) birleştirerek elde edilen öntemin ideal test olmak için gereken güvenilirlik ve eterlilik kriterlerini sağladığını iddia etmektedirler. Çalışma apsamında öne sürülen yöntemin bu kriterleri sağladığını erek teorik gerekse deneysel çalışmalar ile göstermişlerdir. yrıca öne sürülen yöntem, farklı yöntemler ile ıyaslanmıştır. Ancak kullanılan bazı adımlar halen elle erçekleştirildiği için bu adımların otomatik hale getirilmesi eriki çalışmalar olarak ifade edilmiştir. Kilincceker vd. [35] yrıca benzer yaklaşımın gömülü sistemlerin fonksiyonel stleri için de geçerli olduğunu göstermişlerdir.

levcut çalışmalardan farklı olarak, Vos vd. [39], GUI stlerinin test betikleri olmadan yine otomatik olarak erçekleştirilebildiğini göstermiştir. Bunun için TESTAR imli bir araç geliştirilmiştir ve bu araç GUI tabanlı olayları tomatik olarak tespit edebilmekte, ardından bu olayları yine tomatik olarak mevcut GUI üzerinde koşabilmektedir. hahim vd. [40], TESTAR aracını endüstriyel seviyede test mişlerdir.

aldes vd. [38], GUI tabanlı sistemlerin otomatik testlerine önelik 30 yıllık çalışmaları özetleyen kapsamlı bir literatür nalizini ortaya koymuşlardır. Arzu eden okuyucular, bu konu akkında detaylı bilgileri, var olan yöntemleri ve mevcut roblemleri bu çalışmayı okuyarak elde edebilirler.

ablo 1 yukarıda değinilen ve şu anda GUI testinde kullanılan ıodellerin avantaj ve dezavantajlarını özetlemektedir. nerdiğimiz yeni yöntem, mevcut çalışmalardan farklı olarak SG modelini avantajlarından dolayı diğer modellerden önüştürülebilir hale getirerek tekil bir modelleme imkânı ınmaktadır. Böylece diğer modeller ile ifade edilen sistemler ne sürülen yaklaşım ile tekil ESG modeli kullanarak test dilebilir olmaktadır.

3. Önerilen Yöntem

nerilen yöntem, sistemin modelini temsil etmek için JSON eya XML gibi açık standart dosya formatlarının herhangi irinde bir ESG modeli oluşturmanın bir yolunu sağlar. ESG ıodeli FSM, Hiyerarşik FSM (Hierarchical Finite State Machine; HFSM), RE ve EFG'den dönüştürülebilmelidir. Bunların tümü, bir GUI sistemini modellememize izin veren literatürdeki modellerdir. Örneğin, bu bir web sitesinde bir kayıt formu veya bir mobil uygulamada kullanıcı etkileşimlerini kabul eden bir ekran olabilir.

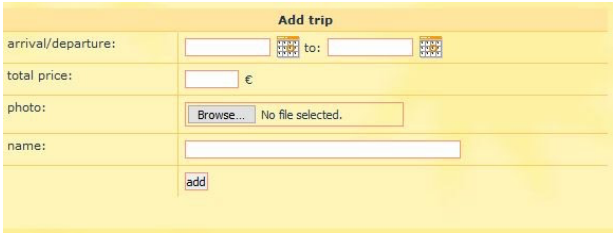
3.1. Kullanılan Modeller ve Dönüşümler

Kullanılan kavramlar bu bölümde resmi (formel) olarak tanımlanmıştır. Bu kavramlar FSM, HFSM, RE ve EFG'dir. Her resmi gösterim için, karşılık gelen modelleri içeren örnek bir GUI sistemi göstereceğiz. Örnek sistem, ISELTA [21] web sitesinin "Special" adı verilen modülünün basitleştirilmiş bir sürümüdür (Şekil 1). ISELTA, otel sağlayıcıları için bir çevrimiçi rezervasyon sistemidir ve buradaki "Special" modülü, yolculuk eklemek için bir formdur. ISELTA uygulaması bölüm 3.4 kapsamında daha detaylı olarak açıklanacaktır. Ayrıca ISELTA uygulamasının gerçek versiyonu yine bölüm 3.4 kapsamında, test koşumu için kullanılacaktır.

3.1.1. FSM

Tanım 1: Aşağıda verilen 5'li küme <Q, Σ, δ, q0, F> bir FSM [20] tanımlar

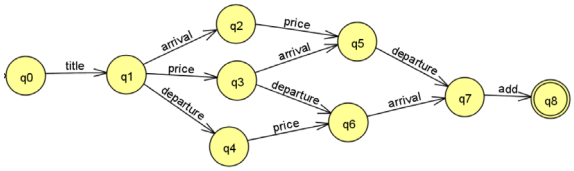
- Q: sonlu bir durum kümesi
- Σ: sonlu bir girdi sembol kümesi (alfabe)
- δ: durum geçiş fonksiyonu
- q0: Q kümesine ait olan başlangıç durumu
- F: Q kümesine ait olan bitiş durumlarının kümesi



Şekil 1: ISELTA Web sitesi "Special" modülü

Örnek 1: Aşağıda verilen 5'li küme ISELTA "Special" modülü için bir FSM tanımlar (bakınız Şekil 2). Burada "t", "d", "p", "a", ve "s" sembolleri sırasıyla (set title, set departure, set price, set arrival, add button) eylemlerini temsil eder. Gerçek sistemde "name" olarak verilen olay, model üzerinde "title" olarak ifade edilmiştir.

- Q: = {q0, q1, q2, q3, q4, q5, q6, q7, q8}
- Σ: = {t, a, p, d, s}
- δ: = {δ(q0, t)=q2, δ(q2, a)=q7, δ(q2, p)=q3, δ(q2, d)=q8, δ(q7, p)=q4, δ(q3, a)=q4, δ(q3, d)=q5, δ(q8, p)=q5, δ(q4, d)=q6, δ(q5, a)=q6, δ(q6, s)=q1}
- q0: = {q0}
- F: ={q1}



Şekil 2: ISELTA "Special" modülü FSM'i

.2. *HFSM*

ınım 2: Aşağıda verilen 6'lı küme <Q, Σ, δ, q0, F, L> bir SM [7] tanımlar.

- Q: sonlu bir durum kümesi
- Σ: sonlu bir girdi sembol kümesi (alfabe)
- δ: durum geçiş fonksiyonu
- q0: Q kümesine ait olan başlangıç durumu
- F: Q kümesine ait olan bitiş durumlarının kümesi
- L: sonlu bir katman kümesi

nek 2: Aşağıda verilen 6'lı küme ISELTA "Special" modül ı bir HFSM tanımlar

- Q: = {q0, q1, q2, q3, q4, q5, q6, q7, q8}
- Σ: = {t, a, p, d, s}
- δ: = {δ(q0, t)=q2, δ(q2, a)=q7, δ(q2, p)=q3, δ(q2, d)=q8, δ(q7, p)=q4, δ(q3, a)=q4, δ(q3, d)=q5, δ(q8, p)=q5, δ(q4, d)=q6, δ(q5, a)=q6, δ(q6, s)=q1}
- q0: = {q0}
- F: = {q1}
- L: = {Ø}

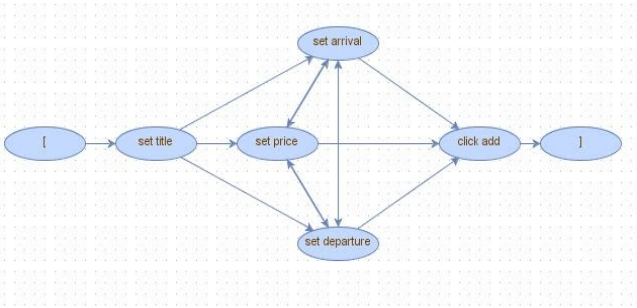
.3. *ESG*

ınım 3: Aşağıda verilen 4'lü küme <E, A, S, F> bir ESG] tanımlar

- E: eylemleri temsil eden sonlu bir düğüm kümesi
- A: A ⊆ N×N olaylar arasındaki ilişkiyi temsil eden yönlendirilmiş sonlu bir ark kümesi
- S: başlangıç eylemini temsil eden boş olmayan eylemler kümesi
- F: bitiş eylemini temsil eden boş olmayan eylemler kümesi

nek 3: Aşağıda verilen 4'lü küme ISELTA "Special" dülü için bir ESG tanımlar (bakınız Şekil 3).

- E: = {[, set title, set arrival, set price, set departure, click add,]}
- A: = {([, set title), (set title, set arrival), (set title, set price), (set title, set departure), (set arrival, set price), (set arrival, set departure), (set arrival, click add), (set price, set arrival), (set price, set departure), (set price, click add), (set departure, set arrival), (set departure, set price), (set departure, click add), (click add,])}
- S: = {[}
- F: = {]}



Şekil 3: ISELTA "Special" modülü ESG'si

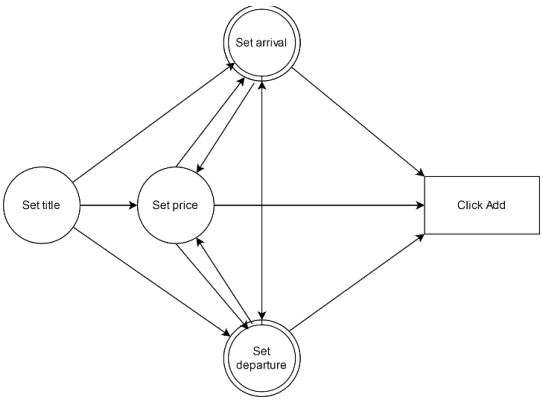
3.1.4. *EFG*

Tanım 4: Aşağıda verilen 4'lü küme <V, E, B, I> bir EFG [14] tanımlar

- V: tüm eylemleri temsil eden bir tepe kümesi
- E: tepeler arasındaki yönlü ayrıtların bir kümesi
- B: modellenmiş GUI'nin başlangıcında mevcut olan bir tepe kümesi
- I: bir GUI bileşeni için yasak olan (gerçekleşmesi mümkün olmayan) eylemler kümesi

Örnek 4: Aşağıda verilen 4'lü küme ISELTA "Special" modülü için bir EFG tanımlar (bakınız Şekil 4).

- V: = {set title, set arrival, set price, set departure, click add}
- E: = {([set title, set arrival), (set title, set price), (set title, set departure), (set arrival, set price), (set arrival, set departure), (set arrival, click add), (set price, set arrival), (set price, set departure), (set price, click add), (set departure, set arrival), (set departure, set price), (set departure, click add)}
- B: = {set title, set arrival, set price, set departure}
- I: = {Ø}



Şekil 4: ISELTA "Special" modülü EFG'si

.1.5. RE

anım 5: Kurallar aracılığıyla bir RE, x, y, z, ... sembollerinin rası ile tanımlanır. Semboller, RE'yi tanımlayan aşağıdaki urallarla ilgili olarak sıfır veya daha fazla kez oluşabilir.

- Birleştirme: "." veya "" (boş) ile gösterilir. Örneğin, "ab", "a" anlamına gelir ve ardından "b" gelir
- Seçim: "+" ile temsil edilir. Örneğin, x + y, "x veya y" anlamına gelir.

rnek 5: Aşağıda verilen örnek ISELTA "Special" modülü in bir düzenli ifade tanımlar.

R: = (tdpas+tapds+(tpda+tpad)s)

.2. Model Dönüşümleri

ölüm 3.1'de verilen formel modeller arası dönüşümlerin asıl sağlanacağı bu bölümde anlatılmaktadır. Önerdiğimiz öntemde kullanılan asıl model ESG'dir. Bu yüzden diğer odellerden ESG modeline dönüşümler için gerekli lgoritmaların sözde kodları devam eden alt bölümlerde österilmektedir. HFSM'den ESG'ye dönüşüm kodu erilmemektedir. Bunun nedeni bu dönüşümün benzerinin SM'den ESG'ye dönüşüm ile kolaylıkla yapılabilmesidir.

.2.1. FSM'den ESG'ye Dönüşüm

u dönüşüm için gerekli sözde kod Şekil 5'te verilmiştir. Bu özde kod girdi (FSM) ve çıktı (ESG) satırları ile başlar. rdından bir döngü ile Lambda fonksiyonu bileşenleri erisinde gezinilmektedir (bakınız satır 3). Bu döngünün ileşenleri Lambda'yı oluşturan alt fonksiyonlardır (f_i). öngü içerisinde FSM'de tanımlanan her bir "t_i" bileşeni SG'nin tepe değerleri kümesi olan "E" içerisine alınır (satır -5). Yine bir döngü sayesinde FSM'in Lambda fonksiyonu ullanılarak FSM durumlar arası komşuluk ilişkisi ile ESG'ye it "A" kümesi (tepeler arası komşuluk) tanımlanır (satır 6-9). on olarak FSM'e ait başlangıç ve bitiş ayrıtları kullanılarak SG'ye ait başlangıç ve bitiş tepesi belirlenir (satır 10-15).

```
1 Girdi: FSM <Q, Sigma, Lambda, q0, F>
2 Çıktı: ESG <E, A, S, F>
3 for each function f_i in Lambda(f(sc, t)=sn)
4     sc_yeni = t_i
5     E = E union (sc_yeni)
6     for each function f_j in Lambda(f(sc, t)=sn)
7         if (sn_i neighbor with sn_j)
8             A = A union (sn_yeni, t_j)
9     end for
0     if (sn_i belongs to F)
1         A = A union (sn_i, "]")
2         F = "]"
3     if (sn_i belongs == q0)
4         A = A union (sn_i, "[")
5         S = "["
6 end for
```

Şekil 5: FSM'den ESG'ye Dönüşüm Sözde Kodu

SM'den ESG'ye dönüşüm için gereken sözde kodun zaman armaşası Lambda fonksiyonu bileşenlerine (n bileşen için) ağlı olarak ikinci derecedendir (quadratic time) ($O(n^2)$).

.2.2. EFG'den ESG'ye Dönüşüm

ekil 6, bu dönüşüm için gereken sözde kodu göstermektedir. özde kodda ilk olarak, EFG girdi değeri, ESG çıktı değeri larak tanımlanır (satır 1-2). Ardından bir döngü ile EFG'ye it tepeler arası ilişkiyi tanımlayan "E_efg" ikililerinin bileşenleri içerisinde gezinilir (satır 3-14). Bu döngü içerisinde elde edilen bileşenler ESG'ye ait bileşenler kümesini (A) tanımlar. Ardından yine bir döngü ile EFG'ye ait tepe değerleri içinde gezinilir (satır 5-11). Bu döngü sayesinde ESG'ye ait tepe kümesi olan "E_esg" elde edilir (satır 6). Ayrıca yine aynı döngü içerisinde, ESG'ye ait başlangıç ve bitiş tepeleri için ilgili bileşenler belirlenir ve bunlar bileşenler kümesi "A"ya eklenir (satır 7-10). Son olarak ESG'ye ait başlangıç ve bitiş tepeleri oluşturulur (satır 12-13).

```
 1 Girdi: EFG <V, E_efg, B, I>
 2 Çıktı: ESG <E_esg, A, S, F>
 3 for each element i of E_efg(current, next)
 4     A  = A union (i)
 5     for each element j of V
 6         E_esg =     E_esg union (j)
 7         if (j is final event)
 8             A = A union (j, "]")
 9         if (j belongs to B)
10             A = A union ("[", j)
11     end for
12     F = "]"
13     S = "["
14 end for
```

Şekil 6: EFG'den ESG'ye Dönüşüm Sözde Kodu

EFG'den ESG'ye dönüşüm için kullanılan sözde kodun zaman karmaşası EFG tepe ilişkileri bileşenleri ve yine EFG tepe kümesi "V" eleman sayısına (n eleman için) bağlı olarak ikinci derecedendir ($O(n^2)$).

3.2.3. RE'den ESG'ye Dönüşüm

Bu dönüşüm için gereken sözde kod Şekil 7'de verilmiştir. Dönüşüm girdi "RE" ve çıktı "ESG" modeli tanımlamaları ile başlar (satır 1-2). Ardından gereken işlemler 3 adımda ifade edilmektedir. İlk adım RE modelinden kararlı olmayan sonlu durum makinesi (NFA: Non-deterministic Finite Automata) elde edilmesidir. Bu dönüşüm için gerekli adımlar Brüggemann-Klein tarafından [28]'de tanımlanmıştır. Bu dönüşüm yöntemi, Otomata Teorisi'ne (Automata Theory) dayanmaktadır. İlk adım için gereken dönüşüm çıktı olarak bir NFA üretmektedir. Elde edilen NFA'nın kararlı sonlu durum makinesine (DFA: Deterministic Finite Automata) dönüşümü sözde kodda ikinci adımda verilmektedir. Okuyucu, bu dönüşüm için detaylı bilgilere [28]'den ulaşabilir. Sonuç olarak elde edilen DFA modelinin ESG modeline dönüşümü 3. adımda verilmektedir. Bu dönüşüm Şekil 5'te verilen FSM'den ESG'ye dönüşüm sözde kodu ile çok benzerdir.

```
1 Girdi:RE
2 Çıktı:ESG <E, A, S, F>
3 Adım1:RE'den kararlı olmayan sonlu
4       durum makinesine(NFA) dönüşüm
5 Adım2:NFA'dan kararlı sonlu
6       durum makinesine(DFA) dönüşüm
7 Adım3:DFA'dan ESG'ye dönüşüm
```

Şekil 7: RE'den ESG'ye Dönüşüm Sözde Kodu

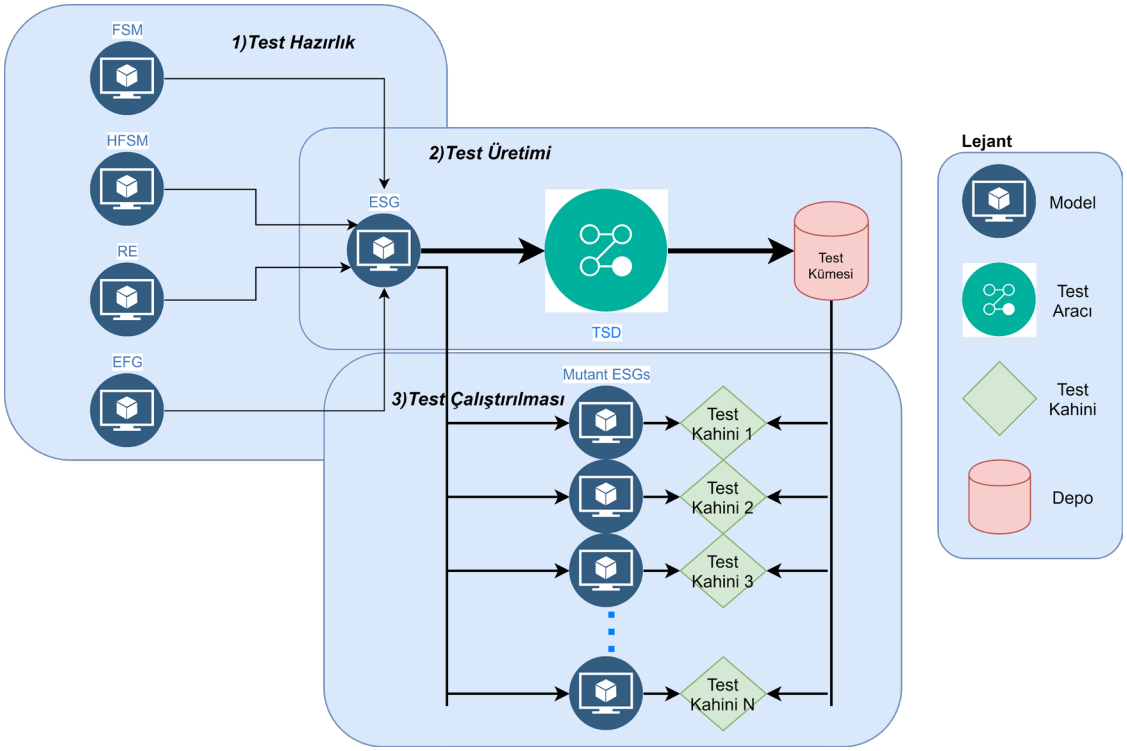
Burada verilen adım 1 için gerekli algoritmanın zaman karmaşası dönüşümlerde kullanılan bileşenlere (n bileşen için) bağlı olarak lineerdir ($O(n)$). Adım 2, genellikle geçerli tipik durumlar için ($O(n^3)$) ve adım 3 ise lineer zamanda gerçekleştirilebilmektedir. Bunun yanında adım 2, en kötü durumda $O(2^n n^2)$ zamanda gerçekleşmektedir. Böylece

'E'den ESG'ye dönüşüm için verilen sözde kodun genel tipik urumlar için zaman karmaşası yine üçüncü derecedendir O(n) + O(n³) + O(n) => O(n³)). Zaman karmaşıklığı detayları çin arzu eden okuyucular [41] çalışması, sayfa 165'i ıceleyebilirler.

.3. Yöntemin Uygulama Adımları ve Örnek Çalışma

Mevcut çalışma, test ettiğimiz sistemin fonksiyonel hatalarını espit etmeyi amaçlamaktadır. Bu doğrultuda, önerilen öntem, test hazırlama, test oluşturma ve test yürütme olmak zere üç adıma bölünür (bakınız Şekil 8). Önerilen yöntemin ir uygulaması ISELTA web sitesi "Special" modülü üzerinde gerçekleştirilmiştir. Mevcut uygulama adımları çalışma sırasında dışardan müdahaleye gereksinim duymaktadır. Ancak komple bir model tabanlı test otomasyon aracı elde etmek için bu üç adım ileride tamamen otomatikleştirilebilir.

ESG modelinden test dizileri oluşturduktan sonra, sistemi test etmek için model tabanlı test yönteminin modele nasıl uygulanacağını göstereceğiz. Sonuçları göstermek için yaklaşımımızın örnek olay incelemesinde ISELTA web sitesinin formlarını kullanıyoruz (bakınız Şekil 1).



Şekil 8: Önerilen Yöntemin Genel Görünümü

.3.1. *Test Hazırlık*

Bölümü 3.2'de ifade edildiği gibi test hazırlık aşamasında SM, HFSM, RE ve EFG modelleri basitlik, genellik ve lçeklenebilirlik avantajları nedeniyle ESG modeline önüştürülür. Dönüşümler için gerekli algoritmalar daha önce ine Bölüm 3.2.'de verilmişti. Görsel olarak ISELTA web itesinin "Special" formunun diğer modelleri de Bölüm .2'deki şekillerde dönüşüm tanımları ve örnekleri aşamasında erilmiştir. Test hazırlık aşamasında diğer modellerden önüşüm yapılacağı gibi ayrıca kullanıcılar sistemlerini SG'de modelleyip doğrudan yönteme de aktarabilirler.

SG modellerinin mutantları, üzerine ekleme (insertion), eğiştirme (replacement) ve çıkarma (omission) mutasyon peratörlerinin uygulanacağı orijinal ESG'den elde edilmiştir. u kapsamda Şekil 3'te verilen ESG modeline bu mutasyon peratörleri uygulanmış ve toplam 20 mutant elde edilmiştir. u mutantlar ESG'de verilen olaylar arası ilişkilerden kaynaklanabilecek fonksiyonel hataları modellemektedir. Bunlardan 2 tanesi ekleme, 9 tanesi değiştirme ve 9 tanesi çıkarma mutantıdır. Değiştirme ve çıkarma mutantları ESG modelinin olası tüm ayrıtlarına uygulanarak elde edilmiştir. Ekleme mutantları elde edilirken oluşan mutantların tamamının hatasız ESG modeline denk modeller ürettiği sonucuna varılmıştır. Bu yüzden deneysel olarak sadece 2 tane ekleme mutantı elde edilmiştir.

3.3.2. *Test Kümesi Üretimi*

Test oluşturma aşamasında, "graph traversal" algoritması kullanılarak ESG modelinden geçerli bir test dizisi seti içeren bir test grubu oluşturulur. Test Kümesi Üretimi işlemini otomatik olarak gerçekleştirmek için Padeborn Üniversitesi Uygulamalı Veri Teknolojisi bölümü tarafından geliştirilen ve Belli vd. [30]'da tanıtılan bir araç kullanılmıştır. Bu araç ESG modelini kullanmaya imkân vermektedir. Aracın kullanımı ile ilgili detaylı bilgilere [28]'den ulaşılabilir.

'est üretim aracı diğer model-tabanlı test araçlarının aksine SG modelini kullanma sayesinde bir optimizasyon lgoritması işleterek optimum test kümeleri üretimine olanak ağlamaktadır. Optimum test kümeleri üretimi için ESG'ye ait aşlangıç ve bitiş tepeleri arasında kalan tepeler arasında tüm epeleri kapsayan, en kısa yolları bulan bir algoritma ullanılmaktadır. Bu sayede diğer model-tabanlı test raçlarından daha sıkı (compact) ve daha etkili test kümeleri retimi mümkün olmaktadır. Ayrıca bu araç sayesinde, lçeklenebilirlik probleminin üstesinden gelmek için katmanlı SG yapısı kurmak ve bu katmanlı yapıdan otomatik olarak kı ve etkili test kümesi elde etmek mümkündür.

u aşamada kullanılan programların girdileri, bir önceki şamada çıkan sonuçların elle programa verilmesi ile ağlanmaktadır. Yöntem kapsamında test üretim aracı ile ıodeller arası dönüşüm işlemlerini gerçekleştiren araç rasında entegrasyon sağlanacak ve böylece diğer modellerden tomatik olarak dönüşüm yapılarak yine otomatik olarak test ümesi elde edilen bir araç geliştirilecektir.

```
[,set title,set price,set arrival,set departure,
set arrival,set price,set departure,set price,click add,],
[,set title,set arrival, click add,],
[, set title, set departure, click add,],
```

Şekil 9: Test Kümesi

rnek çalışma olarak verilen "Special" modülüne ait ESG'den tomatik olarak elde edilen test kümesi Şekil 9'da erilmektedir.

.4. Testlerin Çalıştırılması

.4.1. Model Test Koşumu

'alışmamız ilk aşamada GUI sistemlerini model-tabanlı larak test etmek üzerine kurgulanmıştır. Bu sebeple, ilk şamada test üretim aracının kullanılması sonucu elde edilen st kümeleri gerçek sistem üzerinde çalıştırılmak yerine yine ıodeller üzerinde çalıştırılır. Bu aşamada kullandığımız öntem mutant bazlı model tabanlı test çalıştırma yöntemidir.

latasız ESG modelinden elde edilen test kümeleri, toplam 20 ıutant üzerinde çalıştırılır. Verilen mutant modellerde bu test ümelerini çalıştırma işlemi toplam 20 mutant için yaklaşık larak 8 dakika gibi bir zaman almıştır. Test çalıştırma işlemi snasında test kümesinde verilen her bir test durumu için ESG zerinde ilgili olayın olduğu yere gidilmiştir. Örneğin; test ümesinde {[, set title, set price} test sırası için mutant model zerinde sırası ile bu test durumları takip edilmiştir. Takip tme işlemi için mutant ESG'nin olaylar arası komşuluk işkisi kullanılmıştır. Örneğin; mutant ESG'de "set title" ve set price" olayları komşu değilse test "başarısız" (fail) veya u olaylar komşu ise test "başarılı" (success) olmaktadır. laylar arası komşuluk ilişkisi test çalıştırma işlemi için test âhini (test oracle) vazifesi görmektedir.

'est çalıştırma işleminden elde edilen sonuçlar Tablo 2'de erilmektedir. Çalışma sonuçlarına göre "Ekleme" mutasyon peratörü haricinde tüm hatalar tespit edilmiştir. Hata apsama yüzdeleri "ekleme" operatörü haricinde yüzde 100 ve ıutasyon skorları 1'dir. Yani tüm mutantlar başarı ile tespit dilmiştir. "Ekleme" mutasyon operatörü ile elde edilen mutantların yakalanamamasının sebebi bu mutantların hatasız ESG'ye denk olmasıdır. Toplam mutasyon skoru elde edilirken denk mutantların göz ardı edilmesi de gerektiğinden toplam mutasyon skoru 1'dir.

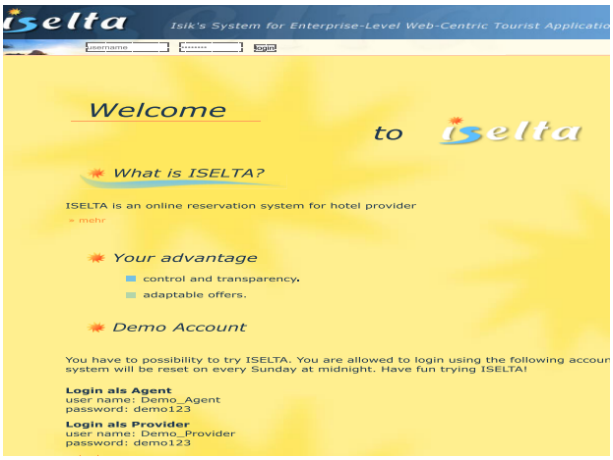
Tablo 2: Model Sistem Test Sonuçları

Mutasyon Operatorü	Mutant Sayısı	Tespit Edilen Hata Sayısı	Hata Kapsama (Yüzde (%))	Mutasyon Skoru
Ekleme	2	0*	0*	0
Çıkarma	8	8	100	1
Değiştirme	8	8	100	1
Toplam	20	16	80	1*

"Ekleme" operatörünün tek başına modele uygulanması sonucu elde edilen tüm mutantlar denk olmaktadır. Bunun sebebi bu operatörün tek başına bir modele uygulanmasının bu modelin hatasız halinde verilen yapısını değiştirmemesi ve sadece ekleme yapmasıdır. Bu yüzden daha sonraki çalışmalarımızda "ekleme" mutasyon operatörünün kullanılmaması düşünülmektedir.

3.4.2. Gerçek Sistem Test Koşumu

Gerçek sistem test koşumları çalışmamızın 2. aşamasıdır. Bu aşama için, "ISELTA" (bakınız Şekil 10) ismi verilen, Paderborn Üniversitesi ve Almanya merkezli ISIK turizm firması tarafından geliştirilen bir turizm web sitesi uygulaması seçilmiştir. Bu web uygulaması sayesinde turizm acentaları ve müşteriler daha kolay bir şekilde iletişim halinde olabilmektedir. Ayrıca acentalar, müşterilerine "Special" adı verilen modül sayesinde özel birtakım kampanyalar sunabilmekte ve müşteriler bu modül üzerinden sunulan kampanyalara rezervasyon yapabilmektedir. ISELTA uygulaması PHP programlama dilinde geliştirilmiştir ve yaklaşık olarak 70000 satır kod içermektedir. Okuyucular dilerlerse, Şekil 10'da verilen demo bilgilerini (kullanıcı adı: Demo_Agent, şifre: demo123 veya kullanıcı adı: Demo_Provider, şifre: demo123) kullanarak sisteme girebilir ve farklı roller için kayıtlar oluşturarak, bunlar için çeşitli durum çalışmaları yapabilirler.



Şekil 10: ISELTA web sitesi

SELTA web sitesi test koşumları için hazırlanan ilgili nateryaller (hatasız model, test kümeleri, test tasarımları ve ullanılan araçlar), [42]'de paylaşılmıştır. Dileyen okuyucular u sayfayı inceleyerek adımları kendi başlarına erçekleştirebilirler.

Deneysel çalışmalar kapsamında, ISELTA web sitesine ait oplam 12 adet hatayı gösteren mutant sistemler kullanılmıştır. Bu mutant sistemlerle ilgili bilgiler [33]'te bulunabilir. Bu atalar literatürde, fonksiyonel hata olarak tanımlanan ve istemin arzu edilen sonuç olayına ulaşmasına ancak istenilen ıktıları üretememesine neden olan durumlardır.

Elde edilen sonuçlar Tablo 3'te verilmektedir. Deneysel alışmalar kapsamında ESG, FSM ve RE modelleri ve bunlara it mutasyon skoru, hata kapsama, test kümesi büyüklüğü, test retim zamanı ve test koşum zamanları kıyaslanmıştır. FSM abanlı test üretimi için Graphwalker [34] isimli model-tabanlı est aracı kullanılmıştır. RE modelinden test üretimi için [35] alışması kapsamında geliştirilen model-tabanlı test üretim racı kullanılmıştır. Model büyüklükleri olarak ESG, 90 tepe e 133 ayrıt, FSM, 60 tepe ve 112 ayrıt, RE ise toplam 283 laydan oluşmaktadır.

Tablo 3: Gerçek Sistem Test Sonuçları

	ESG	FSM	RE
Model Büyüklüğü	90/133	60/112	283
Mutasyon Skoru	1	0.92	0.75
Hata Sayısı/Kapsama	12/12	12/11	12/9
Test Kümesi Büyüklüğü	240	767	228
Test Üretim Zamanı (milisaniye)	320	3870	263
Test Koşum Zamanı (saniye)	116	133	47

Tablo 3'te verilen bilgilere göre ESG modelinden üretilen test ümelerinin mutasyon skoru 1'dir ve tüm hatalar akalanmıştır. Bunların içinde RE modelinden üretilen test ümeleri en düşük mutasyon skoruna ve hata yakalama abiliyetine sahiptir. Test kümesi büyüklüğü olarak ESG ile RE birbirine yakın iken, FSM tabanlı Graphwalker ile erçekleştirilen test kümesi 767 olaydan oluşmaktadır. Ayrıca est üretim zamanı olarak FSM diğerlerinin yaklaşık olarak 10 atıdır. Bunun sebebi Graphwalker aracının ilk çalışma anında eçen zamanın çok fazla olmasıdır. Test koşumları açısından RE tabanlı model en düşük değere sahiptir. Burada ortaya ıkan ilginç sonuç ESG ve FSM'in test koşum zamanlarının ok yakın olmasıdır. Bunun ana sebeplerinden birinin test ümelerinde hatalı olaya gelindiğinde ilgili test dizisinin levam ettirilmek yerine o anda kesilmesi ve devamında gelen est dizisine geçilmesidir. Çünkü hatayı yakalayan test dizisi çin devamında gelen olayların bir önemi olmamaktadır.

Elde edilen sonuçlar kapsamında, ESG modeli ve buna bağlı est üretim algoritması [36], [37] diğer modellere göre gerek laha kompakt gerekse hata yakalama kabiliyeti olarak daha aşarılı test kümeleri üretimine olanak sağlamaktadır.

4. Tartışma

4.1. Sonuçlar ve Çıkarımlar

Uçtan uca model tabanlı bir test oluşturulmuş, bir çalıştırma yöntemi geliştirilmiş ve bu yöntemi örnekleyen bir çalışma ile sonuçlar elde edilmiştir. Diğer modellerden üretilen veya doğrudan sisteme verilen birleşik bir model (ESG) yaklaşımın ana girdisidir. Çalışmada dönüşümleri mümkün kılacak algoritmalar Bölüm 3.2'de karmaşıklık analizleri ve sözde kodlarıyla birlikte sunulmuştur. Bununla beraber, bir test oluşturma yöntemi, ESG modeli kullanılarak incelenmiştir. Test oluşturma yönteminden üretilen test grubunun kalitesi, mutasyon testinin kullanılmasıyla değerlendirilmiştir. Bu çabaların, yazılım mühendisliğinin GUI testi alanında çalışan daha geniş bir kitle için model tabanlı testi daha erişilebilir hale getireceğini umuyoruz.

Yöntemimizin sonuçlarının kalitesine ve yazılım testi alanında çalışmalar yapan insanların bu yöntemi benimsemesine bağlı olarak, daha önce başka çalışmalarda kullanılmış mevcut sistem modelleri, bu modellerin verdiğimiz yöntemler yardımıyla ESG'ye dönüşümü sayesinde kullanabilir hale gelmiştir. Böylece uçtan uca test oluşturma ve çalıştırma yöntemimizin uygulanabilir olması bizim açımızdan çalışmamızın en iyi sonuçlarından biridir. Ayrıca bu çalışma yazılım test topluluğuna model tabanlı testin kullanımının kolaylığı ve bunun değerlendirmesi hakkında bir görüş sunmaya da imkân vermiştir.

Öne sürülen yaklaşım, GUI tabanlı sistemlerin çok ötesinde, makale kapsamında incelenen modellerin diğer tüm uygulama alanları için geçerlidir. Mevcut çalışmanın, test koşum aşaması farklı sistemlere uyarlanarak uçtan uca gereken tüm adımlar gerçekleştirilebilir. Zaten başka alanlara uygulanması, planlanan çalışmalar arasındadır.

4.2. Geçerliliğe Yönelik Tehditler

4.2.1. Sonuç Geçerliliği

Vaka çalışmamızın örneklem büyüklüğü, metodolojiyi genelleştirmek için potansiyel bir tehdittir. Çalışmada kullanılan örneklemimizin küçük olması nedeniyle, yöntemi doğrulamak ve gözden kaçırabileceğimiz olası sorunları bulmak için, her model (FSM, HFSM, EFG, ESG, RE) için büyük boyutta vaka çalışmalarına ihtiyacımız vardır. Bu olası problem nedeniyle, çalışmalarımızı ileride daha büyük test senaryolarıyla genişletmeyi planlıyoruz. Böylece, örnek model boyutu gerçek hayattaki sistemlere benzer boyutta olacak ve bu durumdaki sorunlar tespit edilebilecektir.

4.2.2. İçsel Geçerlilik

Model tabanlı testin yapısı, doğası gereği içsel geçerliliğe yönelik bir tehdittir. Çünkü yaklaşımın tamamı, gerçek GUI programı yerine modeller üzerinde çalışır. Beyaz kutu test (White-Box Testing) yöntemi ile yazılım kodunda testler çalıştırıyormuş gibi model bazlı testlerle bir sistemi tam olarak test etmek mümkün değildir. Bir model, gerçek yazılım davranışının yalnızca bir temsili ve soyutlamasıdır. Test edilen yazılımın karmaşıklığına bağlı olarak, sistemi doğru şekilde temsil edecek doğru bir model oluşturmak zor olabilir. Bu nedenle, sistemin başlangıç modelinin doğruluğu önemlidir ve yöntemimiz için bir tehdittir. Bir model sistemi yanlış şekilde temsil ediyorsa, tüm dönüştürmeler ve test oluşturma / çalıştır

aklaşımı sistemi olması gerektiği gibi kapsamayacaktır. Önerilen yaklaşımın sistemi tam olarak temsil eden uygun modellerde uygulanmasını sağlamak için daha gelişmiş örnek bir sistem için modellerimizi oluşturacağız.

.2.3. Dışsal Geçerlilik

Yaklaşımı çalışmanın içeriği dışında uygulamak, dış geçerliliğe yönelik bir tehdittir. Daha önce de belirtildiği gibi, mevcut çalışma, çoğunlukla oyunların GUI'sinde kullanılan görsel nitelikler ve bunların semantiği gibi diğer hata türleri üzerine, işlevsel ve operasyonel hataları tespit etmeyi amaçlamaktadır. Böylece, mevcut çalışma daha çok menü ağırlıklı uygulamalar için uygundur. Bu tip uygulamalar GUI tabanlı uygulamaların büyük bir çoğunluğunu oluşturmaktadır. Bu, model tabanlı testin ne için oluşturulduğu ile ilgilidir. Modeller işlevsel olarak sistemleri temsil ettiğinden, bir sistemin ekrandaki görsel öğelerinin test edilmesi bu yaklaşım için uygun olmayabilir. Normalde, kod tabanlı bir test yaklaşımı bu tür doğrulamalar için daha uygun olabilir. Bununla birlikte, sıralı ve davranışsal modeller Petri-Nets modellemesinden ziyade bir test yönteminde kullanıldığında, önerilen bu yöntem uygulanabilir.

.2.4. Yapısal Geçerlilik

Modellerin ESG modeline dönüştürülmesi, farklı seviyedeki ifadelerin zorlukları nedeniyle yapısal geçerlilik açısından tehdit oluşturmaktadır. Dönüştürme işleminden sonra, modellerin ifade gücü birleşik modelin ifade gücüne göre artar veya azalır. Bu da çıkarılan modelde bazı sistem özelliklerinin eksik olmasına neden olabilir. Dönüşüm sırasında bu ifade gücünün hangi düzeyde kaybedilebileceğini anlamak için daha büyük örneklem boyutuyla ve farklı durumlarda test edilmesi gereklidir. Daha önce de bahsettiğimiz gibi, ifade gücünün makul bir noktanın ötesine geçmesini önlemek ve bu konunun üstesinden gelmek için çalışmalarımızı daha büyük örneklerle genişletmeyi planlıyoruz.

5. Sonuç

Bu çalışmada, tasarımları yine bu makalede anlatılan ve GUI yazılım testi alanında iyi bilinen GUI test modellerini analiz etmeyi amaçladık. İlk analiz, farklı modellerin kullanımının farklı yetenekler gerektirdiğini ve farklı sözdizim ve semantikle sonuçlandığını göstermektedir. Bu farklılıklar, modellerin temsil etme yeteneklerini ve test üretme ve sıralıştırma gibi diğer süreçlerini etkiler. Deneyimlerimize dayanarak önerilen yaklaşımda ESG'yi seçmemizin temel nedeni hem test üretme hem de çalıştırma açısından diğer modellere kıyasla daha uygun olmasıdır.

Önerilen yaklaşımın test hazırlama adımında, modeller ESG'ye dönüştürülüp test kümesi üretimi için hazır hale getirilmiştir. Ardından test oluşturma adımında orijinal modelden mutanlar üretilmiş ve bu orijinal ESG modelinden test dizileri oluşturulmuştur. Son olarak, üretilen test dizileri, test dizilerinin kalitesini değerlendirmek için ESG modelinin mutantları üzerinde çalıştırılmıştır. Deneysel çalışmalar göstermektedir ki öne sürülen yaklaşım model-tabanlı test için uygundur ve etkilidir. Örnek durum üzerinde gerçekleştirilen değerlendirme çalışmasında, denk mutantlar gelen test sürecinden çıkarıldıktan sonra mutasyon skoru en yüksek seviyede elde edilmiştir. Ayrıca gerçek bir sistem üzerinde gerçekleştirilen deneysel çalışma sonuçları göstermektedir ki, ESG tabanlı test kümeleri diğer yöntemlere göre daha kompakt ve hata yakalama kabiliyeti açısından daha başarılı sonuçlar vermektedir.

Bu çalışmanın devamında öncelikle bu makalede tanıttığımız uçtan uca model tabanlı test üretme ve çalıştırma yönteminin adımlarının otomatik olarak uygulanmasına imkân verecek yeni bir aracın geliştirilmesi hedeflenmektedir. Bir diğer çalışma da ise sonuçların geçerliliğini arttırmak amacıyla daha büyük ölçekli modellerin kullanılması planlanmaktadır. Öte yandan katmanlı ESG modellerinin ve topluluk belirleme (community detection) algoritmalarının kullanılması testlerin ölçeklendirilmesine de katkı verebilir. Bununla ilgili elde ettiğimiz ilk sonuçlar [32]'de aktarılmıştır. Bu çalışmada tanıttığımız yönteme [32]'deki yaklaşımımızı da dahil ederek yine mevcut yöntemin daha büyük modeller üzerine uygulanabilmesini kolaylaştırmayı hedefliyoruz.

6. Kaynaklar

[1] I. Banerjee, B. Nguyen, V. Garousi, and A. Memon, "Graphical user interface (GUI) testing: Systematic mapping and repository," Information and Software Technology 55, no. 10, pp. 1679-1694, 2013.

[2] Harrison, R., Flood, D. & Duce, D. Usability of mobile applications: literature review and rationale for a new usability model. J Interact Sci 1, 1 (2013). https://doi.org/10.1186/2194-0827-1-1

[3] R. K. Shehady, D. P. Siewiorek, "A method to automate user interface testing using variable finite state machines," Proceedings of IEEE 27th International Symposium on Fault Tolerant Computing, Seattle, WA, USA, pp. 80-88, 1997.

[4] T. S. Chow, "Testing software design modeled by finite-state machines," in IEEE Transactions on Software Engineering, vol. SE-4, no. 3, pp. 178-187, May 1978.

[5] L. White, H. Almezen, "Generating test cases for GUI responsibilities using complete interaction sequences," Proceedings 11th International Symposium on Software Reliability Engineering. ISSRE 2000, San Jose, CA, USA, pp. 110-121, 2000.

[6] F. Belli, "Finite state testing and analysis of graphical user interfaces," Proceedings 12th International Symposium on Software Reliability Engineering, Hong Kong, China, pp. 34-43, 2001.

[7] A. M. Memon, M. E. Pollack, M. L. Soffa, "Hierarchical GUI test case generation using automated planning," in IEEE Transactions on Software Engineering, vol. 27, no. 2, pp. 144-155, February 2001.

[8] A. Memon. "An event‐flow model of GUI‐based applications for testing," Software testing, verification and reliability 17.3 pp. 137-157, September 2007.

[9] Q. Xie, A. M. Memon, "Using a pilot study to derive a GUI model for automated testing," ACM Trans. Software Eng. Methodol. 18, 2, pp. 1–35, November 2008.

[10] S. Huang, M. B. Cohen and A. M. Memon, "Repairing GUI test suites using a genetic algorithm," 2010 Third International Conference on Software Testing, Verification and Validation, Paris, pp. 245-254, 2010.

[11] F. Belli, M. Beyazit, N. Güler, "Event-Oriented, model-based GUI testing and reliability assessment-approach and case study," Advances in Computers, 85, pp. 277-326, 2012.

