KTEAD

E-ISSN : 3023-4735
DOI: 10.5281/zenodo.10102956

Kuantum Teknolojileri ve Enformatik Araştırmaları Dergisi

International Peer-Reviewed and Open Access Electronic Journal
Uluslararası Hakemli ve Açık Erişimli Elektronik Dergi

Journal of Quantum Technologies and Informatics Research

JQTAIR

ISSUE 1

VOLUME/CİLT: 2
YEAR/YIL: 2024

HOLISTENCE publications

https:https://journals.gen.tr/index.php/jqtair/

Journal of Quantum Technologies
and Informatics Research

JQTAIR

E-ISSN: 3023-4735
DOI: 10.5281/zenodo.10102956

International Peer-Reviewed and Open Access Electronic Journal
Uluslararası Hakemli ve Açık Erişimli Elektronik Dergi

Volume/Cilt: 2

Issue/Sayı: 1

Year/Yıl: 2024

E-posta: jqtair@gmail.com

Web: https://journals.gen.tr/index.php/jqtair/

İletişim: Adres: Anafartalar Yerleşkesi, B2 Blok, Oda No: 8,
Çanakkale, Türkiye

HOLISTENCE
publications

EDİTÖR'den

"Journal of Quantum Technologies and Informatics Research" dergisinin ikinci sayısını sunmanın gururu içindeyiz. Bu özel sayı, 29 Şubat 2024'te yayımlanarak, kuantum teknolojileri ve enformatik araştırmaları alanında önemli bir dönüm noktasını işaret ediyor. Alanımızın en parlak zihinlerinin katkılarıyla, siber güvenlikten otomatik birim testlerinin oluşturulmasına, kuantum hesaplama uygulamalarından yapay zekâ ve makine öğrenmesine kadar geniş bir konu yelpazesini kapsayan bu sayı, bilimin sınırlarını zorlayan çalışmaları yansıtmaktadır.

Cengiz Çalıkoğlu'nun "Kolektif Siber Güvenliğin Önemi ve Türk Devletler Teşkilatı" başlıklı makalesi, kuantum siber güvenlik teknolojilerinin gelecekteki potansiyelini vurgulayarak, siber güvenlik stratejilerimizin nasıl evrimleşmesi gerektiğini derinlemesine analiz ediyor. Bu çalışma, siber güvenliğin geleceğine dair kapsamlı bir bakış açısı sunarken, kuantum teknolojilerinin bu alandaki önemini ortaya koyuyor. Sevdanur Genç'in "Otomatik Birim Test Oluşturmak İçin Opcode Ayrıştırma Yaklaşımının Geliştirilmesi" başlıklı çalışması ve Osman Semi Ceylan'ın "Kuantum Fourier Dönüşümünün Yüksek Boyutta Cirq Kullanarak Uygulanması" üzerine makalesi, teknolojik yeniliklerin nasıl pratik uygulamalara dönüştürülebileceğini gösteriyor. Ayetullah Ercel ve Emin U Ulugergerli'nin "Öğrenen Makineler Ve Fasiyes Ayrımı; İlk Sonuçlar" başlıklı çalışması ise, yapay zeka ve makine öğrenmesi alanında yeni bir paradigma sunuyor.

Kuantum teknolojileri ve siber güvenlik üzerine genişletilmiş bakış açımız, bu sayının, siber güvenlik protokollerinin ve ağ altyapısının kuantum tehditlerine karşı nasıl yeniden tasarlanması gerektiğine dair değerli içgörüler sağlamasını amaçlamaktadır. Kuantum internetin gelişi ve post-kuantum kriptografinin önemi gibi konular, siber güvenlik uygulamalarında temel bir dönüşümü temsil etmektedir. Bu ilerlemeler, hem bilimsel ve teknolojik anlamda hem de toplumsal ve ekonomik anlamda büyük etkiler yaratmaya devam edecektir.