12] F. Belli, M. Beyazıt, C. J. Budnik, T. Tuglular, "Advances in model-based testing of graphical user interfaces," In Advances in Computers, vol. 107, pp. 219-280. Elsevier, 2017.
13] F. Belli, N. Güler, and M. Linschulte, "Layer-centric testing," FERS-Mitteilungen: Vol. 30, No. 1, pp. 55-62, 2012.
14] A. M. Memon, M. L. Soffa, and M. E. Pollack, "Coverage criteria for GUI testing," Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering, pp. 256-267, September 2001.
15] A. M. Memon, "GUI testing: pitfalls and process," in Computer, vol. 35, no. 8, pp. 87-88, August 2002.
16] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines-a survey," in Proceedings of the IEEE, vol. 84, no. 8, pp. 1090-1123, August 1996.
17] S. Fujiwara, G. v. Bochmann, F. Khendek, M. Amalou and A. Ghedamsi, "Test selection based on finite state models," in IEEE Transactions on Software Engineering, vol. 17, no. 6, pp. 591-603, June 1991.
18] M. Utting, A. Pretschner, and B. Legeard, "A taxonomy of model ‐ based testing approaches," Software testing, verification and reliability 22.5, pp. 297-312, 2012.
19] F. Belli, M. Beyazit and A. Memon, "Testing is an Event-Centric Activity," 2012 IEEE Sixth International Conference on Software Security and Reliability Companion, Gaithersburg, MD, pp. 198-206, 2012.
20] J. E. Hopcroft, R. Motwani, and J. D. Ullman, Automata theory, languages, and computation." International Edition 24.2.2, 2006.
21] ISELTA websitesi, Çevrimiçi olarak mevcut: http://iselta.ivknet.de, Son erişim: 21.03.2022.
22] O. Kilincceker, A. Silistre, M. Challenger and F. Belli, "Random Test Generation from Regular Expressions for Graphical User Interface (GUI) Testing," 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Sofia, Bulgaria, pp. 170-176, 2019.
23] G. Mercan, E. Akgündüz, O. Kilincceker, M. Challenger, and F. Belli, "Android uygulaması testi için ideal test ön çalışması," CEUR Workshop Proceedings, 2018.
24] O. Kilincceker, and F. Belli, "Grafiksel kullanıcı arayüzleri için düzenli ifade bazlı test kapsama kriterleri," CEUR Workshop Proceedings, 2017.
25] O. Kilinccceker, E. Turk, M. Challenger and F. Belli, "Regular expression based test sequence generation for HDL program validation," 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, pp. 585-592, 2018.
26] O. Kilincceker, E. Turk, M. Challenger and F. Belli, "Applying the ideal testing framework to HDL programs," ARCS Workshop 2018; 31th International Conference on Architecture of Computing Systems, Braunschweig, Germany, pp. 1-6, 2018.
27] O. Kilincceker and F. Belli, "Towards uniform modeling and holistic testing of hardware and software," 2019 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, pp. 1-6, 2019.
28] Brüggemann-Klein, A. (1993). Regular expressions into finite automata. Theoretical Computer Science, 120(2), 197-213.
[29] M. Linschulte, On the Role of Test Sequence Length, Model Refinement, and Test Coverage for Reliability (PhD Thesis, Univ. Paderborn), 2014.
[30] F. Belli, A.T. Endo, M. Linschulte, and A. Simao, A Holistic Approach to Model-Based Testing of Web Service Compositions, Software: Practice and Experience, vol.44, no.2, pp. 201–23, 2014.
[31] A. Silistre, O. Kilincceker, F. Belli, M. Challenger and G. Kardas, "Models in Graphical User Interface Testing: Study Design," 2020 Turkish National Software Engineering Symposium (UYMS), Istanbul, Turkey, 2020, pp. 1-6, doi: 10.1109/UYMS50627.2020.9247072.
[32] A. Silistre, O. Kilincceker, F. Belli, M. Challenger, and G. Kardas, "Community Detection in Model-based Testing to Address Scalability: Study Design", 15th Conference on Computer Science and Information Systems (FedCSIS 2020), Track on Software and Systems Engineering, Advances in Software and Systems Engineering (ASSE 2020), Sofia, Bulgaria, 2020, IEEE, pp. 657-660, DOI: 10.15439/2020F163.
[33] O. Kilincceker, A. Silistre, F. Belli and M. Challenger, "Model-Based Ideal Testing of GUI Programs–Approach and Case Studies," in IEEE Access, vol. 9, pp. 68966-68984, 2021, doi: 10.1109/ACCESS.2021.3077518.
[34] Karl, K., 2013. Graphwalker. *URL: www. graphwalker. org* Son erişim: 21.03.2022.
[35] Kilinccceker, O., Turk, E., Challenger, M., & Belli, F. (2018, July). Regular expression based test sequence generation for HDL program validation. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 585-592). Ieee.
[36] Budnik, C. J. (2006). Test generation using event sequence graphs (Doctoral dissertation, University of Paderborn, Germany).
[37] Belli, F., Nissanke, N., Budnik, C. J., & Mathur, A. (2005). Test generation using event sequence graphs. University of Paderborn, Institute for Electrical Engineering and Information Technology.
[38] Rodríguez-Valdés, O., Vos, T. E., Aho, P., & Marín, B. (2021, September). 30 years of automated GUI testing: a bibliometric analysis. In International Conference on the Quality of Information and Communications Technology (pp. 473-488). Springer, Cham.
[39] Vos, T. E., Aho, P., Pastor Ricos, F., Rodriguez ‐ Valdes, O., & Mulders, A. (2021). testar ‐ scriptless testing through graphical user interface. Software Testing, Verification and Reliability, 31(3), e1771.
[40] Chahim, H., Duran, M., Vos, T. E., Aho, P., & Fernandez, N. C. (2020, September). Scriptless testing at the GUI level in an industrial setting. In International Conference on Research Challenges in Information Science (pp. 267-284). Springer, Cham.
[41] Aho, Alfred V., Lam, Monica S., Sethi, Ravi., & Ullman, Jeffrey D. (2007). Compilers Principles, Techniques, & Tools. Addison-Wesley.
[42] Emo dergi Github materyaller, Çevrimiçi olarak mevcut: https://github.com/alpersilistre/gui-emo-dergi, Son erişim: 21.03.2022.
[43] Kilincceker, O., Turk, E., Belli, F. et al. Model-based ideal testing of hardware description language (HDL) programs. Softw Syst Model (2021). https://doi.org/10.1007/s10270-021-00934-6

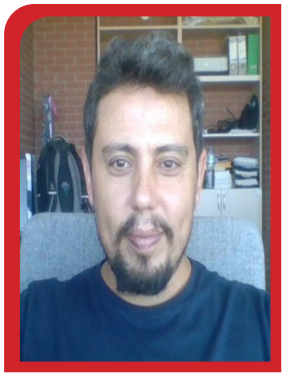
Özgeçmişler

Alper Silistre, 2015 yılında İzmir Ekonomi Üniversitesi'nden yazılım mühendisliği lisans derecesini almıştır. Aynı zamanda Yazılım Mühendisidir. Araştırma ilgi alanları arasında yazılım testi ve yazılım mühendisliği yer almaktadır.

Fevzi Belli (Üye, IEEE) Berlin Teknik Üniversitesi'nden bilgi teknolojisi ve bilgisayar bilimleri alanında B.S., M.S., Ph.D. ve Habilitation (Alman doktora sonrası) derecelerini aldı. Halen Paderborn Üniversitesi ve İzmir Yüksek Teknoloji Enstitüsü'nde Yazılım Mühendisliği Bölümü'nde Fahri Profesör olarak görev yapmaktadır. Araştırma, geliştirme ve yazılım mühendisliği öğretme, doğrulama ve doğrulama, hata toleransı ve kalite güvence konularında 35 yıldan fazla deneyime sahiptir. Uçak endüstrisinde bir programcı olarak başladı ve simülasyon ortamları oluşturmak ve güvenlik açısından kritik özellikleri doğrulamak için programlar yazdı. 1983 yılında Bremerhaven'daki Uygulamalı Bilimler Üniversitesi'nde Profesörlük unvanını aldı; 1989'da Paderborn Üniversitesi'ne geçti. Ayrıca uzun yıllar Maryland Üniversitesi, College Park, MD, ABD ve Avrupa Bölümü'nde Öğretim Üyesi olarak görev yaptı. Aynı zamanda İzmir Ekonomi Üniversitesi Bilgisayar Bilimleri Bölümünün Kurucu Başkanıydı. Yazılım güvenilirliği/hata toleransı, model tabanlı test ve test otomasyonu konularına ilgi ve deneyime sahiptir.

Geylani Kardaş, 2001 yılında Ege Üniversitesi Bilgisayar Mühendisliği Bölümü'nden mezun olduktan sonra bilgi teknolojileri alanındaki yüksek lisans ve doktora derecelerini yine Ege Üniversitesi'nden sırasıyla 2003 ve 2008 yıllarında almıştır. Halen Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü'nde (UBE) doçent doktor olarak görev yapmaktadır ve UBE Yazılım Mühendisliği Araştırma Laboratuvarı'nın (Ege-SERLab) yöneticisidir. Araştırma alanları arasında etmen tabanlı yazılım mühendisliği (AOSE), model güdümlü mühendislik (MDE), alana-özgü diller / alana-özgü modelleme dilleri (DSL'ler / DSML'ler) ve düşük kodlu yazılım geliştirme yer almaktadır. Bu araştırma alanlarında 100'ün üzerinde hakemli makalenin yazarı veya ortak yazarıdır. Dr. Kardaş aynı zamanda Elsevier yayınevinin "Journal of Computer Languages" dergisinin MDE ve DSL bölümünden sorumlu yardımcı editörüdür.

Moharram Challenger (Üye, IEEE) Şubat 2016'da Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü'nden bilgi teknolojisi alanında doktora derecesini aldı. 2005-2009 yılları arasında İAÜ-Shabestar Üniversitesi Bilgisayar Mühendisliği Bölümü'nde Öğretim Üyesi ve Kıdemli Öğretim Görevlisi olarak görev yaptı. 2010-2013 yılları arasında Slovenya ile Türkiye arasında yürütülen ikili bir projede (TÜBİTAK) Araştırmacı ve Takım Lideri olarak görev yaptı. 2012-2016 yılları arasında TÜBİTAK tarafından finanse edilen ulusal bir projeye ve Avrupa'da ITEA ModelWriter ve ITEA Assume adlı iki uluslararası yazılım yoğun projeye liderlik eden UNIT IT Ltd.'de Araştırma ve Geliştirme Direktörü olarak görev yaptı. 2017-2018 yılları arasında Ege Üniversitesi'nde Yardımcı Doçent olarak öğretim üyesi olarak görev yapmıştır. Ocak 2019'dan Temmuz 2020'ye kadar, Flanders Make projelerinde PACo ve DTDesign olarak çalışan Antwerp Üniversitesi'nde Doktora Sonrası Araştırmacı olarak görev yaptı. Halen Antwerp Üniversitesi Bilgisayar Bilimleri Bölümü'nde görev süresi boyunca Yardımcı Doçent olarak görev yapmaktadır. İlgi alanları alana özgü modelleme dilleri, çok etmenli sistemler, siber-fiziksel sistemler ve Nesnelerin İnterneti'dir.



Onur Kılınçeker, M.Sc. (Bilgisayar Müh.), B.Sc. (Matematik) ve Lise. (Elektronik) derecelerine sahiptir. Antwerp Üniversitesi, Bilgisayar Bilimleri Bölümü'nde araştırmacı ve Paderborn Üniversitesi'nde doktora öğrencisidir. İlgi alanları, hem yazılım hem de donanım sistemlerinin model tabanlı doğrulaması ve geçerliliği, model tabanlı testler ve mutasyon testleridir. Şu anda, bütünsel teste (pozitif ve negatif test (diğer adıyla fuzz testi)) ve mutasyon testine dayalı yazılım ve donanım sistemlerinde belirli hataların varlığını ve yokluğunu göstermek için model tabanlı ideal test yöntemleri üzerinde çalışıyor. Aynı zamanda, Simulink modellerinin mutasyon testini endüstriyel ortama entegre etmek için EFFECTS projesinde bilimsel bir araştırmacıdır.

Makale Türü: Araştırma Makalesi
Makale Gönderim Tarihi: 06.03.2021
Makale Kabul Tarihi: 08.04.2022

Bir Uzaktan Program Yükleme ve Yönetim Sistemi

A Remote Program Installation and Management System

Nilgün İncereis[1], Bekir Tevfik Akgün[2]

[1]Bilgisayar Mühendisliği
İstanbul Okan Üniversitesi
niincereis@stu.okan.edu.tr
ORCID:0000-0001-5508-8159

[2]Bilgisayar Mühendisliği
İstanbul Okan Üniversitesi
tevfik.akgun@okan.edu.tr
ORCID: 0000-0002-9726-1340

Özet

Uzaktan program yükleme (OTA) hizmeti sistemi, belirli bir rtamdaki akıllı uçbirimlere, nesnelerin interneti (IoT) ygıtlarına yeni program yükleme, sistemi izleme ve yönetme lemini firma adına güvenlikli ve korumalı bir şekilde yerine etiren bir sistemdir. Önerilen sistem; hizmet yönetimi, IoT önetimi, kullanıcılar, yöneticiler ve uçbirimlerden luşmaktadır. Bu çalışma, önerilen uzaktan yönetim sisteminin ygulamasını ve geliştirilmesini amaçlamaktadır. Uzaktan önetim sisteminin kesintiye uğramaması için OTA işlemleri çbirim uygulama programının izin verdiği güvenli bir bölgede ürütülmektedir. Bu sistemde kullanıcılar, bulut üzerinde ulunan istediği dosyayı cihazına uzaktan yükleyebilir ve ullanıcılar sistemi uzaktan takip edebilir veya bulut ortamına stediği yazılım dosyasını gönderebilir ve isterse sisteme üklediği bu dosyayı cihazına uzaktan yükleyebilir. Önerilen istem gerçeklenmiş ve tasarlanan hizmetlerin başarıyla yerine etirildiği belirlenmiştir.

nahtar kelimeler: *Nesnelerin İnterneti, OTA Güncelleme, Tablosuz Sensör Ağı, İzleme Sistemi, Gömülü Sistem, Uzaktan Yazılım Güncelleme*

Abstract

The remote program installation (OTA) service system is a ystem that performs the process of installing new programs to mart terminal units, IoT devices in a specific environment, onitoring and managing the system on behalf of the company n a secure and protected manner. Recommended system; ervice management consists of IoT management, users, dministrators and terminal units. This study aims to implement nd develop of the proposed remote management system. In rder not to interrupt the remote management system, OTA perations are carried out in a secure area allowed by the terminal application program. In this system, users can remotely upload any file on the cloud to their device, and users can remotely monitor the system or send any software files to the cloud and upload this file to their device remotely. The proposed system has been implemented and it has been determined that the designed services have been carried out successfully.

Keywords: *Internet of Things (IoT), Over The Air (OTA) Update, Wireless Sensor Network, Monitoring System, Embedded System, Remote Firmware Update.*

1. Giriş

Günümüzde uzaktan program yükleme işleminin yaygın kullanılması nedeniyle çok önemli bir rol üstlenir. Nesnelerin İnterneti (IoT) sistemleri, internete bağlı veri ya da bilgi alışverişi yapabilen nesnelerden oluşan sistemlerde yazılım güncellemesi bir zorunluluktur. Böyle teknolojilerin kullanıldığı bir sistemi uygulamak ve geliştirmek avantajlar sağlar.

Kısıtlı bir güvenliğe sahip olunması ve güncellemenin yalnızca belirli kaynaklardan izin verilmesi firmaların uzaktan program yükleme (OTA) işlemlerini kendisinin gerçeklemesini gerektirir. Bu çalışma şekli; yeni sürümle çalıştırma gibi aygıt işlemlerini habersiz yarıda kesmenin firma için önemli bir zarara yol açmadığı durumlarda uygun olacaktır. Ancak sahada çalışan ve kendisine atanmış görevleri kesintisiz, veri kayıpsız ya da kontrol edilen sistemlerin beklenmedik davranışlar olmaması gereken durumlarda ise; aygıtın işlemlerini güvenli bir şekilde ara vermesini sağlamak gerekir. Aygıt üzerinde çalışan bir uygulamanın korumalı bir şekilde ara verilmesi ve yeniden başlatılması ayrıca tasarlanmalıdır.

Bu çalışma; bir uzaktan program yükleme yönetim isteminin tasarlanmasını ve geliştirilmesini kapsar. İlk olarak, nceki yapılan çalışmalar özet halinde verilmiştir. Sonraki ölümlerde, yerel yönetim/uzaktan erişim sistemi ile önerilen zaktan yönetim sistemi üzerine uygulamalar sunulmuştur. Önerilen uzaktan yönetim sisteminin geliştirilmesi için önetimlerin birer bulut hizmeti yazılımına dönüştürülerek amamen bulut üzerinde dağıtık bir sistem olarak luşturulduğunda sistemde oluşabilecek kesintilerin nedenleri e belirlenmiştir. Sistemin deneysel sonuçları verilerek, onuçlar bölümünde güvenlik kapsamındaki eğerlendirilmesinin yanında bu çalışmanın mevcut alışmalara olan üstün yanları ve zayıf yanları üzerinde urulmuştur.

2. Önceki çalışmalar

Nesnelerin İnterneti (IoT) teknolojisi, yerel bir kontrol isteminin kısıtlamalarına maruz kalan uygulamalar için yeni, yileştirilmiş izleme ve kontrol yöntemlerinin geliştirilmesini erekli kılmıştır. IoT ile bağlantılı gömülü uygulamaları zlemek ve kontrol etmek için tahmine dayalı sistem öneren bir alışmaya bakıldığında, kontrol sisteminin tasarımı üzerindeki onuçların ortaya çıkardığı etkilere göre bir ilk örnek model unulmuştur [1]. Bir kontrol sistemi, kontrol döngülerini ullanarak diğer cihazların ve sistemlerin davranışını yönetir, omut verir, yönlendirir ya da düzenler. Bu sistemler, bir ev tipi azanı kontrol edebilen bir termostat kullanan ev ısıtma ontrolünden, işleri ya da makineleri kontrol etmek için ullanan büyük endüstriyel kontrol sistemlerine kadar eğişebilir [2]. Kontrol sistemi, cihazı uygun değer ayarlarında abitleştirilmiş bir dizi kısıtlamaya neden olur. Bununla irlikte, kontrol sisteminin kendisi, sisteme bağlı olmayan bir izi kısıtlamaya maruz bırakır ve bu da uzun yerleşme ürelerine ya da aşımlara neden olur. Bu durumlarda gecikme e ölü zamanlar oluşabilir [3].

Kang, K. ve arkadaşları tarafından [4], farklı hizmet edarikçileri, farklı türde uygulamalar kullanarak farklı izmetler için uygulama türlerini listelemiştir. Hizmet ya da ygulama kalitesinde, güvenilirlik kullanıcılarla ilgili önemli arçalardan biridir. IoT uygulamaları geliştirmek için, endüstri arafından kullanılan çeşitli modellerin analizi yapılarak, nodellerin kullanıldığı tekniklerin karşılaştırılması, özgünlüğü e eksiklikleri durumlarına bakılmıştır [5]. IoT uygulamaları eliştirmek için tasarlanan modellerde, nesneler, aygıtlar, ablosuz iletişim ortamları, ağ geçidi, bulut, uygulama unucusu, yazılımlar, uygulamalar, kullanıcılar, yöneticiler ullanılır. Nesneler, çevredeki verileri toplar ve bunları ablosuz iletişim ortamı ve ağ geçitleri aracılığıyla bir unucuya gönderir. Buluttaki veriler, uygulamalar ve hizmetler racılığıyla depolanıp, işlenip ya da analiz edildikten sonra son ullanıcılara sunulur. Bir sistemdeki uçbirimlerin çok fazla lmasıyla, kolay bağlantı, kontrol sistemi, iletişim ve aynı amanda sağlam ve güvenli bağlantı zorlaşmaktadır. Bu edenle uygun sistem tasarım mimarisinin oluşturulması erekir. Çoğu IoT mimarisindeki zorluk ve sorun, tandardizasyonun olmamasından kaynaklanmaktadır. tandardizasyon eksikliği ve IoT uygulama geliştirmedeki orluklar [6] ve [7]'de açıklanmaktadır.

IoT platformu, makine ve aygıtların bağlanmasını, ardından makine ve sensör verilerinin toplanmasını, işlenmesini, dönüştürülmesini, düzenlenmesini ve saklanmasını sağlayan bir yazılımdır ve genellikle IoT ara katman yazılımı olarak da adlandırılır. Zdravkovic ve diğerleri [8], 16 farklı bulut tabanlı IoT platformunu sunmuşlardır. Buluttaki tüm IoT aygıtlarını kontrol eden esnek bir platform sağladığı söylenen bir çalışmada, çözüm önerisi olarak bir olaya cevap verme sürecini değiştirme esnekliği üzerinde durulmuştur. Yapılan bu çalışmada, bu sistem yerel makine üzerinde çalıştırılmış ve IBM Bluemix platformu yardımıyla da buluta getirilebileceğinden bahsedilmiştir [9]. Uzaktan kontrol edilebilen Wi-Fi, 3G, 4G ya da 5G işlevselliğine sahip akıllı aygıtların sayısı artmaktadır. IoT, bu tür aygıtların uzaktan bağlanmasını ve kontrol edilmesini sağlamaktadır [10]. Buluttaki aygıtların sayısı arttıkça, bunların yazılımının daha kontrollü güncellenmesi gerekir. Uzaktan güncellemenin uygulanmasıyla ilgili [11]'de, sistem programlarının seçici olarak yüklenmesi yapılmıştır.

Yeni teknolojilerin gelmesi ve hızlı bir şekilde ilerlemesiyle, IoT'nin ortaya çıkışı günlük yaşantının her alanında önemli bir rol oynamaktadır. Böylece, ev otomasyonu, güvenlik alanındaki IoT çalışmalar ya da bazı özellikler canlıların yaşam tarzını da kolaylaştırır. Wi-Fi teknolojisi kullanılarak, ev otomasyonu kontrolünün uygulanması ve tasarlanmasında IoT, tümleşik, korumalı ve güvenli sisteme sahip aygıtları yönetir. Örneğin IoT ve mikro denetleyici tabanlı bir NodeMcu (Esp8266) ve bir Android mobil uygulaması kullanılarak güvenlik ve otomasyonda nasıl uygulanabileceği hakkında [12]'de bilgi verilmektedir.

Bulut hizmetleri ile IoT hizmetlerine küresel erişimin mümkün olduğu ve böyle hizmetler için sunucu bakımının zorluğunun ortadan kaldırılabildiği söylenmektedir. Özellikle, böyle sistemlerde genel *ip*'yi ayrı ayrı sağlamadan küresel erişim mümkündür, bu da onu bireysel ya da küçük işletme IoT hizmeti kuruluşu için çok uygun hale getirir [13]. Tasarım aşamasında bulut üzerinde uygulama çalıştırmanın maliyeti, oluşturulan sisteme göre [14]'te tahmin edilmektedir. Dell [15] tarafından kullanılan mimaride ise, ağın kenarındaki verileri güvence altına almak amacıyla uygulamalar için özel ağ geçidi kullanılır.

IoT yönetim sistemlerinin esnekliği, hata toleransı, yüksek algılama doğruluğu, düşük maliyetli ve hızlı dağıtım özellikleri, uzaktan yönetim için birçok yeni ve hayatımızı kolaylaştıran uygulama alanı yaratır. Ancak IoT yönetim sisteminin gerçekleştirilmesi, hata toleransı, ölçeklenebilirlik, maliyet, donanım, çevre ya da güç tüketimi gibi faktörlerin getirdiği kısıtlamaları da karşılamalıdır. Bu kısıtlamalar IoT için yeni kablosuz geçici ağ oluşturma teknikleri oluşturulmasını gerektirir [16].

Uzaktan program yükleme ile ilgili yapılan bir çalışmada [17], sistem tasarımı için, genel IoT sistemleri, yerel sistem, yerel yönetim/uzaktan erişim sistemi, uzaktan yönetim sistemi ve bulut üzerinden dağıtılmış yönetim sistemi olarak incelenmiştir. Burada, sahadaki IoT uçbirimlere, yeni program yükleme işlemlerini firma adına güvenli bir şekilde yerine getiren bir sistem için "kullanıcılar (program kodu) ve OTA hizmeti" ya da "OTA hizmeti ve IoT uç birimleri (donanım)"

ırasındaki güvenli haberleşmenin üzerinde durulur. Güvenlik roblemleri, uçbirimin kötü amaçla saldırıya uğrayabilmesi ya la istenmeyen bir kod yükleme sorunlarının ortaya çıkması eklinde düşünülmektedir. Son yıllarda da OTA (Over the Air) le ilgili bazı çalışmalar [18, 19, 20, 21] incelendiğinde veri letişimi, iletişim modeli, güvenlik ve protokoller kapsamında la giderek artan bir ilgi bulunmaktadır.

3. Kullanılan teknolojiler

Bu bölümde, nesnelerin interneti teknolojisi, bulut eknolojisi, uzaktan program yükleme, ESP8266, OTA ütüphaneleri, fonksiyonları ve işlevleri konuları üzerinde lurulmuştur.

3.1. Nesnelerin interneti teknolojisi

Nesnelerin interneti teknolojisi, kesintisiz bir ortam ağlamak için, veri alışverişi yapmada birbirleriyle etkileşim çinde olan elektronik cihazları ya da nesneleri güvenli bir ekilde yönetmeyi amaçlar. Bunlar kablosuz ya da kablolu ağlar ullanılarak uzaktan algılanabilir ve kontrol edilebilir [22]. nternete bağlandıktan sonra, çeşitli hedeflere, kaynaklara veri üklenebilir ya da indirilebilir. Bu tür sistemler [1], bir merkezi ygıtın bir sunucu olarak hareket ettiği ve aygıtlar arasında stekleri ilettiği bir sunucu istemci mimarisi kullanır.

3.2. Bulut teknolojisi

Bulut bilişim fikrinin temelleri 1950'li yıllarda ortaya ıtılarak veri merkezlerinin modernize edilmesiyle ilk gerçek ulut bilişim hizmeti oluşturulmuştur. Bulut bilişim eknolojisinde büyük veriler, internette depolanabilir ve stenirse bu verilere erişilebilir. Bulut teknolojisinin çeşitleri, şağıdaki şekilde verilmektedir [23].

Public Cloud (Genel Bulut): İnternet üzerindeki sunucular ile urulan bir bulut teknolojisidir.

Private Cloud (Özel Bulut): Bilgileri önemli olan büyük irketlerin tercih ettiği bir bulut teknolojisidir. Tüm bilgilere rişim güvenliği ve gizliliği yüksektir.

Hybrid Cloud (Melez Bulut): Genel ve Özel Bulut irleşiminden ortaya çıkan bulut teknolojisidir.

Community Cloud (Topluluk Bulut): Birkaç şirket ile ortak ullanılan hizmetleri barındıran bulut teknolojisidir. Topluluk iyeleri, uygulama ve verilere erişebilmektedir.

3.3. Uzaktan program yükleme

Wi-Fi ile internete çıkabilen *ESP8266* serisi tüm devre artları, hem boyut olarak küçük hem de fiyat olarak ucuz olduğundan yaygın olarak kullanılmaktadır. *ESP8266*'nın iretiliş amacı, öncelikle Arduino'yu *Wi-Fi* üzerinden internete ağlamaktı. Ancak şimdi, *Xbee*, *Ethernet kartı* gibi pahalı arçalar kullanmak yerine daha ucuz olan bu birimi kullanmak istem maliyetini azaltmaktadır. Arduino IDE üzerinden *ESP8266* kütüphanelerini, yükleyerek, *NodeMCU* kartlar rogramlanır. Bir kez programlandıktan sonra, internet üzerinden OTA ile tekrar programlanabilir duruma gelir. Dosya sistemi; yapılandırma dosyalarını, Web sunucusu içeriğini ve diğer dosya verilerini depolayarak yönetir. Arduino ortamında kullanılan dosya yerleşimi Şekil 1'de [24] göstermektedir.

Şekil 1: Arduino Ortamında Kullanılan Dosya Yerleşimi

3.4. ESP8266

ESP8266, seri haberleşme ile kablosuz olarak internet ağına bağlanabilen bir birimdir. Ucuz ve kolay kullanımından dolayı IoT projelerinde çok yaygın olarak kullanılmaktadır. Arduino'yu internete bağlamanın en kolay ve en ucuz yolu *ESP8266 Wi-Fi* birimini kullanmaktır. Ortamda bulunan kablosuz ağlara bağlanabileceği gibi, kendi internet ağını yayarak diğer aygıtların bu ağa bağlanabilmesine de olanak sağlamaktadır [17].

3.5. OTA kütüphaneleri, fonksiyonları ve işlevleri

ESP8266httpUpdate kütüphanesinde, *ESPhttpUpdate* sınıfı bulunmaktadır. Bu sınıf güncellemeleri kontrol edebilir ve *http web* sunucusundan bir ikili dosya indirebilir. Güncellemeler, ağdaki ya da internetteki herhangi bir *ip* ya da alan adından indirilebilmektedir. Bu kütüphanede basit ve güvenlikli güncelleme işlemlerini yapan fonksiyonlar bulunmaktadır. Basit ve güvenlikli güncelleme işlemlerini yapan fonksiyonlar aşağıda verilmektedir.

a) Basit Güncelleme İşlemi

- ESP8266HTTPUpdate::update(const char * url, const char * current_version, const char * httpsFingerprint)

b) Güvenlikli Güncelleme İşlemi

- ESP8266HTTPUpdate::update(const char * host, uint16_t port, const char * url, const char * current_version, bool https, const char * httpsFingerprint)
- ESP8266HTTPUpdate::update(String host, uint16_t port, String url, String current_version, bool https, String httpsFingerprint)

Basit güncelleyici, işlev her çağrıldığında dosyayı indirir. Güvenlikli güncelleyici ise, güncelleme işlevi için sunucudaki bir komut dosyasını göstermesi mümkündür. Sunucu tarafı komut dosyası, güncelleme yapılıp yapılmayacağını kontrol etmektedir. Sunucu isteği işlemede, basit güncelleyici için sunucunun sadece güncelleme için ikili dosyayı vermesi gerekir. Güvenlikli güncelleme işlemini yapan fonksiyonlardaki *httpsFingerprint* parametresi, sertifikanın *SHA1* parmak izidir. Güvenlikli güncelleme yönetimi için bir *betiğin* (örneğin bir PHP betiği) sunucu tarafında çalıştırılması gerekir. Her güncelleme talebinde *ESP*, *HTTP* başlıklarındaki bazı bilgileri sunucuya gönderir [17]. Örnek başlık verileri aşağıdaki şekilde verilmektedir.

[HTTP_USER_AGENT] => ESP8266-http-Update
[HTTP_X_ESP8266_STA_MAC] => 18:FE:AA:AA:AA:AA
[HTTP_X_ESP8266_AP_MAC] => 1A:FE:AA:AA:AA:AA

HTTP_X_ESP8266_FREE_SPACE] => 671744
HTTP_X_ESP8266_SKETCH_SIZE] => 373940
HTTP_X_ESP8266_CHIP_SIZE] => 524288
HTTP_X_ESP8266_SDK_VERSION] => 1.3.0
HTTP_X_ESP8266_VERSION] => DOOR-7-g14f53a19

Bu bilgilerle *PHP betiği*, bir güncelleme gerekip erekmediğini kontrol edebilir. *MAC* adresine göre farklı ikili losyalar teslim etmek de mümkündür.

3.6. OTA işlemlerinde diğer güvenlik öntemleri

OTA işlemlerinde güvenlik için bazı çözüm önerileri ulunmaktadır. Bunlar, güvenli önyükleme *V1*, güvenli nyükleme *V2* ve OTA aracılığıyla güncelleme şeklindedir.

- *Güvenli Önyükleme V1 (Secure Boot V1):* Güvenli Önyükleme, entegre devre üzerinde sadece kodunuzun alışabilmesini sağlayan bir özelliktir ve *AES* tabanlı güvenli nyükleme şemasını kullanılır. Flash'tan yüklenen veriler her ıfırlamada doğrulanır. Güvenli Önyükleme, Flash Şifreleme zelliğinden ayrıdır ve Flash içeriği şifrelenmeden güvenli nyüklemeyi kullanabilirsiniz. Fakat güvenli bir ortam için her kisinin de aynı anda kullanılması gerekmektedir. [25]

- *Güvenli Önyükleme V2 (Secure Boot V2):* Güvenli Önyükleme V2, *RSA* tabanlı uygulama ve önyükleyici loğrulamasını kullanır. Ayrıca, önyükleyiciyi imzalamadan *SA* şemasını kullanarak uygulamaları imzalamak için bir eferans olarak da kullanılabilir. Güvenli Önyükleme, nyüklenmekte olan her bir yazılım parçasının imzalanmış lup olmadığını kontrol ederek bir aygıtı herhangi bir yetkisiz yani imzasız) kod çalıştırmaya karşı korur. Bir *ESP32*'de, bu azılım parçaları, ikinci aşama önyükleyiciyi ve her bir ygulama ikili dosyasını içerir. İlk aşama önyükleyicinin *ROM* odu olduğu için imza gerektirmediğini ve dolayısıyla leğiştirilemeyeceğine dikkat edilmesi gerekmektedir. Güvenli Önyükleme V2'nin avantajları aşağıdaki gibidir. [26]

- RSA ortak anahtarı cihazda depolanır. İlgili *RSA* özel anahtarı gizli bir yerde tutulur ve cihaz tarafından asla erişilmez.
- Üretim sırasında entegre devre üzerinde yalnızca bir ortak anahtar oluşturulabilir ve saklanabilir.
- Uygulamalar ve yazılım önyükleyici için aynı görüntü formatı ve imza doğrulama yöntemi uygulanır.
- Cihazda hiçbir sır saklanmaz. Bu nedenle, zamanlama veya güç analizi gibi pasif yan kanal saldırılarına karşı duyarsızdır.

- *OTA Aracılığıyla Güncelleme:* IoT cihazları dağıtıldıktan onra, yeniden programlamak ya da güncellemek için fiziksel rişim olmayabilir. Önceden plan yapmak ve gömülü sistemi a da IoT cihazlarını güncellemek için güvenli bir nekanizmaya sahip olmak çok önemlidir. OTA aracılığıyla güncelleme, donanım yazılımını (*Firmware*) şifreler. [27]

4. Yerel yönetim/uzaktan erişim sistemi üzerine bir uygulama

Tümleşik yönetim başlığı altında, genel IoT sistemlerinde tanıtılmış olan yerel yönetim/uzaktan erişim sisteminin [17] önerilen IoT hizmeti şeklinde uygulanması gerçekleştirilmiştir.

4.1. Tümleşik yönetim ortamı

Tümleşik yönetim; yönetimin yürütüldüğü bilgisayarlar ve yazılımlar, kullanıcılar, internet erişim ortamı ve kurulu sahada bulunan uçbirimlerden oluşan bir yapı üzerinde çalışır. Örnek kurulu sahada, uç birimler ve bir adet *PC*, veri tabanı, hizmet yönetimi ve *IoT* yönetimi bulunmaktadır. Burada *PC*'ye uzaktan bağlı bir yönetici vardır.

Tümleşik yönetim sisteminde *Xampp* kullanılmıştır. *Xampp*, yerel web sunucusu oluşturmayı sağlayan, *Apache* web sunucusu ve diğer uygulamaları kapsayan bir dağıtımdır. Bu isterlerle gerçekleştirilen uygulamada, sunucu yazılımı *PHP* dili ile yazılmıştır. Verileri tutmak için hem açık kaynak kodlu olan *PHP* ile hem de *PHP* ile çok iyi bir şekilde çalışan *MySQL* veri tabanı seçilmiştir. Uçbirimler için, *NodeMCU* cihazları kullanılmıştır. Uçbirimlerin yazılım ve donanım olarak program yüklemeye hazır olması gerekir.

4.2. Tümleşik yönetimde yapılan işlemler

Tümleşik yönetimde yapılan temel işlemler, kullanıcı işlemleri, hizmet yönetim işlemleri, IoT yönetim işlemleri ve uçbirim işlemleridir. Tablo 1'de, yapılan temel işlemler ve alt işlemleri verilmektedir.

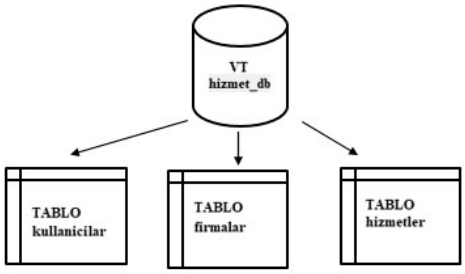
Tablo 1: Tümleşik Yönetimde Yapılan İşlemler

Yapılan Temel İşlemler	Alt İşlemler
Kullanıcı İşlemleri	• Kullanıcı kaydı • Firma kaydı/kendine ait olanları listeleme • Hizmet alma/kendine ait olanları listeleme • IoT uçbirim listesi • Yeni sürüm program gönderme • Uçbirim yeni sürüm program sorgulama/durum
Hizmet Yönetim İşlemleri	• Tüm kullanıcıların listesi • Tüm firmaların listesi • Tüm hizmetlerin listesi • Kullanıcıdan gelen hizmet talebini onaylama • IoT uçbirim listesi • Uçbirim kayıtları/durumları listesi • Kullanıcıdan gelen istekleri IoT yönetime bildirme

IoT Yönetim İşlemleri	• Yeni sürüm program sorgulama/alma • Yeni sürüm program gönderimi (uçbirime) • Uçbirim kaydı alma • Uç birim kayıtlarını/durumlarını hizmet yönetimine bildirme
Uçbirim İşlemleri	• Kendini kayıt etme • Yeni sürüm program sorgulama • Yeni sürüm program alma (OTA)

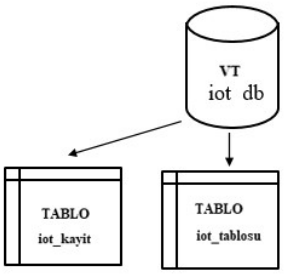
4.3. Tümleşik yönetim veri tabanı tasarımı

Tümleşik yönetimde, hizmet yönetimi için *hizmet_db* adlı veri tabanı oluşturulmuştur ve Şekil 2'de verilmektedir.



Şekil 2: Tümleşik Yönetimdeki *hizmet_db* Adlı Veri tabanı

Tümleşik yönetim sisteminin IoT yönetimi için *iot_db* adlı veri tabanı oluşturulmuştur ve Şekil 3'te verilmektedir.



Şekil 3: Tümleşik Yönetimdeki *iot_db* Adlı Veritabanı

4.4. Tümleşik yönetimde senaryolar

Tümleşik yönetimde senaryolar aşağıdaki şekilde oluşturulmuştur:

a) Kullanıcı kaydı, firma kaydı ve hizmet alımı olmayan kullanıcının sistemde işlem yapamaması.
b) Firma hizmet alım talebini gönderdikten sonra firmanın hizmeti kullanabilmek için yöneticinin onay vermesini beklemesi.
c) Uçbirimlerin bağlantısının kesilmesi durumunda bağlantı tekrar geldiğinde işlemine kaldığı yerden devam edebilmesi.
d) Uçbirimlerin sisteme kendini kaydettirmesi.
e) Kullanıcıların uçbirimlere doğrudan müdahale edememesi ve kullanıcılar isteklerini hizmet yönetimine bildirerek IoT yönetimi üzerinden uçbirimlere iletmesi.

4.5. Kullanıcının sisteme kayıt olma durumu

Geliştirilen sistem; kayıtlı firmalar üzerinden kullanıcılara hizmet sunmaktadır. Hizmet satın alan kullanıcılar, kendi sistemine dosya gönderebilir, uçbirimlere yeni sürüm yükleyebilir, uçbirimleri listeleyebilir ve uçbirimlerin durumlarını görebilir. Bu sistemde, kullanıcı ya da yönetici girişi yapılabilmektedir. Eğer kayıtlı bir kullanıcı değilse yeni kullanıcı kaydı yapılarak giriş yapılabilmektedir. Kullanıcı firma kaydı yapmışsa kayıtlı olan firmalarını listeleme yapabilmektedir. Kullanıcı firma seçerek o firma için hizmet alımı yapabilmektedir. Sistemdeki hizmet işlemleri panelinde, kullanıcı eğer hizmet alımı yaptıysa hizmetleri listeleyebilmektedir.

Sistemdeki firmaların hizmet listesinde hizmet talebi yapılmış fakat hizmeti kullanabilmek için ödeme durumunda yöneticinin onay vermesi beklenmektedir. Yönetici ödeme durumunu ödendi olarak değiştirirse ödemesi yapılan hizmet için kullanıcı, dosya gönderme, yeni sürüm yükleme, cihaz durumu ve cihaz listesi hizmetlerinden var olanları kullanabilmektedir.

4.6. Kullanıcı işlemleri

Kullanıcılar, dosya gönderme, yeni sürüm yükleme, cihaz durum ve cihaz listesi hizmetlerini aldığında ve yönetici ile ilgili gerekli anlaşmalar tamamlandığında firma adına bu hizmetleri kullanabilmektedir. Dosya gönderme hizmeti, kullanıcıların sisteme dosya yükleme yapabildikleri hizmettir. Kullanıcının sisteme gönderdiği dosyanın boyutunda kısıtlama yapılmıştır. Kullanıcı, 4 MB' dan büyük dosyaları sisteme yükleyemez. Bu boyut yönetici tarafından değiştirilebilir.

Yeni sürüm yükleme hizmeti, kullanıcının sistemdeki dosyalardan istediğini istediği uçbirimine yükleme isteğini hizmet yönetimine bildirilmesidir. Hizmet yönetimi de IoT yönetimine bu isteği ileterek IoT yönetimi onay verirse uçbirime yükleme yapılmaktadır. Cihaz durum hizmeti, kullanıcı tarafından herhangi uçbirimin verilerinin gözlenebilmesidir. Burada programa göre kurgulanmış sensörden gelen veriler eş zamanlı gözükmektedir. Cihaz listesi hizmetinde, kullanıcının kendine ait uçbirimlerinin listesini görebilmesidir. Bu listede cihazın aktif ya da pasif olması durumları eş zamanlı gözükmektedir.

Kullanıcı hizmet satın aldığında IoT yönetim paneline erişebilmektedir. Kullanıcı, sisteme dosya göndermek istediğinde hizmet satın alınmış ve yönetici tarafından onay verilmişse, dosya gönderme işlemi kullanıcı tarafından yapılabilmektedir.

4.7. Yönetici işlemleri

Yöneticinin yapabildiği işlemler, tüm kullanıcıları listeleme, tüm firmaları listeleme, hizmetleri listeleme, hizmet yönetimi, firma kaydı şeklindedir.

Tüm kullanıcıların listesine bakıldığında, yönetici ve kullanıcı şeklinde yetki alanı oluşturulmuştur. Yönetici isterse kullanıcıların bilgilerini güncelleyebilmektedir. Yönetici isterse istediği kullanıcıyı sistemden silebilir. Yönetici isterse hizmetlerin bilgilerini güncelleyebilmektedir. Yönetici isterse

stediği hizmeti sistemden silebilir. Kullanıcıların istediği ızmeti kullanabilmesi için hizmet talebi "Var" şeklinde ve ɔdeme durumu yönetici tarafından "Ödendi" olarak naylanmış olması gerekmektedir. Örneğin *kullanıcı_id* değeri *hizmet_id* değeri 30 olan kullanıcı ödeme durumu "Ödendi" azsa da cihaz durum hizmetinde "Yok" yazdığı için cihaz lurum hizmeti kullanılamaz.

4.8. Tümleşik Yönetimde Kullanılan Cihazlar ve Teknolojiler

Tümleşik yönetimde, kurulu saha içinde, veri tabanı, ızmet ve IoT yönetimlerini barındıran PC ve uç birim olarak NodeMCU cihazları kullanılmıştır. NodeMCU üzerine üklenen uygulamalar Arduino'da yazılmıştır. Bu sistemde, veb sunucu yazılımı olan Xampp ile yerel bir şekilde ɔilgisayarda çalışılmıştır. Xampp'da, phpmyadmin kurulu bir ekilde gelmektedir. Phpmyadmin, Php ile yazılmış açık :aynak kodlu bir araçtır. MySQL veri tabanının yönetimi bu ıraç üzerinden sağlanmaktadır.

4.9. Tümleşik yönetimde kullanılan fonksiyonlar ve kütüphaneler

Hizmet Fiyatını Hesaplayan Fonksiyon: Kullanıcı hizmet eçimini yaptıktan sonra fiyatını hesaplayan basit bir fonksiyon azılmıştır. Bu fonksiyonda 4 adet hizmet durumu bulunur.

Tümleşik yönetim sistemi uygulaması için kullanılan onksiyonlar Tablo 2'de açıklamalarıyla verilir.

Tablo 2: Kullanılan Fonksiyonlar

Kullanılan Fonksiyonlar	Açıklama
void wifi_baglan()	WiFi bağlantısı yapar
void iot_guncelle(String ver)	IoT güncelleme işleri yapar
void iot_islemler()	IoT okuma ve IoT yazma işlemi yapar
void updateUI()	Arayüz güncelleme
void handlePage()	El sıkışma

Bu sistemde kullanılan kütüphaneler ise Tablo 3'te erilmiştir.