Yazarlarımızın titiz çalışmaları ve yenilikçi fikirleri, kuantum teknolojileri ve enformatik araştırmaları alanında önemli ilerlemeler kaydedilmesini sağlamaktadır. Bu sayıyı okuyucularımıza sunarken, araştırmaların bilgi birikimimizi artırmanın ötesinde, toplumumuzun karşı karşıya olduğu zorluklara çözümler sunma potansiyeline sahip olduğuna inanıyoruz. "Journal of Quantum Technologies and Informatics Research" olarak, bilimin sınırlarını genişletme yolculuğumuzda sizleri yanımızda görmekten büyük mutluluk duyuyoruz. Her bir makalenin, alanımızda yeni soruları teşvik etmesini ve gelecekteki araştırmalar için ilham kaynağı olmasını umuyoruz.

Doç. Dr. Engin Şahin

Editör

EDITORS/EDİTÖRLER

İletişim: Adres: Anafartalar Yerleşkesi,
B2 Blok, Oda No: 8, Çanakkale, Türkiye
Telefon: 5052423644
E-posta: jqtair@gmail.com
Web: https://journals.gen.tr/index.php/jqtair/

Hakemler

Prof. Dr. Yunus Levent Ekinci,
Bitlis Eren Üniversitesi

Prof. Dr. Safiye Ayşe Göker,
Çanakkale Onsekiz Mart Üniversitesi

Prof. Dr. Emin Ulugergerli,
Çanakkale Onsekiz Mart Üniversitesi

Doç. Dr. Engin Şahin,
Çanakkale Onsekiz Mart Üniversitesi

Doç. Dr. Uğur Ercan,
Akdeniz Üniversitesi

Doç. Dr Mustafa Arslan,
Bilgi Üniversitesi

Dr. Öğr. Üyesi Sevdanur Genç,
Kastamonu Üniversitesi

Öğr. Gör. Dr. Ercan Çağlar,
Çanakkale Onsekiz Mart Üniversitesi

"This page is left blank for typesetting"

Bu sayfa dizgiden dolayı boş bırakılmıştır

CONTENTS / İÇİNDEKİLER

"This page is left blank for typesetting"

Bu sayfa dizgiden dolayı boş bırakılmıştır

Journal of Quantum Technologies and Informatics Research
JQTAIR

KUANTUM TEKNOLOJİLERİ VE ENFORMATİK ARAŞTIRMALARI
Volume / Cilt: 2, Issue / Sayı: 1, 2024, pp. 1-13
E - ISSN: 3023-4735
URL: https://journals.gen.tr/index.php/jqtair/
DOI: https://doi.org/10.5281/zenodo.10102956

Received / Geliş: 22.01.2024
Acccepted / Kabul: 04.02.2024

RESEARCH ARTICLE / ARAŞTIRMA MAKALESİ

Kolektif Siber Güvenliğin Önemi ve Türk Devletler Teşkilatı

The Importance of Collective Cybersecurity and the Organization of Turkic States

Cengiz Çalıkoğlu

Bilgi Üniversitesi, Türkiye, e-mail: ccalikoglu@hotmail.com

Öz

Dijitalleşen dünyada siber güvenliğin önemi her geçen gün artmaktadır. Bu bağlamda ülkeler ve topluluklar, konuyla ilgili gerekli yatırımları yapmakta, önlemler almaya çalışmakta, ulusal ve uluslararası düzeyde siber saldırılarla mücadele etmektedir. Bu çalışmada siber güvenlik, siber saldırılar, ortak siber güvenlik yapıları ve son yıllarda dünyada gerçekleşen siber saldırılarla ilgili bilgiler sunulmuştur. Ayrıca, Türk Devletleri Teşkilatına üye ülkelerin "Kolektif Siber Savunma Gücü" adı ile ortak siber savunma gücü oluşturulması konusunda önerilere de yer verilmiştir. Bu çalışma, devletler, araştırmacılar, bilim insanları ve bu alana ilgi duyan kişiler için önemli bilgiler içermekte ve bundan sonra yapılacak çalışmalara kaynak oluşturmaktadır.

Anahtar Kelimeler: Siber Güvenlik, Siber Savunma, Siber Tehditler, Türk Devletleri Teşkilatı, Kolektif Siber Savunma Gücü, TDT- Kolektif Siber Savunma Gücü, TDT- KSSG

Abstract

In the digitalized world, the significance of cybersecurity is progressively increasing. In this context, countries and communities are making necessary investments, attempting to take precautions, and collaborating to combat both national and international cyberattacks. This paper provides information on cybersecurity, cyberattacks, collaborative cybersecurity frameworks, and recent cyberattacks worldwide. Additionally, recommendations are made regarding the establishment of a joint cyber defense force under the name "Collective Cyber Defense Force" for member countries of the Turkic Council. This study contains valuable information for governments, researchers, scholars, and individuals interested in this field, serving as a resource for future research endeavors.

Keywords: Cybersecurity, Cyber Defense, Cyber Attacks, Organization of Turkic States, Collective Cyber Defense Force, OTS- Collective Cyber Security Force, OTS- CCFS

Citation/Atıf: ÇALIKOĞLU, C. (2024). Kolektif Siber Güvenliğin Önemi ve Türk Devletler Teşkilatı. *Kuantum Teknolojileri ve Enformatik Araştırmaları*. 2(1): 1-13, DOI: 10.5281/zenodo.10102956

Corresponding Author/ Sorumlu Yazar:
Cengiz Çalıkoğlu
E-mail: ccalikoglu@hotmail.com

GİRİŞ

Günümüzde, hızla dijitalleşen dünya düzeninde siber güvenlik, ulusal ve uluslararası güvenliğin temel bir bileşeni olarak ön plana çıkmaktadır. Siber alan, sadece bireysel kullanıcılar için değil, aynı zamanda devletler ve çok uluslu örgütler için de stratejik bir öneme sahiptir. Türk Devletleri Teşkilatı (TDT) kendi üye devletlerinin siber güvenlik alanında karşılaştığı tehditlere karşı koyma ve bölgesel iş birliğini güçlendirme yönünde önemli adımlar atmaktadır. Bu çalışma, TDT üyesi ülkeler arasında siber güvenlik entegrasyonunun güçlendirilmesi için yenilikçi önerileri ele almakta ve böylece siber tehditlere karşı birlikte mücadele etme kapasitesinin artırılmasını hedeflemektedir. Siber güvenlik, sadece teknolojik bir konu olmanın ötesinde ekonomik, sosyal ve politik boyutlarıyla da derinlemesine incelenmesi gereken bir alandır. Bu bağlamda TDT›nin siber güvenlik stratejisi, bölgesel ve küresel siber tehditlerin doğası ve bu tehditlere karşı alınabilecek tedbirler üzerine yoğunlaşmaktadır. Dijitalleşen dünyada, siber güvenlik entegrasyonu, üye ülkeler arasındaki bilgi paylaşımını, ortak eğitim programlarını, ortak siber tatbikatları, teknolojik iş birliğini ve hukuki düzenlemeleri içermelidir. Bu temeller çerçevesinde ortak siber güvenlik yapısı oluşturulmasıyla birlikte, gelecekteki değişen teknolojilere derin teknoloji odağının olması kritiktir. Bu entegrasyon, aynı zamanda, TDT ülkelerinin siber alanda karşılaştıkları ortak tehditlere karşı koymada daha etkili ve koordineli bir yaklaşım geliştirmelerini sağlayabilir.