Tablo 3: Kullanılan Kütüphaneler

Kullanılan Kütüphaneler	Açıklama
#include <ESP8266WiFi.h>	WiFi ile ilgili işlevleri yapmak için gereklidir.
#include <WiFiClient.h>	Web tarayıcısına istek göndermek için gereklidir.
#include <ESP8266WebServer.h>	ESP8266WebServer, web sunucusu
#include <ESP8266HTTPClient.h>	ESP8266HTTP, http İstemcisi
#include <ESP8266HTTPUpdateServer.h>	ESP8266HTTPUpdate Server, Güncelleme Sunucusu
#include <ESP8266httpUpdate.h>	ESP8266httpUpdate, http güncelleme
#include <EmbAJAX.h>	Ajax kütüphanesi

Bu çalışmada, IoT aygıtlarının yeni program güncellemesi kablosuz olarak yapılması gerekir. Bu da, aygıtların kötü amaçla saldırıya uğrayabileceği ya da başka bir program yüklenme ihtimallerini ortaya çıkarmaktadır. Saldırıya uğramanın olasılığını azaltmak için, bu yüklemeleri bir şifreyle, belirli bir OTA bağlantı noktası vb. seçerek korunmalıdır. Güvenliği artırmak için, Arduino OTA kütüphanesi ile sağlanan işlevsellik kontrol edilebilir [24].

void setPort(uint16_t port);
void setHostname(const char hostname);*
void setPassword(const char password);*

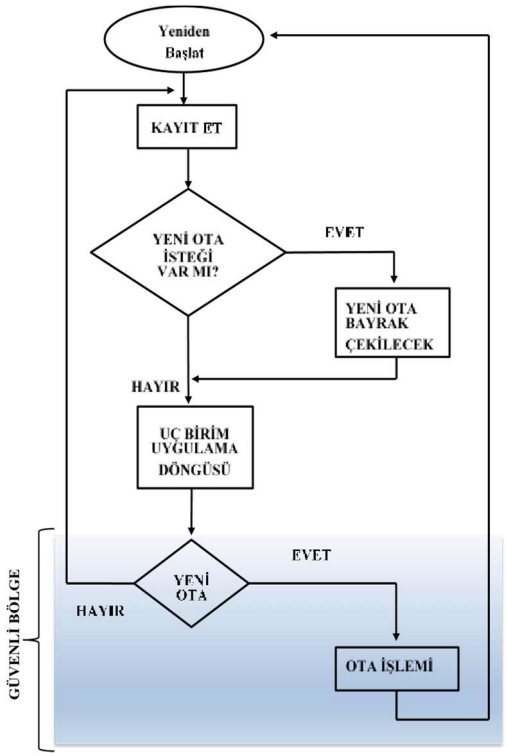
Bunun gibi, belli koruma işlevleri önceden oluşturulmuştur ve geliştirici tarafından herhangi bir ek kodlama gerektirmez. *Arduino OTA* ve *espota.py*, karşıdan yüklemeyi doğrulamak için *Digest-MD5* kullanılır. Aktarılan verilerin bütünlüğü, *ESP* tarafında *MD5* kullanılarak doğrulanır [24].

4.10. Tümleşik yönetimin çalışma prensibi

Bu çalışmanın birinci aşamasında, hizmet yönetimi ve IoT yönetimi aynı *PC* üzerinde bulunmaktadır. Her iki yönetimin de ayrı ayrı veri tabanı oluşturulmuştur. Aslında tek bir veri tabanı da yeterli olur. Çünkü bütün sistem yereldedir. Sonraki aşama uygulamalarına yönelik veri tabanları ve yönetim sistemleri ayrı ayrı oluşturulur. Bu sistemde, kullanıcılar uzaktan sisteme erişebilmektedir. Kullanıcıların doğrudan IoT yönetime müdahale edebilmesi güvenlik nedeniyle istenmez. Sisteme çok fazla cihaz bağlanması durumunda, bu cihazlara tek tek elle program yüklemek iş yükünü artırır ve zaman kaybına neden olur. Belki de bu cihazlara yüklenmesi gereken programların eş zamanlı yüklenmesi gerekebilir. Parça parça programlar büyük bir sistemi oluşturabilir.

.11. Uç birimde OTA sağlanması

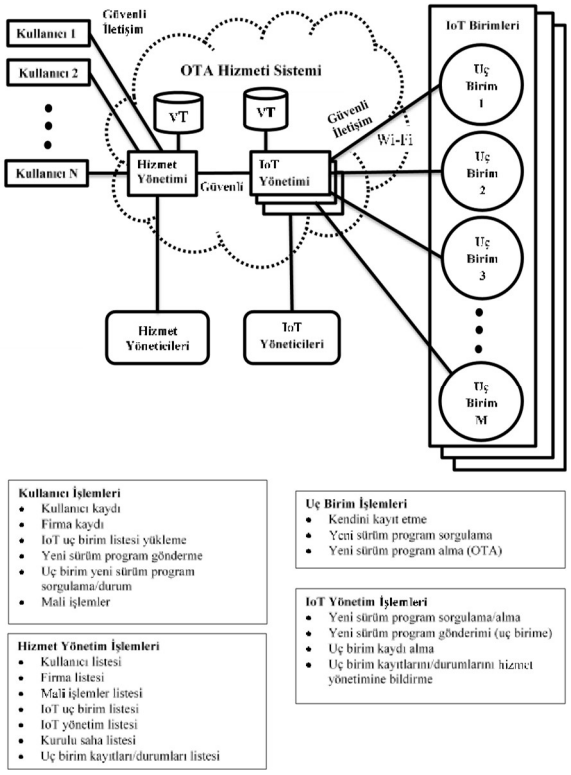
Şekil 4'te verilen, uçbirim uygulama programı bir sonsuz öngü içinde çalışır. Uygulamanın kesintiye uğramaması için)TA işlemleri uçbirim uygulama programının izin verdiği üvenli bir bölgede yürütülür. Eğer bir OTA işlemi bir başka eğişle yeni bir program yüklenmesi gerçekleşecek ise uçbirim ygulama programının gereken önlemlerini örneğin aktif ıkışlarının pasife alınması ve verilerini almasına, saklamasına zin verilir. Uygulama programının izin verdiği aşamada OTA erçekleşir ve sistem yeniden başlat noktasından (Reset) alışmaya başlar.



Şekil 4: Uçbirimde OTA Gerçekleştirilmesi

5. Önerilen uzaktan yönetim sistemi üzerine bir uygulama

Uzaktan program yükleme (OTA) hizmeti sistemi, ahadaki IoT uçbirimlere yeni program yükleme işlemini firma dına güvenilir bir şekilde yerine getiren bir sistemdir. Önerilen zaktan program yükleme hizmetinin sistem tasarımı Şekil 5'te erilmektedir. OTA hizmeti sistemi; hizmet yönetimi, IoT önetimi, kullanıcılar, yöneticiler ve uçbirimlerden luşmaktadır. Hizmet ve IoT yönetimlerinin, ayrı ayrı hizmet e IoT yöneticileri vardır. OTA hizmeti sistemi olarak, çbirimlerle kullanıcı arasında, doğrudan veri alışverişi üvenlik nedeniyle kısıtlanmış ve işlemlerin hizmet yazılımı racılığı ile kullanıcılar, kendi uygulamalarına ait işlemlerinin, aberleşmesinin yapılması sağlamıştır. OTA hizmeti aberleşmesini ayrı bir ortam üzerinden yürütür. OTA uygulamasında, kullanıcı A'nın istediği uçbirime program yüklemesi için, kullanıcı A, "Sisteme kayıt yaptırdı mı?", "Kayıtlı ise sözleşmeleri imzaladı mı?" veya "Almak istediği hizmet için ücretini ödedi mi?" gibi mali aşamaları da geçmesi, sonra da isteğini hizmet yönetimine bildirmesi istenir. Eğer gelen uygun bir istek ise hizmet yönetimi isteği ve program dosyasını IoT yönetimine aktarır. IoT yönetimi de gelen programı uçbirimlere gönderir. Sonuç olarak, uçbirimde gerçek program yüklemesi yapılır.



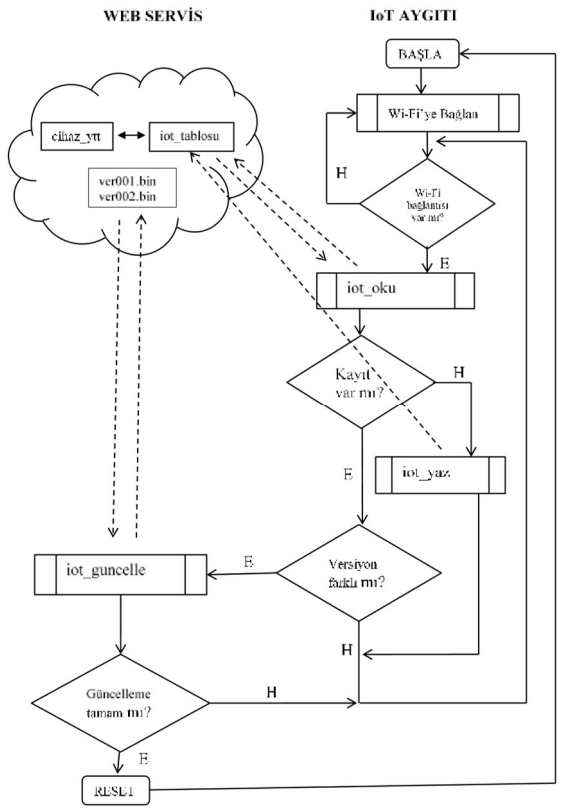
Şekil 5: Önerilen Uzaktan Program Yükleme Hizmeti

Burada Tümleşik sisteme benzer olarak *PHP* ve *MySQL* tabanında uygulanması gerçekleşmektedir. Bu seçim; Uzaktan Yönetim Sisteminin bir PC üzerinde gerçeklemesi sağlandığı gibi kolaylıkla Bulut üzerine alınarak Web Hizmeti Sunucuları üzerinde de gerçekleşmesine olanak sağlamaktadır. Çalışma öncelikle *PC* üzerinde uzaktan yönetim sisteminin gerçeklemesi üzerinde tamamlanmıştır. Daha sonra bir Web Sunucu üzerine yazılımlar taşınarak yönetim sisteminin çalışması gözlenmiştir.

5.1. Önerilen uzaktan yönetim sisteminin çalışma prensibi

Önerilen uzaktan yönetim sisteminin çalışma prensibi Şekil 6'da gösterilmektedir. Bu sistem çalışmaya başladığında önce, IoT tarafında *Wi-Fi* bağlantısı fonksiyonu çalışır. *Wi-Fi* bağlantısı yoksa tekrar başa döner. Eğer *Wi-Fi* bağlantısı var ise *iot_oku* fonksiyonu çalışır. *iot_oku* fonksiyonu bulut üzerindeki *iot_tablosu* ile bağlantılıdır. *iot_oku* fonksiyonu,

uluta istek gönderir ve cevabını alır. Daha sonra *iot_oku* onksiyonunun işleri bitince kayıt varsa ve sürüm farklıysa *ot_guncelle* fonksiyonu çalışır. Eğer güncelleme tamamsa ihaz *reset*'e gider ve sonra tekrar sistemin çalışması başa löner. Sistemin çalışmasındaki diğer bir yol ise, sistem alışmaya başladığında *Wi-Fi* bağlantısı varsa, *iot_oku* onksiyonu işini bitirince eğer kayıt yoksa *iot_yaz* fonksiyonu alışır. *iot_yaz* fonksiyonu işlerini bitirince *Wi-Fi* bağlantısı var ı noktasına geri döner. Sistemin çalışmasındaki diğer tüm ollar şekil üzerinde gösterilmektedir. Bu sistemde, *ot_oku.php*, *iot_yaz.php* ve *iot_guncelle.php* dosyaları bulut zerinde bulunmaktadır ve PHP dilinde yazılmıştır. Aynı amanda *ver001.bin* ve *ver002.bin* uzantılı dosyaları da bulut zerinde bulunmaktadır. *iot_guncelleme* fonksiyonu ile bulut zerinde bulunan sürüm cihaz üzerindekinden farklı ise üncelleme başarılı şekilde yapılabilmektedir.



Şekil 6: Önerilen Uzaktan Yönetim Sisteminin Çalışma rensibi

Aşağıdaki Tablo 4'te sistemde kullanılan tetikleyiciler ve QL sorguları gösterilmektedir.

*Tablo 4:*Tetikleyiciler

Tetikleyiciler	Sql Kod
istekEkle	INSERT INTO cihaz_ytt(ip, iot_adi, ver, yeniver, Date, Time, durum, TimeStamp, islem) VALUES (NEW.ip, NEW.iot_adi, NEW.ver, NEW.yeniver, NEW.Date, NEW.Time, NEW.durum, NEW.TimeStamp, 'EKLE')
istekGuncelle	INSERT INTO cihaz_ytt(ip, iot_adi, ver, yeniver, Date, Time, durum, TimeStamp, islem) VALUES (NEW.ip, NEW.iot_adi, NEW.ver, NEW.yeniver, NEW.Date, NEW.Time, NEW.durum, NEW.TimeStamp, 'GÜNCELLE')
istekSil	INSERT INTO cihaz_ytt(ip, iot_adi, ver, yeniver, Date, Time, durum, TimeStamp, islem) VALUES (OLD.ip, OLD.iot_adi, OLD.ver, OLD.yeniver, OLD.Date, OLD.Time, OLD.durum, OLD.TimeStamp, 'SİL')

5.2. Önerilen uzaktan yönetim sistemindeki kullanıcı ekranı

Bu sistemde kullanıcılar, uzaktan bulut üzerinde bulunan istediği dosyayı cihazına uzaktan yükleyebilir ve kullanıcılar sistemini uzaktan takip edebilir veya bulut ortamına istediği yazılım dosyasını (OTA hizmeti bulunan ".bin" dosyası) gönderebilir ve isterse sisteme yüklediği bu dosyayı cihazına uzaktan yükleyebilir. Bu sistem bu tür hizmet almak isteyenler için hazırlanmış bir uzaktan yönetim sisteminin hizmet yazılımıdır. Sisteme kaydı olmayan sistemi hiç kullanamaz ya da sistemin sınırlı özelliklerini kullanabilir.

5.3. IoT cihaz yönetim paneli

Kullanıcılar veya yöneticiler, IoT cihaz yönetim panelinden sisteme bağlı cihazları görebilir.

5.4. IoT cihazların durum takibi

Bu sistemde IoT cihazların takibi yapılmaktadır. Cihazlar sisteme hazır duruma getirildikten sonra bu sisteme bağlandığında cihazların bilgileri veri tabanına otomatik olarak kaydedilir. Cihazın *ip* adresi, *MAC* adresi, üzerindeki sürüm adı, tarih, zaman ve durum bilgileri tutulur. Cihaz için yeni sürüm talebi yapıldığında belli bir süre sonra sistem yeni sürümü cihaza yükleyinceye kadar tekrar tekrar çalışacaktır. Eğer yeni sürüm talebi verildikten sonra internet bağlantısını kesersek ve sonra cihazı tekrar sisteme bağlarsak cihaz kaldığı yerden yeni sürüm talebini yükleyinceye kadar tekrar tekrar çalışacaktır.

5.5. Kullanıcıların sisteme dosya yükleyebilme ekranı

Kullanıcıların sisteme dosya yükleyebilecekleri ekranlar oluşturulmuştur ve nasıl yükleme yapılacağı bu ekranlardan görülmektedir.

5.6. Yönetici ekranı

Yönetici sisteme bağlı olan tüm cihazları yönetici ekranından görebilir. Sisteme 3 adet *NodeMCU* cihaz bağlandıktan sonra sistemde bu cihazların aktif olduğu gözlenmiştir. Bu sistemde cihazların sürümlerini değiştirebileceğimiz ekran da bulunmaktadır.

Yerel yönetim/uzaktan erişim sisteminde, IoT yönetimi PC üzerinde bulunmaktadır. Kullanıcılar PC'ye bağlı uygulamalar ya da programlar vasıtasıyla isteklerini IoT yönetimine iletirler. *PC* yerine *RaspberryPi* gibi cihazlar kullanılarak IoT yönetimi işlemleri daha küçük ve maliyeti daha az olan cihazlar da kullanılabilir. Aynı zamanda birden çok *PC* kullanmak yerine *RaspberryPi* kullanılması hizmet almak isteyen kişilerin ya da şirketlerin maliyetini düşürür.

5.7. Uzaktan yönetim sisteminin bir web sunucusuna taşınması

Bu çalışmada *nilguninceeis.com* alan adı, web sunucu hizmeti "www.ekonomikhost.net" [28] adresinden sağlanmıştır.

Öncelikle alınan hizmet karşılığında verilen *http://nilgunincereis.com:8880* adresinden, *Plesk* web sunucu paneline [29] girilerek, kullanıcı adı ve şifre ile giriş yapılmaktadır.

6. Önerilen uzaktan yönetim sisteminin geliştirilmesi

Önerilen uzaktan yönetim sisteminin geliştirilmesi için yönetimlerin birer bulut hizmeti yazılımına dönüştürülerek tamamen bulut üzerinde dağıtık bir sistem olarak oluşturulduğunda sistemde oluşabilecek kesintilerin nedenleri belirlenmiştir. Bunlar:

- IoT yönetimi ve uçbirimler arasında güvenli iletişimin kopması ya da iletişimin bilinmeyen güvensiz bir iletişime dönüşmesi,
- OTA hizmeti sistemi içindeki hizmet yönetimi ile IoT yönetimi arasında güvenlik protokolünün kopması ya da bilinmeyen bir protokole dönüşmesi,
- Kullanıcılar ve hizmet yönetimi arasında kullanıcıdan kaynaklı güvenli iletişimin bozulması,
- Kullanıcılar ve hizmet yönetimi arasında güvenli iletişimin kopması ya da iletişimin bilinmeyen güvensiz bir iletişime dönüşmesi,
- İletişimin gecikmesi,

olarak belirlenmiştir.

Daha güvenli bir sistem oluşturmak için güvenlik seviyeleri oluşturulmalı ve bu seviyelerde uygulanması gereken tedbirler belirlenerek hangi seviyede ise o seviyedeki tedbirler uygulanmalıdır.

7. Deneysel sonuçlar

Sistemin deneysel sonuçlarını görebilmek için oluşturulan sistem 21 gün boyunca incelenmiştir. Bu süre boyunca sisteme 4 adet *NodeMCU* cihazı bağlanmış ve sistemde bu cihazların aktif ya da pasif olduğu durumlar görülmektedir. Bu cihazların 455 işlemi aktif durumdayken 45 işlemi pasif durumdadır. Bunun yanında sistemde ekleme, silme ve güncelleme işlem türlerini ve 15 farklı *ip* bilgisini içeren toplam 500 işlem yapıldığı tespit edilmiştir.

Tablo 5'te görüldüğü gibi, sistem üzerindeki programların *Versiyon 1* ve *Versiyon 2* türlerinin işlem durumları incelenmiştir. *Versiyon 1*'deki işlemlerin ekleme sayısı 16, silme sayısı 13, güncelleme sayısı 299 iken, *Versiyon 2*'deki işlemlerin ekleme sayısı 13, silme sayısı 13, güncelleme sayısı 146 olmuştur. *Versiyon 1*, *Versiyon 2*'ye göre %91 daha fazla işleme sahiptir.

Tablo 5: Versiyonların İşlem Durumlarının Sayıları

	Versiyon 1	Versiyon 2
Ekleme	16	13
Silme	13	13
Güncelleme	299	146
Toplam İşlem	328	172

OTA kapsamındaki uygulamalarda; güvenlik, güvenilirlik, doğrulama, içerik dağıtım ağı (CDN) ve bulut yükü dengesi gibi güncel problemler için çözüm önerilerinin deneysel çalışmaları da yapılması gerekmektedir. Bu çalışmada, OTA işlemlerindeki güvenlik yöntemlerinin önemi vurgulanmaktadır.

8. Sonuçlar

Günümüzde kullanımı yaygınlaşmış IoT sistemlerine ve uçbirimlerine yönelik doğrudan program yükleme işlemleri firma içinde kapalı olarak yapılmaktadır. Firma içinde yapılmasının en önemli nedeni güvenlikli bir ortamda program yüklemenin basitçe gerçeklenmesidir. Ancak uygulamada çeşitli sahalara yerleştirilmiş IOT cihazlarına yeni bir program sürümünün doğrudan yüklenmesi maliyetli ve kolay olmayan bir işlem olacaktır. Bu çalışmada uzaktan program yüklemeyi karşılayan OTA işlemlerinin bir hizmet olarak sunulması önerilmiştir. Güvenlikli uzaktan program yükleme işlemlerinin firma içinde tasarlama, geliştirme ve bakım maliyetlerinin karşılanması gerekmektedir. Firma içinde çözülmesi gereken OTA işlemlerinin, her türlü çalışma türlerine göre kesintisiz ve hatasız yerine getirilmesini ve güvenlik koşullarını da sağlayan genel kullanıma açık, maliyeti düşük hizmetler olarak sunan bir bilişim sistemi tasarlanıp gerçeklenmiştir. Bu çalışma ile ilk defa bir açık kullanımlı uzaktan program yükleme hizmeti sistemi geliştirilmiştir. Geliştirilen bilişim sistemi benzer başka problemlere de uyarlanabilir. Sistemin faydalanıcıları olarak; özellikle gelişmiş OTA sistemi gerçekleme ve sürdürme maliyetlerine girmesi mümkün olmayan mikro düzeyde veya küçük düzeyde firmaların projelerinde kullanımına bir hizmet sunulmuş olmaktadır. Bu ortamda küçük adımlarla başlayan bir firma projeleri geliştikçe artan OTA hizmeti gereksinimi

arşılanmakta ve sadece kendisine yönelik bir alt sistem urulmasına imkân verilecek şekilde büyümesi ölçeklenebilmektedir. Çalışmada tasarlanan sistem, *MYSQL* eri tabanı kullanan *PHP* dilinde yazılmış bir uzaktan yönetim istemi yazılımıdır. Bu sistemin geliştirilmesi için yönetim azılımları birer bulut hizmeti yazılımına dönüştürülebilir.

Güvenlik kapsamı bakımından program yükleme işlemi bir ifreyle ve belirli bir OTA bağlantı noktası seçimi ile orunmaktadır. Güvenliği daha da artırmak için, *Arduino OTA* ütüphanesi ile sağlanan işlevsellikler kontrol edilmektedir. Bu alışmanın mevcut çalışmalara olan üstün yanları, belli koruma şlevleri önceden oluşturularak, geliştirici tarafından her hangi ir ek kodlamaya gerek duyulmamasıdır. Ayrıca, Önerilen zaktan yönetim sisteminin kesintiye uğramaması için OTA şlemleri uçbirim uygulama programının izin verdiği güvenli ir bölgede yürütülmüştür. Gerçeklenen sistemde tüm ileşenlerin sayısı arttıkça zaman bakımından kayıp vermeden aha güvenli bir sistemi oluşturma ve yönetmenin zorluğu ardır. Çalışmada bunları karşılayacak ileri uygulama önerileri e verilmiştir.

9. Kaynaklar

1] S., Wagle, T., Sathe, G., Vamburkar, A., Gaikaiwari, "Regression Based Prediction Algorithm for Remote Controlling of IoT Based Applications", 2015 International Conference on Computing and Network Communications (CoCoNet), IEEE, 2015.

2] Wikipedia – Control Systems, 2021.

3] M., Bak, "Control systems with constraints", PhD Thesis, Technical University of Denmark, 2000.

4] K., Kang, Z., Pang, L. D., Xu , L., Ma, C., Wang, "An Interactive Trust Model for Application Market of the Internet of Things", IEEE Transactions On Industrial Informatics, VOL. 10, NO. 2, 2014.

5] A., Behura, A., Narayan, A. K., Ray S. K., Pani, "A Complete Model for IOT Application", Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS 2017), ISBN:978-1-5386-1959- 9, IEEE Xplore Compliant - Part Number:CFP17M19-ART, 2017.

5] S. A., Al-Qaseemi, H.A., Almulhim, M.F., Almulhim, S.R., Chaudhry, "IoT Architecture Challenges and Issues: Lack of Standardization", Future Technologies Conference, San Francisco, United States, IEEE, 2016.

7] C., Wang, M., Daneshmand, M., Dohler, X., Mao, R. Q., Hu, H., Wang, "Guest editorial - special issue on Internet of things (IoT): Architecture protocols and services", IEEE Sensors Journal, vol. 13, no. 10, pp. 35053510, 2013.

8] M., Zdravkovic, M., Trajanovic, J., Sarraipa, R., Lezoche M., Jardim-Gonçalves, "Survey of Internet-of-Things platforms", 6th International Conference on InformationSociety and Techology, ICIST 2016, Kopaonik, Serbia, pp. 216-220, 2016.

9] A., Rajalakshmi, H., Shahnasser "Internet of Things using Node-Red and Alexa", 2017 17th International Symposium on Communications and Information Technologies (ISCIT), 2017.

[10] S., Dey, A., Roy, S., Das, "Home automation using Internet of Things", Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual, 2016.

[11] G., Cuijuan, M., Changyun, W., Zhigang, X., Lina, "Research and Implementing of Remote Update for Network Telephone System Program Based on Ethernet", 2010 2nd International Conference on Signal Processing Systems (ICSPS), 2010.

[12] W., Tarneberg, V., Chandrasekaran, M., Humphrey, "Experiences creating a framework for smart traffic control using AWS IOT", IEEE/ACM 9th International Conference on Utility and Cloud Computing, 2016.

[13] D. H., Kang, M. S., Park, H. S., Kim, D.Y., Kim, S.H., Kim, "Room temperature control and fire alarm/suppression IoT service using MQTT on AWS", International Conference on Platform Technology and Service, 2017.

[14] H., He, Z. H., Ma, X., Li, H., Chen, W., Shao, "An approach to estimating cost of running cloud applications based on AWS", 19th Asia-Pacific Software Engineering Conference, 2012.

[15] https://www.businesswire.com/, "Dell Enters Embedded PC Market with New Embedded Box PCs, Helping Smart Systems Connect to the Internet of Things", 06.03. 2021,

[16] I.F., Akyildiz, W., Su , Y., Sankarasubramaniam, E., Cayirci, "Wireless sensor networks: a survey", Elsevier, Computer Networks, 38 (2002) 393–422, 2002.

[17] N. İncereis, B. T., Akgün, "A Remote Program Loading Service Design and Implementation", Turkish National Software Engineering Symposium (UYMS), IEEE, 2020.

[18] A., Kanda, T., Kurafuji, K., Takeda, T., Ogawa, Y., Taito, K., Yoshihara, M., Nakano, T., Ito, H., Kondo, T., Kono, "A 24-MB Embedded Flash System Based on 28-nm SG MONOS Featuring 240-MHz Read Operations and Robust Over-the-Air Software Update for Automotive Applications", IEEE Solıd-State Cırcuıts Letters, Vol. 2, NO. 12, 2019.

[19] X., Cao, J., Xu, K., Huang, "Cooperative Interference Management for Over-the-Air Computation Networks", IEEE Transactıons On Wıreless Communıcatıons, Vol. 20, NO. 4, 2021.

[20] A., Ghosal, S., Halder, M., Conti, "STRIDE: Scalable and Secure Over-The-Air Software Update Scheme for Autonomous Vehicles", ICC 2020 - 2020 IEEE International Conference on Communications (ICC), IEEE, 2020.

[21] M. L., Manna, P., Perazzo, L., Treccozzi, G., Dini, "Assessing the Cost of Quantum Security for Automotive Over -The-Air Updates", 2021 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2021.

[22] J. A., Stankovic," Research directions for the internet of things", Internet of Things Journal, IEEE, 2014.

[23]https://www.endustri40.com/bulut-bilisim-cloudcomputing-nedir/, 04.03.2021.

[24]https://media.readthedocs.org/pdf/arduinoesp8266/latest/arduinoesp8266.pdf, Ivan Grokhotkov, ESP8266 Arduino Core Documentation Release 2.5.038-g95cf925, Feb 26, 2019.

[25]https://docs.espressif.com/projects/esp-idf/en/latest/esp32/security/secure-boot-v1.html, 27.01. 2022.

[26]https://docs.espressif.com/projects/esp-idf/en/latest/esp32/security/secure-boot-v2.html, 27.01. 2022.

[27]https://www.lab4iot.com/2021/02/21/esp32-secure-firmware-update-over-the-air-ota/, 27.01. 2022.

[28] www.ekonomikhost.net, Ekonomikhost İnternet ve Bilişim Hizmetleri, 2021.

[29] https://www.plesk.com, Plesk International GmbH, 2021.

Özgeçmişler

Bekir Tevfik Akgün, İstanbul D.M.M.A. Elektrik Mühendisliği Lisans eğitimini 1981 yılında, İstanbul Teknik Üniversitesi Kontrol ve Bilgisayar Mühendisliği Yüksek Lisans ve Doktora eğitimini 1984 ve 1991 yıllarında tamamlamıştır. 1992 yılında Yrd. Doç., 1996 yılında Doçent ve 2002 yılında Profesör unvanını almıştır. 1982-1986 yılları arasında Yıldız Teknik Üniversitesi'nde Araş. Gör., 1986-2000 yılları arasında İstanbul Teknik Üniversitesi'nde Araş. Gör. ve Öğretim Üyesi olarak çalışmıştır. 2000-2008 yılları arasında Yıldız Teknik Üniversitesinde Öğretim Üyesi olarak akademik ve yönetici görevlerini yürütmüş ve Dekanlık görevini tamamlayarak emekli olmuştur. 2009 yılından günümüze İstanbul Okan Üniversitesi Bilgisayar Mühendisliği Bölümünde Öğretim Üyesi olarak çalışmaktadır. Ohio State Üniversitesinde kısa süreli konuk Öğretim Üyesi olarak araştırma etkinliklerinde bulunmuştur. Uzmanlık alanları arasında Gömülü Sistemler, Akıllı Sistemler, Bilgisayarda Grafik ve Bilgisayar Oyunları yer almaktadır.

Nilgün İncereis, Ondokuz Mayıs Üniversitesi Fizik Öğretmenliği, Bilgisayar ve Öğretim Teknolojileri Öğretmenliği lisans eğitimlerini ve Fizik Yüksek Lisans eğitimini 2009, 2012 ve 2012 yıllarında, İstanbul Okan Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans eğitimini 2019 yılında, Abant İzzet Baysal Bilgisayar Mühendisliği lisans eğitimini 2020 yılında tamamlamıştır. 2020 yılından beri İstanbul Okan Üniversitesi Bilgisayar Mühendisliği'nde doktora yapmaktadır. 2013-2020 yılları arasında İstanbul Okan Üniversitesi'nde, 2020-2021 yılları arasında Zonguldak Bülent Ecevit Üniversitesi'nde öğretim görevlisi olarak çalışmıştır. 2021 yılından günümüze Bartın Üniversitesi Uzaktan Eğitim ve Uygulama ve Araştırma Merkezinde öğretim görevlisi olarak çalışmaktadır. Çalışma alanları bulut bilişim, nesnelerin interneti, makine öğrenmesi ve derin öğrenme üzerinedir.

Makale Türü: Araştırma Makalesi
Makale Gönderim Tarihi: 07.03.2021
Makale Kabul Tarihi: 25.03.2022

Tersine Mühendislik Yöntemi ile Test Senaryo Üreten ve Yürüten Çerçeve: Finansal Bir Uygulamada Vaka Çalışması

Test Case Generation and Execution Framework with Reverse Engineering Method: Case Study for a Financial Application

Emine Dumlu Demircioğlu[1], *Oya Kalıpsız*[1]

[1]Bilgisayar Mühendisliği
Yıldız Teknik Üniversitesi
eminedumlu@gmail.com, oyakalipsiz@gmail.com

Özet

'u makalede, birbiriyle API mesajlarını kullanarak 'etişimde bulunan istemci-sunucu mimarisine dayalı istemler için, farklı veri formatlarını destekleyen bir mesaj üdümlü test senaryosu üreten ve yürüten çerçeve yaklaşımı nerilmektedir. Temel olarak, çerçeve ağ üzerinde istemci-unucu uygulamaları arasındaki iletişimden elde edilen ağ g dosyasından tersine mühendislik yöntemi ile API esajlarını elde ederek yürütülmesini sağlamaktadır.

'üyük ölçekli iş uygulamalarının fonksiyonel doğruluğunun krarlı bir şekilde manuel olarak test edilmesi zaman alan ve ataya açık bir süreçtir. Bu sürecin uçtan uca (test senaryosu retiminden yürütülmesine, doğrulanması ve hata aporlanması kadar geçen sürecin) otomatikleştirilmesi azılım testinde verimliliği artırmaktadır. Günümüzde API stinde kullanılan mevcut test araçları, tüm iş alanlarına ntegre edilememektedir. Dahası, var olan test otomasyon raçları daha çok HTTP protokolü üzerinden erişilebilen VEB-API'leri içindir. Bu çalışmanın motivasyonu da, 'etişim olarak API mesajlarını kullanan istemci-sunucu imarisine dayalı uygulamalara yönelik regresyon testi tomasyonu çerçevesi azlığından kaynaklanmaktadır.

'u makalede önerdiğimiz yaklaşımımızın etkinliğini eğerlendirmek için finansal bir sisteme uyguladık.

nahtar kelimeler: Uygulama Programı Arayüzü Testi, Otomasyon Testi, İstemci-Sunucu Uygulmaları, Vaka 'alışması

Abstract

n this paper, a framework that generates and executes a essage driven test scenario that supports different data rmats is proposed for the systems based on client-server rchitecture that communicate with each other by using ifferent data formats. Basically, the framework provides the eneration of API messages by using reverse engineering ethod from the network log file obtained from the ommunication between client-server applications on the etwork. Repetitive manual testing of the functional correctness of large-scale business applications is a time-consuming and error-prone process. Automating this process by ensuring end-to-end increases software testing efficiency. The studying is motivated due to the fact that there is a lack of regression test automation framework in a specific domain: client-server apps which uses API messages for the communication, such as financial applications. We have applied the proposed testing framework in this paper into a financial systems in order to evaluate the effectiveness of the framework.

Keywords: API Testing, Automation Testing,Client-Server applications, Case Study

1. Giriş

Yazılım testi, yazılım ürünündeki hataları tespit etmek, ürünün kalitesinden emin olmak amacıyla manuel veya otomatik olarak yürütülebilen yazılım geliştirme yaşam döngüsünde büyük öneme sahip olan bir süreçtir. Yazılım geliştirme süreç modeli olarak artırımlı süreç modelini kullanan şirketlerde artırımlı olarak geliştirilen yazılım, her artırımda gelen yeni gereksinimler veya yazılım değişikliği talepleri mevcut yazılımda değişiklik yapılmasını gerektirir. Yapılan değişiklikler ve geliştirilen yeni fonksiyonlar devreye alım öncesi doğrulanma ve etkili bir şekilde regresyon testlerinin gerçeklenmesi ihtiyacını ortaya çıkarmaktadır. Regresyon testi, yazılıma yeni eklenen özelliklerin veya yazılımdaki değişikliklerin mevcut sistemde herhangi bir yan etkiye sebep olup olmadığını doğrulamak amacıyla gerçekleştirilen testtir [1] [2]. Günden güne artırımlı süreç modeli kullanımındaki artış, etkili bir regresyon testi stratejisi ihtiyacını da ortaya çıkarmıştır. Bu testleri özellikle büyük ölçekli iş uygulamalarında manuel şekilde gerçekleştirmek epey zordur [3] [4] [5]. Binlerce farklı test senaryosunun manuel oluşturulması, yürütülmesi ve doğrulanması insan kaynağına büyük yatırım yapmayı gerektirmektedir [26]. Yeni bir yazılım versiyonunun devreye alım öncesi yapılan bu testler, elle yazılmış veya öndecen kaydedilmiş istekleri yeni sürüme göndermek ve belirli yanıtları almak şeklinde gerçekleştirilebilmektedir. Bu süreç,

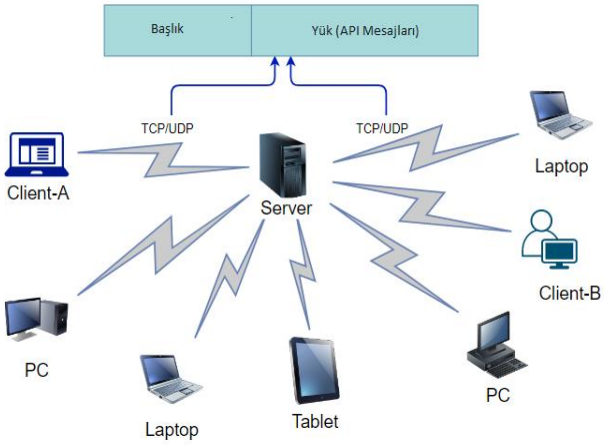
şitli test araçları ile desteklense bile zahmetli bir iştir. Bu
bir test tipinde, yeni sürümün kontrol edilmesi bazı iş
ınlarında tamamen otomatik olacak şekilde
rçekleşememektedir [21]. Bunun yanında, manuel testin
şarısının, manuel testi yürüten kişinin o günkü ruh haline
ğlı olduğu da bilinmektedir. Dikkat eksikliği, odak azalması
gecikme gibi faktörler yazılım testinin verimliliğini
cilemekte ve tüm projeye zarar verebilirmektedir [8][9].
st otomasyonu, test senaryolarının (test cases)
ışturulmasından yürütülmesine, doğrulanması ve hata
orlaması da dahil olmak üzere uçtan uca otomasyon olarak
şünülmelidir [9]. Otomatik olarak test senaryolarının nasıl
ışturulacağına ilişkin yapılan çalışmalar, otomasyon
stinde önemli bir yere sahiptir ve hala günümüzde
ıştırma konusudur [11].
ı çalışmanın amacı da, istemci-sunucu mimarisine dayalı
stemler için, sunucu üzerindeki API versiyonu
ncellemeleri sonrası bir regresyonun oluşup oluşmadığını
spit etmek için bir regresyon test otomasyonu uygulaması
klaşımı önermektedir. Motivasyonumuz, iletişim olarak
PI mesajlarını kullanan sistemlere yönelik literatürde
gresyon testi otomasyon çerçevesi azlığından
ynaklanmaktadır.
ılışmamız üç aşamadan oluşmaktadır:

1. Test senaryolarının üretilmesi,
2. Üretilen test senaryolarının otomatik olarak yürütülmesi,
3. Testlerin doğrulanması ve hata raporlama

ıklaşımımız test senaryolarının yeniden kullanımını
steklemekle birlikte API mesajları ile iletişimde bulunulan
TCP sunucusunun regresyon testlerinde %100
omasyonu sağlamaktadır.
ı makalede ise, test senaryolarının otomatik olarak
etilmesi, yürütülmesi ve doğrulanmasını sağlayan çerçeve
klaşamımız anlatılacaktır.
ı makalenin yazım organizasyonu şu şekildedir; Bölüm
de; çalışmaya ilişkin temel kavramlar, Bölüm 3'te API
stine yönelik literatür taraması yapılmıştır. Bölüm 4'te;
erdiğimiz çerçeveye ilişkin detaylar, bölüm 5'te çerçevenin
gulanması ve vaka analizine ilişkin sonuçlar
ylaşılacaktır. Bölüm 6'da ise sonuçlar ve gelecek
lışmalarımız hakkında bilgi verilecektir.

2. Temel Kavramlar

z konusu hedef sistem mimarisini Şekil 1'de görebilirsiniz.
emci-Sunucu arasındaki iletişim TCP veya UDP protokolü
erinden kurulmaktadır. İstemci uygulaması, sunucu IP'si ve
nucu üzerinde önceden tanımlanan port bilgisini kullanarak
ğlantı gerçekleştirmektedir. TCP (Transmission Control
otocol), internet üzerinde veri aktarımında kullanılan
venilir, bağlantı temelli bir protokoldür. UDP (User
atagram Protocol) ise, IP üzerinde çalışan bağlantısız
nelli bir protokoldür. İstemci ve sunucu arasındaki iletişim,
CP veya UDP veri paketleri kullanılarak sağlanmaktadır.
r veri paketi başlık ve veri alanı (payload) bölümünden
ışmaktadır. Başlık bölümü, kaynak/hedef port numaralarını
erir. TCP başlığı ayrıca ne kadar verinin gönderildiğini
cip etmek için sıra numaraları bilgisini de
lundurmaktadır. Veri bölümü ise, iletişimde kullanılan
verileri (API mesajlarını) taşır [30] .



Şekil 1 Hedef Sistem Mimarisi

Bu makalede önerilen yaklaşım, veriye paket seviyesinde ulaşmak ve test senaryolarını, tersine mühendislik yöntemi ile elde etmektir. Bu amaçla Wireshark aracı [12] kullanılarak istemci-sunucu arasındaki iletişimde kullanılan veri paketleri yakalanmıştır. Wireshark bir paket analizi aracıdır. Veri paketleri, bu araç kullanılarak bir PCAPs (Packet Captures) dosyasında saklanır. PCAPs dosyası, istemci-sunucu arasındaki iletişimi gösteren bir ağ log dosyasıdır. Örnek bir PCAPs dosyası içeriği Şekil 2'de gösterilmiştir. Dosyadaki her veri paketi, kullanılan taşıma katmanı protokolüne bağlı olarak bir TCP veya UDP veri paketidir ve bu paketler iletişimde kullanılan API mesajlarını içerir.

3. İlgili Çalışmalar

API testi, uygulama programı arayüzünün test edilmesini sağlayan ve iş mantığının işlevselliğini, güvenilirliğini, güvenliğini sağlamak için büyük önem taşıyan bir test türüdür. API'lerin, bulut teknolojilerinin ve birbirine bağlı uygulamaların sayısının artmasıyla birlikte API testi, yazılım testinin kritik bir parçası haline gelmiştir. Bu test, kullanıcı arayüzü olmadığı için mesaj seviyesinde gerçekleşir. İş mantığı katmanının içeriği, uygulamaların başarılı çalışması için çok önemlidir. Bir API mesajındaki hata büyük ölçekli yazılım hatalarına sebep olabilir. Bundan kaynaklı, API mesajlarının kalitesini sağlamak için API testi zorunlu hale gelmiştir. API testi ve bu amaç için var olan araç desteği konusunda araştırma topluluğunda daha fazla çalışmaya ihtiyaç bulunmakta ve endüstriyel uygulamada test otomasyonunun gerçek uygulamalarına ilişkin deneyim raporlarının nadir olduğuna inanılmaktadır [31].

Literatürde var olan API testi araçları çoğunlukla bir uygulama katmanı protokolü olan HTTP protokolü ile erişilebilien Web API'leri içindir; özellikle Rest-API en yaygın kullanılan API'dir [5][13][14]. REST tabanlı web uygulamaları temel olarak REST API yanıtları vermek için JSON veya XML formatlarını kullanır [13]. Bu tür test

raçları, kullanılabilirlik açısından tüm iş alanlarına entegre dilememektedir.

.iteratürde, SoapUI [25], Postman [24], jMeter ve diğerleri ibi regresyon testlerinde kullanılabilecek pek çok test aracı ulunmaktadır. Bu tür test araçları ile sabit bir test havuzu luşturulup, regresyon testleri yürütülebilir. Ancak test enaryolarının otomatik olarak üretilmesi hala araştırma onusudur. Bu tür test araçları daha çok HTTP protokolü ile rişilebilen Web tabanlı sunucu testlerinde kullanılmaktadır e daha uygun olduğu bilinmektedir [21]. TCP sunucu estlerine entegre edilmesinde bir takım kısıtlamalar ulunmaktadır. Ayrıca var olan otomasyon araçlarını ullanmak için test script kodunun yazılması ve bunun eğişen gereksinimlerde bakımının yapılması gerekmektedir.

ayrıca mevcut API testi araçlarının yüksek ek yük, uzun test erisi hazırlama ve yeniden kullanım zorluğu gibi bazı orlukları bulunmaktadır [15][19][20]. Günümüzde yaygın larak kullanılan Jmeter test otomasyon uygulamasını ele ldığımızda da bir takım zorluklar gözlemlemekteyiz. Bu ygulama TCP sunucu testlerini desteklese bile söz konusu edef sistemimize uyarlama aşamasında başarısız olmuştur. Manuel olarak oluşturulan test senayrolarının onaylanma şamasında bir takım kısıtlamalar ile karşılaşılmıştır.

ithub'da "goreplay" [27], "pollyjs" [28] ve "node-replay" 29] gibi ağ paketlerini yakalıp, test sistemine karşı oynatan est araçları da bulunmaktadır. Bu araçlar, daha çok yük estleri için kullanılmakta ve HTTP trafiğini tekrar ynatmaktadır. Uygulamanın fonksiyonel işlevselliği test dilmemektedir. Önerdiğimiz yaklaşım ile ağ paketleri test istemine karşı tekrar oynatılmakta ve test sisteminin onksiyonel işlevselliği test edilmektedir. Literatür aramasına göre, ağ paketlerinden testlerin oluşturulması, egresyon testinde uygulanması ve kullanılmasına yönelik ir çalışmaya rastlayamadık. Ağ log dosyası standart bir osya olup, çalışmamıza özel değildir. Ağ log dosyası API stek ve yanıt mesaj çiftlerini içermektedir.

Bu makalede, yukarıda tanımı yapılan sistemler için iteratürde mevcut API test aracı azlığından ötürü ağ üzerinde aket seviyesinde mesajları yakalayan ve regresyon testinde ullanılacak test senaryoları üreten bir tersine mühendislik aklaşımı önermektedir. Kullanılacak verinin paket eviyesinde yakalanması fikri, hem daha uygulama bağımsız lmayı hedeflememiz hem de farklı istemcilerden gelen istek nesajlarını en kolay şekilde elde etmemizi sağlamasından aynaklanmaktadır.

4. Test Senaryo Üreten Çerçeve

Bu bölümde, ağ log dosyasından API mesajların elde dilmesine ilişkin uyguladığımız yöntemin ana iş akışına yer ermekteyiz. Elde ettiğimiz API mesajları, test senaryolarına arşılık gelmektedir. Önerdiğimiz çerçeveyi Java'da eliştirdik. Ağ log dosyasını okuyabilmek için "pkts.io" açık aynak kodlu kütüphaneyi kullandık [16]. Temel olarak, nerdiğimiz çerçeve istemci-sunucu arasındaki iletişim aketlerinin yakalanıp saklandığı bir Pcaps dosyasını girdi osyası olarak alır. Bu dosya, çerçeve tarafından analiz edilir ve istemci-sunucu arasındaki haberleşme mesajlarını elde etmek için tersine mühendislik yaklaşımını uygulanır. Böylece test edilecek sisteme doğru bu dosyayı tekrar oynatmak için (replay) test senaryolarından oluşan test grubu (test suite) üretilmiş olmaktadır. Bir PCAPs dosyasında, farklı protokollere ait API mesajları olabilir. Bu durumda çerçeve her farklı protokol için farklı test grupları oluşturur. Test grubu içerisinde yer alan her test senaryosu istemciden sunucuya gönderilen istek mesajını ve sunucudan istemciye dönen yanıt mesajlarını içerir. Çerçeve bu eşleştirmeyi Kaynak IP:Port ve Hedef Ip:Port bilgisine bakarak yapmaktadır.