Ülkeler bir araya gelerek siber saldırılara karşı yeni iş birlikleri gerçekleştirmekte veya ülkeler tarafından kurulmuş Kuzey Atlantik Anlaşma Teşkilatı (The North Atlantic Treaty Organization-NATO), Avrupa Birliği (AB) gibi organizasyonlarda siber güvenlik bölümleri oluşturmuşlardır. Bu çalışmada önerilen yapı vasıtasıyla TDT üye ülkelerinin mevcut siber güvenlik yapıları, karşılaştıkları zorluklar ve fırsatlar detaylı bir şekilde analiz edilebilecektir. Ayrıca, siber güvenlik politikalarının oluşturulmasında ve uygulanmasında üye ülkeler arasında daha iyi bir uyum ve koordinasyon sağlamayı hedeflemektedir. Etkili bir siber güvenlik entegrasyonu için, teknik kapasite geliştirme, ortak risk değerlendirme yöntemleri ve acil durum müdahale ekiplerinin oluşturulması gibi unsurların üzerinde durulmuştur.

TDT ülkeleri arasında "Kolektif Siber Savunma Gücü" adı ile bir yapı oluşturulması önerilmiştir. Kolektif Siber Savunma Gücü siber güvenlik konusunda birlikte çalışarak, TDT ülkelerinin siber alanda daha dirençli ve hazırlıklı olmalarını sağlamayı amaçlamaktadır. Siber tehditlerin ulusal sınırları aşan doğası göz önünde bulundurulduğunda, bu tür bir entegrasyonun, bölgesel ve küresel düzeyde siber güvenliği artırıcı etkisi olacağı düşünülmektedir. Bu makale, TDT ülkelerinin siber güvenlik alanında karşılaştıkları mevcut zorlukları tartışacak ve yenilikçi bir entegrasyon modeli önererek, bu zorlukların üstesinden gelmelerine yardımcı olacak stratejiler sunmaktadır.

SİBER GÜVENLİK, SİBER SALDIRILAR VE JEOPOLİTİK OLAYLAR

Siber güvenlik, elektronik ağlar ve sistemler üzerinden bireyler ve kuruluşlar arasında paylaşılan hassas ve gizli verileri korumak için benimsenen bir dizi önlem ve uygulama (Türk Dil Kurumu, 2024) olarak ifade edilebilir. Bilgilerin güvenli bir şekilde iletilmesini ve depolanmasını sağlamanın yanı sıra verilere yetkisiz erişim, kullanım, değişiklik veya imhanın önlenmesini içerir. Siber güvenliğin birincil amacı veri, yazılım ve donanım gibi dijital varlıkların bütünlüğünü, kullanılabilirliğini ve gizliliğini potansiyel siber tehditlere ve saldırılara karşı korumaktır.

Siber saldırı ise bir veya birden fazla bilgisayar sistemlerine veya ağlarına yetkisiz erişim sağlamak için saldırganlar tarafından gerçekleştirilen bir dizi kötü niyetli faaliyet (Türk Dil Kurumu, 2024) olarak açıklanabilir. Bu saldırılar verilerin çalınması, değiştirilmesi veya imha edilmesi gibi farklı şekillerde olabilir. Saldırganlar, hedef sistemdeki açıklardan faydalanmak ve kendi çıkarları için kullanabilecekleri hassas bilgilere erişim sağlamak için çeşitli teknikler kullanırlar. Siber saldırılar bireyler, işletmeler ve hükümetler için ciddi bir tehdit oluşturmaktadır ve bunlara karşı korunmak için önleyici tedbirler almak çok

önemlidir.

Siber güvenlikle ilgili literatüre baktığımızda her geçen gün bu konuda yapılan akademik çalışmaların arttığı görülmektedir. Son zamanda yapılan çalışmalara (Gündüz & Daş, 2022; Nezgitli & Benzer, 2020; Karasoy & Babaoğlu, 2021; Kurnaz & Önen, 2019; Altın, 2023; Köker, 2022; Atakan, 2021; Kişman & Güleç, 2021; Eldem, 2021; Paltacı, 2022) (Ada & Çakır, 2017; Acar & Pekcandanoğlu, 2020; Göçoğlu & Aydın, 2019; Renda, 2022; Gündoğdu, 2023; Yılmaz, 2018; Dolma, 2023; Çıtak, 2021; Güntay, 2018; Ünal, Kanat, & Gürkaynak, 2023) örnek gösterilebilir.

Bu çalışmada, koşullar, durumlar ve özelliklerin ortaya koyulması, mevcut olayların daha önceki benzer olay ve koşullarla ilişkileri dikkate alınarak açıklanmasını hedefleyen (Kaptan, 1991) betimleme yöntemi kullanılmıştır.

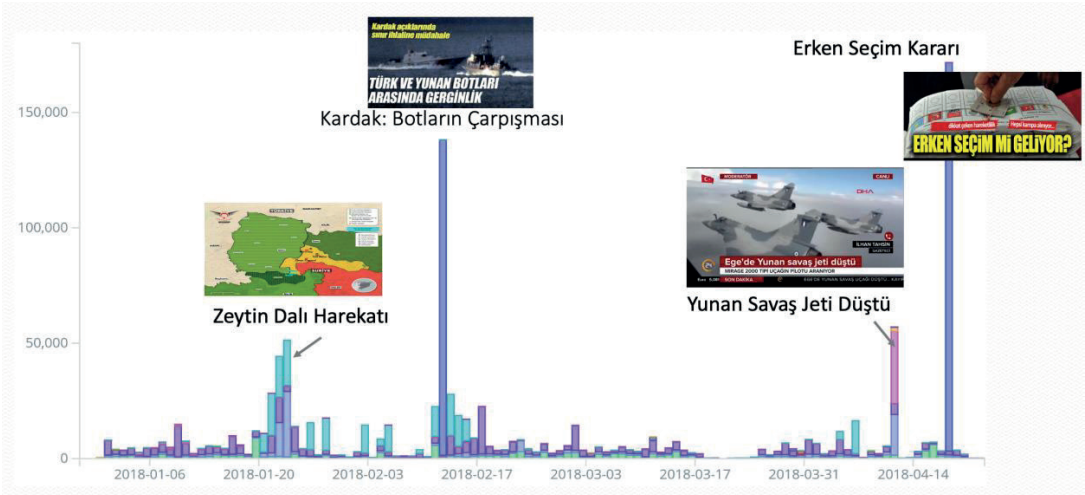
Siber saldırılar ülkelerin altyapılarını etkilemekte ve bu saldırılar genel olarak askeri, finansal ve kritik altyapılar olarak üç başlıkta ele alınabilir. Enerji üretim ve dağıtım, su ve gaz dağıtım altyapıları genel yapılara örnek gösterilebilirken, askeri komuta kontrol ve iletişim sistemler askeri yapılarak örnek gösterilebilir. Havalimanları, bankacılık ve ödeme sitemleri, Telekom ve iletişim altyapıları ise finansal altyapılara örnek gösterilebilir.

Jeopolitik olayların olduğu ülkelerdeki olaylar ile siber hareketlilikler arasında bir korelasyon olduğu söylenebilir. Siber saldırıların durumlarıyla ilgili bazı veriler Şekil 1-3 de sunulmuştur.

Jeopolitik olaylar ve siber saldırılar arasında bir korelasyon olduğu söylenebilir. Türkiye 20 Ocak 2018 tarihinde Zeytin Dalı Harekâtı (Milli Savunma Bakanlığı, 2018) gerçekleştirmiştir. Şekil 1. de görüldüğü üzere 6 Ocak 2018 tarihinde siber saldırılar 50.000'nin altındayken, Zeytin dalı harekatının başladığı 20 Ocak 2018 tarihinde siber hareketlilik günlük olarak 50.000'nin üzerine çıktığı görülmektedir. Türkiye Cumhurbaşkanı ve 57. dönem milletvekilliği seçimi 2018 yılında yapılmıştır (Yüksek Seçim Kurumu, 2019). Yine şekil 1 de görüldüğü üzere seçimler öncesinde Türkiye ile ilgili siber hareketlilik 50.000'nin altıdayken, seçimlerle ilgili haberlerle birlikte 14 Nisan 2018 tarihinde Türkiye ile ilgili siber hareketlilik günlük 150.000'nin üzerine çıkmıştır. Bu ve benzeri örneklerden yola çıkarak jeopolitik olayların başladığı zamanlarda siber hareketliliklerin arttığı görülmektedir.