PCAPs dosyasında yer alan bir veri paketi okunduğunda, çerçeve bu veri paketini analiz etmeye başlar. İlk olarak, iletişimde hangi taşıma katmanı protokolünün (TCP/UDP) kullanıldığını belirler. Eğer TCP kullanılmış ise, çerçeve bu veri paketinin yeniden iletilen (retransmitted packet) bir paket olup olmadığına karar verir. Bir paketin yeniden iletilen paket olup olmadığı bilgisi, paketin üzerinden yer alan sıra numaralarından ve verinin boyutu bilgisinden hesaplanmaktadır. Eğer bir paketin yeniden iletilen bir paket olduğu tespit edilirse, ilgili paketin önceden test grubuna eklendiği bilindiği için o pakette yer alan mesajlar yeni bir test senaryosu olarak test grubuna eklenmez. Bu durum aynı test senaryosunun üretilmesinin önüne geçer. UDP bağlantısız temelli bir protokol olduğundan, yeniden iletim söz konusu değildir.

Bu aşamadan sonra, hedef sistem için iletişimde hangi uygulama katmanı protokollerinin kullanıldığı tespit edilmektedir. Bu aşamada sezgisel arama yöntemi (heuristic search methods) uygulanmıştır. Belirlenen mesaj desenleri, paketin veri bölümünde aranır. Örneğin, finansal verinin iletimi için kullanılan FIX protokolüne göre, her FIX protokolü mesajı "8 = FIXT" ile başlar ve "10 = XYZ" ile biter [17]. Bir veri paketinin payload bölümünde, FIX protokolüne ait bir API mesajının olup olmadığı aşağıdaki desen uygulanarak arama yapılmıştır:

```
Pattern.compile("8=FIXT.*?\\x0110=.{3}\\x01")
```

Sezgisel arama yöntemi, sırayla ağ log dosyasından elde edilen veriden ilgili mesaj desenini üretmeye çalışır. Farklı uygulama alanları için, farklı desenler belirlenerek çerçeveye entegre edilebilir.
Kullanılan uygulama katmanı protokolü tespit edildikten sonra, paket içerisinde yer alan veri çıkarılır ve söz konusu sistem için belirlenen mesaj formatına göre mesajlar yeniden oluşturulur. Bir mesaj oluşturulurken, paketin içerisindeki veri bölümü parçalanmış olabilir. Çünkü paket üzerindeki veri bölümünün taşıyabileceği maksimum veri boyutu bulunmaktadır. Eğer parçalanma tespit edilirse, veri bölümü hafızada saklanır ve anlamlı mesaj oluşabilmesi için bir sonraki paketin veri bölümü ile birleştirilir.
Son olarak, paketin varış zamanı, Kaynak IP:Port, Hedef IP:Port bilgileri de paket üzerinden elde edilir. Bu bilgiler doğrultusunda mesajların istemci tarafından üretilen bir istek mesajı veya sunucudan alınan bir yanıt mesajı olup olmadığına karar verilir. Yukarıda verdiğimiz bilgiler

doğrultusunda çerçeveyi geliştirmek için kullandığımız algoritma aşağıda verilmiştir.

orithm 1: createTestSuite
ıt: pcapFile
eat
1. packet ← getNextPacket();
2. extractPacketDetail (packet);
3. messages ← constructMessagesInPacket(packet);
4. addToTestSuite(messages);
l end of pcapFile

Algoritma 1, girdi olarak PCAPs doyasını alan ve dosyada yer alan her paketi okuyarak dosya sonuna kadar döngü kuran createTestSuite rutinini gösterir. Her iterasyonda bir sonraki paketi almak için getNextPacket() fonksiyonu çağrılır. İkinci satırda paketin varış zamanı, Kaynak/Hedef IP/Port ve kullanılan taşıma katmanı protokolü (TCP/UDP) bilgilerini elde etmek için extractPacketDetail(packet) fonksiyonu çağrılmaktadır. 3. satırda ise paketin veri bölümünde yer alan API mesajları, hedef sistem için belirlenen mesaj desenleri kullanılarak sezgisel arama yöntemi ile uygun mesaj formatlarına dönüştürelerek test senaryoları oluşturur. 3. adımda elde edilen API mesajları test grubuna eklenir.

Bu yaklaşım farklı uygulama alanlarına kolay bir şekilde adapte edilebilir. Bunun için constructMessagesInThePacket metodu söz konusu hedef sistemin mesaj yapısına uygun olarak uygulanması gerekmektedir. Bu yaklaşımı kullanarak test senaryoları paket düzeyinde otomatik olarak oluşturulabilir. Her test senaryosu, istemciden sunucuya gönderilen istek mesajları ve sunucudan gelen beklenen yanıt mesajlarını içerir.
Bazı uygulama katmanı protokolleri, ortam ile ilgili çeşitli veriler içerir ve bu nedenle bu veriler tekrar kullanılamaz. Örneğin, HTTP yanıtı, hedef sayfanın son değiştirilme zamanını, yanıt tarihini veya oturum bilgilerini içerir. Tüm paketleri herhangi bir analiz yapmadan kullanmak, bu değişken değerler nedeniyle onaylanma hatasına (doğrulama hatasına) neden olabilir. Bu yaklaşım kullanılarak geliştirilecek test otomasyonunda, bu tür çevre ile ilgili değerlerin, değişken verilerin çıkarılması gerekir. Bu nedenden ötürü, tüm TCP/UDP paketlerini kaydetmek ve bunları test ortamına karşı yürütmek yerine, yukarıda bahsedilen uygulama katmanı mesajlarının çıkarılması adımları gerçekleştirilir.

5. Vaka Çalışması

Önerilen yaklaşımın etkinliğini ölçmek amacıyla, borsalarda iletişim protokolleri olarak kullanılan FIX ve OUCH API'yi geliştirdiğimiz çerçeveye uyarladık. İletişim protokolleri olarak kullanılan FIX-API ve OUCH-API protokolleri tamamen birbirinden farklı yapıda mesaj formatlarına sahiptir. FIX (Financial Information Exchange), finansal verinin değişiminde kullanılan bir iletişim protokolüdür. FIX protokolü mesaj yapısına ait tüm bilgiler FIX spesikasyonu dökümanında yer almaktadır [17]. OUCH protokolü de iletişimde kullanılan bir emir iletimi protokolü olup, istemcinin emir girmesine, mevcut emirlerini iptal etmesine, güncellemesine ve bunlara ilişkin işlem bilgilerini almasına izin veren bir protokoldür. OUCH protokolü, istemci uygulaması ile OUCH sunucusu arasında geçen mantıksal mesajlardan oluşmaktadır [18]. FIX protokolü metin (text) tabanlı, OUCH protokolü ise binary tabanlı bir protokoldür.

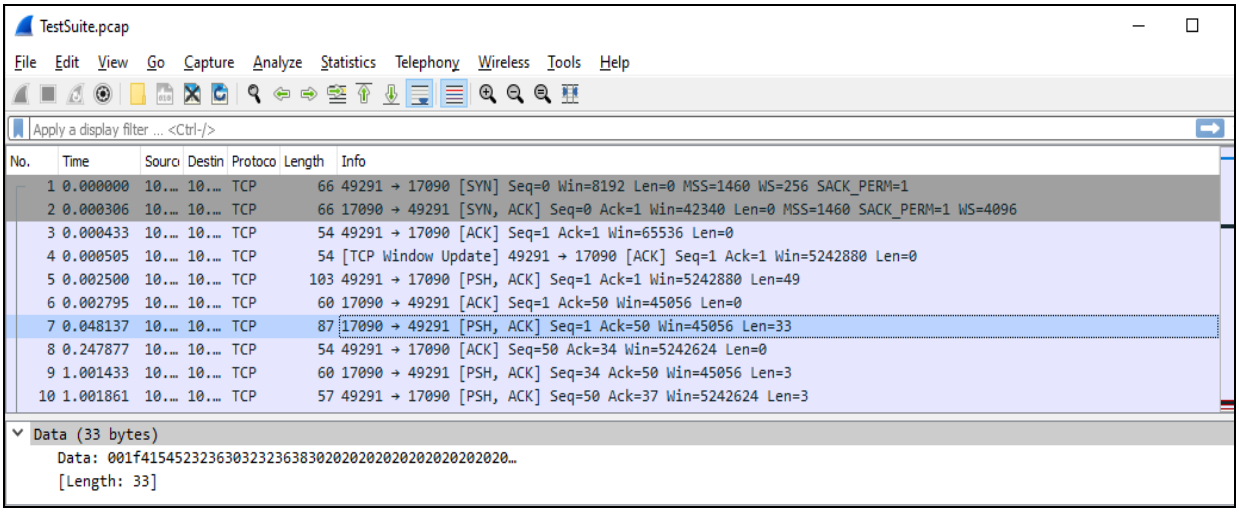
5.1. Verinin Hazırlanması

Çerçevede kullanılacak PCAPs dosyasının elde etmek için, keşif testi uygulanarak test senaryolarının bir listesi hazırlanmıştır. İnsan bilgisine dayalı olarak manuel olarak oluşturulan test senaryosu tasarımı, keşif testi olarak bilinir [9]. Testçiler hazırlanan test senaryolarını test edilen sisteme manuel olarak yürütürken, ağ paketleri Wireshark aracıyla yakalanır. Böylelikle PCAPs dosyası elde edilerek, geliştirilen çerçeveye girdi olarak verilir.

PCAPs dosyası oluşturmanın bir başka yolu ise, üretim/canlı ortamdaki istemci-sunucu arasındaki trafik dinlenerek gerçekleştirilebilir. Böylece gerçek veriler elde edilerek, gerçek verilerden test senaryoları oluşturulur. Bu yöntem ile bir istemcinin üretim ortamındaki sunucuya karşı olan davranışı da yakalanmış olur. Böylece gerçek ortamda bir istemcinin davranışı kaynaklı oluşabilecek bir hatanın hızlı bir şekilde tespiti için, ilgili istemcinin davranışını test ortamına karşı simüle edilmesi sağlanabilir.

5.2. Çerçevenin Uygulanması

Bu çalışmada, hem FIX protokolü hem de OUCH protokolüne ait API mesajlarını içeren TestSuite.pcap dosyası kullanılmıştır. Şekil 2, TestSuite.pcap dosyasındaki ilk 10 TCP paketini göstermektedir. Çerçeve bu dosyayı okuyarak 2 farklı test grubu oluşturur. Tablo 1'de OUCH protokülüne ait TestSuite.pcap dosyasından üretilen seçilmiş 5 adet test senaryosu gösterilmektedir. Tablodaki her satır, istemciden sunuya gönderilen istek mesajını ve sunucudan alınan yanıt mesajını gösterir. Örneğin 1. test senaryosuna göre, tablodaki verileri kullanarak "LoginRequestPacket" mesajı test sunucusuna tekrar gönderilirse, sunucunun da yanıt olarak "LoginAcceptedPacket" mesajı dönmesi beklenecektir. 2. test senaryosuna göre, "UnsequencedData-NewMsg" paketi tablodaki veriler ile sunucuya gönderilirse, sunucunun da "SequenceData-AcceptedMsg" paketi dönmesi beklenmektedir.
Tablo 2'de ise FIX protokolü için seçilmiş 5 adet test senaryosunu görebilirsiniz. Örneğin 1. test senaryosundaki veriler, yeni sürüm yüklenen test sistemine karşı yürütüldüğünde, sunucudan alınan mesajın, tabloda yer alan mesaj ile uyumlu olması beklenir. Beklenmedik bir mesajın gelmesi durumunda regresyon olduğu düşünülmektedir.
Tablodaki test senaryoları, test sistemine karşı otomatik yürütülmeden önce, bazı değişken verilerin (örneğin mesajın oluşturulma zamanı, checksum bilgisi gibi) olabileceği bilinmektedir. Bu değerler, API spesifikasyon dökümanından belirlenmiş ve doğrulama hatalarından kaçınmak için validasyon aşamasında çıkarılmıştır.



Şekil 2 TestSuite.pcap dosyası – Wireshark Ekran Görüntüsü

↓↓

Tablo 1. TestSuite.pcap dosyasından üretilen seçilmiş OUCH-API Test Senaryoları

Hedef IP:Port	İstek (Request) API Mesajı	Yanıt (Response) API Mesajı / Beklenen Mes
10.A.B.C: 1709	**PacketType: LoginRequestPacket** [USER1,ABC, TR18,1]	**PacketType: LoginAcceptedPacket** [TR18, 1]
10.A.B.C: 1709	**PacketType: Unsequenced Data Packet-NewMsg** [**O**, 21, 7798, Buy, 22, 600,,NoChange, XY, Pass, 6789, 0, Client, Normal_hours]	**PacketType: Sequence Data Packet-AcceptedMsg** [**A**, 21, 7098, Buy, 715305981147226, NoChange, XY OnBook, Pass, 6789, 22, 0, Client, Normal_hours]
10.A.B.C: 1709	**PacketType: Unsequenced Data Packet-UpdateMsg** [**U**,21,1b,99,0,NoChange,0,Empty]	**PacketType: Sequence Data Packet-ReplacedMsg** [**U**, 1b, 21, 7798, Buy, 715305981145226, 99, 600, NoChange, XY, OnBook, Pass, 6789, Client]
10.A.B.C: 1709	**PacketType: Unsequenced Data Packet-CancelMsg** [**X**, 21]	**PacketType: Sequence Data Packet-CanceledMsg** [**C**,21,7798,Buy,71530598114735226,Canceled]
10.A.B.C: 1709	**PacketType: LogoutRequestPacket**	**PacketType: LogoutResponsePacket**

Tablo 2 TestSuite.pcap dosyasından üretilen seçilmiş FIX-API Test Senaryoları

Hedef IP:Port	İstek (Request) API Mesajı	Yanıt (Response) API Mesajı / Beklenen Me
10.A.B.C: 1200	**NewMsg:** ***8=FIXT.1.1*** **35=D** 34=1 49=X 50=F6 52=20211019-10:32.06 56=TEST 1=XY **11=10** 38=22 40=2 44=5.000 54=1 55=A.E 59=0 60=20211019-10:32:06.843 70=ASD ***10=044***	**ExecutionReport: AcceptedMsg:** ***8=FIXT.1.1*** 35=8 34=1 49=TEST 56=X 57=F6 1= 6=0 **11=10** 14=0 17=152 22=M 37=**6363BF122222** 38=22.0000000 39=0 40=2 44=5.0000000 48=616 55=A.E 151=20.0000000 60=20211019-10:32:06.9 ***10=032***

10.A.B.C: 1200	**CancelMsg:** **8=FIXT.1.1 35=F** 34=2 49=X 50=F6 52=20211019-10:32:15 56=TEST **11=23** 37=**6363BF1222228** 38=22 54=1 55=A.E 60=20211019-10:32:15.380 **10=184**	**ExecutionReport: CanceledMsg** **8=FIXT.1.1** 35=8 34=2 49=TEST 56=X 57=F6 1=X 6=0 **11=23** 14=0 17=158 22=M 37=**6363BF122222** 38=22.0000000 39=4 41=10 48=616 54=1 55=A.E 70=ASD 150=4 151=0 60=20211019-10:32:15.450 **10=252**
10.A.B.C: 1200	**NewMsg:** **8=FIXT.1.1 35=D** 34=3 49=X 50=F6 52=20211019-10:32:09 56=TEST 1=XY **11=11** 38=77 40=2 44=11.000 54=1 55=A.E 59=0 60=20211019-10:32:16.200 70=ASD **10=022**	**ExecutionReport: AcceptedMsg** **8=FIXT.1.1** 35=8 34=3 49=TEST 56=X 57=F6 1=X 6=0 **11=11** 14=0 17=157 22=M 37=**6363AC111111** 38=77.0000000 39=0 40=2 44=11.0000000 48=616 55=A.E 59=0 70=ASD 119=220.0000000 150=0 151=20.0000000 60=20211019-10:32:16.500 **10=0**
10.A.B.C: 1200	**UpdateMsg:** **8=FIXT.1.1 35=G** 34=4 49=X 56=TEST 50=F6 52=20211019-10:32:16.43=N 37=**6363AC11111181 11=12** 55=A.E 60=20211019-10:32:16.800 38=27 59=0 54=1 40=2 44=11.000 **10=035**	**ExecutionReport: UpdatedMsg** **8=FIXT.1.1** 35=8 49=TEST 56=X 34=4 57=F6 37=**6363AC11111181 11=12** 17=89485 150=5 39=(1=XY 55=A.E 48=616 22=M 54=1 38=27.0000000 44=11.000 59=0 151=27.0000000 14=0 6=0 60=20211019-10:32:16.900 **10=166**
10.A.B.C: 1200	**CancelMsg:** **8=FIXT.1.1 35=F** 34=5 49=X 50=F6 52=20211019-10:32:15.492 56=TEST **11=13** 37=**6363AC11111181** 38=20 54=1 55=A.E 60=20211019-10:32:17.200 **10=184**	**ExecutionReport: CanceledMsg** **8=FIXT.1.1** 35=8 34=5 49=TEST 56=X 57=F6 1=X 6=0 **11=13** 14=0 17=158 22=M 37=**6363AC111111** 38=20.0000000 39=4 41=40360 48=616 54=1 55=A 70=ASD 150=4 151=0 60=20211019-10:32:17.250 **10=252**

Tüm API mesajlarını oluşturup eşleştirdikten sonra, çerçeve bu test senaryolarını test ortamında yeniden oynatmaya başlar. Her test senaryosunun yukarıdaki tabloda da gözükeceği gibi hedef IP adresi ve Port bilgisi vardır. Bu bilgileri kullanarak çerçeve hedef IP:Port'a bağlanır ve test sunucusuna, test senaryolarında yer alan istek API mesajlarını göndermeye başlar. PCAPs dosyasından çıkarılan her test senaryosu Tablo 1 ve Tablo 2'de de görüleceği gibi beklenen/cevap API mesajına sahiptir. Çerçeve test ortamından dönen gerçek yanıt mesajlarını aldıktan sonra, gerçek ve beklenen mesajları karşılaştırarak her test senaryosunu doğrular.

Hangi mesaj alanlarının doğrulanma sürecine dahil edileceğini, spesifikasyona dayalı doğrulama yaklaşımı kullanarak belirledik. Bu aşamada her API protkolü için XML dosyaları oluşturduk. OUCH API için hazırladığımız örnek bir XML dosyasının içeriğini Tablo 3'de görebilirsiniz. Oluşturulan OUCH_API.xsd ve FIX_API xsd dosyaları çerçeveye input olarak verilmiştir ve çerçevenin derlenme aşamasında XML dosyasından ilgili mesaj sınıfları otomatik olarak üretilmiştir. Bununla birlikte her protokolde yer alan mesajların hangi alanlarının doğrulama sürecine dahil edileceği bilgisi de use="required" özelliği ile tanımlanmıştır.

Tablo 3. OUCH protokolü için doğrulama aşamasında kullanılan XML dosyası

OUCH_API.xsd
:element name="CanceledMsg"> <xs:complexType> <xs:attribute name="msgType" fixed="C"/> <xs:attribute name="timeStamp" type="xs:long"/> <xs:attribute name="orderToken" type="xs:string" /> <xs:attribute name="orderBookId" type="xs:int" **"required"** /> <xs:attribute name="side" type="xs:string" **"required"** /> <xs:attribute name="orderId" type="xs:long" /> <xs:attribute name="canceledReason" type="xs:int" **"required"** /> </xs:complexType> **'xs:element>**
element name="AcceptedMsg"/>
element name="ReplacedMsg"/>
element name="RejectedMsg"/>

Doğrulanma süreci tamamlandıktan sonra, çerçeve test sonuçlarının özetini bir rapor olarak üretmekte ve ilgili kişilere bildirim göndermektedir.

6. Sonuç ve Gelecek Çalışmalar

Tekrarlı bir şekilde yapılan manuel testler özellikle büyük ölçekli iş uygulamalarında oldukça zordur. Bu tür uygulamalarda istemci davranışını simüle etmek için binlerce farklı test senaryosu olabilir ve bu senaryoların manuel olarak gerçekleştirilmesi zaman alıcı bir süreçtir. Bu tür uygulamaların test edilmesinde test otomayonu kritik bir öneme sahiptir. Literatüre göre HTTP protokolü dışında kendi uygulama katmanı protokollerine sahip TCP sunucular için var olan test otomasyon uygulamalarının tüm iş alanlarına entegre edilemediğini görmekteyiz ve entegrasyonunda bir takım kısıtlamalar ile karşılaştık. Bu nedenle, bu tür sistemlerin regresyon testleri için istemci ve sunucu arasındaki ağ paketlerini kullandığımız yeni bir yaklaşım önerdik.

Bu makalede istemci-sunucu mimarisine dayalı sistemler için test senaryosu üretimi yapan ve test sistemine karşı tekrar yürüten bir yaklaşım önerdik. Bilgilerin paket düzeyinde kullanılması, uygulama bağımsızlığını sağlamıştır. Bu çerçeveyle, test senaryolarını otomatik olarak paket seviyesinde oluşturarak, API testindeki otomasyonu artırıyoruz. Ayrıca yaklaşımımız, test senaryolarının yeniden kullanımını da sağlamaktadır.

Gelecek çalışmalar olarak iki çalışma yapmayı planlamaktayız. İlk olarak, farklı iş alanlarına yönelik yeni bir API protokolünü çerçeveye tanıtmak ve uyarlamaktır. İkinci olarak ise, üretim ortamından alınan gerçek veri paketlerini test sürecinde kullanmaktır. Bu sayede gerçek verilerden test senaryoları oluşturulabilecek ve böylece bir istemcinin gerçek ortamdaki davranışı, test ortamında otomatik olarak simüle edilebilecektir.

7. Kaynaklar

[1] A. K. Sultanía : Developing software product and test automation software using Agile methodology, Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT), Hooghly, pp. 1-4. (2015)

[2] S.Dalal, K.Solanki, : Challenges of Regression Testing: A Pragmatic Perspective in International Journal of Advanced Research in Computer Science, vol.9, no.1, February (2018)

[3] Z. Liu, Q. Chen and X. Jiang : A Maintainability Spreadsheet-Driven Regression Test Automation Framework, IEEE 16th International Conference on Computational Science and Engineering, Sydney, NSW, pp. 1181-1184. DOI= 10.1109/CSE.2013.175 (2013)

[4] Bangare, Sunil & Borse, Seema & Bangare, Pallavi & Nandedkar, Shital. (2012). AUTOMATED API TESTING APPROACH. International Journal of Engineering Science and Technology. 4.

[5] Isha, A. Sharma and M. Revathi, "Automated API Testing," 2018 3rd International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2018, pp. 788-791.

[6] R. M. Sharma, : Quantitative Analysis of Automation and Manual Testing, International Journal of Engineering and Innovative Technology (IJEIT) Volume 4, Issue 1, (2014)

[7] X. Han, N. Zhang, W. He, K. Zhang and L. Tang, "Automated Warship Software Testing System Based on LoadRunner Automation API," 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, 2018, pp. 51-55.

[8] Gonçalves, Wellington & Barreto de Almeida, Carlos & Araújo, Ladyanny & Ferraz, Mateus & Xandú, Rogerio & Junior, Ivaldir. (2017). The Impact of Human Factors on the Software Testing Process: The Importance of These Factors in a Software Testing Environment. Journal of Information Systems Engineering & Management. 2. 10.20897/jisem.201724.

[9] V. Garousi and F. Elberzhager, : Test Automation: Not Just for Test Execution, in IEEE Software, vol. 34, no. 2, pp. 90-96, Mar.-Apr. DOI= 10.1109/MS.2017.34. (2017)

[10] J. Itkonen and M.V. Mantyla,: Are test cases needed? Replicated comparison between exploratory and test-case based software testing, Emprical Software Engineering (2014)

[11] Fernandez-Sanz, L., & Misra, S. (2012). Practical application of UML activity diagrams for the generation of test cases. Proceedings of the Romanian academy, Series A, 13(3), 251-260.

[12] Wireshark Aracı: https://www.wireshark.org/ (son erişim 09 Şubat 2022)

[13] K. Sneha and G. M. Malle, : Research on software testing techniques and software automation testing tools, International Conference on Energy, Communication, Data Analytics and Soft Computing, Chennai, pp. 77-81. DOI= 10.1109/ICECDS.2017.8389562 (2017)

14] N. Bhateja, : A Study on Various Software Automation Testing Tools 2015 International Journal of Advanced Research in Computer Science and Software Engineering Volume 5, Issue 6, June (2015)
15] D. Xu, W. Xu, M. Kent, L. Thomas and L. Wang, "An Automated Test Generation Technique for Software Quality Assurance," in IEEE Transactions on Reliability, vol. 64, no. 1, pp. 247-268, March 2015, doi: 10.1109/TR.2014.2354172.
16] Java Library for reading and writing PCAPs., https://github.com/aboutsip/pkts (son erişim 08 Şubat 2022)
17] FIXAPI-Protocol: https://www.borsaistanbul.com/files/genium-inet-fix-protocol-specification.pdf
18] OUCHAPI-Protocol: https://www.borsaistanbul.com/files/OUCH_ProtSpec_BIST_va2414.pdf
19] Y. Chen, Y. Gao, Y. Zhou, M. Chen and X. Ma, "Design of an Automated Test Tool Based on Interface Protocol," 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Sofia, Bulgaria, 2019, pp. 57-61, doi: 10.1109/QRS-C.2019.00024
20] K. V. Aiya and H. Verma, "Keyword driven automated testing framework for web application," 2014 9th International Conference on Industrial and Information Systems (ICIIS), Gwalior, 2014, pp. 1-6, doi: 10.1109/ICIINFS.2014.7036478.
21] Patrice Godefroid, Daniel Lehmann, and Marina Polishchuk. 2020. Differential regression testing for REST APIs. In Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2020). Association for Computing Machinery, New York, NY, USA, 312–323. DOI:https://doi.org/10.1145/3395363.3397374
22] E. Viglianisi, M. Dallago and M. Ceccato, "RESTTESTGEN: Automated Black-Box Testing of RESTful APIs," in 2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST), Porto, Portugal, 2020 pp. 142-152.
23] H. Ed-Douibi, J. L. C. Izquierdo, and J. Cabot. Automatic generation of test cases for REST APIs: A specificationbased approach. In 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC), pages 181–190. IEEE, 2018.
24] Postman-API Development Environment. https://www.getpostman.com/
25] SOAP: https://www.soapui.org/
26] E. Çelik, S. Eren, E. Çini and Ö. Keleş, "Software test automation and a sample practice for an enterprise business software," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 141-144, doi: 10.1109/UBMK.2017.8093583.
27] GoReplay: https://github.com/buger/goreplay (son erişim 03 Ocak 2022)
28] Pollyjs: https://github.com/Netflix/pollyjs (son erişim 10 Ocak 2022)
29] NodeReplay: https://www.npmjs.com/package/replay (son erişim 7 Şubat 2022)
[30] TCP/UDP: https://www.cs.nmt.edu/~risk/TCP-UDP%20Pocket%20Guide.pdf (son erişim 7 Şubat 2022)
[31] V. Garousi et al., "Automated Testing of Simulation Software in the Aviation Industry: An Experience Report," in IEEE Software, vol. 36, no. 4, pp. 63-75, July-Aug. 2019, doi: 10.1109/MS.2018.227110307.

Özgeçmişler

Emine Dumlu Demircioğlu, lisans eğitimini Ege Üniversitesi Bilgisayar Mühendisliği, yüksek lisans eğitimini ise Sabancı Üniversitesi Bilgisayar Bilimleri ve Mühendisliği'nde tamamlamıştır. Şuanda Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği'nde doktora öğrencisidir ve Borsa İstanbul'da yazılım geliştirme uzmanı olarak çalışmaktadır.

Oya Kalıpsız, yüksek lisans eğitimini Istanbul Teknik Üniversitesi'nde tamamlamıştır. Doktora derecesini ise İstanbul Üniversitesi Sayısal Yöntemler alanında almıştır. Şuanda Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği bölümünde Profesör olarak çalışmalarına devam etmektedir. İlgi alanları: Yazılım Mühendisliği, Veritabanı Sistemleri, Veri Madenciliği, Sistem Analizi ve Yönetim Bilişim Sistemleri'dir.

Bazı Alt Uzaylarda Kriptografik Açıdan Eniyilenmiş Büyük S-kutuları
Cryptographically Optimized Large S-boxes in Some Subspaces

Selçuk Kavut
Bilgisayar Mühendisliği Bölümü, Balıkesir Üniversitesi, Balıkesir, Türkiye
skavut@balikesir.edu.tr

Öz

Arama uzayının büyüklüğünden dolayı sezgisel arama lgoritmaları, güçlü kriptografik özelliklere sahip S-kutularını lde etmek için literatürde genellikle sekiz ve daha küçük oyutlardaki uzaylarda uygulanmıştır. Bununla birlikte, oyutun artmasıyla doğrusal olmama, farksal birbiçimlilik ve ebirsel bağışıklık özelliklerinin iyileşebileceği bilinmektedir. 'alışmamızda bu durum ele alınarak, bildiğimiz kadarıyla ilk 'efa on boyutlu uzay için arama gerçekleştirilmiştir. Özel larak, kriptografik açıdan zengin olan bazı alt uzaylarda astgele ve sezgisel aramalar yürütülerek, her iki alt uzay için lde edilen en iyi sonuçlar AES S-kutusunun kriptografik zellikleri ile karşılaştırılmıştır. Bunun sonucunda, cebirsel nşa yöntemlerinin yanı sıra, rastgele veya sezgisel arama lgoritmaları ile on boyut için bahsedilen alt uzaylarda ulunan S-kutularının doğrusal, farksal ve cebirsel riptanalize karşı AES S-kutusundan daha dayanıklı labileceği deneysel olarak gösterilmiştir. Ayrıca, sezgisel rama algoritmasının ters fonksiyondan başlayarak arama aptığında, ters fonksiyon ile aynı veya çok yakın kriptografik zelliklere sahip S-kutularını üretebildiği gözlenmiştir.
nahtar Kelimeler: S-kutusu, sezgisel arama, doğrusal lmama, farksal birbiçimlilik, cebirsel bağışıklık

Abstract

Due to the size of the search space, heuristic search lgorithms are applied for the spaces in dimensions less than ine to obtain cryptographically strong S-boxes in literature. lowever, it is known that increasing dimension can improve onlinearity, differential uniformity and algebraic immunity roperties. We here perform a search in dimension ten for the 'rst time to our knowledge. Specifically, implementing andom and heuristic searches within some cryptographically ich subspaces, the best obtained results are compared with ryptographic properties of AES S-box. Consequently, beside lgebraic constructions, we show that the S-boxes found by andom or heuristic searches in the mentioned subspaces for imension ten can be more resistant than AES S-box against inear, differential and algebraic cryptanalyses. Further, we bserve that when heuristic search is started by the inverse unction, S-boxes having the same or almost the same roperties as those of the inverse function can be generated.
ey Words: S-box, heuristic search, nonlinearity, differential niformity, algebraic immunity

1. Giriş

S-kutuları (yer değiştirme kutuları) simetrik kriptosistemlerde genellikle doğrusal olmama uygulayan tek bileşenlerdir ve farksal [1], doğrusal [2], cebirsel [3, 4] ve yüksek mertebeden farksal kriptanaliz [5] gibi saldırı yöntemlerinin başarısı S-kutularının kriptografik dayanıklılığına bağlıdır. Bu nedenle kullanılan S-kutusunun kriptografik özellikleri, tüm kriptosistemin güvenliği açısından kritik bir öneme sahiptir. S-kutusu tasarımı simetrik kriptografide karşılaşılan en zor problemlerden birisidir ve literatürde bulunan güçlü kriptografik özelliklere sahip S-kutusu inşaları [6] azdır. $n\times m$ büyüklüğünde bir S-kutusu, n biti m bite gönderen bir fonksiyon olarak tanımlanır. Çalışmamızda, $n\times n$ büyüklüğünde bijektif S-kutuları, diğer bir ifadeyle n boyutlu $GF(2)^n$ vektör uzayındaki permütasyonlar ele alınmıştır.

Kriptosistemlerde kullanılan S-kutularının sağlaması gereken ve (bir özelliği iyileştirirken başka bir özelliğin kötüleşmesi anlamında) birbiriyle çelişen birçok kriptografik özellik bulunduğundan, bütün kriptografik özellikler bakımından en iyi S-kutusu tasarlamak mümkün değildir [7]. Bu nedenle, kriptografik özellikler arasında dengeleme yapılması kaçınılmazdır. S-kutuları rastgele üretme, cebirsel inşa ve sezgisel/evrimsel arama yöntemleri ile elde edilebilmektedir. Bunlardan rastgele üretme kolay ve hızlı bir yöntem olmasına rağmen, bulunan S-kutularının kriptografik özellikleri çoğunlukla zayıftır. Günümüzde en popüler S-kutusu büyüklüğü olan 8×8 durumu için, rastgele arama yöntemi doğrusal olmama değeri 98'e kadar [8] (bilinen en iyi değer 112) ve farksal birbiçimlilik değeri 10 ile 18 arasında [9] (bilinen en iyi değer 4) S-kutuları üretebilmektedir. Cebirsel inşalar [6] ise genellikle güçlü temel kriptografik özelliklere sahiptir. Örneğin, AES S-kutusunun [10] kullandığı $GF(2)^8$ uzayında tanımlı ters fonksiyon [11], doğrusal olmama ve farksal birbiçimlilik gibi kriptografik özellikler bakımından literatürde bilinen en iyi değerlere sahiptir. Bununla birlikte, bu tür inşa yöntemleri literatürde fazla bulunmamaktadır ve etkisi henüz günümüzde sınırlı kalmakla beraber, kullanılan cebirsel yapıların cebirsel saldırılar [3, 4] açısından zayıflığa yol açabileceği bilinmektedir. S-kutusu tasarımında yaygın olarak kullanılan (bahsedildiği gibi, AES S-kutusunun da kullandığı) ters fonksiyon, cebirsel bağışıklık açısından en iyi dayanıklılığı sağlamamaktadır. Ayrıca, ters fonksiyonun yan kanal analizi karşısında dayanıklı olmadığı bilinmektedir [7, 12]. Diğer bir yaklaşım olan sezgisel arama yöntemleri, cebirsel yapısı daha karmaşık S-kutuları üretebilmekte, fakat arama uzayının çok büyük olmasından dolayı (bkz. Tablo 1), genellikle $n \leq 8$ için uygulanmaktadır. Doğrusal olmama ve farksal birbiçimlilik

emel kriptografik özellikler açısından ele alırsak, sezgisel yöntemleri ile elde edilen sonuçlar, rastgele arama mlerinin ürettiği sonuçları iyileştirmekte; bununla birlikte, un artmasıyla cebirsel inşa yöntemleri ile ulaşılan en iyi lara ulaşamamaktadır. Özel olarak, 8 boyutlu durum için sal olmama değeri 104'e kadar [9] ve farksal imliliği 6'ya kadar [13] olan S-kutuları elde ilmektedir. Cebirsel inşa yöntemlerinin bahsedilen grafik özellikleri daha iyi olabilmektedir; bununla e, sezgisel arama yöntemleri ile yeni S-kutularının nını gerektiren senaryolar mevcuttur. Bunların başında, ğin, daha önce belirtilen yan kanal analizine karşı ıklılık veya cebirsel bağışıklık gibi) cebirsel inşa mlerinin sağlayamadığı kriptografik özellikleri, ters düşen temel kriptografik özelliklerle birlikte dengelemek veya emek gerekliliği düşünülebilir. Temel kriptografik klerin en iyi olmasından çok, tüketilen güç, donanım alanı cikme gibi gerçeklenme açısından en iyi performansı S-kutusunun tasarımı bir başka senaryo olabilir. Ayrıca, grafik özellikleri bakımından cebirsel inşa yöntemlerine özelliklere sahip (örneğin, belirli bir şirket veya kuruma an) S-kutularının tasarlanması amacıyla sezgisel arama mleri kullanılabilir.

-kutusu tasarımı sezgisel arama algoritmaları için kle boyut arttıkça zorlaşan bir problemdir. Yakın ıda, Jacobovic vd. [14, 15] tarafından Boole fonksiyonları kutuları tasarımının neden zor bir problem olduğunu ak için maliyet ortamı analizi yapılmıştır. Özel olarak, S- ı için gerçekleştirilen çalışmada [14] seçilen maliyet iyonları ve komşuluk türleri için, neredeyse her başlangıç sının farklı bir yerel en iyiye gittiği gözlenmiş ve maliyet ında verimli bir şekilde gezinmenin zorluğu deneysel gösterilmiştir. S-kutusu tasarımında sezgisel algoritma an ilk çalışma, Milan [8] tarafından yapılan tepe ma yöntemi ile rastgele üretilen 8×8 büyüklüğündeki S- ının doğrusal olmama değerlerinin iyileştirilmesidir. sında Milan vd. [16] genetik algoritma ile birlikte tepe ma yöntemi kullanarak doğrusal olmama ve mutlak ge değerlerini iyileştirmişlerdir. Bu iki çalışmada, sal olmama ve mutlak gösterge değerleri doğrudan et fonksiyonu olarak kullanılmıştır. Clark vd. [17], ı-Hadamard spektrumundaki tüm değerleri eniyileyen bir et fonksiyonu kullanarak, tepe tırmanmayla takip edilen na benzetimi algoritması ile [16] çalışmasındaki sonuçları irmişlerdir. Tesař [18], tüm ağaç arama tekniği ile irdiği genetik algoritmayı n = 6, 7, 8 için uygulamış ve i sonuçlardan daha iyi doğrusal olmama değerleri elde ir. Kazymyrov vd. [19] dereceli azalma yöntemi ile sal olmama, farksal birbiçimlilik, cebirsel derece ve el bağışıklık özellikleri bakımından en iyi olan 8×8 lüğündeki S-kutuları için arama gerçekleştirmişlerdir. a bu çalışmada, başlangıç fonksiyonu olarak ters iyonun seçilmesi durumda, [18] çalışmasından daha k doğrusal olmama değerine sahip S-kutularının duğu belirtilmiştir. Doğrusal olmama ve farksal imlilik özellikleri güçlü olan fakat permütasyon olmayan kuvvet fonksiyonlarından sezgisel arama ile bijektif S- ı elde etme yöntemi Mamadolimov vd. [20] tarafından miştir. Bu yöntem, Isa vd. [21] tarafından tüm kuvvet iyonlarına genelleştirilmiş ve S-kutusunu bijektif hale ek için geliştirdikleri (fazlalık giderme olarak ndirilen) algoritma GF(2)^8 uzayında uygulanmıştır. Isa 22] daha sonra arıların sallanma dansından esinlenerek adıkları sezgisel arama algoritmasını, başlangıç iyonu olarak seçtikleri ters fonksiyona eşdeğer olan üç i bir polinoma uygulayarak önceki sonuçları rmişlerdir. Ivanov vd. [23], ters fonksiyondan elde ettikleri başlangıç popülasyonunu kullanan genetik algoritma ile 8×8 büyüklüğündeki S-kutuları için en yüksek doğrusal olmama değeri olan 112'ye ulaşmışlardır. Aynı çalışmada, benzer yaklaşımla 16×16 büyüklüğündeki S-kutuları için de güçlü kriptografik özellikler elde edilmiştir. Bahsedilen çalışmalardan başlangıç olarak rastgele üretilen S-kutularının kullanıldığı arama algoritmalarının tümü, 8×8 büyüklüğündeki tüm bijektif S-kutularının oluşturduğu (büyüklüğü 2^{1684} olan) arama uzayında koşulmuştur. Bu şekilde yürütülen (ters fonksiyondan başlayarak veya belirli bir matematiksel inşa yöntemini kullanarak arama yapmayan) sezgisel aramaların ürettiği S-kutularının en iyi doğrusal olmama ve farksal birbiçimlilik değerlerinin sırasıyla 104 ve 6 olduğu görülmektedir. Bununla birlikte, Döngüsel Simetrik S-Kutuları (DSSK'lar), DSSK'ların bağlaşımları ve k-DSSK'ların (burada k, $n\times n$ büyüklüğündeki S-kutusu için n'yi bölen ve 1'den büyük sabit bir tamsayıdır) oluşturduğu alt uzaylarda yapılan aramalar sonucunda daha yüksek doğrusal olmama değerleri elde edilebilmektedir [13, 24].

Tablo 1. Bijektif S-kutuları için arama uzaylarının büyüklükleri.

Arama Uzayı	Değişken Sayısı (n)		
	6	8	10
Tüm uzay	2^{296}	2^{1684}	2^{8769}
DSSK uzayı	$2^{47.9}$	$2^{208.3}$	$2^{872.4}$
Bağlaşım uzayı	$2^{61.3}$	$2^{243.7}$	$2^{976.1}$
2-DSSK uzayı	$2^{97.4}$	$2^{412.2}$	$2^{1754.3}$
(n/2)-DSSK uzayı	$2^{141.2}$	$2^{824.7}$	$2^{4345.2}$

DSSK'lar ilk olarak 2008'de Rijmen vd. [25] tarafından tanımlanmıştır. Bu tanım tek çıkışlı döngüsel simetrik Boole fonksiyonları (DSBF) tanımının çok çıkışlı Boole fonksiyonları olan S-kutularını kapsayacak şekilde genişletilmesi olarak görülebilir. Bir $n\times n$ S-kutusu, $1 \le i \le n$ olmak üzere, her bir girişi i kere döngüsel olarak kaydırıldığında karşılık gelen çıkışı da i kere döngüsel olarak kayıyorsa döngüsel simetrik olarak isimlendirilir; diğer bir ifadeyle, $n\times n$ DSSK'lar $\pi(x_0, x_1, \ldots, x_{n-1}) = (x_1, x_2, \ldots, x_{n-1}, x_0)$ permütasyonuna göre simetrik S-kutuları olarak düşünülebilir. Benzer şekilde, büyüklüğü $n\times n$ olan k-DSSK'lar ise $\pi(x_0, x_1, \ldots, x_{n-1}) = (x_k, x_{k+1}, \ldots, x_{n-1}, x_0, x_1, \ldots, , x_{k-1})$ permütasyonuna göre simetrik S-kutuları olarak tanımlanır. DSSK'ların kuvvet fonksiyonlarından ve bunların toplamından üretilen S-kutuları ile doğrusal ilişkili olduğu gösterilmiştir [25]. Temel kriptografik özellikler doğrusal dönüşüm altında değişmediğinden, yüksek doğrusal olmama, düşük farksal birbiçimlilik ve yüksek cebirsel derece gibi istenilen kriptografik özellikleri barındıran ters fonksiyon, Dobbertin, Gold, Kasami fonksiyonları ve benzeri inşalar [6], birer DSSK olarak düşünülebilir. Bu yüzden DSSK sınıfı bahsedilen kriptografik özelliklere sahip S-kutuları açısından zengin bir sınıf oluşturmaktadır. İki tane $(n-1)\times(n-1)$ DSSK'nın bağlaşımı olarak tanımlanan [24] $n\times n$ S-kutuları, $\pi(x_0, x_1, \ldots, x_{n-1}) = (x_0, x_2, \ldots, x_{n-1}, x_1)$ permütasyonuna göre simetrik S-kutularıdır. $n = 6$ için DSSK'lar ve bağlaşımlar üzerine daha önce yapılan çalışmalarda [24, 26], verimli bir tüketici arama algoritması gerçekleştirilmiş ve bu alt uzaylarda ters fonksiyonla aynı doğrusal olmama ve farksal birbiçimlilik değerlerine sahip olan, aynı zamanda ters fonksiyonla afin ilişkili olmayan S-kutuları elde edilmiştir. Ayrıca, bağlaşım yöntemi ile oluşturulan 8×8 S-kutuları için gerçekleştirilen arama algoritması ile doğrusal olmama değeri 106 olan S-kutuları üretilmiştir [24]. 8×8 büyüklüğündeki simetrik S-kutuları üzerine yapılan çalışmada [13] ise, DSSK'lar için uygulanan arama algoritması ile 108 doğrusal olmama değerine ulaşılmıştır.