Jeopolitik Olaylar ile Siber Hareketlilik arasında korelasyon bağı olduğunu gösteren ikinci bir örnek Şekil 2 de sunulmuştur. Türkiye'nin zeytin dalı operasyonuna devamı olarak, Afrin operasyonu süresince de siber hareketliliğin artarak devam ettiği görülmektedir. Suriye'de savaş ve operasyonlar sürecinde Amerika Birleşik Devletleri (ABD) başkanı Donalt Trump "hazır ol Rusya, füzelere gelecek" başlıklı bir açıklama yapmıştır (NTV , 2018). Bu açıklamayla birlikte şekil 2'de de görüldüğü üzere siber hareketliliğin artığı görülmektedir.

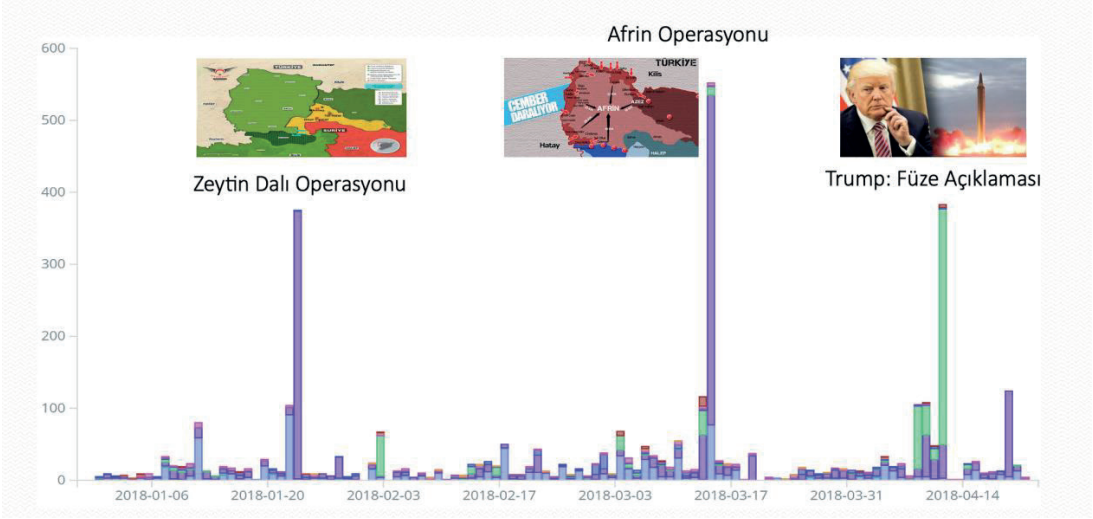
Şekil 1. Jeopolitik Olaylar ve Siber Hareketlilik *(Taş, 2017)*