Blok şifreler ağırlıklı olarak 8 ve daha küçük boyutlu ıaylarda tanımlı olan S-kutularını kullanmaktadır. Bununla likte, daha büyük boyutlarda S-kutusu kullanan blok şifreler bulunmaktadır (örneğin, Kasumi [27] 9×9 S-kutusu llanmaktadır). Bu çalışma, bildiğimiz kadarıyla, 10×10 S-uları için rastgele veya sezgisel aramanın gerçekleştirildiği çalışma niteliğindedir. Özel olarak, 10 boyutlu durumda ma uzayı büyüklükleri sırasıyla $2^{872.4}$ ve $2^{976.1}$, $2^{1754.3}$ ve $^{45.2}$ olan DSSK'ların, bağlaşımların, 2-DSSK'ların ve 5-SSK'ların oluşturduğu alt uzaylarda, bijektif S-kutuları tgele arama yöntemi ve en dik iniş prensibine dayalı arama oritması [28] ile aranmıştır. AES S-kutusunun giriş ve çıkış leri arasındaki en yüksek doğrusal ilişki olasılığının 0.5625 en yüksek farksal olasılığının (S-kutusunun girişine gulanan bir fark ile çıkışında elde edilen farkın aynı kalma sılığı) 0.015625 olduğu bilinmektedir. Gerçekleştirdiğimiz ma sonucu üretilen 10×10 büyüklüğündeki S-kutuları için saplanan en iyi doğrusallık ve farksal olasılıkları ise sırasıyla 546875 ve 0.0078125 olarak bulunmuştur. AES S-usundan daha düşük olan olasılık değerleri, arama yöntemi bahsedilen alt uzaylarda üretilen S-kutularının doğrusal ve ksal kriptanalize karşı daha dayanıklı olabildiğini stermektedir. Bunun yanı sıra, AES S-kutusunun cebirsel ğışıklık açısından (8 boyutlu durum için) en iyi özellikleri ğlamadığı bilinmektedir. Çalışmamızda elde edilen S-ularının ise hem AES S-kutusundan daha iyi cebirsel ğışıklık özelliklerine sahip oldukları hem de 10 boyutlu tum için cebirsel bağışıklık açısından en iyi oldukları lunmuştur. Ayrıca, büyük S-kutularının yan kanal analizine şı dayanıklılığı artırabildiği bilinmektedir [29] ve DSSK'lar lece yörünge temsilcileri ile ifade edilebildiğinden donanım yazılım açısından verimli bir şekilde gerçeklenebilirler [25]; nedenle, çalışmamızda elde edilen sonuçlar pratik açıdan da em taşımaktadır. Diğer taraftan, sezgisel arama oritmasının başlangıç S-kutusu $GF(2)^{10}$ uzayında tanımlı s fonksiyon olarak alındığında, ters fonksiyon ile aynı veya cın kriptografik özelliklere sahip S-kutularının da elde lebildiği gözlenmiştir.

S-kutularının kriptografik özellikleri, arama algoritmasında llanılan maliyet fonksiyonu ve simetrik S-kutuları üzerine bir ıraki bölümde sunulan temel bilgilerden sonra, Bölüm 3'te llandığımız arama algoritmasının genel yapısı ile birlikte çekleştirme detayları verilmektedir. Bölüm 4'te arama oritması ile elde edilen kriptografik özellikler sunularak S S-kutusunun özellikleri ile karşılaştırılmış ve sonuç lümü ile makalemiz sonlandırılmıştır.

2. Temel bilgiler

Bir Boole fonksiyonu $f : GF(2)^n \to GF(2)$, n biti bir bite nderen bir fonksiyondur ve tek çıkışlı Boole fonksiyonu rak da isimlendirilir. f fonksiyonun Hamming ağırlığı, ğruluk tablosundaki birlerin sayısı olarak tanımlanır. ğruluk tablosunda birlerin sayısı sıfırların sayısına eşit olan ole fonksiyonuna dengeli denir. Kriptografik açıdan llanılabilir olması için, Boole fonksiyonun dengeli olması ekmektedir. İki Boole fonksiyonu arasındaki Hamming aklık, doğruluk tablolarında aynı pozisyonlarda bulunan klı bitlerin sayısı olarak tanımlanır.

$n\times m$ büyüklüğünde bir S-kutusu $S : GF(2)^n \to GF(2)^m$ ise, biti m bite gönderen çok çıkışlı bir Boole fonksiyon olarak ımlanır. Herhangi bir S-kutusu S, $x = (x_0, x_1, \ldots, x_{n-1}) \in (2)^n$ olmak üzere, n-değişkenli Boole fonksiyonlarının şturduğu bir kombinasyon, diğer bir ifadeyle $S(x) = (f_0(x), c), \ldots, f_{m-1}(x))$ olarak düşünülebilir. Buradaki f_i fonksiyonları ($i = 0, 1, \ldots, m-1$) koordinat fonksiyonları, bu fonksiyonların sıfırdan farklı ($2^m - 1$ tane) lineer kombinasyonları ise bileşen fonksiyonları olarak isimlendirilir. Sıfır vektöründen farklı bir $\omega = (\omega_0, \omega_1, \ldots, \omega_{m-1}) \in GF(2)^m$ için karşılık gelen bileşen fonksiyonu f_ω ile gösterilir ve aşağıdaki eşitlik ile elde edilir:

$$f_\omega(x) = \omega_0 f_0(x) \oplus \omega_1 f_1(x) \oplus \ldots \oplus \omega_{m-1} f_{m-1}(x). \quad (1)$$

Bu bölümde S-kutuları için verilen doğrusal olmama, mutlak gösterge ve cebirsel derece özellikleri, tek çıkışlı Boole fonksiyonları için tanımlanan kriptografik özelliklerin m-bit çıkışlı S-kutularına genişletilmesi olarak görülebilir.

2.1 Cebirsel derece

Herhangi bir Boole fonksiyon $f : GF(2)^n \to GF(2)$, çıkış bitlerinin oluşturduğu 2^n uzunluğundaki doğruluk tablosu ile veya cebirsel normal biçim olarak isimlendirilen, GF(2) üzerinde çok değişkenli bir polinom ile eşsiz şekilde gösterilebilir. Değişken sayısı n için, cebirsel normal biçimin genel ifadesi aşağıdaki gibidir:

$$f(x) = a_0 \oplus a_1 x_0 \oplus \ldots \oplus a_n x_{n-1} \oplus a_{12} x_0 x_1 \oplus a_{13} x_0 x_2 \oplus \ldots \oplus a_{12\ldots n} x_0 x_1 \ldots x_{n-1}, \quad (2)$$

burada $a_0, a_1, \ldots, a_{12}, a_{13}, \ldots, a_{12\ldots n} \in GF(2)$ eşsiz sabitlerdir ve Möbius dönüşümü ile doğruluk tablosundan elde edilebilir. f Boole fonksiyonunun cebirsel derecesi d_f, cebirsel normal biçimindeki terimlerin sahip olduğu en yüksek değişken sayısıdır. Derecesi en fazla bir olan fonksiyonlar afin fonksiyonlar, sabit terimi sıfır ($a_0 = 0$) olan afin fonksiyonlar ise doğrusal fonksiyonlar olarak adlandırılır.

$n\times m$ büyüklüğünde bir S-kutusu S için cebirsel derece (d_S) değeri, bileşen fonksiyonların aldığı kriptografik açıdan en kötü değer olarak tanımlanır. Diğer bir ifadeyle,

$$d_S = \min_{\omega \neq \mathbf{0} \in GF(2)^m} d_{f_\omega}. \quad (3)$$

Bir S-kutusunun yüksek mertebeden farksal kriptanaliz [5] yöntemine karşı dayanıklı olabilmesi için yüksek cebirsel dereceye sahip olması beklenir.

2.2 Doğrusal olmama

f Boole fonksiyonunun Walsh-Hadamard dönüşümü (veya spektrumu), tüm doğrusal fonksiyonlar ile korelasyonunu görmemizi sağlayan bir dönüşümdür:

$$W_f(w) = \sum_{x \in GF(2)^n} (-1)^{f(x)} (-1)^{w \cdot x}. \quad (4)$$

NL_f doğrusal olmama değerini, spektrumdaki mutlak değerce en büyük değer belirler ve aşağıdaki gibi hesaplanır:

$$NL_f = 2^{n-1} - \frac{1}{2} \max_{w \in GF(2)^n} |W_f(w)|. \quad (5)$$

Diğer bir ifadeyle, bir Boole fonksiyonun doğrusal olmama değeri, tüm afin fonksiyonlara olan Hamming uzaklıklarının en küçüğüdür. Çift değişken sayısı n için, doğrusal olmama değeri açısından en iyi olan Boole fonksiyonları bükük fonksiyonlar olarak isimlendirilir ve (Parseval teoreminden) n-değişkenli bir bükük fonksiyonun tüm spektrum değerleri mutlak değerce $2^{n/2}$'ye eşittir. Fakat bükük fonksiyonlar dengeli değillerdir ve cebirsel dereceleri düşüktür.

Herhangi bir $n\times m$ S-kutusu S için doğrusal olmama (NL_S) değeri, bileşen fonksiyonlarının aldığı en düşük doğrusal olmama değeri olarak tanımlanır. Diğer bir ifadeyle,

$$NL_S = \min_{\omega \neq \mathbf{0} \in \mathrm{GF}(2)^m} NL_{f_\omega}. \quad (6)$$

Bir Boole fonksiyonun doğrusal olmama özelliği, tüm afin fonksiyonlara uzaklıklarının en küçüğü olarak tanımlandığından, doğruluk tablosundaki $2^n - NL_f$ tane bitin afin bir fonksiyonla aynı olduğu ve bu nedenle afin bir fonksiyonla aynı çıktıyı üretme olasılığının (diğer bir ifadeyle doğrusallık olasılığının) $(2^n - NL_f)/2^n$ olduğu görülmektedir. Doğrusal ilişkinin azalması için, bu değerin 0.5'e yaklaşması gerektiğine dikkat edilmelidir. Bir S-kutusu için ise doğrusallık olasılığı, bileşen fonksiyonlarının en kötü (en yüksek) doğrusallık olasılığıdır ve bu değer p_d ile gösterilir. Doğrusal kriptanaliz [2] karşısında dayanıklılık için, S-kutusunun yüksek doğrusal olmama değerine sahip olması beklenir.

Bir S-kutusunun bileşen fonksiyonlarının tüm doğrusal fonksiyonlara yakınlıkları, Doğrusal Yaklaşım Tablosu (DYT) [2, 30, 31] ile ölçülmektedir. $n \times m$ büyüklüğündeki bir S-kutusu S için karşılık gelen DYT, $2^n \times 2^m$ büyüklüğündedir ve bu tablonun her bir elemanı aşağıdaki eşitlik ile bulunur:

$$DYT(u, v) = \#\{x \in \mathrm{GF}(2)^n : u \cdot x = v \cdot S(x)\} - 2^{n-1}, \quad (7)$$

burada $u \in \mathrm{GF}(2)^n$, $v \in \mathrm{GF}(2)^m$ ve "$\cdot$" işlemi iç çarpım işlemidir. Diğer bir ifadeyle, $v \cdot S(x)$ bileşen fonksiyonunun $u \cdot x$ doğrusal fonksiyonuna eşit olma olasılığı $DYT(u, v)/2^n + 0.5$ olur. Ayrıca, S-kutusunun doğrusal olmama değeri, DYT değerleri kullanılarak aşağıdaki eşitlikle bulunabilir:

$$NL_S = 2^{n-1} - \max_{(u,v) \neq (\mathbf{0},\mathbf{0})} |DYT(u, v)|. \quad (8)$$

DYT'deki mutlak değer bakımından en yüksek değer ne kadar küçükse, p_d olasılığının da 0.5'e o kadar yakındır.

2.3 Farksal birbiçimlilik

Herhangi bir $n \times m$ S-kutusu S'nin farksal birbiçimlilik değeri δ_S, $S(x) \oplus S(x \oplus \gamma) = \beta$ eşitliğini sağlayan $x \in \mathrm{GF}(2)^n$ girişlerinin en yüksek sayısıdır ve bu durumda S'ye farksal-δ_S birbiçimlidir denir. Bu tanımdan, S-kutusunun girişine uygulanan bir fark ile karşılık gelen çıkış farkının değişmeme olasılığının $\delta_S/2^n$ olduğu görülmektedir. Bu değer p_f ile gösterilir ve en düşük 2^{n-1} olabilmektedir. Bir S-kutusunun farksal kriptanalize [1] karşı dayanıklı olabilmesi için düşük farksal birbiçimlilik değerine sahip olması beklenir.

Her bir giriş ve çıkış farkı (γ, β) çifti için $S(x) \oplus S(x \oplus \gamma) = \beta$ eşitliğini sağlayan x girişlerinin sayısı Fark Dağılım Tablosu (FDT) ile verilmektedir [1, 30, 31]. S'nin FDT'si $2^n \times 2^m$ büyüklüğündedir ve tablo elemanları aşağıdaki eşitlik ile bulunur:

$$FDT(\gamma, \beta) = \#\{x \in GF(2)^n : S(x) \oplus S(x \oplus \gamma) = \beta\}. \quad (9)$$

Farksal birbiçimlilik değeri, FDT değerleri kullanılarak aşağıdaki gibi bulunabilir:

$$\delta_S = \max_{(\gamma,\beta) \neq (\mathbf{0},\mathbf{0})} FDT(\gamma, \beta). \quad (10)$$

2.4 Cebirsel bağışıklık

S-kutularının giriş ve çıkış bitleri arasında düşük dereceye sahip çok değişkenli polinomlar ile tanımlanan ilişkilerin varlığı, cebirsel saldırılarda kullanılabilmektedir. Büyüklüğü $n \times m$ olan herhangi bir S-kutusu S, $(x_0, x_1, \ldots, x_{n-1}) \in \mathrm{GF}(2)^n$ giriş bitlerini ve $(y_0, y_1, \ldots, y_{m-1}) \in \mathrm{GF}(2)^m$ çıkış bitlerini temsil etmek üzere, aşağıda verilen (r tane) eşitliklerden oluşan bir sistem ile tanımlanabilir:

$$\begin{aligned} &g_0(x_0, x_1, \ldots, x_{n-1}, y_0, y_1, \ldots, y_{m-1}) = 0, \\ &g_1(x_0, x_1, \ldots, x_{n-1}, y_0, y_1, \ldots, y_{m-1}) = 0, \\ &\vdots \\ &g_{r-1}(x_0, x_1, \ldots, x_{n-1}, y_0, y_1, \ldots, y_{m-1}) = 0. \end{aligned} \quad (11)$$

S-kutusunun cebirsel bağışıklığı (I_S) sistemdeki tüm polinomların en düşük derecesi olarak tanımlanır:

$$I_S = \min_{0 \leq i \leq r-1} \deg(g_i). \quad (12)$$

Sistemi oluşturan bağımsız eşitliklerin sayısı ise N_S parametresi ile verilmektedir. Bir S-kutusunun cebirsel saldırılar karşısında dayanıklı olması için cebirsel bağışıklık değerinin yüksek ve bununla birlikte eşitlik sayısının düşük olması beklenir. Çalışmamızda ele aldığımız 8×8 ve 10×10 S-kutuları için ulaşılabilecek en iyi (I_S, N_S) değerleri sırasıyla (3, 441) ve (3, 327)'dir [32, 33].

2.5 Mutlak gösterge

Özilinti fonksiyonu $r_f(d)$, f Boole fonksiyonu ile girişine $d \in \mathrm{GF}(2)^n$ farkı uygulandığında elde edilen versiyonu arasındaki korelasyonu verir:

$$r_f(d) = \sum_{x \in \mathrm{GF}(2)^n} (-1)^{f(x)} (-1)^{f(x \oplus d)}. \quad (13)$$

Özilinti spektrumuyla ilişkili olan ve global çığ etkisi karakteristiği [34] olarak adlandırılan iki önemli kriptografik özellik, mutlak gösterge ve kareler toplamı göstergesidir. Mutlak gösterge AI_f (özilinti fonksiyonu d = (0, .., 0) için her zaman 2^n'ye eşit olduğundan) r_f (0, …, 0) haricinde özilinti spektrumunda bulunan mutlak değerce en büyük değer olarak tanımlanır:

$$AI_f = \max_{d \neq \mathbf{0} \in \mathrm{GF}(2)^n} |r_f(d)|, \quad (14)$$

burada $\mathbf{0}$ = (0, …, 0).

$n \times m$ büyüklüğünde bir S-kutusu S için mutlak gösterge (AI_S) değeri, bileşen fonksiyonların mutlak gösterge değerlerinin en düşüğü olarak tanımlanır:

$$AI_S = \min_{\omega \neq \mathbf{0} \in \mathrm{GF}(2)^m} AI_{f_\omega}. \quad (15)$$

Çalışmamızda maliyet fonksiyonu seçiminde kullanılan ve global çığ etkisi karakteristiklerinden diğeri olan kareler toplamı göstergesi, özilinti spektrumundaki değerlerin karelerinin toplamıdır:

$$\sum_{d \in \mathrm{GF}(2)^n} r_f^2(d). \quad (16)$$

Mutlak ve kareler toplamı göstergelerinin düşük olması, S-kutusunun iyi difüzyon özellikleri taşıdığını gösterir.

2.6 Maliyet fonksiyonu

Kareler toplamı göstergesi ile Walsh-Hadamard spektrumu aşağıda verilen teorem ile ilişkilendirilir.

Teorem 1 [35].

$$\sum_{d \neq \mathbf{0} \in \mathrm{GF}(2)^n} r_f^2(d) = 2^{-n} \sum_{w \in \mathrm{GF}(2)^n} \left(W_f^2(w) - 2^n\right)^2. \quad (17)$$

Teorem 1'den, özilinti değerlerindeki (mutlak değerce) minimizasyonun, aynı zamanda Walsh-Hadamard değerlerini (doğrusal olmama yönünden en iyi olan) bir bükük fonksiyon spektrumuna yaklaştıracağı görülmektedir. Bu nedenle, arama algoritmamızda S-kutusunun bileşen fonksiyonlarının kareler

toplamı göstergeleri minimize edilmeye çalışılmış ve maliyet fonksiyonu olarak bu göstergelerin toplamı seçilmiştir.

2.7 Simetrik S-kutuları

Herhangi bir S-kutusu $S : \mathrm{GF}(2)^n \to \mathrm{GF}(2)^n$, $\pi(x_0, x_1, \ldots, x_{n-1})$ bir permütasyon olmak üzere, her $x = (x_0, x_1, \ldots, x_{n-1}) \in \mathrm{GF}(2)^n$ için $S(\pi(x)) = \pi(S(x))$ koşulunu sağlıyorsa π permütasyonu altında simetrik S-kutusu olarak isimlendirilir. π permütasyonuna göre, bir x vektörü tarafından üretilen yörünge

$$G_n(x) = \{\pi^k(x) \mid 1 \le k \le n\} \tag{18}$$

ile tanımlanır ve yörüngede bulunan vektörlerin sözlüksel (leksikografik) sıralanışında ilk sırada yer alan vektör yörünge temsilcisi olarak isimlendirilir. $n \times n$ büyüklüğünde simetrik bir S-kutusu için toplam yörünge sayısı g_n ile gösterilir.

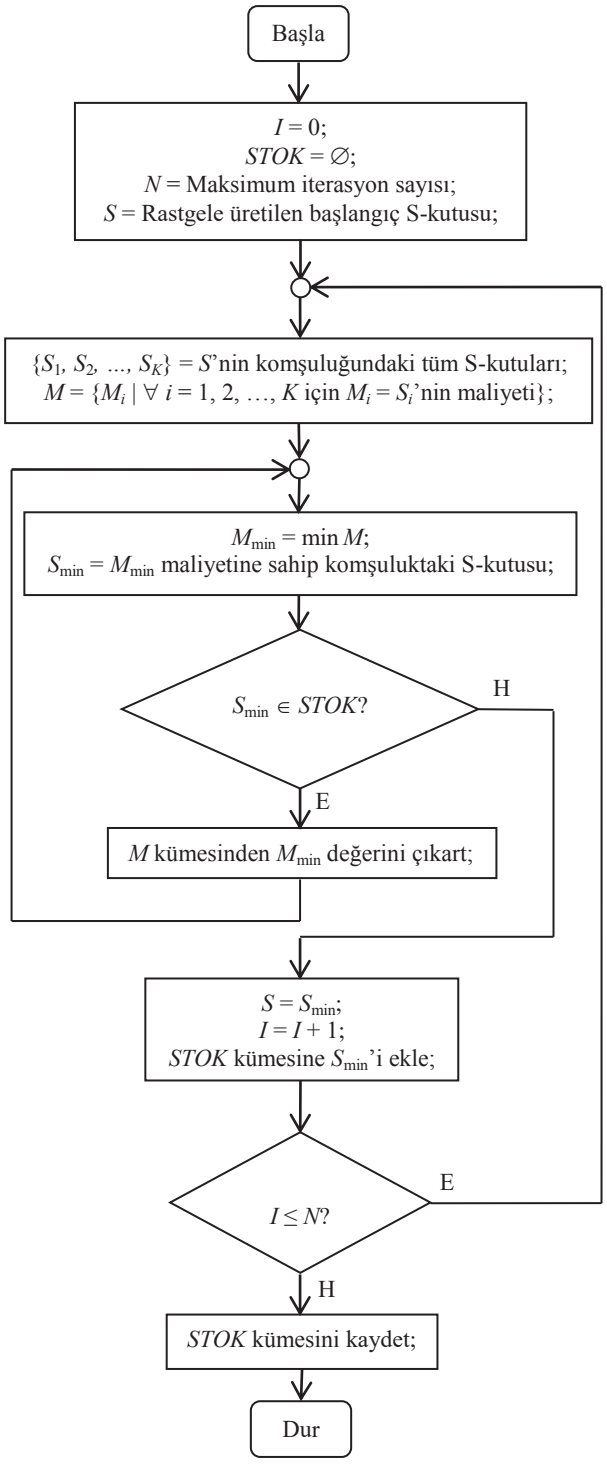
DSSK'lar $\pi(x_0, x_1, \ldots, x_{n-1}) = (x_1, x_2, \ldots, x_{n-1}, x_0)$ permütasyonu altında ve bağlaşımlar $\pi(x_0, x_1, \ldots, x_{n-1}) = (x_0, x_2, \ldots, x_{n-1}, x_1)$ permütasyonu altında simetrik S-kutuları olarak düşünülebilir. Değişken sayısı n çift olmak üzere, 2-DSSK'lar ve $(n/2)$-DSSK'lar ise sırasıyla $\pi(x_0, x_1, \ldots, x_{n-1}) = (x_2, x_3, \ldots, x_{n-1}, x_0, x_1)$ ve $\pi(x_0, x_1, \ldots, x_{n-1}) = (x_{n/2}, x_{n/2+1}, \ldots, x_{n-1}, x_0, x_1, \ldots, x_{n/2-1})$ permütasyonlarına göre simetriktir. $n = 10$ durumunda $\mathrm{GF}(2)^{10}$ vektör uzayı, DSSK'lar için 1, 2, 5 ve 10 büyüklüğünde sırasıyla 2, 1, 6 ve 99 yörüngeye, bağlaşımlar için ise 1, 3 ve 9 büyüklüğünde sırasıyla 4, 4 ve 112 yörüngeye bölüntülenmektedir.

3. Arama algoritması

DSSK'lar ve bağlaşımlar için gerçekleştirilen en dik iniş prensibine dayalı sezgisel arama algoritmasının akış diyagramı Şekil 1'de verilmiştir. Algoritma, DSSK'lar veya bağlaşımların oluşturduğu alt uzayda rastgele üretilen bir S-kutusu (S) ile başlamaktadır ve arama aynı alt uzayda gerçekleşmektedir. İterasyon sayısı (N) denemelerimizde 400 olarak alınmıştır. Aynı iterasyon çıktılarının üretilmesini engellemek için her iterasyon çıktısı *STOK* kümesine kaydedilir. Algoritmanın her bir iterasyonunda, iterasyon girişindeki S-kutusunda aşağıda verilen değişiklikler yapılarak, iterasyon çıktısı için aday S-kutularının bulunduğu ve bunların her biri için maliyet fonksiyonunun hesaplandığı bir komşuluk oluşturulmaktadır.

- Büyüklüğü birden fazla olan çıkış yörüngelerinin olası tüm permütasyonları ile yer değiştirmesi. Diğer bir ifadeyle, her $1 \le k < t$ değeri için çıkış yörüngesinde bulunan tüm $S(x)$ vektörleri $\pi^k(S(x))$ olarak değiştirilir, burada t değeri $\pi^t(x) = x$ eşitliğini sağlayan en küçük değerdir. Örneğin, DSSK'lar için 10 büyüklüğünde 99 yörünge bulunmaktadır; buna göre, böyle bir yörüngedeki vektörler en fazla 9 kere kaydırılabilir ve böylelikle $9 \times 99 = 891$ komşu elde edilir. Bu adımda DSSK'lar için 916 ve bağlaşımlar için 904 komşu üretilir.
- Aynı büyüklükte iki farklı çıkış yörüngesinin permütasyonları göz önüne alınmadan birbiri ile değiştirilmesi. Örneğin bu şekilde, DSSK'lar için 10 büyüklüğündeki 99 yörüngeden $99 \times 98/2 = 4851$ komşu elde edilir. Böylelikle DSSK'lar için 4867, bağlaşımlar için 6228 komşu bulunur.

Bu iki adımın sonunda komşulukta bulunan S-kutularının toplam sayısı (Şekil 1'deki K parametresi) DSSK'lar için 5783 ve bağlaşımlar için 7132 bulunmaktadır. Her iterasyonda, iterasyon girdisi S'nin komşuluğunda bulunan S-kutularının maliyetleri hesaplanır ve *STOK* kümesinde bulunmayan en düşük maliyetli S-kutusu, iterasyon çıktısı olarak *STOK* kümesine eklenir.



Şekil 1. En Dik İniş Prensibine Dayalı Arama Algoritması.

Komşuluktaki herhangi bir S-kutusu için algoritmada kullanılan maliyet fonksiyonu, (6×6 ve 8×8 S-kutuları için yapılan çalışmalarda [13, 26] iyi sonuçlar veren) bileşen fonksiyonlarının kareler toplamı göstergelerinin toplamıdır:

$$\sum_{\omega \neq \mathbf{0} \in \mathrm{GF}(2)^n} \sum_{d \neq \mathbf{0} \in \mathrm{GF}(2)^n} r_{f_\omega}^2(d), \quad (19)$$

burada $d = \mathbf{0}$ için özilinti değeri sabit olduğundan hesaplamaya katılmamaktadır.

Algoritma C dilinde gerçeklenmiş ve Windows 8.1 Pro işletim sistemi, Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50GHz işlemci ve 16 GB RAM'e sahip bir bilgisayarda bütün çekirdekler kullanılarak iki hafta çalıştırılmıştır. N = 400 için algoritmanın tek çekirdek üzerinde bir kere koşulması DSSK'larda 2 saat sürerken, bağlaşımlarda 2.5 saat, 2-RSSB'lerde 9 saat ve 5-RSSB'lerde ise yaklaşık 6 gün sürmektedir. Algoritmanın kodları [36]'da verilen bağlantıdan indirilebilmektedir.

3.1 Zaman ve bellek karmaşıklığı

Algoritma kısır döngüye girmemek için, her iterasyon sonucunu önceki iterasyon sonuçları ile karşılaştırmaktadır. Bu nedenle, genel olarak $n \times n$ büyüklüğündeki S-kutuları için algoritmanın koşulduğu düşünülürse, N iterasyon için $2^n N(N-1)/2$ karşılaştırma işlemi yapılmaktadır. Herhangi bir iterasyonda, önceki iterasyon sonuçlarından birisi üretilirse karşılaştırma işlemi tekrar etmektedir; bununla birlikte, yapılan tekrarın işlem yükü açısından etkisi sınırlıdır. Ayrıca, her iterasyonda her bir komşuluk için (19) ile verilen maliyet fonksiyonu hesaplanmaktadır. Maliyet fonksiyonunu bileşen fonksiyonlarının kareler toplamı göstergelerini kullanarak hesaplamak, özilinti değerlerini elde etmek için iki kere Walsh-Hadamard dönüşümü almayı gerektirir. Bunun yerine, ((17) eşitliğinden) doğrudan Walsh-Hadamard spektrumlarını kullanarak hesaplamak daha verimlidir. n değişkenli bir Boole fonksiyonun Walsh-Hadamard dönüşümünün $n2^n$ toplama ve çıkarma işlemi ile elde edilebildiği bilinmektedir. Mutlak Walsh-Hadamard değerlerinin dağılımı afin dönüşüm altında değişmez olduğundan, simetrik bir S-kutusunun maliyet fonksiyonunu hesaplamak için bütün bileşen fonksiyonlar yerine sadece birbiri ile afin ilişkili olmayan ($g_n - 1$ tane) bileşen fonksiyonların Walsh-Hadamard spektrumlarını elde etmek yeterlidir. Bu spektrumların tümü elde edildikten sonra, (17) eşitliğini kullanarak maliyet fonksiyonunun hesaplanması için, kare alma işlemlerinin yapılması gerektiği görülmektedir. Buna karşın, olası tüm Walsh-Hadamard dönüşümü değerleri için $\left(W_f^2(w) - 2^n\right)^2$ işleminin karşılık gelen sonuçları önceden bir diziye atılarak, çarpma işlemi yapılmadan $2^n(g_n - 1)$ toplama işlemi ile maliyet fonksiyonu hesaplanabilir. Bunun sonucunda, N iterasyon ve K komşu için $(n+1)2^n(g_n - 1)NK$ toplama ve çıkarma işleminin yapılması gerektiği görülmektedir. Bu nedenle, simetrik S-kutuları için komşu sayısı $K \approx \binom{g_n}{2} + ng_n$ olduğundan, sabit iterasyon sayısı için asimptotik zaman karmaşıklığı $O(n2^n g_n^3)$ olarak elde edilir.

Kriptografik elemanların tasarımında yaygın olarak kullanılan tavlama benzetimi ve tepe tırmanma gibi diğer benzer arama yöntemleri ile karşılaştırıldığında, en dik iniş prensibine dayalı arama algoritmasında olduğu gibi tüm komşulukta arama yapmadıkları için, bu yöntemlerin daha verimli oldukları düşünülebilir. Bununla birlikte, tepe tırmanma algoritmasının zayıf yönü yerel minimumdan kaçamamasıdır. Tavlama benzetimi algoritması ise tüm komşulukta arama yapmamaktadır ve bu nedenle daha iyi sonuçları kaçırma olasılığı bulunmaktadır. Ayrıca, başlangıç sıcaklığı, soğutma çarpanı, (bir iterasyonda rastgele üretilen) komşu sayısı gibi parametrelerinin ayarlanmasına ihtiyaç duymaktadır. Gerek tavlama benzetimi gerek tepe tırmanma yöntemlerinde, belirli bir komşu üretme operatörü ile elde edilebilen tüm komşuluğa bakıldığı varsayıldığında, asimptotik karmaşıklıklarının en dik iniş prensibine dayalı arama algoritmasınınki ile aynı olduğu görülmektedir. Burada elde edilen avantaj, yerel minimuma takılmadan her zaman komşuluk içerisindeki en iyi çözümün üretilmesidir. Bellek açısından değerlendirdiğimizde, en dik iniş prensibine dayalı arama algoritması her iterasyon çıktısını kaydetmektedir ve bu yüzden belirlenebilecek en yüksek iterasyon sayısı kullanılan bellek kapasitesi ile sınırlıdır. İterasyon çıktılarını kaydetmek için ihtiyaç duyulan bellek miktarının $n2^n N$ bit olduğu kolaylıkla görülebilir. Bunun yanı sıra, simetrik S-kutuları yörünge temsilcileri ile temsil edilebildiğinden, sadece $ng_n N$ bit, tüm iterasyon çıktılarını kaydetmek için yeterlidir. Örneğin, döngüsel simetrik S-kutuları için $g_n \approx \frac{2^n}{n}$ olduğundan, gerek duyulan belleğin $2^n N$ bit olduğu görülür. Bu gereksinim, genellikle makul büyüklükteki S-kutuları ve iterasyon sayıları için karşılanabilir niteliktedir.

4. Bulgular

En dik iniş prensibine dayalı sezgisel arama ve rastgele arama algoritmalarından elde edilen en iyi sonuçlar ile birlikte AES S-kutusunun kriptografik özellikleri Tablo 2'de sunulmuştur. Tablo 2'den, bağlaşımlar için elde edilen sonuçların, DSSK'lar için elde edilen sonuçlara yakın olduğu gözlenmektedir. Sezgisel aramanın rastgele aramadan daha iyi sonuçlar verdiği, her iki alt uzayda da en yüksek olmama değerinin 456, en düşük farksal birbiçimliliğin 8 ve en düşük mutlak göstergenin 168 bulunduğu görülmektedir; bununla birlikte DSSK alt uzayında bulunan 456 doğrusal olmama değerine sahip sonucun farksal birbiçimlilik değeri (δ_S = 10) daha iyi çıkmıştır.

Tablo 2. (NL_S, AI_S, δ_S, d_S), (p_d, p_f) ve (I_S, N_S) sonuçlarının karşılaştırılması.

	DSSK'lar	Bağlaşımlar
Rastgele arama	(450, 200, 12, 9) (448, 200, 10, 9) (440, 176, 14, 9)	(450, 304, 12, 9) (448, 334, 10, 9) (440, 176, 12, 9)
En iyi (p_d, p_f)	(0.560546875, 0.009765625)	
En iyi (I_S, N_S)	(3, 327)	
Sezgisel arama	(456, 192, 10, 9) (454, 184, 8, 9) (448, 168, 10, 9)	(456, 192, 12, 9) (454, 184, 8, 9) (448, 168, 10, 9)
En iyi (p_d, p_f)	(0.5546875, 0.0078125)	
En iyi (I_S, N_S)	(3, 327)	
	2-DSSK'lar	5-DSSK'lar
Rastgele arama	(448, 224, 10, 9) (444, 184, 10, 9) (442, 176, 12, 9)	(442, 232, 14, 9) (440, 216, 12, 9) (432, 200, 12, 9)
En iyi (p_d, p_f)	(0.5625, 0.009765625)	
En iyi (I_S, N_S)	(3, 327)	
Sezgisel arama	(454, 208, 8, 9) (454, 176, 10, 9) (444, 176, 8, 9)	(450, 200, 10, 9)
En iyi (p_d, p_f)	(0.556640625, 0.0078125)	
En iyi (I_S, N_S)	(3, 327)	
AES S-kutusu	(112, 32, 4, 7)	
(p_d, p_f)	(0.5625, 0.015625)	
(I_S, N_S)	(2, 39)	

Rastgele üretme yöntemi ile elde edilen sonuçlardan mutlak göstergesi en düşük (AI_S = 176) olanların doğrusal olmama değerleri düşük (NL_S = 440) olduğundan, AES S-

utusundan daha kötü doğrusallık olasılığına (p_d = 1024−440)/1024 = 0.5703125) sahip oldukları gözlenmektedir. 48 doğrusal olmama değeri AES S-kutusu ile aynı (p_d = .5625) ve 450 doğrusal olmama değeri ise AES S-kutusundan aha iyi (p_d = 0.560546875) doğrusallık olasılığı vermektedir. 'arksal birbiçimlilik açısından bakıldığında ise rastgele üretme le bulunan 10, 12 ve 14 değerlerinin tümü AES S-kutusundan aha düşük p_f olasılığı üretmektedir. Sezgisel arama lgoritmasının her iki alt uzay için de kriptografik özellikleri yileştirerek, AES S-kutusundan daha iyi doğrusallık ve farksal lasılıkları bulduğu görülmektedir.

Diğer taraftan, n çift olmak üzere, GF(2)^n uzayında tanımlı ers fonksiyonun $2^{n-1}-2^{n/2}$ doğrusal olmama değerine sahip arksal-4 birbiçimli olduğu bilinmektedir [11]. Sezgisel arama lgoritmasında, başlangıç S-kutusu olarak rastgele üretilen bir -kutusu (DSSK veya bağlaşım) yerine ters fonksiyon ullanıldığında, ters fonksiyon ile aynı veya yakın kriptografik zelliklere sahip S-kutuları üretilmiştir. Özel olarak, arama lgoritması 400 iterasyon için koşulduğunda, bulunan doğrusal lmama değerleri 470, 472, 474, 476, 478, 480 ve farksal irbiçimlilik değerleri 4, 6, 8, 10, 14 olarak elde edilmiştir.

Karşılaştırma amaçlı olarak 2-DSSK'ların ve 5-DSSK'ların luşturduğu alt uzaylarda yürüttüğümüz arama sonuçlarına akıldığında, (Tablo 1'de sunulan) özellikle 5-DSSK'lar için rama uzayının büyüklüğünden dolayı diğer alt uzaylarda elde dilen sonuçlara ulaşılamadığı; bununla birlikte her iki alt uzay çin de sezgisel arama algoritmasının, rastgele üretme yöntemi le elde edilen sonuçları iyileştirdiği gözlenmektedir. Tablo 'de sunulan sonuçlar cebirsel bağışıklık açısından ele lındığında, 10 boyutlu alt uzayların tümü için elde edilen S-utularının cebirsel bağışıklıklarının en iyi olduğu ve AES S-utusundan daha iyi cebirsel bağışıklık sağladıkları örülmektedir.

Tablo 2'de verilen sonuçlardan DSSK'lar için elde edilen en iyi) 456 ve 454 doğrusal olmama değerlerine sahip S-utularını sırasıyla S_1 ve S_2 ile, bağlaşımlar için elde edilen ynı doğrusal olmama değerlerine sahip S-kutularını sırasıyla $_3$ ve S_4 ile, ve AES S-kutusunu AES ile gösterelim. Bu S-utularının FDT ve DYT'leri, S-kutuları ile birlikte [36]'da erilen bağlantıdan indirilebilmektedir. Bu tablolar çok büyük lduğundan, burada (10 boyutlu durum için her biri 64×1024, 8 oyutlu durum için ise her biri 16×256 büyüklüğünde) 16 arçaya bölüntülenerek, her bir bölüntüdeki mutlak değerce en üyük değer FDT için Tablo 3'te ve her bir bölüntüdeki en üyük değer DYT için Tablo 4'te sunulmuştur.

Tablo 3. Elde edilen en iyi sonuçların ve AES S-kutusunun FDT'lerinin karşılaştırılması.

Bölüm #	S_1	S_2	S_3	S_4	AES
1	8	8	10	8	4
2	10	8	10	8	4
3	10	8	10	8	4
4	8	8	10	8	4
5	10	8	10	8	4
6	10	8	10	8	4
7	8	8	10	8	4
8	8	8	10	8	4
9	10	8	12	8	4
10	8	8	8	8	4
11	10	8	8	8	4
12	8	8	8	8	4
13	8	8	8	8	4
14	8	8	8	8	4
15	8	8	8	8	4
16	8	8	12	8	4

Tablo 4. Elde edilen en iyi sonuçların ve AES S-kutusunun DYT'lerinin karşılaştırılması.

Bölüm #	S_1	S_2	S_3	S_4	AES
1	56	58	56	58	−16
2	−56	−58	−56	58	−16
3	−56	−58	56	58	−16
4	56	58	56	−58	−16
5	56	−58	56	58	16
6	56	−58	56	−58	16
7	56	−56	56	−58	−16
8	−56	58	−56	−58	−16
9	56	58	56	56	16
10	−56	−58	56	58	16
11	56	−58	−56	56	−16
12	56	−56	56	58	−16
13	56	58	56	58	16
14	−56	56	−56	56	−16
15	−56	58	56	58	−16
16	−56	58	−56	58	16

FDT ve DYT dağılımları incelendiğinde, farksal birbiçimlilik değeri 8 olan S-kutularının, farksal birbiçimlilik değeri 10 ve 12 olan S-kutularına göre daha tekdüze FDT'lere sahip oldukları; benzer şekilde, doğrusal olmama değeri 456 olan S-kutularının, doğrusal olmama değeri 454 olan S-kutularına göre daha tekdüze DYT'lere sahip oldukları oldukları görülmektedir. AES S-kutusunun doğrusal olmama ve farksal birbiçimlilik değerlerinin en iyiye yakın olmasından dolayı, tabloların bütününe [36] bakıldığında hem FDT hem de DYT'sinin arama algoritması ile elde edilen S-kutularına göre daha tekdüze olduğu gözlenmektedir.

5. Sonuç

10×10 büyüklüğündeki S-kutuları için DSSK'lar ve bağlaşımların sırasıyla $2^{872.4}$ ve $2^{976.1}$ olan arama uzaylarında uyguladığımız rastgele veya sezgisel arama yöntemleri ile bulunan S-kutularının, doğrusal, farksal ve cebirsel kriptanalize karşı AES S-kutusundan daha dayanıklı olabileceği gösterilmiştir. Bu çalışmada elde edilen kriptografik özelliklerin ve kullanılan metodun, büyük S-kutularının aranmasında farklı sezgisel arama yöntemlerinin geliştirilmesini motive edebilecek nitelikte olduğu düşünülmektedir. Gözlemlerimiz, elde edilen S-kutularının AES S-kutusuna göre üstünlüğünü yansıtmaktan çok, S-kutusu büyüklüğünün doğrusal olmama, farksal birbiçimlilik ve cebirsel bağışıklık gibi kriptografik özelliklerin sağladığı kriptanaliz karşısında dayanıklılığa etkisinin bağımsız biçimde onaylanması olarak yorumlanmalıdır.

Kaynakça

[1] E. Biham, A. Shamir. Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4(1):3-72, 1991.

[2] M. Matsui. M. Linear cryptanalysis method for DES cipher. In: EUROCRYPT'93, LNCS, vol. 765, pp. 386-397, Springer, 1994.

[3] N.T. Courtois, J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In: Advances in Cryptology - ASIACRYPT 2002, LNCS, vol. 2501, pp. 267-287, Springer, 2002.

[4] N.T. Courtois. General principles of algebraic attacks and new design criteria for cipher components. In: Advanced Encryption Standard - AES 2004, LNCS, vol. 3373, pp. 67-83, Springer, 2005.

5] X. Lai. Higher order derivatives and differential cryptanalysis. In: "Symposium on Communication, Coding and Cryptography", in honor of J. L. Massey on the occasion of his 60'th birthday, The Springer International Series in Engineering and Computer Science, vol. 276, pp. 27-233, Springer, 1994.

6] C. Carlet. Vectorial Boolean functions for cryptography. In: Yves Crama, Peter L. Hammer (Eds.), Chapter of the Monography "Boolean Models and Methods in Mathematics, Computer Science, and Engineering", Cambridge University Press, pp. 398-469, 2010.

7] C. Carlet. On highly nonlinear S-boxes and their inability to thwart DPA attacks. In: Proceedings of INDOCRYPT'05, LNCS, vol. 3797, pp. 49-62, Springer, 2005.

8] W. Millan. How to improve the nonlinearity of bijective S-boxes. In: Australasian Conference on Information Security and Privacy, vol. 1438, pp 181-192, Springer, 1998.

9] S. Picek, M. Cupici L. Rotim. A New Cost Function for Evolution of S-Boxes. Evolutionary Computation, 24(4):695-718, 2016.

10] J. Daemen, V. Rijmen. AES Proposal: Rijndael. NIST Publication, 1999.

11] K. Nyberg. Differentially uniform mappings for cryptography. In: Proceedings of EUROCRYPT'93, LNCS, vol. 765, pp. 55-64, Springer, 1994.

12] M. A. Evci, S. Kavut. DPA resilience of rotation-symmetric S-boxes. In: Proceedings of IWSEC 2014, LNCS, vol. 8639, pp. 146-157, Springer, 2014.

13] S. Kavut, S. Tutdere. Highly nonlinear (vectorial) Boolean functions that are symmetric under some permutations. Advances in Mathematics of Communications, 14 (1):127-136, 2020.

14] D. Jakobovic, S. Picek, M. S. R. Martins, M. Wagner. A characterisation of S-box fitness landscapes in cryptography. In: Proceedings of Genetic and Evolutionary Computation Conference – GECCO'19, pp. 285-293, 2019.

15] D. Jakobovic, S. Picek, M. S. R. Martins, M. Wagner. Toward more efficient heuristic construction of Boolean functions. Applied Soft Computing, vol. 107, 107327, 2021.

16] W. Millan, L. Burnett, G. Carter, A. Clark, E. Dawson. Evolutionary heuristics for finding cryptographically strong S-boxes. In: International Conference on Information and Communications Security, LNCS, vol. 1726, pp 263-274, Springer, 1999.

17] J. A. Clark, J. L. Jacob, S. Stepney. The design of S-boxes by simulated annealing. New Generation Computing, 23(3):219-231, 2005.

18] P. Tesař. A new method for generating high non-linearity s-boxes. Radio Engineerng, 19(1):23-26, 2010.

19] O. V. Kazymyrov, V. N. Kazymyrova, R. V. Oliynykov. A method for generation of high-nonlinear S-Boxes based on gradient descent. Mat. Vopr. Kriptogr., 5(2):71-78, 2014.

20] A. Mamadolimov, H. Isa, M. S. Mohamad. Practical Bijective S-box Design, arXiv:1301.4723v1, 2013.

21] H. Isa, N. Jamil, M. R. Z'aba. S-box construction from non-permutation power functions. In: Proceedings of the 6th International Conference on Security of Information and Networks, pp. 46-53, 2013.

22] H. Isa, N. Jamil, M. R. Z'aba. Construction of cryptographically strong S-boxes inspired by bee waggle dance. New Generation Computing, 34(3):221-38, 2016.

23] G. Ivanov, N. Nikolov, S. Nikova. Reversed genetic algorithms for generation of bijective s-boxes with good cryptographic properties. Cryptography and Communications, 8:247-276, 2016.

[24] S. Kavut, S. Baloğlu. Results on symmetric S-boxes constructed by concatenation of RSSBs. Cryptography and Communications, 11:641-660, 2019.

[25] V. Rijmen, P. S. L. M. Barreto, D. L. G. Filho. Rotation symmetry in algebraically generated cryptographic substitution tables. Information Processing Letters, 106:246-250, 2008.

[26] S. Kavut. Results on rotation-symmetric S-boxes. Information Sciences, 201:93-113, 2012.

[27] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, 3G Security, Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification, V.3.1.1, 2001.

[28] M. Bartholomew-Biggs. Chapter 5: The steepest descent method, nonlinear optimization with financial applications. pp. 51-64. Springer, 2005.

[29] L. Goubin, A. Martinelli, M. Walle. Impact of sboxes size upon side channel resistance and block cipher design. In AFRICACRYPT'13, LNCS, vol. 7918, pp. 240-259, Springer, 2013.

[30] B. Aslan, M. T. Sakalli, E. Bulus. Classifying 8-bit to 8-bit S-Boxes based on power mappings from the point of DDT and LAT Distributions. In: Proceedings of Arithmetic of Finite Fields – WAIFI 2008, LNCS, vol. 5130, pp. 123-133, Springer, 2008.

[31] H. Heys, C. M. Adams. A tutorial on linear and differential cryptanalysis. Cryptologia, 26(3):189-221, 2002.