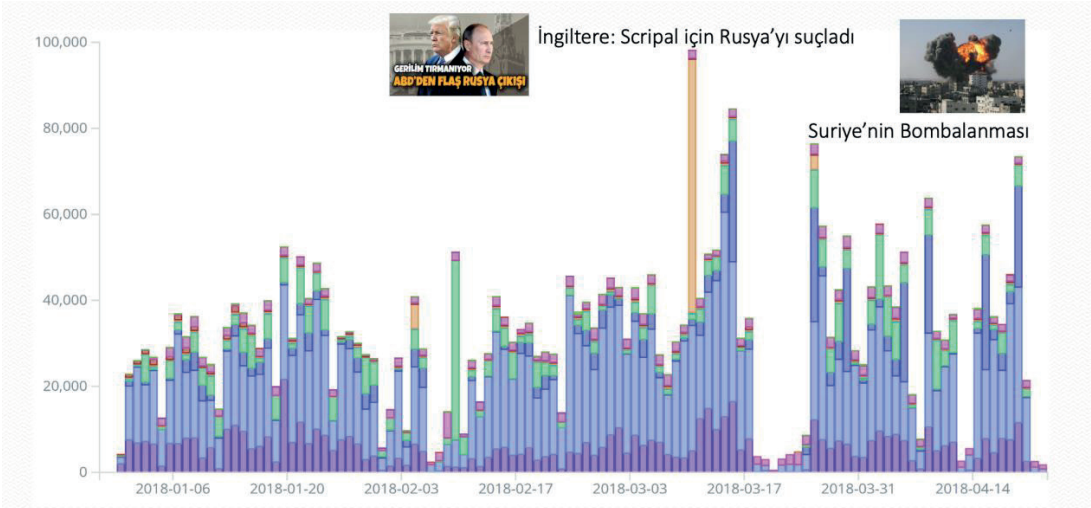
Jeopolitik olaylar ve siber hareketlilikle ilgili olarak son örnek ise Rusya ile İngiltere arasında casus krizi örnek verilebilir. 2018 yılında İngiltere ajan Sergey Skripal'ın Rus yapımı kimyayalar madde ile zehirlendiği iddia etmiştir (Anadolu Ajansı, 2018). Konuyla ilgili Şekil 3- 'te görüldüğü üzere 3 Mart 2018 tarihinde Rusya da siber hareketliliğin 100.000'in üzerine çıktığı görülmektedir. Tablo detaylı olarak incelendiğinde Ocak 2018 de en yüksek siber hareketlilik 55.000 civarındayken, ajan krizi ile birlikte Mart ayında yapılan açıklamalardan sonra siber hareketlilik % 82 artarak 100.000'in üzerine çıkmıştır.

Bu örneklerden yola çıkarak Siber saldırılar ile jeopolitik olaylar arasında karmaşık bir ilişki bulunduğu söylenebilir. Çağdaş dünya düzeninde, siber saldırılar sadece teknik birer ihlal değil, aynı zamanda devletler arası ilişkilerin bir parçası haline gelmiştir. Jeopolitik gerilimler, uluslararası sahnede meydana gelen siber saldırılarda artan bir rol oynamaktadır. Siber saldırıların potansiyel yıkıcılığı, geleneksel savaşlar kadar ciddi sonuçlar doğurmaktadır. Bu saldırılar, devletlerin kritik altyapılarına, askeri sistemlerine ve ekonomik yapılarına yönelik olarak gerçekleştiğinde, uluslararası arenada güç dengelerini etkilemektedir. Güçlü devletler, siber yeteneklerini ulusal güvenlik stratejilerinin bir parçası olarak kullanabilir. Bu devletler, casusluk, bilgi çalma ve diğer siber operasyonları yürüterek ulusal çıkarlarını koruma amacını taşıyabilir. Bazı durumlarda ise birden fazla devlet, ortak hareket ederek karmaşık ve

Şekil 2. Jeopolitik Olaylar ve Siber Hareketlilik *(Taş, 2017)*



Şekil 3. Jeopolitik Olaylar ve Siber Hareketlilik *(Taş, 2017)*



sofistike siber saldırılar düzenleyebilirler. Bu durum, uluslararası ilişkilerde yeni bir boyut kazandırarak, siber alanın giderek daha önemli bir stratejik unsura dönüşebilir.