[32] O. Kazymyrov. Methods and tools for analysis of symm etric cryptographic primitives. PhD thesis, The Selmer Center, Department of Informatics, University of Bergen, Norway, 2014.

[33] A. M. Eilertsen, O. Kazymyrov, V. Kazymyrova, M. Storetvedt. A Sage library for analysis of nonlinear binary mapping. In: Pre-proceedings of Central European Conference on Cryptology – CECC'14, pp. 69-78, 2014.

[34] X. M. Zhang and Z. Yheng. GAC – the criterion for global avalanche characteristics of cryptographic functions, Journal for Universal Computer Science, 1(5):316-333, 1995.

[35] M. D. Yücel. Alternative nonlinearity criteria for Boolean functions. Electrical and Electronics Engineering Department, Middle East Technical University, Memorandum No. 2001-1, 2001.

[36] GitHub, URL: https://github.com/Selcuk-kripto/sbox10, (Erişim tarihi: 27, 02, 2022).

Özgeçmişler

Selçuk Kavut, Ankara Üniversitesi Elektronik Mühendisliği Bölümü'nden lisans derecesini 1998 yılında, Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik ve Elektronik Mühendisliği Anabilim Dalı'ndan yüksek lisans ve doktora derecelerini sırasıyla 2002 ve 2008 yıllarında almıştır. 2009-2014 yılları arasında Gebze Yüksek Teknoloji Enstitüsü'nde Öğr. Gör. Dr. olarak çalıştıktan sonra Balıkesir Üniversitesi'ne geçmiş olup, halen Bilgisayar Mühendisliği Bölümü'nde Doç. Dr. olarak çalışmaktadır. Çalışma alanları kriptoloji ve kodlama teorisi üzerinedir.

Makale Türü: Araştırma Makalesi
Makale Gönderim Tarihi:14.04.2022
Makale Kabul Tarihi: 20.05.2022

Şebekeden Bağımsız Güneş/Rüzgâr/Biyogaz/Yakıt Hücresi/Batarya Tabanlı Hibrit Enerji Sisteminin Tekno-Ekonomik Analizi: Muğla Zaferler Köyü Vaka Çalışması

Techno-Economic Analysis of an Off-Grid Solar/Wind/Biogas/Fuel Cell/Battery Based Hybrid Energy System: Muğla Zaferler Village Case Study

Aykut Fatih Güven[1], Cüneyt Hatipoğlu[2]

[1,2] Yalova Üniversitesi, Mühendislik Fakültesi, Enerji Sistemleri Mühendisliği Bölümü, Yalova/TÜRKİYE
afatih.guven@yalova.edu.tr, hatip.cuneyt@gmail.com

Öz

ünya çapında hızla artan enerji tüketimi, fosil yakıtlar ve ra gazları nedeniyle sürdürülebilir bir dünya için alternatif erji kaynaklarına ihtiyaç duyulmaktadır. Geleneksel enerji rlıklarına bağımlılığı azaltmak için şebekeden bağımsız nilenebilir enerji çerçevesi kullanılabilir. Bunun yanında ibrit Yenilenebilir Enerji Sisteminin (HRES) ani yük ğişimlerine karşı duyarlı ve düşük maliyetli olması içinde en i şekilde boyutlandırılması gerekmektedir. Bu bağlamda lışmada, 220 hane ve 590 kişiden oluşan, 2021 yılına ait talama 1.087,26 kWh/gün ve 162,15 kW elektrik pik yükü lebine sahip Muğla ili Köyceğiz ilçesine bağlı olan Zaferler yü için şebekeden bağımsız HRES 3 farklı senaryo uşturularak araştırılmıştır. HOMER (Hybrid Optimization odel for Electric Renewable) programı kullanılarak seçilen lot bölgenin enerji ihtiyacının HRES ile 3 farklı senaryoya re optimal olarak sağlanıp sağlanamayacağı analiz lmiştir. Şebekeden bağımsız HRES için yakıt Hücresi ve neratör içeren ama dizel jeneratör içermeyen senaryo 3 en timum sonuç olarak değerlendirilmiştir. Optimizasyon nucunda, sistemin birim enerji maliyeti ve net bugünkü ğer maliyeti 0,152 $ ve 2.69 milyon $ olarak hesaplanmıştır. rada, güneş paneli % 78.9, rüzgar türbini % 1.52, biyogaz neratörü % 1.15 ve yakıt hücresi jeneratörü % 18.4 katkı anı ile ihtiyaç olan toplam enerjiyi karşılamaktadır. Genel arak çalışmada yenilenebilir enerji alanında kurulabilecek brit enerji sistemlerine iyi bir örnek teşkil ederek özellikle ke olarak bu alanda büyük bir potansiyelimiz olduğunu istermekte ve araştırmacıların da bu alanda çalışma pmalarını teşvik edici özelliktedir.

nahtar Kelimeler: Hibrit enerji sistemleri, nilenebilir enerji optimizasyonu, maliyet analizi, OMER.

Abstract

Alternative energy sources are needed for a sustainable world due to rapidly increasing energy consumption, fossil fuels and greenhouse gases worldwide. Off-grid renewable energy framework can be used to reduce dependency on traditional energy assets. In addition, the Hybrid Renewable Energy System (HRES) should be optimally sized to be sensitive to sudden load changes and low cost. In this context, in the study, 3 different off-grid HRES scenarios were created for Zaferler village, which consists of 220 households and 590 people, and has an average electricity demand of 1,087.26 kWh/day and 162.15 kW electricity peak load for 2021, which is connected to the town of Köyceğiz in Muğla province. By using the HOMER (Hybrid Optimization Model for Electric Renewable) program, it has been analyzed whether the energy need of the selected pilot region can be optimally met with HRES according to 3 different scenarios. For off-grid HRES, scenario 3 with fuel Cell and generator but no diesel generator was evaluated as the most optimal result. As a result of the optimization, the unit energy cost and net present value cost of the system were calculated as $0.152 and $2.69 million. Here, solar panel 78.9%, wind turbine 1.52%, biogas generator 1.15% and fuel cell generator 18.4% contribute to the total energy needed. In general, the study sets a good example for hybrid energy systems that can be established in the field of renewable energy, showing that we have a great potential in this field, especially as a country, and it encourages researchers to work in this field.

Keywords: Hybrid energy systems, renewable energy optimization, cost analysis, HOMER.

1. Giriş

Enerji insanların yaşam tarzını ve kalitesini belirleyen nemli bir faktör olup, ekonominin temel girdisini luşturmakta ve medeniyetin sürekliliğindeki en önemli ereksinimdir. Dünya nüfusu ve bağlantılı olarak enerji ıtiyacı sürekli olarak artmaktadır. Gelişmekte olan ülkeler in enerji, kırsal alanların kalkınması için önemli bir çenektir. Geleneksel kaynaklar bu enerjiyi kesintisiz olarak darik etmek için yeterli değildir. Son zamanlarda dünya eneline bakıldığında enerji krizleri ortaya çıkmaya aşlamıştır. Bu nedenle yenilenebilir enerji kaynakları bugün e gelecekte tek umut kaynağımız olacaktır [1]. Dünya zerinde farklı formlarda ve miktarlarda her ülkede enilenebilir enerji kaynağı mevcuttur. Türkiye'nin üneşlenme süresi ve düşen radyasyon değerlerine akıldığında ülkenin tüm bölgelerinin birbirlerine göre vantajları olmakla birlikte güneş enerjisi için uygun olduğu örülmektedir. Bu bağlamda Türkiye birçok yenilenebilir nerji kaynağını bünyesinde barındırmaktadır. Bu enilenebilir enerji kaynakları coğrafi bölgeye uygun olarak çileceğinden dolayı, farklı bölgelerde farklı yenilenebilir nerji kaynaklarının kombinasyonlarından yararlanarak hibrit stem kurmak da mümkündür.

enilenebilir enerji kaynakları içerisinde güneş ve rüzgar nerjisi kaynakları, güneş ışınımı ve rüzgar hızı değerleri aatler veya günler arasında büyük ölçüde değişebilir. Bu aynı amanda, özellikle bu tür sistemlerin büyük ölçekli yatırımları lanlandığında, güvenilir ve istikrarlı bir enerji sistemi için orunlara neden olan ölçülemez bir belirsizlik yaratır. enilenebilir enerji kaynaklarının değişken doğasından dolayı rtaya çıkan belirsizlikler, yedekleme üniteleri kullanma orunluluğunu da beraberinde getirmektedir. Bu da üretim aliyetini artırmaktadır [2]. Bu nedenle, yenilenebilir enerji aynaklarına ait sayısal verilerin ölçülmesi, elektrik enerjisi ve nerji üretim sistemlerinin planlanması, yönetimi ve verimli alışması sağlanarak belirsizlik azaltılır. Yenilenebilir enerji aynaklarının istikrarsız ve değişken doğasının üstesinden elmek için temel ve ana çözüm, HRES olarak adlandırılan irden fazla yenilenebilir enerji kaynağı kullanmaktır. Bir nerji sisteminde daha fazla enerji kaynağının kullanılmasıyla, ıtiyaç duyulan enerji, tek bir yenilenebilir enerji kaynağına öre daha az maliyetle, ihtiyaç duyulan saatler ve mevsimler in üretilebilir. Bununla birlikte, belirli bir konumda yük ılebini karşılamak için en uygun boyutlandırmanın elirlenmesi, enerji kaynaklarının değişken doğası, başarılı bir aliyet modelinin hesaplanmasının zorluğu ve optimum oyutlandırmada optimizasyon algoritmalarının uzun işlem asamakları nedeniyle zordur [3]. Bu zorluğun üstesinden elmek için yazılım programlarına ihtiyaç duyulmaktadır. iteratürde birçok optimizasyon sürecine fayda sağlayan azılımlar olmasına rağmen, bu yazılımların içinde HOMER n fazla kullanılan, farklı kombinasyonların daha kolay ve assas şekilde değerlendiren bir yazılımdır.

HRES'lerde şebekeye bağlı veya şebekeden bağımsız olarak tasarlanmış sistemin en iyi çalışma koşullarını elde etmek için HOMER yazılımın kullanıldığı farklı çalışmalar bulunmaktadır [4-11]. Akan çalışmasında, Türkiye'de Tekirdağ ilinin kırsal bir bölgesinde şebekeden bağımsız bir müstakil konut için rüzgâr-güneş yenilenebilir enerji kaynakları ile oluşturulan bir hibrit sistemin tekno-ekonomik analizlerini HOMER yazılımı ile gerçekleyerek, güneş enerjisi %61,8, rüzgar enerjisi %38,2 katkı oranı ile yük talebini karşılamıştır [12]. Rajbongshi ve arkadaşları Hindistan'ın Jhawani köyünde elektriksiz hanelerin ihtiyacını karşılamak üzere, HOMER yazılımı kullanarak hibrit sistem tasarlamışlardır. Çalışma sonucu olarak, güneş paneli/biyokütle/dizel tabanlı hibrit sistemin çok uzak köyler şebekeden bağımsız olarak uygulanabilir ve güvenli olduğunu göstermiştir. Ancak önerilen hibrit sistemlerin pratik zorluklarını anlamak için uygulamaya ihtiyaç duyulduğu da ayrıca vurgulanmıştır [13]. Shadid Jaman, yapmış olduğu çalışmada Bangladeş'e bağlı St. Martin adasının bulunduğu konuma göre en optimum hibrit sistem olan güneş paneli/yakıt hücresi önermiş ve sistemin tekno-ekonomik bileşenlerinin analizini ve boyutlandırılmasını HOMER yazılımı ile gerçekleştirerek simülasyonlarda önemli ölçüde CO2 emisyonunu azaldığını gözlemlemiş ve kırsal alanlar için hidrojen temelli hibrit sistemin şebeke bağlantılı mümkün olabileceğini belirtmiştir . Khan ve Iqbal , Newfoundland, Kanada'daki uygulamalar için hidrojeni enerji taşıyıcı olarak kullandıkları bir hibrit enerji sistemi için fizibilite çalışması hazırlamışlardır. Benzetim ve optimizasyon için HOMER programını kullanarak çeşitli yenilenebilir ve konvansiyonel enerji çözümleri ve farklı enerji depolama yöntemleri değerlendirilmiştir. Çalışma sonuçlarında bugünkü fiyatlarla rüzgar-dizel akümülatör sistemi uygun çözüm görünmüş ancak yakıt pillerinde % 15 civarında bir maliyet düşüşü yaşandığı takdirde rüzgar-yakıt pili sistemini daha cazip hale geleceği belirtilmiştir [14].

Bu çalışmada ise Muğla ilinin Köyceğiz ilçesine bağlı Zaferler köyünün 2020 yılındaki elektrik kullanım istatistiklerine bakılarak bölgenin enerji ihtiyacının karşılanması için şebeke bağlantısız Rüzgâr türbini/Güneş paneli/Jeneratör/Batarya hibrit enerji sistemi tasarımı ve tekno-ekonomik analizi HOMER yazılımı kullanılarak yapılmıştır. Sistem tasarımlarında kullanılan jeneratör bileşeninin 3 farklı türü olan biyogaz, yakıt hücresi ve dizel kullanılarak 3 farklı senaryo uygulanmıştır. Oluşturulan senaryolar arasında en uygun senaryo seçilmiştir. Seçilen senaryolarda, her bir bileşenin sisteme olan etkileri teknik ve ekonomik açıdan incelenmiş ve en optimum sonuç ortaya konulmuştur. Sonuç olarak, ekonomi ve çevre açısından en uygun sistem tasarımı ve optimizasyonu ortaya çıkarılmıştır.

2. Materyal ve Yöntem

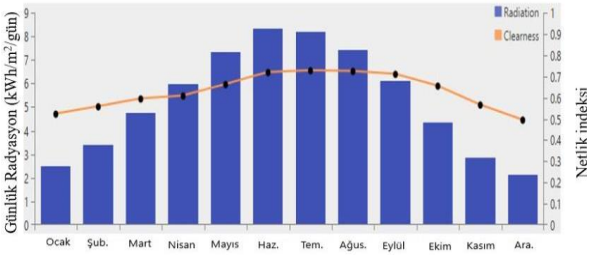
Bu çalışmada, Türkiye'nin Ege Bölgesinde bulunan Muğla ili Köyceğiz ilçesinde yer alan Zaferler köyünün $(36°57.5'N, 28°37.0'E)$ enerji ihtiyacının karşılanması adına hibrit sistem ekonomik analizi yapılmıştır. Zaferler köyünde 220 hane mevcuttur ve 590 kişi köyde yaşamaktadır. Çalışmanın uygulanacağı yörede yaşayan yerel halk geçinimini çiftçilikle sağlamaktadır. Yöre halkı tüm yıl boyunca erken saatlerde kalkmakta, öğlene kadar çalışmakta, birkaç saatlik dinlenmenin ardından tekrar çalışmaya koyulmaktadır. Akşam saatlerine kadar çiftçilikle uğraşan yerel halk gece erken saatler de yatmaktadır.

Yöre halkının 2021 yılı aylık enerji kullanımına ait veriler Aydem Elektrik Dağıtım A.Ş'den alınmıştır. Bu veriler kullanılarak köyün günlük ortalama enerji ihtiyacı 1087,26 kWh olup, günlük tepe değeri ise 162,15 kW olarak hesaplanmıştır. Köy halkının yıl içerisinde en fazla enerji kullanımı Eylül ayı, en düşük ise Mart ayımda gerçekleşmektedir. Yıl içerisinde yaz mevsiminde gerçekleşen tarla sulama enerji ihtiyacında artışa neden olmaktadır. Bu nedenle 5 aylık günlük enerji kullanımı diğer aylara göre fazladır.

Tablo 1: Zaferler Köyü aylık ortalama enerji kullanım değerleri

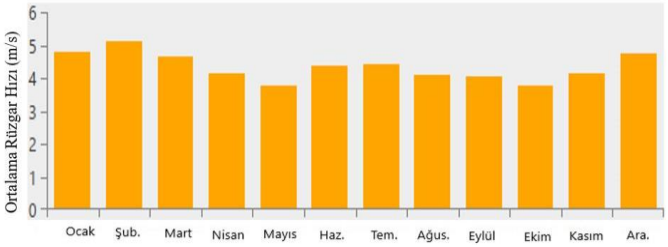
Tarih	Enerji Kullanımı(kWh)	Tarih	Enerji Kullanımı(kWh)
01/2021	784,6666667	07/2021	1261,333333
02/2021	932,8	08/2021	1386
03/2021	718,6666667	09/2021	1562
04/2021	982,6666667	10/2021	1232
05/2021	1173,333333	11/2021	902
06/2021	1144	12/2021	960,6666667

HOMER'da kullanılan meteorolojik veriler ''NASA "Yüzey Meteorolojisi ve Güneş Enerjisi" veri tabanından elde edilmiştir. Buradan elde edilen solar ışıma ve hava sıcaklığı değerleri için 22 yıllık (Temmuz 1983-Temmuz 2005) verilerin ortalaması, rüzgâr hızı değerleri için 10 yıllık (1983-1993) verilerin ortalaması alınmaktadır. Çalışmada ele alınan bölgenin solar enerji potansiyeli yaz aylarında fazlayken, kış aylarında azalmaktadır. Bölgenin yıllık ortalama günlük solar enerji potansiyeli Şekil 2'de de görüldüğü gibi 5,27 kWh/m2/gün olarak belirlenmiştir.



Şekil 1: Aylık ortalama solar radyasyon (kWh/m2/gün) profil

Bölgenin aynı şekilde rüzgâr hızı profili NASA'dan elde edilmiştir. Bu verilere göre bölgenin ortalama rüzgâr hızı 4,34 m/s'dir. Bölgenin rüzgâr hızı profili Şekil 3'te gösterilmiştir. Solar enerjinin yetersiz kaldığı durumlarda rüzgâr enerjisinin tamamlayıcı enerji kaynağı olacağı göz önündedir. Be nedenden dolayı hibrit sistemlerde en çok kullanılan enerji kaynakları olmuşlardır.



Şekil 2: Aylık ortalama rüzgâr hızı (m/s) profili

Şekil 3: Bölgenin aylara göre günlük sıcaklık değerleri (℃).

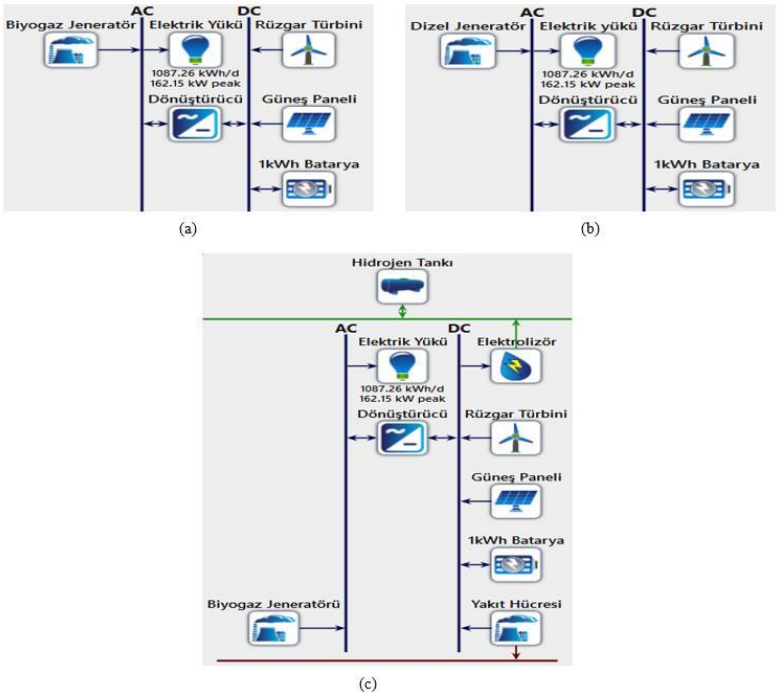
Çalışma yapılacak bölgenin yıllık sıcaklık ortalaması 8,58℃ olarak gözlemlenmiştir. Bölgenin aylara göre günlük ıcaklık değerleri Şekil 4'te gösterilmiştir.

2.1 HOMER (Hybrid Optimization of Multiple Electric Renewables) programı

Temel bir bilgisayar yazılımı olan HOMER programı, Birleşik Devletler Ulusal Yenilenebilir Enerji Laboratuvarı NREL) tarafından geliştirilmiş bir mikro güç optimizasyon rogramıdır. HOMER programı dünyanın en gelişmiş nodelleme yazılımıdır [15]. Bu program, sistemlerin fiziksel avranışlarını, işletme ve kurma maliyeti toplamı olan yaşam oyu maliyeti ve enerji birim maliyetini (COE) farklı ombinasyonlar için bulmaktadır. Ek olarak, sistem nodellenirken yapılan herhangi bir değişikliğin ve elirsizliğin anlaşılmasına yardımcı olur. HOMER yazılımı bir ükü besleyen rüzgar türbinlerinin, fotovoltaik panellerin, idroelektrik santrallerin, yakıt hücrelerinin, pistonlu motor eneratörlerinin, biokütle gücünün, akülerin ve hidrojen epolama sistemlerinin bulunduğu şebeke bağlantılı ve şebeke ağlantısız çalışan sistemleri modelleyebilir. Sistem tasarımı aparken birçok değişken vardır. HOMER yazılımı tasarımı orlaştıran bu gibi sorunların önüne geçmesi için tasarlanmış lup üç temel görevi gerçekleştirirler. Bunlar; hassaslık, imülasyon ve optimizasyon analizleridir. Hassaslık analizi ısmında HOMER programı, girişlerdeki değişikliklerin ve elirsizliklerin sistem üzerindeki etkilerini ölçmek için çok ayıda optimizasyon işlemi gerçekleştirir. Optimizasyon işlemi, sistemi tasarlayan kişinin belirlediği bütün değişkenlerin optimum değerlerini belirler. Hassaslık analizinde ise kişinin elinde olmayan değişkenlerin etkilerinin değerlendirilmesinde yardımcı olur. Simülasyon süreci kısmında ise HOMER programı yılın her saati için sistem konfigürasyonu performansını, sistemin teknik fizibilitesini ve yaşam süresi maliyetini belirleyebilmek için modeller.

2.2 Hibrit sistem bileşenleri

Hibrit sistemin temel bileşenleri içinde güneş panelleri, rüzgar türbinleri, dizel jeneratörler, yakıt hücresi, biyogaz jeneratörler, dönüştürücüler ve bataryalar mevcuttur. Dönüştürücüler, istenilen elektrik yüküne bağlı olarak alternatif akıma (AA) veya doğru akıma (DA) dönüştürmek için kullanılır. Tasarlanacak olan sistemde alternatif akıma ihtiyaç vardır. Bu nedenle sisteme dönüştürücü eklenmiştir. Sistem talep edilen yükü karşılamak için yakıt hücresi, rüzgâr türbini ve güneş panelleri kullanmaktadır. Fakat bu bileşenler enerji ihtiyacını karşılamakta yetersiz kalırsa dizel jeneratör ve biyogaz jeneratörü devreye girecektir. Bütün bu bileşenler sisteme dahil edilmiş ve HOMER programında simülasyonları elde edilmiştir. Ele alınan bölgenin yük ihtiyacını karşılamak için 3 farklı simülasyon yapılmıştır. Bunlar; güneş paneli-rüzgâr türbini-dizel jeneratör-batarya, güneş paneli-rüzgâr türbini-biyogaz jeneratör-batarya ve güneş paneli-rüzgâr türbini-yakıt hücresi-biyogaz jeneratörü-batarya şeklindedir. Bu farklı senaryolara ait HOMER şematik diyagramı Şekil 4'te verilmiştir.



Şekil 4: (a) Güneş paneli-rüzgâr türbini-biyogaz jeneratör-batarya (b) güneş paneli-rüzgâr türbini-dizel jeneratör-batarya (c) güneş paneli-rüzgâr türbini-yakıt hücresi-biyogaz jeneratörü-batarya hibrit sistem modeli

2.2.1.Rüzgâr türbini

Rüzgâr türbinleri, rüzgâr hızıyla pervanelerin önmesinden oluşan mekanik enerjiyi elektrik enerjisine önüştüren yenilenebilir bir enerji kaynağıdır. Genel olarak üzgâr türbinleri, kule jeneratör, hız dönüştürücüleri, elektrik e elektronik devreler ve pervanelerden oluşan ekipmanlar ütününden meydana gelir. Her enerji kaynağı türüne göre nerji maliyeti değişkenlik göstermektedir [16]. Birim enerji naliyetini düşürmek için en uygun yollardan birisi de rüzgâr ürbinleridir. Fakat rüzgâr türbinlerinin kurulum maliyeti üksek olduğu için düşük kapasiteli rüzgâr türbinleri lverişsizdir. Bu nedenle, çalışmada rüzgâr türbinleri 10 W'lık kullanılmıştır. Bu seçimle kurulum maliyetini üşürmesinin yanında aynı zamanda bölge için verimli bir nerji kaynağı olmuştur. Çalışmada, rüzgâr türbininin ömrü 20 e 25 yıl, türbin yüksekliği ise 24 ve 30 m farklı değerlerde eçilmiştir. Rüzgâr türbinin ekonomik özellikleri Tablo 2'de erilmiştir.

Tablo 2: Rüzgar türbininin ekonomik verileri.

Rüzgar türbini	Generic 10 kW
Kurulum maliyeti/kW	7.000 $
Kurulum maliyeti	50.000 $
Yenileme maliyeti	50.000 $
Yıllık operasyon ve bakım maliyeti	500 $
Ömrü	20 ve 25 yıl
Türbin yüksekliği	24 ve 30 m

2.2.2.Güneş paneli

Güneş enerji kaynağının en önemli parçalarından birisi üneş panelidir. Güneş panelindeki fotovoltaik hücreler, gelen ışınımı direkt olarak doğru akıma çevirirler. Güneş enerjisi ünyanın enerji ihtiyacını karşılayabilecek bir yenilenebilir nerji kaynağıdır [17]. Doğal olarak, Türkiye'nin de güneş nerji potansiyeli yüksek olmasından dolayı tercih dilebilecek en uygun yenilenebilir enerji sistemidir [18]. HOMER programının kullanılmasıyla tasarlanacak sistemde 'eimer SGM360M paneli tercih edilmiştir. Sistemde ullanılan panelin ömrü 30 yıl olarak alınmıştır. Kullanılacak anelin 1 kW için maliyeti 650$, yenileme maliyeti 650$, bir ıl için operasyon ve bakım maliyeti 20$ olarak kabul edilerek HOMER programında simülasyonlar gerçekleştirilmiştir. istemde kullanılacak olan panelin azaltma faktörü yani kir, ıcaklık, gölge ve eskime gibi nedenlerden dolayı çıkışında luşacak kayıpların dikkate alınmasını sağlayan değer %80 larak alınmıştır. Peimer SG360M panelinin teknik özellikleri şağıda Tablo 3'te verilmiştir.

Tablo 3: Peimer SGM360M güneş panelinin teknik özellikleri.

Panel tipi	Düz plaka
Değerlendirilen kapasite (kW)	1
Sıcaklık katsayısı	-0,352
Verimlilik (%)	18,5
Ağırlık (kg)	22,5
STC güç derecesi (W)	360

2.2.3.Batarya

Sistem modellemesinde bataryalar, yenilenebilir enerji kaynakları tarafından üretilen enerjiyi, üretilen enerji yük ihtiyacından fazla olduğu zaman depolayan ve aynı zamanda üretilen enerji sisteme yeterli olmadığında depoladığı enerjiyi sisteme aktaran elemanlardır. Bataryalar tarafından depolanan enerji ve sisteme aktarılan enerji DA gerilimindedir. Bataryaların maliyeti oldukça fazla olmasından dolayı batarya sayısının sisteme olan etkisi oldukça önemlidir [19]. Sistemde kullanılacak olan bataryaların 1 kWh başına maliyeti 450$, yenileme maliyeti 450$, yıllık operasyon ve bakım maliyeti ise 20$ olarak kabul edilerek simülasyona dahil edilmiştir. Bataryaların ömrü 15 ve 25 yıl olarak eklenmiştir. Aynı zamanda bataryanın deşarj derinliği %80 olarak simülasyonda kullanılmıştır. Sistemde kullanılan bataryanın teknik özellikleri Tablo 4' de verilmiştir.

Tablo 4: Generic 1 kWh Li-Ion batarya teknik özellikleri.

Nominal Voltaj (V)	6
Nominal kapasite (kWh)	1
Nominal kapasite (Ah)	167
Gidiş-dönüş verimliliği (%)	90
Maksimum şarj akımı (A)	167
Maksimum deşarj akımı (A)	500

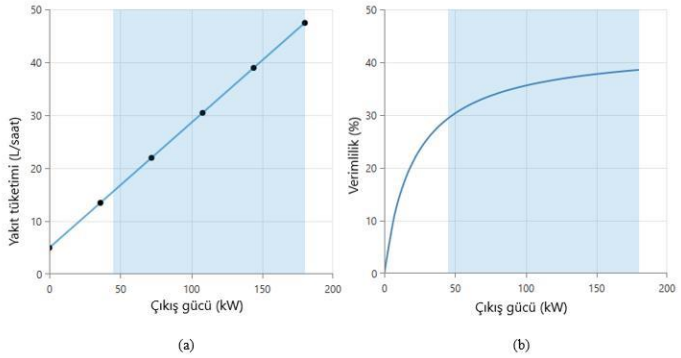
2.2.4.Dizel jeneratör

Çalışmada kullanılan jeneratörlerden biri dizel jeneratördür. HOMER programında tanımlı Autosize Genset jeneratör alınmış olup, sistemde 180 kW'lık dizel jeneratör kullanılmıştır. Dizel jeneratörün kW başına maliyeti 550$, yenileme maliyeti 550$ olarak alınmıştır. Aynı zamanda birim saatte operasyon ve bakım maliyeti 0,030$ olarak alınmıştır. Jeneratörün çalışma ömrü ise 15.000 saat olarak alınmıştır. Dizel jeneratör saatte 4,96 litre yakıt harcamaktadır. Dizel jeneratörde kullanılacak dizelin fiyatı 20 Aralık 2021 tarihinde Merkez Bankası'ndan alınan değerler kullanılarak (1$=17 TL) 0,68$/L değeri alınmıştır. Dizel yakıtın emisyon ve yakıt özellikleri aşağıda Tablo 5'te verilmiştir.

Tablo 5: Sistemde kullanılan dizel yakıtın emisyon ve özellikleri.

Emisyon değerleri		Yakıt özellikleri	
CO (g/L yakıt)	16,5	Alt ısıl değer (MJ/kg)	43,2
Partikül miktarı (g/L yakıt)	0,1	Yoğunluk (kg/m3)	820
NOx (g/L yakıt)	15,5	Karbon içeriği (%)	88
Yanmayan HC (g/L yakıt)	0,72	Sülfür içeriği (%)	0,4

Dizel jeneratörün tüketeceği yakıta göre elde edeceği çıkış ücü ve çıkış gücüne göre verimliliği Şekil 5'te verilmiştir.



Şekil 5: (a) dizel jeneratörün tüketeceği yakıta göre çıkış gücü (b) dizel jeneratörün çıkış gücüne göre verimliliği.

2.2.5.Biyogaz jeneratörü

Sistemde kullanılan diğer bir jeneratör tipi biyogaz eneratörüdür. Biyogaz jeneratörünün yakıtı kızılçam odun eletidir. Çalışma yapılacak bölgede kızılçam ormanları nevcuttur. Kızılçam ormanlarından hasat edilen dal odunları Muğla il merkezinde bulunan ticari pelet üretimi erçekleştiren Yücel Kereste İşletmesinde kızılçam peleti aline getirilmektedir. Üretilen kızılçam peletlerinin özkütlesi, ül bırakma yüzdesi ve uçucu madde miktarı sırasıyla 0,671 /cm3, %4,91 ve %73,9 olarak bulunmuştur. Aynı zamanda C, I, O, N miktarları ise sırasıyla %47.52, %5.15, %42.16, %0.26 olarak bulunmuştur. Kızılçam peletinin ısıl değeri ise 7-18 MJ/kg aralığındadır [20]. Simülasyona kızılçam eletinin ısıl değeri 17 MJ/kg olarak dahil edilmiştir. Biyogaz eneratörü için kızılçam peleti işletmeden istenilen miktarda lınacaktır. Kızılçam peletinin 1 ton için maliyeti 50$ olarak esaplanıp simülasyonlar gerçekleştirilmiştir. Tercih edilen iyogaz jeneratörünün 1kW için maliyeti 1000$, yenileme naliyeti 1000$ ve operasyon-bakım maliyeti 1 saat için 0,30$ larak hesaplanmıştır. Sistemde kullanılan biyogaz jeneratörü 0kW kapasitelidir. Jeneratörün kullanım ömrü ise 20.000 saat larak belirlenmiştir.

2.2.6.Yakıt Hücresi

Yakıt hücresi, kaynak olarak hidrojen kullanan ve ullandığı hidrojenden ısı ve elektrik üreten elektrokimyasal ir sistemdir. HOMER programında tanımlı olan Generic Fuel ell yakıt hücresi sisteme dahil edilmiştir. Kullanılan yakıt ücresinin kapasitesi 30 kW, kullanım ömrü 50.000 saat, urulum maliyeti 2000$/kW, operasyon ve bakım maliyeti ,010$/saat/kW, yenileme maliyeti 2000$/kW alınarak imülasyon işlemleri gerçekleştirilmiştir.

2.2.7.Elektrolizör

Elektroliz işlemini gerçekleştiren ve işlem sonunda idrojen oluşumunu sağlayan sistem bileşenine elektrolizör denir. Sistemde kullanılan Elektrolizör HOMER programında tanımlı Generic Elektrolizer olup, verimliliği %85, kullanım ömrü 15 yıl olarak alınmıştır. Elektrolizörün kapasitesi 100 kW olarak sisteme dahil edilmiştir. Elektrolizörün ekonomik özellikleri, kurulum maliyeti 1000$/kW, operasyon ve bakım maliyeti 0,10$/saat/kW, yenileme maliyeti 500$/kW olarak değerlendirilmiştir.

2.2.8.Hidrojen tankı

Elektrolizörde elde edilen hidrojeni sıkıştırılmış gaz halinde depolamaya yarayan en basit yöntem hidrojen tank kullanımıdır. Yapılan araştırmalara göre hidrojen tankının 1kg için 1000-1100 arasında maliyeti olduğu görülmüştür [21]. Hidrojen tankının ekonomik verileri , kurulum maliyeti 1000$/kg, operasyon ve bakım maliyeti 10$/saat/yıl, yenileme maliyeti 500$/kg ve ömrü 25 yıl seçilmiştir.

Ekonomik verileri verilen hidrojen tankı, HOMER programına tanımlı olan Generic Hidrojen Tank olarak sisteme dahil edilmiştir. Hidrojen tank boyutu 0 kg, 100 kg ve 200 kg olacak şekilde tercih edilip simülasyonlar gerçekleştirilmiştir. Sisteme 3 farklı tank boyutu tanımlanmasının sebebi, en uygun tank boyutunun bulunmasıdır.

2.2.9.Dönüştürücü

AA ve DA arasındaki enerji akışını sağlayabilmek için bir güç dönüştürücü gereklidir. HOMER'da dönüştürücü olarak tanımlanan eleman, hem DA-AA dönüştüren evirici hem de AA-DA dönüştüren doğrultucu gibi çalışmaktadır. Sisteme eklenen dönüştürücü, HOMER'da tanımlı Generic System Converter olarak simülasyonları yapılmıştır. Sisteme dahil edilen dönüştürücünün verimi %95, ömrü 15 yıl, kurulum ve yenileme maliyetleri 600$/kW, operasyon ve bakım maliyeti 40$/yıl olarak belirlenmiştir.

2.3.Ekonomik Analiz Girdileri

HOMER programı maliyet hesaplamalarında yıllık gerçek faiz oranını kullanmaktadır. Gerçek faiz oranı bulunurken

güncel faiz oranı ve enflasyon oranlarından yararlanılır. Yıllık gerçek faiz oranı aşağıda verilen denklem ile bulunur.

$$i = (i' - f)/(1 + f) \quad (1)$$

Burada; i yıllık gerçek faiz oranı, i' güncel faiz oranı ve f ise yıllık enflasyon oranı olarak tanımlanmaktadır. Bu çalışma kapsamında güncel faiz oranı %15.00, enflasyon oranı %19.80 olarak alınmıştır. Bu verilere göre gerçek faiz oranı -%4.01 olarak bulunmuştur.

2.4.Hibrit Sistem Modeli

Bu çalışmada Muğla'nın Köyceğiz ilçesine bağlı Zaferler Köyü'nde şebekeden bağımsız 3 farklı hibrit enerji sisteminin HOMER programı yardımıyla simülasyon ve analizleri yapılmıştır. Yapılan simülasyonda 3 senaryo içinde birim enerji maliyeti ve net şimdiki maliyet değerleri elde edilmiştir. Bu senaryolarda belirtilen optimum maliyet sonuçları Tablo 6'da verilmiştir.

Tablo 6: Hibrit enerji sistemlerinin optimizasyon sonuçları.

Senaryo	Rüzgâr Türbini	Güneş Paneli (kW)	Biyogaz Jeneratör (kW)	Dizel Jeneratör (kW)	Yakıt Hücresi Jeneratör (kW)	Dönüştürücü (kW)	Batarya	Optimizasyon Stratejisi	Net Bugünkü Maliyet ($)	Birim Enerji Maliyeti ($)
1	1	546	30	-	-	134	1.077	Döngü Şarj	3.26 M	0,185
2	1	465	-	180	-	119	978	Yük İzleme	2.93 M	0,166
3	1	320	30	-	30	113	431	Yük İzleme	2.69 M	0,152

Bütün senaryolarda güneş paneli/rüzgâr türbini/batarya/jeneratör kullanılmıştır. Oluşturulan hibrit sistemler arasındaki fark jeneratör tipleridir. Senaryo 1'de biyogaz jeneratörü, senaryo 2'de dizel jeneratörü, senaryo 3'te ise biyogaz jeneratörü ve yakıt hücresi kullanılarak hibrit sistemler oluşturulmuştur. Senaryo 3'te kullanılan biyogaz jeneratörü ve yakıt hücresi diğer seçeneklere göre uygun olup, aynı zamanda optimizasyon stratejisinde yük izleme seçilmesi sistemde ihtiyaç duyulan batarya sayısını da azaltmaktadır. Bu sebeplerden dolayı sistemin net bugünkü maliyeti ve birim enerji maliyeti diğer sistemlere göre oldukça düşük çıkmaktadır. Hibrit sistemlerin maliyetleri karşılaştırıldığında, net bugünkü maliyeti 2.69 milyon $ ve birim enerji maliyeti 0,152 $ olan senaryo 3 en uygun model olmaktadır. Sistemlerde optimizasyon yapılırken batarya ve rüzgâr panelinin yaşam ömrü, rüzgâr türbininin kule yüksekliği değiştirilmiştir. Seçilen senaryonun hibrit sistem şematik görünümü Şekil 4 c'de verilmiş olup köy halkının elektrik ihtiyacını karşılayacak şekilde tasarlanmıştır.

3. Bulgular ve Tartışma

Sistem senaryolarının optimizasyon sonuçlarının yanında aynı zamanda bileşenlerinin de optimizasyon sonuçları incelenip değerlendirilmesi gerekir. Yapılan simülasyonlar ışığında, belirlenen parametreler dahilinde seçilen bölgeye en uygun sistem senaryo 3 olmuştur. Aynı zamanda senaryo 3'ün emisyon değerlerinin de incelenmesi gerekir. Senaryo 3'ün emisyon değerleri Tablo 7'de verilmiştir.

Tablo 7: Seçilen hibrit sistemin emisyon değerleri.

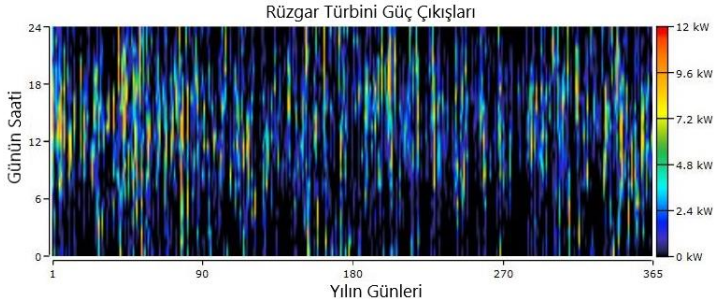
	Değer kg/yıl
Karbon Dioksit	53.163
Karbon Monoksit	5,52
Yanmamış Hidrokarbon	0
Partikül Madde	0
Sülfür Dioksit	0
Nitröz Oksit	0,577

Seçilen sistemin emisyon değerleri diğer sistemlere göre daha azdır. Her ne kadar karbon dioksit miktarı fazla çıksa da aslında biyogaz jeneratöründe kullanılan yakıt kızılçam peleti olduğundan karbon ayak izi neredeyse sıfırdır. Tüm bunlar göz önüne alındığında sistemin yenilenebilir enerji faktörü %67,1 olarak bulunmuştur. Yenilenebilir enerji faktörünün bu şekilde çıkması yine kullanılan kızılçam peletinin etkisidir. Kullanılan

ızılçam peletinin daha temiz bir yakıt olması sağlanabilir. Bu ayede yenilenebilir enerji faktörü arttırılabilir. Sürdürülebilir ir dünya için bunun gibi iyileştirmeler tüm sistemler için aima düşünülmeli ve daha temiz enerji elde etmek için ğraşılmalıdır.

Tasarlanan 3 sistem arasında senaryo 3'ün yılın farklı gün ve saatlerinde rüzgâr türbininden elde edilen çıkış gücü Şekil 6'da verilmektedir. Seçilen bölgede, elde edilen rüzgâr türbininin çıkış gücü o anki rüzgar hızı ve sistemin ihtiyacı olan üretim miktarına bağlıdır.

3.1.Rüzgâr Türbininin Sisteme Etkisi

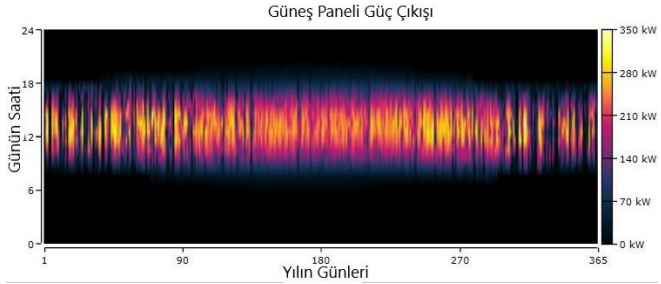


Şekil 6: Seçilen sistemin yılın farklı zamanlarında rüzgâr türbini güç çıkış değerleri.

Şekilde gözüktüğü üzere sisteme entegre edilen rüzgar ürbini, yılda 6.489 saat çalışarak 10.765 kWh/yıl elektrik nerjisi üretmektedir. Üretilen elektrik enerjisinin birim naliyeti 0,170 $/kWh, rüzgar türbinin ortalama güç çıkışı 1,23 W, kapasite faktörü %12,3 olmaktadır. Rüzgar türbini örüldüğü üzere yılın ilk aylarında daha fazla kullanılmıştır. Bunun nedeni kış aylarında güneş panellerinin sağlayamadığı eya kullanımda olmadığı durumlarda rüzgar türbinin ihtiyaç lan elektrik enerjisini üretmesinden kaynaklıdır.

3.2.Güneş Panellerinin Sisteme olan Etkisi

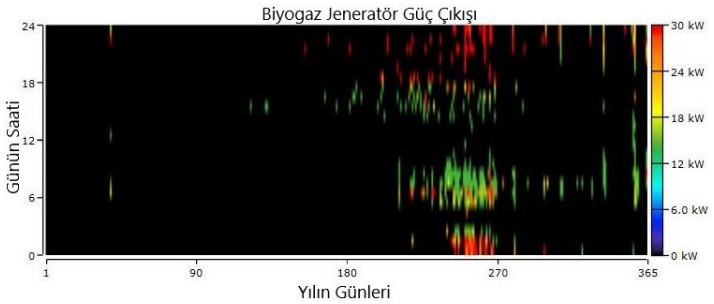
Şekil 7'de senaryo 3'te kullanılan güneş panellerinin yılın arklı gün ve saatlerinde elde ettiği çıkış gücü değerleri gözükmektedir. Seçilen bölgenin konumu itibariyle, güneş panellerinin elde ettiği güç miktarı sistemin ana enerji kaynağıdır. Aynı zamanda panellerin kapasitesi yüksek olmasından dolayı kısa sürede daha fazla enerji üretmektedir. Sistem için seçilen güneş panelleri yılda 4.387 saat çalışarak toplamada 558.809 kWh/yıl elektrik enerjisi üretmektedir. Panellerin günlük ortalama güç çıkışı 1.531 kWh ve kapasite faktörü %19,9 olmaktadır. Sisteme entegre edilen güneş panellerinin maksimum çıkış gücü ise 334 kW olarak ortaya çıkmıştır. Panellerin ürettiği elektrik enerjisinin birim maliyeti 0,0160 $/kWh olmaktadır. Şekil 7'de de görüldüğü üzere günün belli saatleri güç çıkışı elde eden panellerin tek başına yeterli olmadığı gözükmektedir.



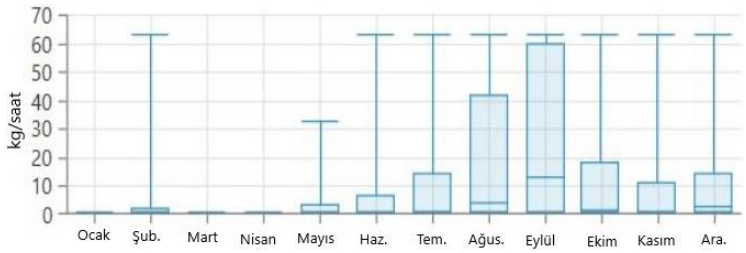
Şekil 7: Yılın farklı zamanların sisteme entegre edilen güneş panellerinin çıkış gücü değerleri.

3.3.Biyogaz Jeneratörünün Sisteme olan Etkisi

Şekil 8'de görüldüğü üzere biyogaz jeneratörü en yüksek üç çıkışını gece saatlerinde elde etmiştir. Aynı zamanda aptığı üretimin büyük kısmını yaz aylarında kullanıcıya edarik etmiştir. Bunun sebebi yöre halkının yaz aylarında aptığı tarla sulama olayıdır. Sulamadan kaynaklı elektrik ullanımının artması biyogaz jeneratörünün güç çıkışında büyük role sahiptir. Sisteme entegre edilen biyogaz jeneratörü yılda 201 kez çalıştırılmış olup, yılda 399 saat çalıştırılmış ve toplamda 8.123 kWh elektrik enerjisi üretmiştir. Jeneratörün spesifik yakıt tüketimi 2,15 kg/kWh olup yıllık yakıt tüketimi ise 23.6 ton olarak ortaya çıkmıştır. Kullanılan yakıt olan kızılçam peletinin aylık kullanım miktarları Şekil 8'de verilmiştir.



Şekil 8: Sisteme entegre edilen biyogaz jeneratörünün çıkış gücü değerleri.

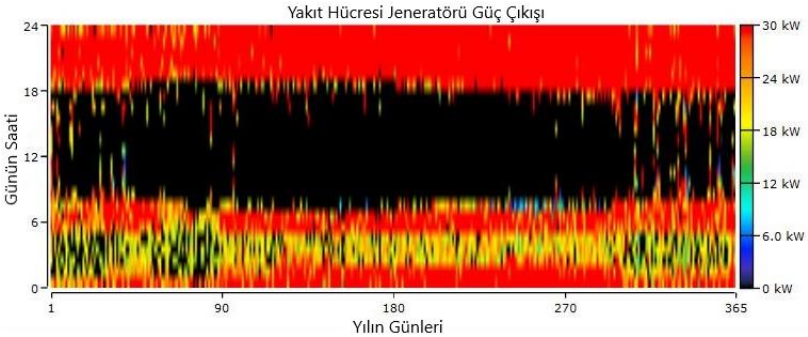


Şekil 9: Biyogaz jeneratörünün kullandığı kızılçam peletinin aylık tüketimi.

Şekil 9'da görüldüğü üzere yaz aylarında jeneratörün ullandığı yakıt miktarı artmış ve en yüksek eylül ayında yakıt ullanılmıştır. Kızılçam peletinin ortalama günlük jeneratör eslemesi 0,0646 ton ve ortalama saatlik jeneratör beslemesi ,00269 ton olarak ortaya çıkmıştır.

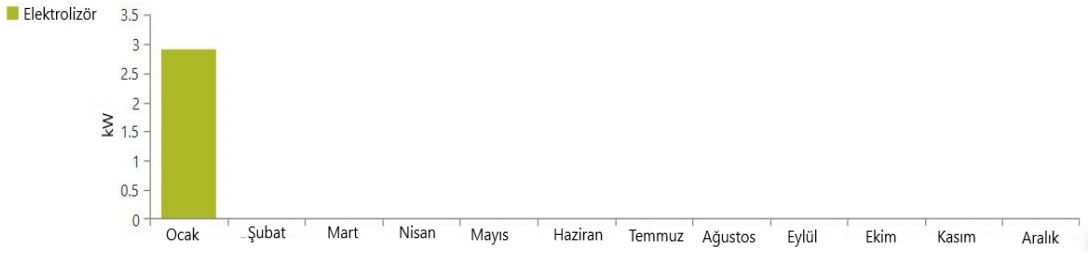
3.4.Yakıt Hücresi Jeneratörünün Sisteme olan Etkisi

Yakıt hücresi jeneratörü yakıt olarak hidrojen kullanılır. Kullanılacak olan hidrojen, elektrolizör vasıtasıyla suyun elektrolizi ile meydana gelir. Suyun elektrolizi sırasında kullanılacak olan elektrik enerjisi sistemdeki fazla elektrik enerjisinden sağlanır. Elde edilen hidrojeni depo etmek için hidrojen tankı kullanılmıştır. Yani sisteme entegre edilen bu yenilenebilir enerji sistemi bir nevi enerji depolama ve enerji üretme sistemidir. Çalışmada kullanılan yakıt hücresi jeneratörünün çıkış gücü değerleri Şekil 10'da verilmiştir.



Şekil 10: Sisteme entegre edilen yakıt hücresi jeneratörünün çıkış gücü değerleri.

Şekil 10'da görüldüğü üzere yakıt hücresi jeneratörünün ıkış gücü değerleri yılın farklı gün ve saatlerine göre erilmiştir. Sisteme entegre edilen bu jeneratör yılda 4.947 aat çalışarak toplamda 130.361 kWh/yıl elektrik enerjisi retmiştir. Aynı zamanda yakıt hücresi jeneratörünün toplam akıt tüketimi 27.376 m3 hidrojen ve spesifik yakıt tüketimi ,210 m3/kWh olarak bulunmuştur. Sisteme dahil edilen bu eneratör Şekil 12'de görüldüğü üzere, günün öğle saatlerinde fazla çalışmamıştır. Genellikle akşam saatlerinden başlayıp öğle saatlerine kadar çalışmıştır. Bunun sebebi ise güneş panellerinin öğle saatlerinde ihtiyaç olan enerjiyi karşılamaya başlamasıdır. Bütün bunların yanında hidrojen üretmek için gereken elektrolizörün performansını da incelemek yerinde olur. Elektrolizörün aylık harcadığı enerji miktarı Şekil 11'de verilmiştir.



Şekil 11: Sisteme entegre edilen elektrolizörün aylık enerji kullanımı.

Şekil 11'de verilen grafiğe göre elektrolizör sadece Ocak yında üretim yapmıştır. Sisteme entegre edilen elektrolizör ıllık 90 kg üretim yapmıştır. Sistem simülasyonu yapılırken 00 kW kapasiteli elektrolizör seçilmiştir. Burada 3-5 kW apasiteli elektrolizör bu sisteme yeterli olacaktır. Sistem ileşenlerinden biri olan hidrojen tankı üretilen hidrojeni erektiği zaman kullanılması için depo etmektedir. Daha önceden hidrojen tankı için girilmiş olan tank boyutları 100 kg ve 200 kg olacak şekilde seçenek belirlenmişti. Şekil 11'de görüldüğü üzere elektrolizör yalnızca ocak ayında üretim yapmış olup, bu üretim 90 kilogramdır ve hidrojen tankında depolanmaktadır. Bütün bunlar göz önüne alındığında 100 kg boyutundaki hidrojen tankı yeterli olacaktır. Şekil 12'de hidrojen tankının aylık depoladığı hidrojen miktarı verilmiştir.

Şekil 12: Sisteme entegre edilen hidrojen tankının aylık depo ettiği hidrojen miktarı.

istem dahilinde mevcut 100 kg hidrojen tankı 3.333 kWh nerji depo etme özelliğine sahiptir.

3.5.Batarya Kullanımının Sisteme Etkileri

Sistem içerisinde batarya kullanımı sistem maliyetini nemli derecede arttırmıştır. Tablo 8'de 3 senaryo için stemlerin net bugünkü değerleri ve birim enerji fiyatları erdikleri batarya sayılarıyla birlikte verilmiştir.

Görüldüğü üzere batarya sayısının artması sistemin aliyetini önemli oranda arttırmıştır. Batarya sayısının 1.077 lduğu senaryo 1 de enerji maliyeti 0,185 $ iken, 431 batarya ayısına sahip senaryo 3 ise 0,152 $ birim enerji maliyetine ahiptir. Bu bilgilere dayanarak senaryo 3'teki batarya sayısının miktarı ile birlikte birim enerji maliyeti ve net bugünkü değer de düşmüştür. Sisteme entegre edilmiş bataryaların yıl içerisindeki şarj durumunu saatlik olarak Şekil 13'te gösterilmiştir.

Tablo 8: Farklı senaryoların batarya sayısı, NBD ve birim enerji maliyeti.

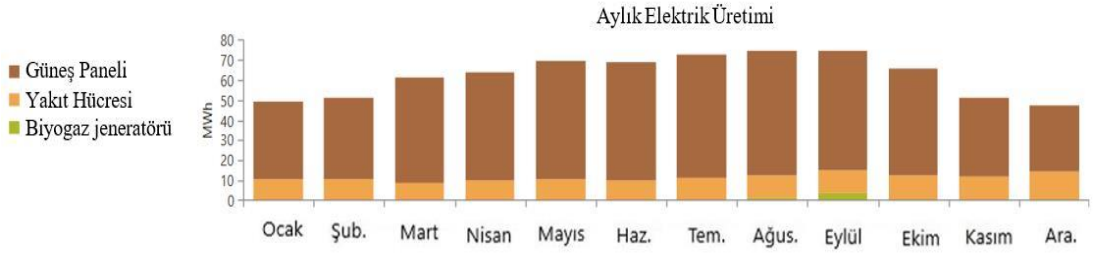
Senaryolar	Batarya Sayısı	Net Bugünkü Değer ($)	Birim Enerji Maliyeti ($)
1	1.077	3.26 M	0,185
2	978	2.93 M	0,166
3	431	2.69 M	0,152

Şekil 13: Bataryaların yıl içerisindeki şarj durumu.

örüldüğü üzere bataryalar günün gündüz vaktinde şarj lmakta ve akşam vakitlerinde kullanılmaktadır. Bataryalara ıllık 76.732 kWh enerji girdisi olup, yıllık 7.690 kWh kayıp ardır.

4.Sistem Bileşenlerinin Üretim Dağılımı

Sistem bileşenlerinin toplam üretime ne kadar payı olduğu eğerlendirme için önemlidir. Şekil 14'te sistem bileşenlerinin ylık enerji üretimi verilmiştir.



Şekil 14: Sistem bileşenlerinin aylık enerji üretim dağılımı.

Görüldüğü üzere elektrik üretiminin büyük bir kısmını üneş paneli ve yakıt hücresi sağlamıştır. Tablo 9'da toplam enerji üretiminin bileşenler tarafından ne kadarlık bir paya sahip olduğu verilmiştir.

Tablo 9: Sistem bileşenlerinin enerji üretimindeki payları

Üretim	Güneş Paneli	Rüzgar Türbini	Biyogaz Jeneratörü	Yakıt Hücresi Jeneratörü	Toplam
kWh/yıl	558.809	10.765	8.123	130.361	708.057
%	78,9	1,52	1,15	18,4	100

Tablo 9'da verilen değerlere göre güneş paneli ardından akıt hücresi jeneratörü üretimde yüksek enerji üretim payına ahiptir. Biyogaz jeneratörü ve rüzgar türbini enerji üretim ayında çok küçük bir paya sahip olduğu görülmektedir. Bütün u sistemin fazla elektrik üretimi yılda 279.354 kWh, arşılanmamış elektrik yükü yılda 132 kWh ve kapasite ıkıntısı yılda 386 kWh olarak bulunmuştur. Seçilen sistemin enilenebilir enerji faktörü %67,1 olarak elde edilmiştir.

4. Sonuç

Bu çalışmada, Muğla ilinin Köyceğiz ilçesine bağlı olan Zaferler köyünün elektrik ihtiyacının karşılaması amacıyla ebeke bağlantısız bir hibrit enerji sistem analizleri yapılmıştır. oplanan veriler HOMER programında kullanılarak farklı istemler karşılaştırılmış ve simülasyonları yapılmıştır. OMER programında 3 farklı senaryonun simülasyonu erçekleştirilmiştir. Bütün senaryolarda rüzgâr türbini, güneş aneli ve bataryalar kullanılırken, senaryo 1'de biyogaz eneratörü, senaryo 2'ye dizel jeneratörü ve senaryo 3'e yakıt ücresi ve biyogaz jeneratörü eklenmiştir. Bütün senaryoların imülasyon sonuçlarına göre, birim enerji maliyeti ve net ugünkü değerleri göz önüne alındığında senaryo 3 en uygun ibrit sistem olarak karşımıza çıkmıştır. Senaryo 3'ün birim nerji maliyeti, net bugünkü değeri, kurulum maliyeti ve perasyon ve bakım maliyeti sırasıyla 0,152 $, 2.69 milyon $, 09.980 $, 42.222 $ olarak hesaplanmıştır. Her hanenin aklaşık 12.300 $ yatırım yapmasıyla 25 yıllık hem elektrik em de tarla sulama ihtiyacını karşılayabileceği üşünülmektedir. Senaryo 3'ün bileşenlerinden olan güneş aneli % 78.9, rüzgar türbini % 1.52, biyogaz jeneratörü % .15 ve yakıt hücresi jeneratörü % 18.4 olacak şekilde ihtiyaç lan toplam enerjiyi karşılamıştır. Her ne kadar biyogaz jeneratörü ve rüzgâr türbininin toplam enerji ihtiyacını karşılamada payları düşük olsa da ana enerji üretim kaynağı olan güneş panelinin enerji üretmediği zamanlarda ihtiyaç olan yükü karşılamışlardır. Bunun yanında biyogaz jeneratörünün yakıt olarak kullandığı kızılçam peleti üretimi için çevrede bolca bulunan kızılçam artıkları bu sistem bileşeninin avantajlarından birisidir. Aynı zamanda sistem bileşeninde bulunan yakıt hücresi jeneratörü fazla üretilen enerjiyi hidrojen olarak depo ettiği ve ihtiyaç zamanında kullandığı için sistemde batarya sayısı diğer senaryolara göre oldukça düşüktür. Bu durumda senaryo 3'ün maliyetini düşürmüş ve sistemin daha avantajlı bir sistem olmasını sağlamıştır. Bu değerlendirmeler kapsamında, seçilen bölgeye güneş paneli/rüzgar türbini/biyogaz jeneratörü/yakıt hücresi jeneratörü/batarya sistemi simülasyonları gerçekleştirilen sistemler arasında kurulumu yapılabilecek en uygun sistemdir. Fakat hibrit sistem konfigürasyonlarının pratik zorluklarını anlamak için uygulamaya ihtiyaç vardır.

Teknolojinin gelişmesiyle birlikte yenilenebilir enerji bileşenlerinin düşmesi ve fosil yakıtların tükenmesinden dolayı fiyat artışı yakın bir gelecekte hibrit enerji sistemlerinin ürettiği enerjinin birim maliyetinin geleneksel yolla üretilen enerjinin birim maliyetinden daha uygun olacağını ortaya koymaktadır.

5. Kaynakça

[1] S. Jaman, "Techno-Economic Analysis Of A Solar Pv-Fuel Cell Based Hybrid Energy System For St. Martin Island Using Homer," Seu J. Sci. Eng., Vol. 12, No. 1, 2018.

[2] A. Maleki, M.G. Khajeh, M. Ameri, Optimal sizing of a grid independent hybrid renewable energy system

incorporating resource uncertainty, and load uncertainty, Int. J. Electr. Power Energy Syst. 83 (2016) 514–524.
3] G. Tina, S. Gagliano, S. Raiti, Hybrid solar/wind power system probabilistic modelling for long-term performance assessment, Sol. Energy. 80 (2006) 578–588.
4] A. Tabak, "Konya İlinde Bir Fabrikanın Enerji Talebinin Karşılanması Için Hibrit Enerji Üretim Sisteminin Analiz Ve Tasarımı Analysis And Design Of A Hybrid Energy Production System To Meet The Energy Demand Of A Plant In Konya," 2021.
5] B. E. Türkay And A. Y. Telli, "Economic Analysis Of Standalone And Grid Connected Hybrid Energy Systems," Renew. Energy, Vol. 36, No. 7, Pp. 1931–1943, 2011.
6] D. Ribó-Pérez, Á. Herraiz-Cañete, D. Alfonso-Solar, C. Vargas-Salgado, And T. Gómez-Navarro, "Modelling Biomass Gasifiers In Hybrid Renewable Energy Microgrids; A Complete Procedure For Enabling Gasifiers Simulation In Homer," Renew. Energy, Vol. 174, Pp. 501–512, 2021.
7] M. S. Hossain, A. G. Alharbi, K. Z. Islam, And M. R. Islam, "Techno-Economic Analysis Of The Hybrid Solar Pv/H/Fuel Cell Based Supply Scheme For Green Mobile Communication," Sustain., Vol. 13, No. 22, Pp. 1–29, 2021.
8] Ş. Emeç and G. Akkaya, "Techno-economic analysis of a university's electrical energy consumption with hybrid systems," J. Inf. Optim. Sci., vol. 42, no. 2, pp. 417–430, 2021.
9] A. Singh and P. Baredar, "Techno-economic assessment of a solar PV, fuel cell, and biomass gasifier hybrid energy system," Energy Reports, vol. 2, pp. 254–260, 2016.
10] B. K. Das, R. Hassan, M. S. Islam, and M. Rezaei, "Influence of energy management strategies and storage devices on the techno-enviro-economic optimization of hybrid energy systems: A case study in Western Australia," J. Energy Storage, vol. 51, no. March, p. 104239, 2022.
11] S. Madhura and V. Boddapati, "Optimal sizing and assessment of a hybrid energy based AC microgrid," Mater. Today Proc., vol. 49, pp. 326–332, 2021.
12] A. E. Akan, "Techno-Economic Analysis Of An Off-Grid Hybrid Energy System With Homer Pro," Icontech Int. J., Vol. 5, No. 3, Pp. 56–61, 2021.
13] R. Rajbongshi, D. Borgohain, And S. Mahapatra, "Optimization Of Pv-Biomass-Diesel And Grid Base Hybrid Energy Systems For Rural Electrification By Using Homer," Energy, Vol. 126, Pp. 461–474, 2017.
14] P. Nema and S. Dutta, "Feasibility Study of 1 MW Standalone Hybrid Energy System: For Technical Institutes," Low Carbon Econ., vol. 03, no. 03, pp. 63–68, 2012.
15] S. Dursun, "Optimal Wind / Pv / Biomass Hybrid Power System For Forest Mugla Journal Of Science And Technology Optimal Wind / Pv / Biomass Hybrid Power System For Forest," Vol. 2, No. June, Pp. 43–47, 2016.
[16] M. A. Vaziri Rad, M. Panahi Vaghar, A. Kouravand, E. Bellos, and A. Kasaeian, "Techno-economic evaluation of stand-alone energy supply to a health clinic considering pandemic diseases (COVID-19) challenge," Sustain. Energy Technol. Assessments, vol. 51, no. August 2021, p. 101909, 2022.
[17] S. Basu, A. John, Akshay, And A. Kumar, "Design And Feasibility Analysis Of Hydrogen Based Hybrid Energy System: A Case Study," Int. J. Hydrogen Energy, Vol. 46, No. 70, Pp. 34574–34586, 2021.
[18] A. F. Güven And M. Mete, "Balikesiili Erdeilçesiiçin Bağimsihibritenerjsisteminfizibilite Çalişmasi Ve Ekonomianalizi," Konya J. Eng. Sci., Vol. 8055, Pp. 1063–1076, 2021.
[19] M. F. Roslan, M. A. Hannan, P. Jern Ker, R. A. Begum, T. M. Indra Mahlia, and Z. Y. Dong, "Scheduling controller for microgrids energy management system using optimization algorithm in achieving cost saving and emission reduction," Appl. Energy, vol. 292, no. October 2020, 2021.
[20] T. Türkoğlu And C. Gökoğlu, "Kızılçam Ormanları Hasat Artıklarından Yapılan Odun Peletinin Yakıt Özelliklerinin Belirlenmesi," Süleyman Demirel Üniversitesi Fen Bilim. Enstitüsü Derg., Vol. 21, No. 1, P. 58, 2016.
[21] F. Dawood, G. M. Shafiullah, And M. Anda, "Standalone Microgrid With 100% Renewable Energy: A Case Study With Hybrid Solar Pv-Battery-Hydrogen," Sustain., Vol. 12, No. 5, 2020.

Özgeçmişler

Aykut Fatih Güven, Lisans ve Yüksek Lisans eğitimlerini sırasıyla 2000 ve 2004 yıllarında Karadeniz Teknik Üniversitesi Elektrik Mühendisliği bölümünde tamamlamıştır. 2018 yılında başladığı doktora eğitimine Kocaeli Üniversitesi Elektrik Mühendisliği Anabilim Dalında devam etmektedir. Halen Yalova Üniversitesi Enerji Sistemleri Mühendisliği bölümünde Öğretim Görevlisi olarak çalışmaktadır. Araştırma alanları; yenilenebilir enerji, hibrit enerji sistemleri yönetimi, güç sistemi analizi ve meta-sezgisel optimizasyon algoritmaları.

Cüneyt Hatipoğlu, Lisans eğitimini, 2022 yılında Yalova Üniversitesi Enerji Sistemleri Mühendisliği bölümünde tamamlamıştır. Araştırma Alanları; hibrit enerji sistemleri, biyokütle ve yenilenebilir enerji kaynakları.

Makale Türü: Araştırma Makalesi
Makale Gönderim Tarihi:07.03.2022
Makale Kabul Tarihi:29.05.2022

Güç Sisteminde Oluşan Harmonik ile Ara Harmoniklerin Modellenmesi ve Simülasyonu

Modeling and Simulation of Harmonic and Interharmonics in the Power System

Sabir Rüstemli[1], Behçet Kocaman[1], Sinan Tekev[2]

[1]Elektrik-Elektronik Mühendisliği Bölümü Mühendislik Mimarlık Fakültesi
Bitlis Eren Üniversitesi, Bitlis, Türkiye
srustemli@beu.edu.tr, bkocaman@beu.edu.tr

[2]Elektrik-Elektronik Mühendisliği Anabilim Dalı Lisansüstü Eğitim Enstitüsü
Bitlis Eren Üniversitesi, Bitlis, Türkiye
sinantekev@hotmail.com

Öz

lektrik tesislerinin güvenli olarak çalışmasında, tesisin arımında ve işletiminde güç kalitesini etkileyen rametrelerin dikkate alınması gerekir. Bu parametrelerden ri de nonlineer özellikli elemanların oluşturduğu harmonik ara harmoniklerdir. Güç sisteminde oluşan harmonik ve a harmonikler, güç kalitesini önemli ölçüde etkilemektedir. u araştırmada, güç sisteminde oluşan harmonik ve ara rmoniklerin etkilerinin tespiti için MATLAB/Simulink ogramı yardımıyla farklı nonlineer yüklere karşı seri aktif iç filtresi modellenip simülasyon yapılmıştır. Nonlineer kleri olan güç sistemindeki harmoniğin filtre yapılmadan ce ve filtre yapıldıktan sonra elde edilen sonuçlar analiz lmiştir. Yapılan analizde, seri aktif güç filtresi llanıldıktan sonra 3. harmonik için % 5.11 olan harmonik ozulmanın % 0.09'a düştüğü ve 3.5 ara harmonik için % 16 olan harmonik bozulmanın % 0.17'e düştüğü tespit lmiştir.

nahtar Kelimeler: Güç sistemi, harmonik, ara rmonik, seri aktif güç filtresi.

bstract

order for electrical installations to operate safely, some ctors must be taken into account during the design and eration of the plant. One of these factors is the harmonics d interharmonics created by the elements with nonlinear aracteristics, which are among the parameters that termine the power quality. The harmonics and terharmonics that occur in the power system significantly fect the power quality. In this study, a serial active power ter was modeled and simulated against different nonlinear ads with the help of MATLAB/Simulink program to termine the effects of harmonics and interharmonics in the wer system. The results obtained before and after filtering e harmonic in the power system with nonlinear loads were alyzed. In the analysis, it was determined that after using e serial active power filter, the harmonic distortion, which was 5.11% for the 3rd harmonic, decreased to 0.09%, and the harmonic distortion, which was 4.16% for the 3.5 interharmonic, decreased to 0.17%.

Keywords: Power system, harmonic, inter harmonics.

1. Giriş

Elektrik enerjisine olan talep, insan ve sanayileşmeye bağlı olarak her geçen gün artmaktadır. Bu artan talep ile birlikte, enerjinin daha kaliteli ve daha güvenilir olmasına gereksinim duyulmaktadır. Elektrik enerjisi, üretildiği santralden tüketim noktasına ulaştırılıncaya kadar ki aşamalarda akım ve gerilimin saf sinüs dalgası şeklinde olması gerekmektedir. Ancak bunun sağlanması daima mümkün olmayabilir. Çünkü teknolojinin gelişmesiyle birlikte artan elektronik cihazlar ve lineer olmayan (nonlineer) elektrik devre elemanları güç sistemlerinde harmonik oluşturmaktadır. Harmonik, saf sinüs dalgası (sinüsoidal) şeklini bozan, istenmeyen, frekansı değişmiş dalga şekilleridir [1].

Hem düşük hem de yüksek frekanslarda harmonikler ve ara harmonikler; gerilimde istenmeyen dalgalanma, ekipmanın aşırı ısınması, şebekede artan kayıplar, iletişim sistemlerinde parazit, kontrol sistemlerinde ve dijital sayaçlarda hatalar gibi etkiler oluşturmaktadır [2-4]. Endüstride kullanılan transformatör, ark fırını, dönüştürücü, güç elektroniği devre elemanları gibi harmonik oluşturan cihazlar, güç kalitesinin bozulmasına neden olmaktadır. Son yıllarda tüm dünyada güç kalitesini iyileştirmek ve harmonik ve ara harmoniklerin elimine edilmesi için artan bir çaba görülmektedir. Bunun için enerjinin sürekliliğine, güç faktörü değerinin 1'e yakın olmasına, faz gerilimlerinin dengeli olmasına, akım ve gerilimde oluşan toplam harmonik değerlerinin standartlarda istenilen değerlerde kalmasına yönelik çalışmalar yapılmaktadır. Ara harmonikler, enerji sisteminin temel frekansının tamsayı katı olmayan

frekanslardan oluşurlar. Ara harmoniğin, gerilim dalgalanması temel frekansın % 0.2'nin ve 200 Hz'den büyük bileşenler için ise 0.3%'ün altında olması istenmektedir. Ara harmonikler, harmoniklerden kaynaklı birçok problemle birlikte ışık titremesi gibi ek problemler de oluşturmaktadır [5].

Ara harmonikler temel olarak, tesislerin hızlı bir akım değişiminin neden olduğu genlik ve/veya faz değişikliğinden dolayı sistem frekansı ve harmoniklerinin yan bantları etrafında yer almasından ve yarı iletken cihazlar kullanan statik dönüştürücülerde asenkron anahtarlama frekansının güç sistemi ile senkronize olmaması durumundan olmak üzere iki şekilde oluşmaktadır.

Harmonik ve ara harmonik değerlerini azaltmak/elimine etmek için kullanılan en etkili yöntemlerden biri de harmonik filtre kullanılmasıdır [6,7].

Güç sisteminde oluşan harmonikler; teknik ve ekonomik sorun olan ek kayıp ve gerilim düşümlerine, rezonans olaylarına, güç faktörü değerinin değişmesine neden olmaktadır. Alternatörlerin alternatif akım (AC) üretimi sırasında alternatörler için alınan önlemler ile, elde edilen dalganın mümkün mertebede sinüzoidal dalgaya yakın elde edilmesi sağlanır. Fakat, aynı şebekede bulunan lineer olmayan alıcılar (yükler) lineer olan yüklerle etki edebilmesi mümkündür [8-10].

Sanayi tesislerinde gerilim dalgasında oluşan bozulma (THD_v) değerinin % 5'den fazla olmaması gerekir. Bu değerden fazla çıkması durumunda, pasif filtre yardımıyla bu değer indirilebilir. Bunun yanında havalimanı ve hastane gibi yerlerde gerilim değerinde oluşan bozulma değerinin %3'ün altında olması gerekir. THDv değerinin istenilen değerlerin üzerinde çıkması halinde, çeşitli yapı ve şekilde filtre tasarımları mümkündür. Genel manada aktif ve pasif olmak üzere iki tür filtreleme tekniği kullanılmaktadır. Yapılan filtreyi oluşturan bileşenlerin direnç (R), bobin (L) ve kondansatör (C) gibi pasif devre elemanlarından oluşması durumundaki fitrelere pasif filtre, filtrelerin kontrollü akım veya gerilim değerine sahip olduğu filtrelere ise aktif güç filtresi denir. Her zaman pasif filtrelerden çözüm bulunmayabilir. Bu durumda, aktif filtrelerin kullanılmasına gerek duyulmaktadır. Bu amaç için aktif güç filtresi, çalışma karakteristiği gereği sistemin belirtilen noktalarından elde edilen akım ve gerilim verilerinden dalga şekli bozulmalarını tespit eder ve bunlara kendi dahili güç kaynağı üzerinden ters yönlü etki oluşturarak bozulmaları giderir.

Pasif filtreler, seri ve paralel (şönt) olarak iki şekilde yapılmaktadır. Seri filtre olarak tasarlanan, filtrelerdeki rezonans hali görülmemesine rağmen, tam yükteki akımı taşıma ve hat gerilim değerine göre izolasyon zorunluluğu bulunmaktadır [11]. Güç sisteminde seri filtre devresinin uygulanmasında; sistemde bulunan bütün alıcıların çektiği akımların seri pasif filtrenin üzerinden geçmesidir. Ayrıca, tam hat gerilimlerinin yalıtılmasına ihtiyaç durulması ve gerilim düşümünün ortaya çıkmasıdır [12].

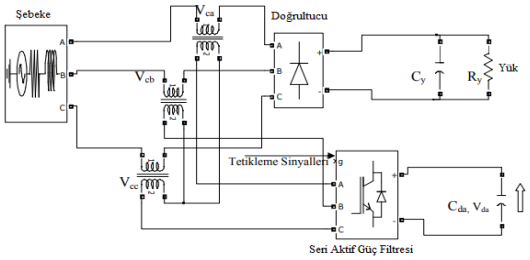
Harmoniği oluşturan kaynağa paralel olarak bağlanan pasif filtrelere paralel pasif filtreler denir. Paralel pasif filtre (PPF) büyük salınımlı harmoniklerin kompanzasyonu için modellenmektedir. Pasif filtrenin kullanıldığı bir güç sisteminde, filtre kullanılmadan önce güç katsayısı 0.6877 ve THD_I değeri % 12.71 olarak ölçülmüştür. Pasif filtre kullanıldıktan sonra aynı güç sistemindeki güç katsayısı 0.99 ve THD_I değeri % 3.591 olduğu tespit edilmiştir [13].

Şebekede bulunan harmonik değerlerin hesaplanarak, bunlara aynı genlik değerinde ters yönde harmonik üreten filtrelere aktif filtre denilmektedir. Aktif filtre olarak tasarlanmış bir filtrenin çalışma prensibi, pasif filtre olarak tasarlanmış bir filtrenin çalışma prensibi birbirlerinden tamamıyla farklı olmaktadır. Aktif filtre yardımıyla güç kalitesini etkileyen harmonikler elimine edilmektedir. Aktif filtreleri yapıları açısından; seri, paralel ve hibrit aktif güç filtresi olarak üç kısımda incelemek mümkündür. Bu filtreler, kontrol yöntemi açısından ise açık ve kapalı çevrim aktif güç kontrol filtre sistemi olmak üzere iki kısımda incelenmektedir.

Şebekeye paralel olarak bağlanan aktif filtrelere paralel aktif güç filtresi (PAGF) denilmektedir. Bu filtreler, endüstri tesislerinde en yaygın kullanım alanı bulan en önemli filtrelerdir [14]. Ayrıca, bu filtrelerle akım harmoniklerinin filtrelenmesi, reaktif güç kompanzasyonun sağlanması, yük akımının dengelemesi ve nötr akım kompanzasyonun sağlanması gibi akıma bağlı harmoniklerin yok edilmesi sağlanmaktadır [15,16]. Paralel aktif filtre ile, tesisin iç şebekesinde yoğun olarak bulunan 5. ve 7. akım harmonik bileşenlerin, kompanzasyon kapalı sistemde %10 seviyelerinden %3 seviyelerine ve gerilim harmonik bileşenlerinin, % 9 seviyelerinden %5,5 seviyelerine düşmüştür [9].

Şebekeyle bağlı bir transformatör üzerinden seri bağlanan filtrelere seri aktif güç filtresi (SAGF) denilmektedir. Bu filtre, genellikle güç kaynakları ve yükler arasında bağlanır ve gerilim harmonik bozulmalarını düzeltmek için yeterlidir [17]. SAGF, temelde anlık gerilimin giriş ve çıkışının yük süresince tam sinüzoidal gerilimin dalga şeklinin kararlı kalmasını sağlamak, şebekede oluşabilecek gerilim dengesizliğini yok etmek ve gerilim düşümlerinin elimine etmek için kullanılır [18,19]. SAGF devrede olmadan önce %THD değerleri; kaynak gerilimi için 41,23 ve akım harmonikleri için ise 19.46 olarak ölçülmüştür. SAGF devreye alındıktan sonra kaynak gerilim ve akımları için harmonik bozulmalar azalmış ve %THD değerleri; kaynak gerilimi için 0.33, akım harmonikleri için ise 0.02 olarak ölçülmüştür [20]. Hibrit bir yapıda kurgulanan SAGF ve paralel pasif filtreden oluşan hibrit bir yapıya ait benzetim çalışması yapılmıştır. Burada, filtre kullanılmadan önce % THD değerleri; a fazının 23.24, b fazının 23.79 ve c fazının ise 23.28 olarak ölçülmüştür. Hibrit filtre kullanıldığında % THD değerleri; a fazının 4.93, b fazının 4.84 ve c fazının ise

.75 olmuştur [21.] SAGF'in blok şeması Şekil 1'de erilmiştir.



Şekil 1: SAGF blok şeması

Gerilim değişimlerine duyarlı cihazlar için saf sinüzoidal alga şekli çok önemlidir. Bu dalga şekli, sistemde bulunan armonik ve ara harmoniklerin filtre edilmesiyle ağlanmaktadır. PAGF'nin SAGF'ye göre en belirgin kusuru, ltre üzerinde bulunan çıkış gerilim dalga formunun inüzoidal olan şeklini devam ettirememesidir. Belirtilen usur, harmonik ve ara harmoniklerin filtrelemesinde AGF'ın tercih edilmesini sağlamaktadır [18]. Bununla irlikte, SAGF transformatördeki bağlantıların tümünü yük kımlarının üzerinden akıtacağından, büyük kapasiteli lmalıdır. Harmonikli akım değerlerinin büyük olması urumunda, seçilecek aktif güç filtresi kapasitesinin de rtması gerekmektedir.

Aktif güç filtresi ile pasif güç filtresine ait bağlantı apılarının birleştirilmesinde oluşan filtreye hibrit aktif güç ltresi (HAGF) denilmektedir. Maliyet azaltmak ve etkinliği rtırmak amacıyla tasarlanan HAGF, gerilim regülasyonunda, üç kompanzasyonunda, kaynak ve harmonik kaynaklı yük rası izolasyonlarda veya sadece harmonik ompanzasyonunda kullanılmaktadır. Yüksek sıralı armoniklerin kompanzasyonunda, anahtarlama frekansları ınırlandırılmaktadır. Aynı zamanda, burada kullanılan AGF, PPF ile kaynak empedansının arasında olabilecek ezonansı yok edebilmek adına kullanıldığı bilinmektedir 14]. HAGF'nin tekniği için belirgin kusur, PPF için fazla ıktarda güç bileşeninin bulunmasıdır. Dolayısıyla, PPF ürekli sisteme bağlı olduğundan önceden lineer olmayan kaynakları bilinen alıcılarda kullanılması uygun olmaktadır [22].

Enerji sistemlerinde oluşan harmonik değerlerini belirli sınırlarda olması için farklı standartlar tarafından belirli sınırlama konulmuştur. Bu sınırlama toplam harmonik distorsiyonu (THD) değerine göre yapılır.

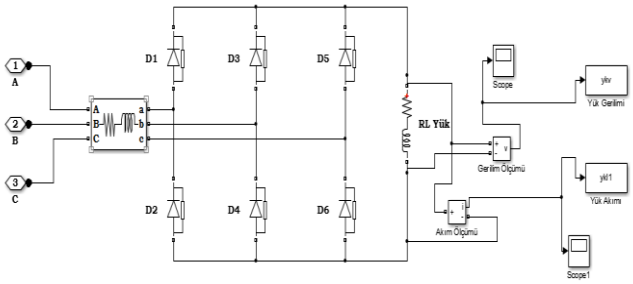
$$THD_V = \frac{\sqrt{\sum_{n=2}^{\infty}(V_n)^2}}{V_1} \quad ; \quad THD_I = \frac{\sqrt{\sum_{n=2}^{\infty}(I_n)^2}}{I_1} \qquad (1)$$

Harmonik bileşenin efektif değerinin, temel bileşen efektif değerine oranı olan THD, gerilim ve akım için Denklem (1)'deki gibi ayrı ayrı tanımlanır ve yüzde cinsinden ifade edilir. IEC standartlarına göre THD değeri, gerilimler için % 3, akımlar için ise % 6'dır.

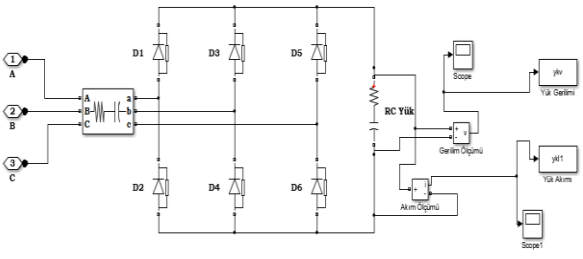
Yapılan çalışmada, endüktif ve kapasitif yük durumunda güç sistemindeki harmonik ve ara harmoniklerin kıyaslanıp incelenmesi yapılarak alınması gereken tedbirler belirtilmiştir. Güç sistemi üzerinde oluşan harmonik ve ara harmonik sonucunun görülmesi için MATLAB / Simulink programı yardımıyla modellenip, simülasyonu yapılmıştır. Ayrıca sonuçlar analiz edilerek gerekli önerilerde bulunulmuştur.

2. Materyal ve Yöntem

Güç sisteminde oluşan harmonik ve ara harmoniklerin modellenmesi ve simülasyonunda MATLAB/Simulink programı kullanılmıştır. Harmonik ve ara harmoniklerin elimine edilmesi için seri aktif güç filtresinden faydalanılmıştır. Yükün fazlara dengeli bir biçimde dağıldığı kabul edilen örnek olarak alınan sistemde THD azaltılması üzerinde durulmuştur. Yapılan analiz neticesinde grafikler incelenmiş, harmonik ve ara harmonikler tespit edilip eliminasyonu yapılmış ve sonuçlar karşılaştırılarak gösterilmiştir. Lineer olmayan endüktif ve kapasitif yükün MATLAB/Simulink programında hazırlanmış blok şeması, Şekil 2'de verilmiştir.



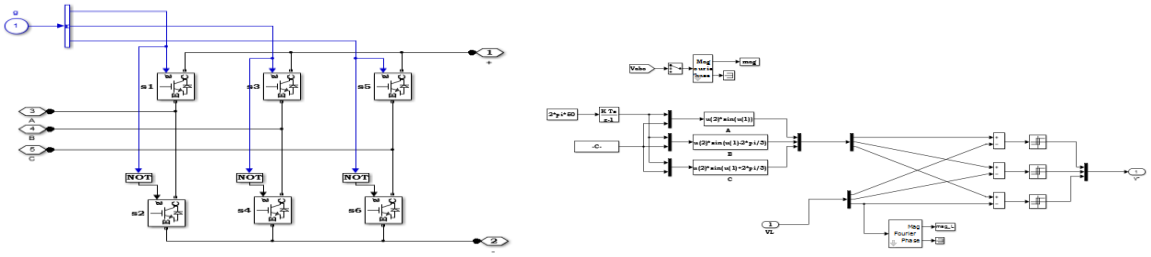
(a) Lineer olmayan endüktif yük



(b) Lineer olmayan kapasitif yük

Şekil 2: Lineer olmayan endüktif (a) ve kapasitif (b) yükün MATLAB/Simulink blok şeması

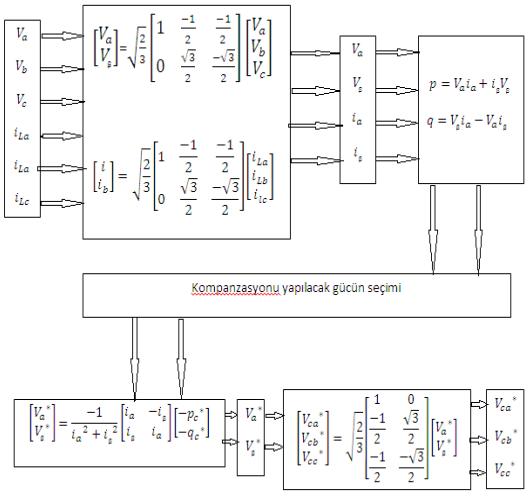
Çalışma kapsamında uygulanan seri aktif güç ltresinin tasarımında, literatürde mevcut olan filtre tasarım matematiksel modellerinden faydalanılmıştır. Bu kapsamda yapılan tasarım, Şekil 3'te verilmiştir.



(a) SAGF tetikleme şeması (b) SAGF iç yapısı blok şeması

Şekil 3: Seri aktif güç filtresi tetikleme ve iç yapısı blok şeması

Üç fazlı seri aktif güç filtresinin kontrol yapısına yönelik uşturulmuş blok diyagramı Şekil 4'te verilmiştir.



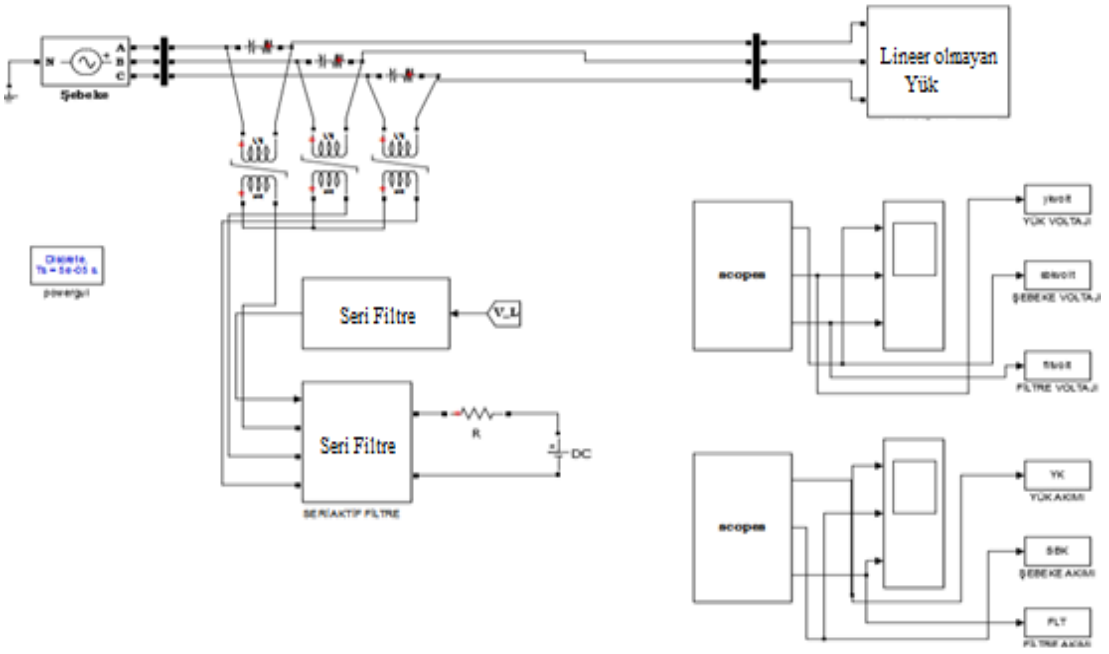
Şekil 4: Üç fazlı aktif güç filtresi temel kontrol blokları

Tasarlanan sistemde, üç fazlı bir dağıtım şebekesi dikkate alınmış ve her bir fazın gerilimi, 380 V olarak alınmıştır. Kullanılan yük, fazlara eşit şekilde dağıtılmıştır. Tasarlanan sistem de bulunan parametreler, Tablo 1'de verilmiştir.

Tablo 1: Tasarlanan sistemdeki SAGF'ye ait parametreler

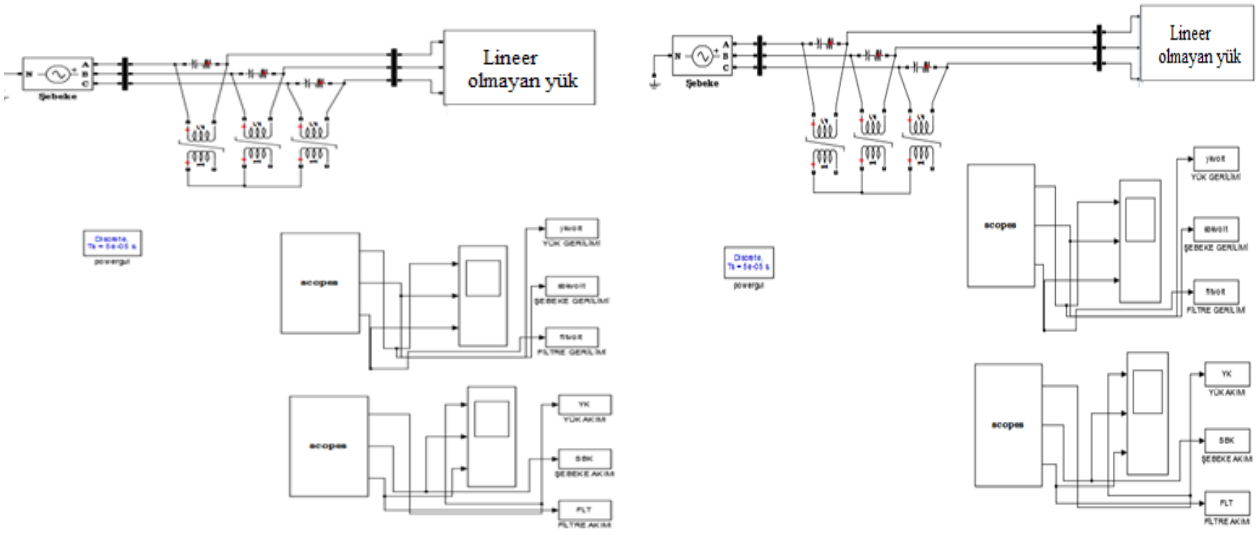
SAGF Parametreleri	Değer
Şebekeye Ait Gerilim ve Akım (Va, Vb, Vc, Ik)	380 V, 60A
Şebeke Frekansı	50 Hz
Şebekeye Ait Empedans (Rk, Ck)	1 Ω, 100e-6 F
Endüktif Yüke Ait Empedans (Ry, Ly)	60 Ω, 15e-3 H
Kapasitif Yüke Ait Empedans (Ry, Cy)	60 Ω, 1e-6 F

Güç sistemi için MATLAB/Simulink programında sistem yükü tasarlanmıştır. Buna ilişkin tasarlanan blok şeması, Şekil 5'te verilmiştir.



Şekil 5: Güç sistemine ilişkin MATLAB/Simulink blok şeması

Lineer olmayan endüktif ve kapasitif yüklü ve filtre ullanılmadan önceki güç sistemine ilişkin MATLAB/Simulink blok şeması, Şekil 6'da verilmiştir

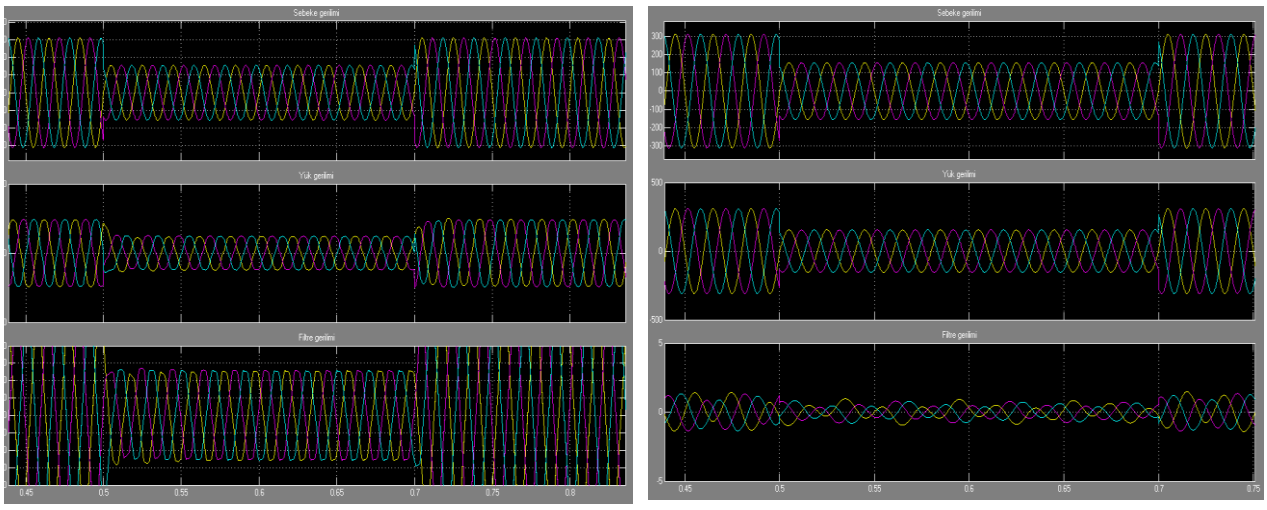


(a) (b)

Şekil 6: Filtre kullanılmadan endüktif yüklü (a) ve kapasitif yüklü (b) güç sistemine ilişkin MATLAB/Simulink blok şeması

3. Bulgular ve Tartışma

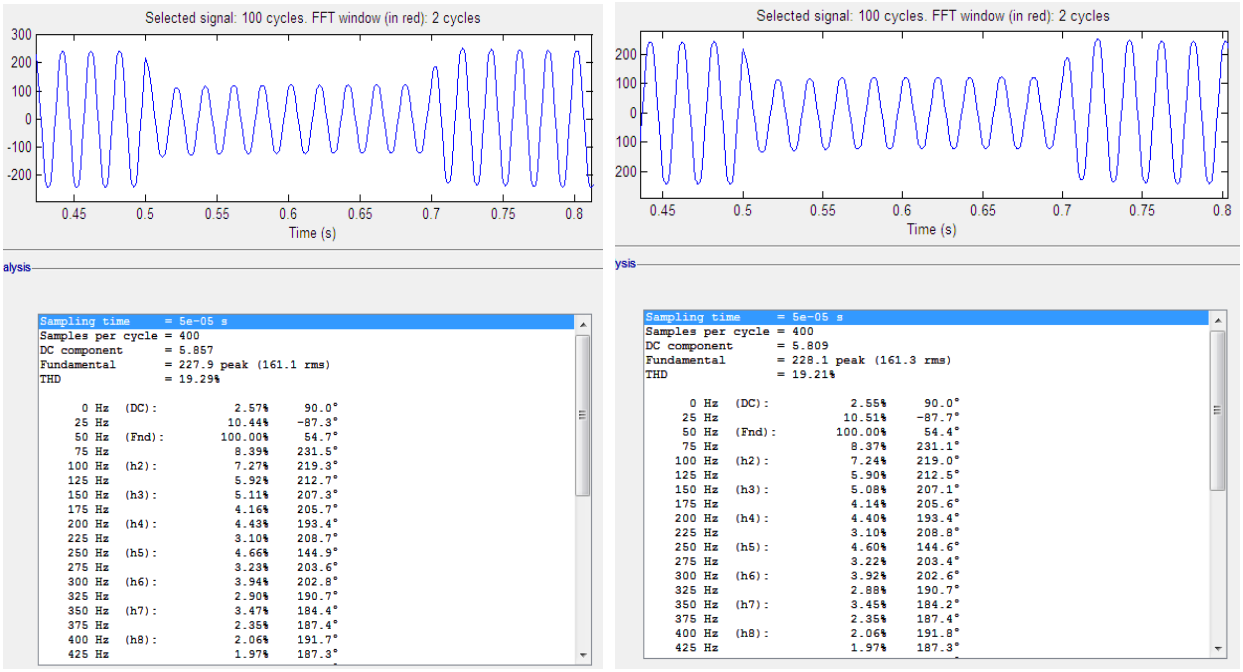
Yapılan çalışmada, Şekil 5'te verilen güç sistemi, ATLAB/Simulink programı yardımıyla SAGF'nin sistem ükü (endüktif ve kapasitif) tasarlanmış ve simülasyon için veriler alınmıştır. Filtrenin değişen gerilim karşısındaki tolerasyon aralığı tespit edilmiş olup analiz sonucunda istenmeyen gerilim dengesizlikleri yok edilmiştir. Filtre kullanılmadan endüktif ve kapasitif yüklü güç sistemine ilişkin gerilim dalga formları Şekil 7'de verilmiştir.



(a) (b)

Şekil 7: Filtresiz endüktif yüklü (a) ve kapasitif yüklü (b) güç sistemine ilişkin gerilim dalga formları

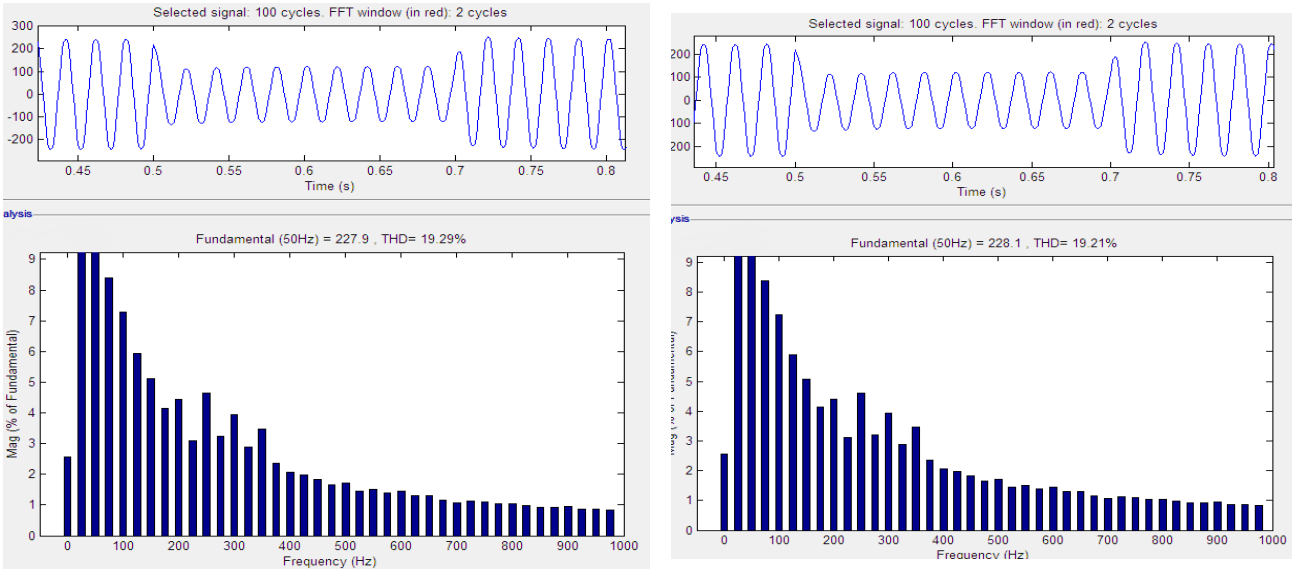
Şekil 7'de görüldüğü gibi, tasarlanan filtre devrede lmadığından güç sistemine ilişkin olan gerilim dalga ormalarının örnek alınan (0.4-0.8) t süreleri boyunca armonikten dolayı genliğinin 250 V'tan 150 V'a azalmıştır. ayrıca gerilim dalga formu sinüzoidal olmaktan uzaklaşmıştır. Endüktif ve kapasitif yüklü filtre kullanılmayan güç sisteminin hızlı fourier dönüşümü (Fast Fourier Transform–FFT) sistemine ait genlik analiz diyagramı Şekil 8'de verilmiştir.



(a) (b)

Şekil 8: Filtresiz endüktif yüklü (a) ve kapasitif yüklü (b) güç sisteminin FFT sistemi genlik analiz diyagramı

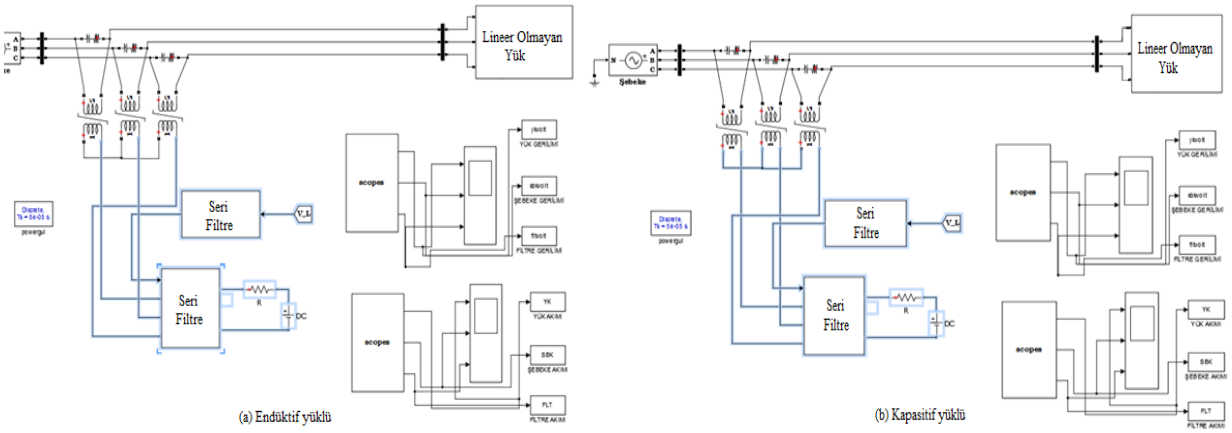
Şekil 8'de görüldüğü gibi filtrelenmemiş sistemin FFT sistemi analiz diyagramında görülen endüktif yüklü için THD_v oranının %19.29 ve kapasitif yüklü için THD_v oranının %19.21 olduğu görülmektedir. Ayrıca n. harmonik dışında kalan ara harmoniklerin de var olduğu görülmektedir. Endüktif ve kapasitif yüklü filtre yapılmayan güç sisteminin FFT sistemi ve genlik spektrumu görünümü Şekil 9'da verilmiştir.



(a) (b)

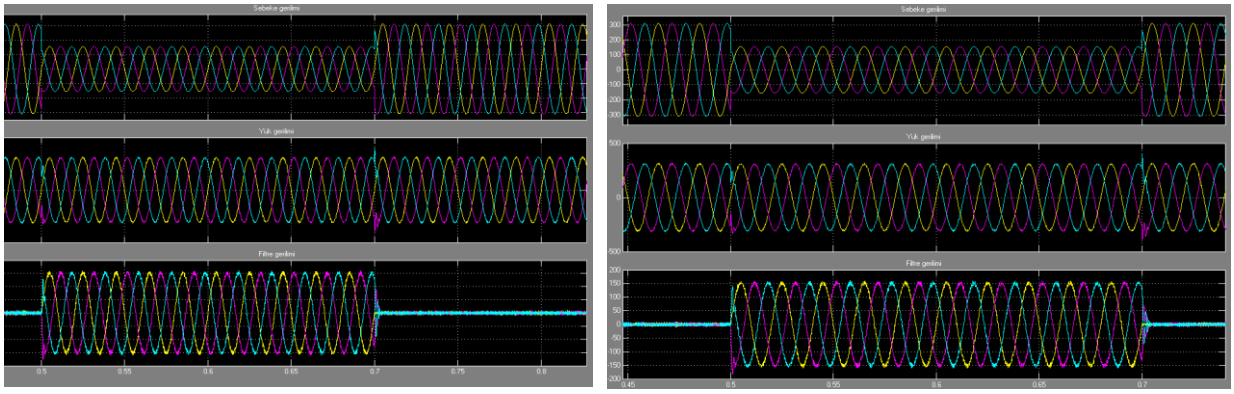
Şekil 9: Endüktif yüklü (a) ve kapasitif yüklü (b) filtre yapılmayan güç sisteminin FFT sistemi genlik spektrumu

SAGF, endüktif ve kapasitif yüklü olan güç sistemine ilişkin MATLAB/Simulink blok şeması, Şekil 10'da verilmiştir.



Şekil 10: SAGF endüktif yüklü (a) ve kapasitif yüklü (b) güç sistemine ilişkin MATLAB/Simulink blok şeması

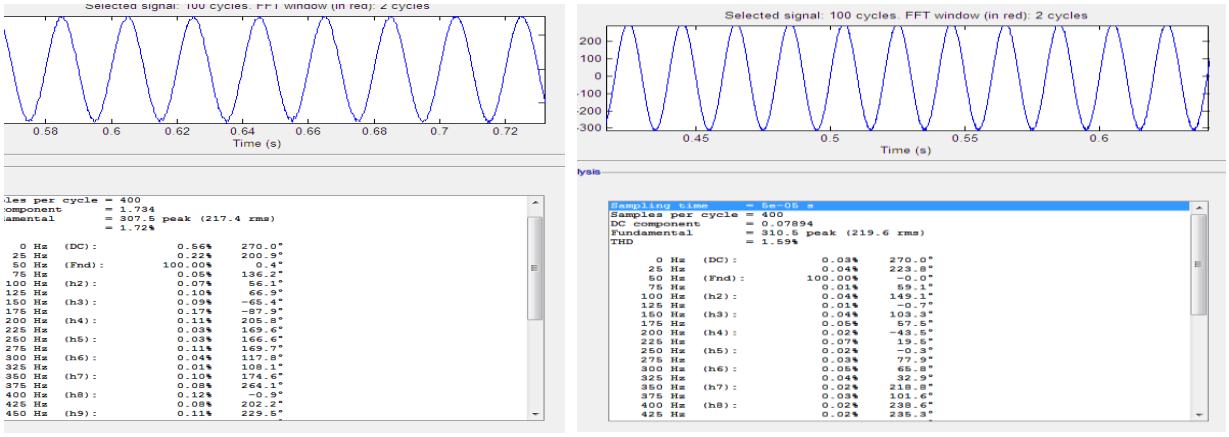
ndüktif ve kapasitif yüklü filtre yapılan güç sistemine ilişkin gerilim dalga verileri, Şekil 11'de verilmiştir.



(a) (b)

Şekil 11: Endüktif yüklü (a) ve kapasitif yüklü (b) filtreli sistem gerilim dalga verileri

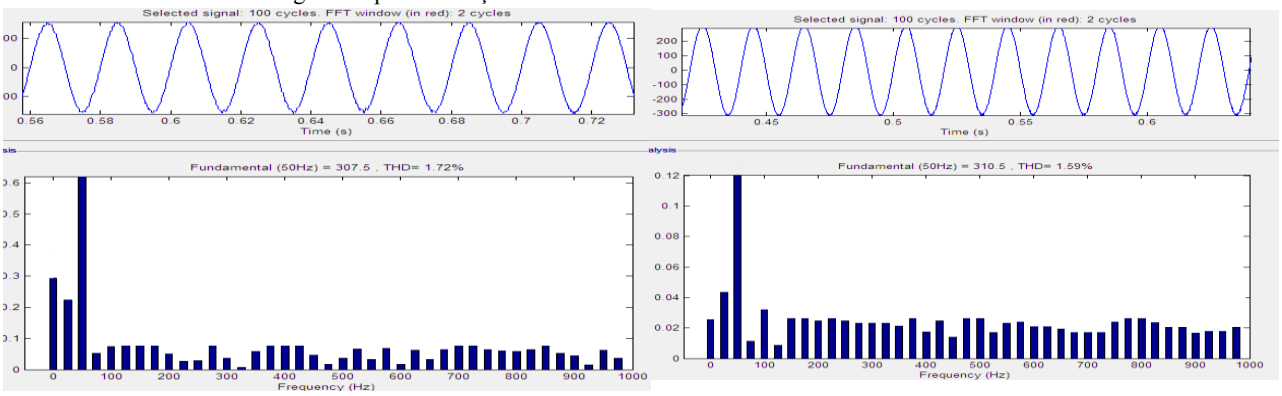
ekil 11'de görüldüğü gibi SAGF, endüktif ve kapasitif i güç sisteminde olduğunda gerilim dalga formunun lığı için örnek alınan (0.4-0.8) t sürelerindeki yüklerde, V gerilim değerinin korunması için gerilim genliğinin 100 V arttırılıp harmonik ve ara harmoniklere karşı koyulduğu görülmüştür. Endüktif ve kapasitif yüklü filtre yapılan güç sisteminin FFT sistemine ait analiz diyagramı, Şekil 12'de verilmiştir.



(a) (b)

Şekil 12: Filtre yapılan endüktif yüklü (a) ve kapasitif yüklü (b) güç sisteminin FFT sistemi genlik analiz diyagramı

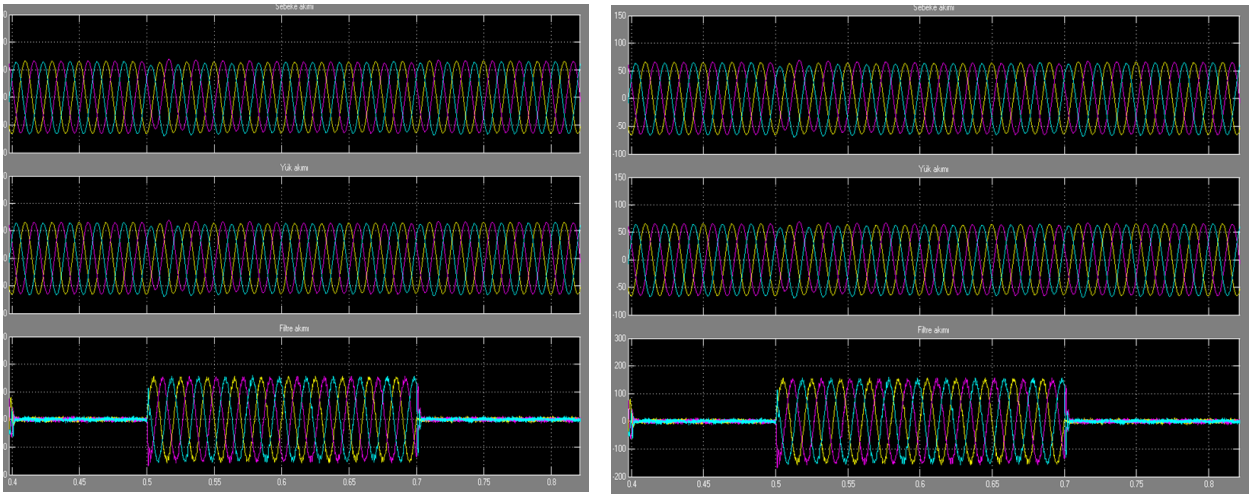
Endüktif ve kapasitif yüklü filtre yapılan güç isteminin FFT sistemine ait genlik spektrumu Şekil 13'te görülmektedir.



(a) (b)

Şekil 13: Filtre yapılan endüktif yüklü (a) ve kapasitif yüklü (b) güç sisteminin FFT sistemi genlik spektrumu

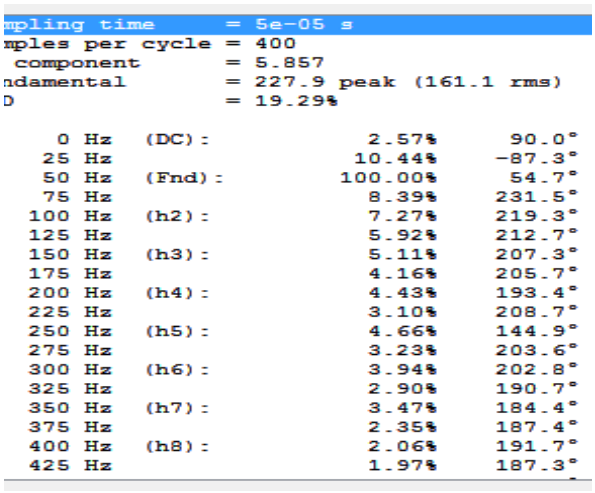
Şekil 12'deki analiz diyagramında ve Şekil 13'teki genlik pektrumunda endüktif ve kapasitif yüklü sistemde örnek lınan (0.5-0.7) t süre aralığında genlik değerini koruduğu 3., ., 7., 9. harmoniklerin sıfırlara çekildiği ve aradaki 3.5, 4.5, .5, 7.5'deki ara harmonik değerlerinde tanımlanan nonlineer engesizlikler yok olduğu görülmektedir. Bilgiler üç fazlı yük sisteminden alınmıştır. Güç sistemi için temel frekans 50 Hz'dir. Seri aktif güç filtre devredeyken sinüzoidal dalga formunu koruduğu ve nonlineer yüke karşı koyup harmonik ve ara harmonikleri yok ettiği görülmüştür. Endüktif ve kapasitif yüklü güç sisteminin akım dalga verileri, Şekil 14'te verilmiştir.



(a) (b)

Şekil 14: Endüktif yüklü (a) ve kapasitif yüklü (b) filtreli sistem akım dalga verileri

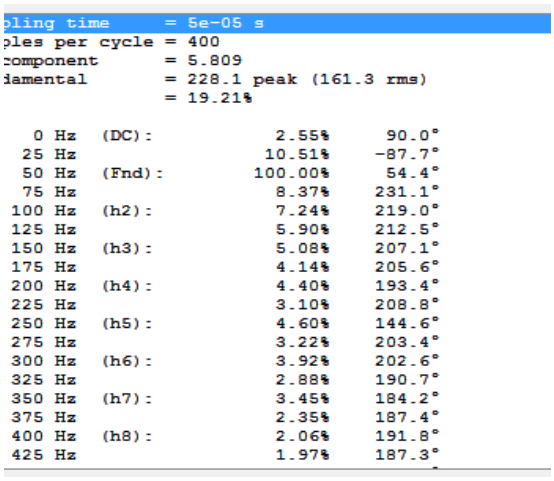
Şekil 15'te SAGF kullanılmadan önce olan endüktif ükün, Şekil 16'da SAGF kullanılmadan önce olan kapasitif ükün, Şekil 17'de SAGF kullanıldıktan sonra endüktif ükün, Şekil 18'de SAGF kullanıldıktan sonra kapasitif ükün harmonik analizleri verilmiştir. Filtre yapılmadan önce e filtre yapıldıktan sonra bulunan veriler karşılaştırılmış ve üç sistemi değerleri Tablo 2'de verilmiştir. Elde edilen FFT nalizi yardımıyla harmonik değerlerdeki sonuçların çok üşük bir seviyede olduğu görülmüştür. Bununla birlikte AGF sistemde iken harmonik ve ara harmonikerin yok edildiği tespit edilmiş ve karşılaştırmaları çizelgeler şeklinde verilmiştir.



```
mpling time      = 5e-05 s
mples per cycle  = 400
 component       = 5.857
ndamental        = 227.9 peak (161.1 rms)
D                = 19.29%

    0 Hz   (DC):          2.57%    90.0°
   25 Hz                 10.44%   -87.3°
   50 Hz   (Fnd):       100.00%    54.7°
   75 Hz                 8.39%    231.5°
  100 Hz   (h2):         7.27%    219.3°
  125 Hz                 5.92%    212.7°
  150 Hz   (h3):         5.11%    207.3°
  175 Hz                 4.16%    205.7°
  200 Hz   (h4):         4.43%    193.4°
  225 Hz                 3.10%    208.7°
  250 Hz   (h5):         4.66%    144.9°
  275 Hz                 3.23%    203.6°
  300 Hz   (h6):         3.94%    202.8°
  325 Hz                 2.90%    190.7°
  350 Hz   (h7):         3.47%    184.4°
  375 Hz                 2.35%    187.4°
  400 Hz   (h8):         2.06%    191.7°
  425 Hz                 1.97%    187.3°
```

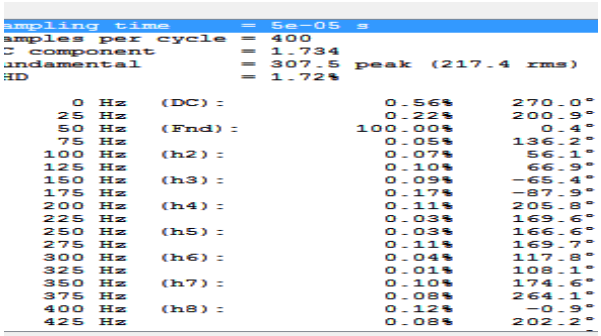
15: Filtre yapılmadan önce endüktif yükün FFT sistem mi harmonik analizi



```
pling time      = 5e-05 s
ples per cycle = 400
component       = 5.809
damental        = 228.1 peak (161.3 rms)
                = 19.21%

  0 Hz   (DC):          2.55%    90.0°
 25 Hz                 10.51%   -87.7°
 50 Hz   (Fnd):       100.00%    54.4°
 75 Hz                 8.37%    231.1°
100 Hz   (h2):         7.24%    219.0°
125 Hz                 5.90%    212.5°
150 Hz   (h3):         5.08%    207.1°
175 Hz                 4.14%    205.6°
200 Hz   (h4):         4.40%    193.4°
225 Hz                 3.10%    208.8°
250 Hz   (h5):         4.60%    144.6°
275 Hz                 3.22%    203.4°
300 Hz   (h6):         3.92%    202.6°
325 Hz                 2.88%    190.7°
350 Hz   (h7):         3.45%    184.2°
375 Hz                 2.35%    187.4°
400 Hz   (h8):         2.06%    191.8°
425 Hz                 1.97%    187.3°
```

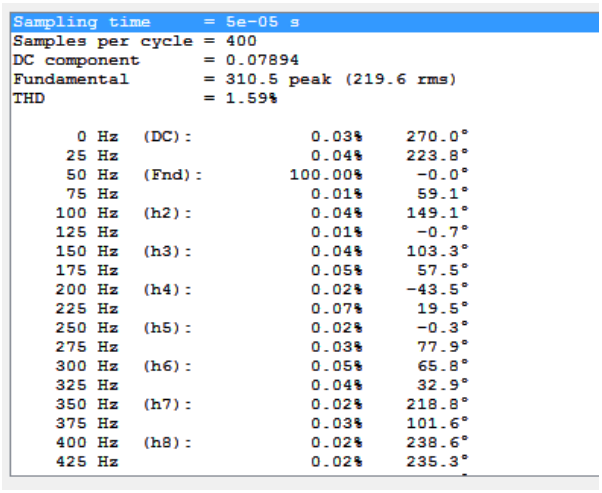
16: Filtre yapılmadan önce kapasitif yükün FFT sistem mi harmonik analizi



```
ampling time      = 5e-05 s
amples per cycle  = 400
C component       = 1.734
undamental        = 307.5 peak (217.4 rms)
HD                = 1.72%

    0 Hz   (DC):          0.56%   270.0°
   25 Hz                  0.22%   200.9°
   50 Hz   (Fnd):       100.00%     0.4°
   75 Hz                  0.05%   136.2°
  100 Hz   (h2):          0.07%    56.1°
  125 Hz                  0.10%    66.9°
  150 Hz   (h3):          0.09%   -65.4°
  175 Hz                  0.17%   -87.9°
  200 Hz   (h4):          0.11%   205.8°
  225 Hz                  0.03%   169.6°
  250 Hz   (h5):          0.03%   166.6°
  275 Hz                  0.11%   169.7°
  300 Hz   (h6):          0.04%   117.8°
  325 Hz                  0.01%   108.1°
  350 Hz   (h7):          0.10%   174.6°
  375 Hz                  0.08%   264.1°
  400 Hz   (h8):          0.12%    -0.9°
  425 Hz                  0.08%   202.2°
```

:il 17: Filtre yapıldıktan sonra endüktif yükün FFT sistem ilimi harmonik analizi



```
Sampling time      = 5e-05 s
Samples per cycle = 400
DC component       = 0.07894
Fundamental        = 310.5 peak (219.6 rms)
THD                = 1.59%

    0 Hz   (DC):          0.03%   270.0°
   25 Hz                  0.04%   223.8°
   50 Hz   (Fnd):       100.00%    -0.0°
   75 Hz                  0.01%    59.1°
  100 Hz   (h2):          0.04%   149.1°
  125 Hz                  0.01%    -0.7°
  150 Hz   (h3):          0.04%   103.3°
  175 Hz                  0.05%    57.5°
  200 Hz   (h4):          0.02%   -43.5°
  225 Hz                  0.07%    19.5°
  250 Hz   (h5):          0.02%    -0.3°
  275 Hz                  0.03%    77.9°
  300 Hz   (h6):          0.05%    65.8°
  325 Hz                  0.04%    32.9°
  350 Hz   (h7):          0.02%   218.8°
  375 Hz                  0.03%   101.6°
  400 Hz   (h8):          0.02%   238.6°
  425 Hz                  0.02%   235.3°
```

Şekil 18: Filtre yapıldıktan sonra kapasitif yükün FFT sistem gerilimi harmonik analizi

Tablo 2: Filtresiz ve filtreli durum için sistemin gerilim ve THD_v değerleri

Sistem		**Sistem Gerilim THD_v (%)**	**Sistem Gerilim (V)**
Endüktif Yük Durumu	Filtresiz	19.29	217.4 RMS
	Filtreli	1.72	161.1 RMS
	Azalma	91.08	
Kapasitif Yük Durumu	Filtresiz	19.21	219.6 RMS
	Filtreli	1.59	161.3 RMS
	Azalma	91.34	

Sistemden alınan ve Şekil 15-19'da görülen veriler ayrıntılı analizinde, önemli bilgileri elde edilmiştir. Bu bilgilerden harmonik ve ara harmonik değerlerinin eliminasyonu görülmüştür. Ayrıca harmonik ve ara harmonik bozulmadaki değerler farklı yüklerdeki karşılaştırmalı olarak gösterilmiş ve kazanımlar verilmiştir. Endüktif ve kapasitif yüklerin SAGF'ye karşı verilen gerilimin sonuçları Tablo 3'te verilmiştir.

Tablo 3: Endüktif ve kapasitif yüklerin SAGF'ye karşı verilen gerilimin sonuçları

	Endüktif Yükün Olması Durumunda		**Kapasitif Yükün Olması Durumunda**	
Harmonik	**Filtre Yapılmadan Önce Yükteki Harmonik Bozulma (%)**	**Filtre Yapıldıktan Sonra Yükteki Harmonik Bozulma (%)**	**Filtre Yapılmadan Önce Yükteki Harmonik Bozulma (%)**	**Filtre Yapıldıktan Sonra Yükteki Harmonik Bozulma (%)**
3	5.11	0.09	5.08	0.04
3.5	4.16	0.17	4.14	0.05
5	4.66	0.03	4.60	0.02
5.5	3.23	0.11	3.22	0.03
7	3.47	0.10	3.45	0.02
7.5	2.35	0.08	2.35	0.03

Tablo 3'te görüldüğü gibi, endüktif yükte SAGF ullanılmadan önce 3. harmonik değerindeki harmonik ozulma değeri % 5.11 olmuştur. SAGF'nin sisteme dahil dilmesiyle 3. harmonik değerindeki harmonik bozulma eğeri % 0.09'a düşmüş ve % 98.24 oranında azalmıştır. asif filtre kullanılarak yapılan harmonik eliminasyon alışmasında 3. harmonik bileşende % 59.51 oranında azalma lduğu tespit edilmiştir [23]. Endüktif yükte ara harmonik lan 3.5 değerindeki harmonik bozulma değeri % 4.16 iken AGF'nin sisteme dahil edilmesiyle bu ara harmonikteki armonik bozulma değeri % 0.17'a düşmüş ve % 95.91 ranında azalmıştır. Benzer şekilde 5. ve 7. harmonik ile 5.5 e 7.5 ara harmoniklerde de SAGF kullanıldıktan sonra % 96'nın üzerinde azalma oluşmuştur. Buradan aktif filtre ullanılmasıyla harmonik ve ara harmonik değerlerinin daha azla düştüğü görülmüştür.

Kapasitif yükte SAGF kullanılmadan önce 3. harmonik eğerindeki harmonik bozulma değeri % 5.08 olmuştur. AGF'nin sisteme dahil edilmesiyle 3. harmonik değerindeki armonik bozulma değeri % 0.04'a düşmüş ve % 99.21 ranında azalmıştır. Kapasitif yükte ara harmonik olan 3.5 eğerindeki harmonik bozulma değeri % 4.14 iken SAGF'nin isteme dahil edilmesiyle bu ara harmonikteki harmonik ozulma değeri % 0.05'e düşmüş ve % 98.79 oranında zalmıştır. Benzer şekilde 5. ve 7. harmonik ile 5.5 ve 7.5 ara armoniklerde de SAGF kullanıldıktan sonra % 98'in zerinde azalma oluşmuştur. Buradan aktif filtre ullanılmasıyla harmonik ve ara harmonik değerlerinin daha azla düştüğü görülmüştür.

4. Sonuç

Elektrik enerji sistemlerinin sorunsuz olarak çalışması için sistemin belirli bir kalite koşuluna sahip olması gerekiyor. Elektrik şebekesinde var olan harmonik akımlar elektrik şebekesinin empedansında gerilim düşümünün oluşmasına neden olmaktadır. Bu durum gerilimin dalga formunun bozulmasına, kayıpların artmasına, elektrik şebekesinde kullanılan cihazların arızalanmasına yol açmaktadır. Burada endüktif ve kapasitif yük durumu için filtre kullanmadan önceki ve filtre kullanıldıktan sonra harmonik ve ara harmoniklerdeki bozulmaları göre bilmek için sistem MATLAB/Simulink programında modellenerek simülasyonu yapılmıştır. Filtre kullanmadan önce endüktif yüklü nonlineer yük için sistemin THDv oranının % 19.29 olduğu, filtre kullanıldıktan sonra sistemin THDv oranının % 1.72 seviyesine düştüğü yani % 91.08 oranında azaldığı tespit edilmiştir. Filtre kullanmadan önce kapasitif yüklü nonlineer yük için sistemin THDv oranının % 19.21 olduğu, filtre kullanıldıktan sonra sistemin THDv oranının % 1.59 seviyesine düştüğü yani % 91.34 oranında azaldığı tespit edilmiştir. Endüktif yük kullanılması durumunda, SAGF'nin sisteme dahil edilmesiyle harmonik ve ara harmonikteki harmonik bozulma değeri % 96'ın üzeri oranında azalmıştır

Kapasitif yük kullanılması durumunda, SAGF'nin sisteme dahil edilmesiyle harmonik ve ara harmonikteki harmonik bozulma değeri % 98'in üzeri oranında azalmıştır.

Buradan da gerek sistemin tasarım aşamasında gerekse de sistemin işletme aşamasında harmonik analizi ile birlikte ara harmonik analizlerinin de yapılaması ve filtrelemeye önem verilmesi kanaatine varılmıştır. Seri aktif güç filtresi

ullanmanın daha faydalı olacağı görülmüştür.

Yapılan bu çalışmayla, seri aktif güç filtresinin de armonik kaynağı gibi davrandığı gözlemlenmiştir. Ancak istemdeki harmonikleri yok edeceği için bu şekilde dalga ormu üretmesi gerektiği tespit edilmiştir. Elektrik güç istemlerinde harmonik ve ara harmoniklerin olması, elektrik üç sistemlerinin çalışmayacağı anlamına gelmez. Bu ususların sistemin tasarım aşamasında dikkate alınması erekir.

5. Kaynakça

1] M.R. Tür, F.Yaprdakdal, "Yenilenebilir Enerji Kaynaklarına Dayalı Bir Sistemde Güç Kalitesi Analizi, Kontrolü ve İzlemesi", *Gazi University Journal of Science Part C: Design and Technology*, 8 (3), pp. 572-587, 2020.

2] A. Kalair, N. Abas, A.R. Kalair, Z. Saleem, N. Khan, "Review of harmonic analysis, modeling and mitigation techniques", *Renew. Sustain. Energy Rev*.,78, pp.1152–1187, 2017.

3] A. Otcenasova, A. Bolf, J. Altus, M. Regula, "The Influence of Power Quality Indices on Active Power Losses in a Local Distribution Grid", *Energies*, 12(7), pp.1389, 2019.

4] S. Adak, H. Cangi, B. Eid, "Developed analytical expression for current harmonic distortion of the PV system's inverter in relation to the solar irradiance and temperature", *Electrical Engineering*, 103, pp. 697–704, 2021.

5] H.C. Lin, "Sources, effects, and modelling of interharmonics", *Mathematical Problems in Engineering*, vol.2014, pp.1-10, 2014.

6] S. Rüstemli, E. Okuducu, M.N. Almalı, S.B. Efe, "Reducing The Effects Of Harmonics On The Electrical Power Systems With Passive Filters", *Bitlis Eren Univ J Sci & Technology,* 5(1), pp. 1 – 10, 2015.

7] S. Rüstemli, M.S.Cengiz, "Passive Filter Solutions and Simulation Performance in Industrial Plants", *Bitlis Eren Univ J Sci & Technology*, 6 (1), pp. 39-43, 2016.

8] S. Adak, H. Cangi, A.S.Yılmaz, "Design of an LLCL type filter for stand-alone PV systems' harmonics", *Journal of Energy Systems*. 3(1), pp. 36-50, 2019.

9] E. Orucu, "Endüstriyel Tesislerde Aktif Harmonik Filtre Uygulaması İle Elektrik Güç Kalitesinin Düzenlenmesi", *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi,* Denizli, Türkiye, 2020.

[10] S.B. Efe, A. Dalcalı, "Elektrik Makinalarında Harmonik ve Ara-Harmonik Analizi" *Electronic Letters on Science and Engineering*, 17 (2), pp. 117-125, 2021.

[11] F. Kürker, R. Taşaltın, K. Karadağ, "Elektrik Tesisinde Harmonik İncelemesi ve Harmonik Filtreli Kompanzasyon", *Harran Üniversitesi Mühendislik Dergisi*, 3 (3), pp. 43-51, 2018.

[12] S. Rüstemli, S. Tekev, "Güç Sistemindeki Harmoniklerin Bilgisayar Destekli Modellenmesi", *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 12(5), pp. 711-718, 2021.

[13] S. Adak, H. Cangi, A.S.Yılmaz, "Doğrusal Olmayan Yüklerde Güç Faktörünün Düzeltilmesi ve Harmonik Bileşenlerin Süzülmesi", *Gazi University Journal of Science Part C: Design and Technology*, 7 (1), pp. 153-164, 2019.

[14] M. Kashif, M.J.Hossain, F. Zhuo, S. Gautam, "Design and implementation of a three-level active power filter for harmonic and reactive power compensation", *Electric Power Systems Research*, 165, pp.144–156, 2018.

[15] V.N. Gali, N. Gupta, R.A.Gupta, "Mitigation of power quality problems using shunt active power filters: A comprehensive review", *12th IEEE Conference on Industrial Electronics and Applications (ICIEA), Cambodia*, June 18-20, pp. 1100-1105, 2017.

[16] M.T.L. Gayatri, A.M. Parimi, K.A.V. Pavan, "A review of reactive power compensation techniques in microgrids", *Renewable and Sustainable Energy Reviews*, 81, pp.1030–1036, 2018.

[17 L. F. J. Meloni, F. L. Tofoli, A. J. J Rezek, E. R. Ribeiro, "Modeling and Experimental Validation of a Single-Phase Series Active Power Filter for Harmonic Voltage Reduction", *IEEE Access*, 7, pp. 151971–151984, 2019.

[18] W. Śleszyński, A. Cichowski, P. Mysiak, "Current harmonic controller in multiple reference frames for series active power filter integrated with 18-pulse diode rectifier", *Bulletin of The Polish Academy of Sciences Technical Sciences*, 66(5), pp. 699-704, 2018.

[19] R. Jadeja, A. Faldu, T. Trivedi, S. Chauhan, V. Patel, "Compensation of harmonics in neutral current using active power filter for three phase four wire system",

Gazi University Journal of Science, 31 (3), pp. 846-861, 2018.

20] M. O. Mahmoud, W. Mamdouh, H. Khalil, "Power System Distortion Mitigation by Using Series Active Power Filter", International Journal of Industry and Sustainable Development (IJISD), 1 (2), pp. 36-48, 2020.

21] M.T. Akçay, U. Arifoğlu, "Anlık Güç Kuramı ile Dengesiz Yükler İçin SAGF Uygulaması", *Haliç Üniversitesi Fen Bilimleri Dergisi*, 2 (1), pp. 1-16, 2019.

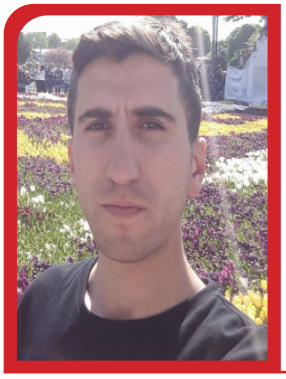
[22] S. Agrawal, D.K. Palwalia, M. Kumar M, "Performance Analysis of ANN Based three-phase four-wire Shunt Active Power Filter for Harmonic Mitigation under Distorted Supply Voltage Conditions", *IETE Journal of Research*, pp.1-9, 2019.

[23] S. Adak, "Güç Sisteminde Triplen Harmoniklerin Eliminasyonu", *SETSCI Conference Indexing System*, 3, pp. 111-116, 2018.

Özgeçmişler

Sabir RÜSTEMLİ, Lisans ve Yüksek Lisans eğitimini, 1990 yılında Azerbaycan Devlet Petrol Akademisinde tamamlamıştır. Doktora Eğitimini 1995 yılında Azerbaycan Devlet Petrol Akademisi ve Azerbaycan Bilimler Akademisinde tamamlamıştır. 2005-2012 yılları arasında Yüzüncü Yıl Üniversitesi Elektrik-Elektronik Mühendisliği bölümünde Profesör olarak çalışmıştır. 2012 yılından itibaren Bitlis Eren Üniversitesi Mühendislik Mimarlık Fakültesi Elektrik-Elektronik Mühendisliği bölümünde Profesör olarak çalışmaktadır. Araştırma alanları; yüksek gerilim, iletim dağıtım sistemleri ve yenilenebilir enerji kaynaklarıdır.

Behçet KOCAMAN, Lisans eğitimini, 1993 yılında Yıldız Teknik Üniversitesi Elektrik Mühendisliği bölümünde tamamlamıştır. Yüksek Lisans ve Doktora eğitimlerini sırasıyla 1997 ve 2015 yıllarında Kocaeli Üniversitesi Elektrik Mühendisliği Anabilim Dalında tamamlamıştır. 1993-2015 yılları arasında ise Yüzüncü Yıl Üniversitesi/ Bitlis Eren Üniversitesi Tatvan Meslek Yüksekokulu Elektrik-Enerji bölümünde Öğretim Görevlisi olarak çalışmıştır. 2015-2021 yılları arasında Bitlis Eren Üniversitesi Mühendislik Mimarlık Fakültesi Elektrik-Elektronik Mühendisliği bölümünde Dr. Öğr. Üyesi olarak çalışmıştır. Halen aynı bölümünde Doçent olarak görev yapmaktadır. Araştırma alanları; enerji verimliliği, aydınlatma, reaktif güç kompanzasyonu ve yenilenebilir enerji sistemleridir



Sinan TEKEV, Lisans eğitimini 2014 yılında Pamukkale Üniversitesi Elektrik-Elektronik Mühendisliği bölümünde tamamlamıştır. Yüksek Lisans eğitimini, 2021 yılında Bitlis Eren Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalında tamamlamıştır. Araştırma alanı, güç kalitesidir.