SİBER GÜVENLİK VE İŞ BİRLİĞİ

Siber güvenlik kişilerin, firmaların, kamu kuruluşları ve diğer organizasyonların hassas verilerini, her türlü altyapılarını tehlikelere karşı korumak ve sürdürmek için önemlidir. Bu tehditlere karşı kişiler ve kurumlar tedbirleri almaya çalışırken yetersiz kalabilmektedir. Bu durumda ülke politikalarında ve stratejilerinde siber tehditlere karşı gerekli hukuksal ve teknolojik önlemler alınmaya çalışılmaktadır. Bu çalışmalar ülke bazında bazı durumlarda yeterli olmamakta, ülkeler arasında iş birliğini zorunlu kılmaktadır. Ayrıca, ülkelerin oluşturmuş NATO gibi ortak savunma güçleri içerisinde siber güvenlik konusunda bölümler oluşturulmakta ve iş birliği yapılmaktadır. Siyasi ve ekonomik örgütlenme örgütü olaraktan AB ülkeleri kendi aralarında siber güvenlik yapısıyla ilgili her türlü çalışmaları yapmakta ve gerekli önlemler konusunda iş birliğine gitmektedir. Bu çalışma kapsamında siber güvenlik ve iş birliği ile ilgili NATO ve AB'deki çalışmalara yer verilmiştir.

Avrupa Birliği Siber Güvenlik Ajansı (ENISA)

ENISA, 2004 yılında Yunanistan'ın başkenti Atina'da kurulmuştur. Bu oluşumun temel amacı, Avrupa'nın siber güvenlik kapasitesini güçlendirmektir. Bağımsız bir ajans olarak faaliyet gösteren ENISA, Avrupa Birliği üye devletleri arasında siber güvenlik iş birliğini artırmayı, kabiliyetleri güçlendirmeyi ve dirençli bir siber güvenlik ekosistemi oluşturmayı amaçlamaktadır. Saldırıya uğrayan ülkelere yardım sağlama, üye ülkelerin siber olaylara müdahale ekipleri ile iş birliği yapma ve ulusal kapasitelerin geliştirilmesine destek olma gibi görevlerle hareket eden ENISA, kolektif siber güvenliği artırmak adına bilgi paylaşımı, eğitim, iş birliği ve ortak tatbikatlar düzenlemektedir. Yönetim kurulu üyeleri her AB üye devletinden seçilirken, ajansın bağımsızlığını sağlamak ve görevlerini etkin bir şekilde yerine getirmek için İcra Direktörü tarafından yönetilmektedir. ENISA'nın kuruluşu, Avrupa genelindeki siber güvenlik endişelerine karşı bir yanıt olarak ortaya çıkmış olup, özellikle İngiltere, Almanya, Fransa gibi ülkelerde yaşanan ciddi siber güvenlik saldırılarından alınan derslerle şekillenmiştir (ENISA, 2024).

ENISA, AB ülkelerinde gerçek zamanlı olarak siber saldırı izlemeleri yapmamaktadır. Siber saldırıların anlık olarak izlenmesi ve yönetilmesi her bir AB üyesi devletinin sorumluluğundadır. Diğer taraftan saldırıya uğrayan bir AB üyesi talep etmesi durumunda destek vermektedir. Böylece kolektif siber güvenlik sağlanmakta ve iş birliği yapılmaktadır. ENISA'nın diğer önemli bir faaliyeti ise AB ülkeleri arasında bilgi paylaşımı ve olası siber saldırılarla ilgili tatbikatlar yapmaktır. Bu pratik uygulamalar olası saldırılara karşı ülkeleri hazır ve güvende tutmaktadır. ENISA, karasal olarak bir ordusu olmamasına rağmen, dijital olarak siber uzayda ortak ordu mantığıyla sahip olduğu söylenebilir

NATO Müşterek Siber Savunma Mükemmeliyet Merkezi (CCDCOE)

NATO, 4 Nisan 1949 tarihinde 12 ülke tarafından imzalanan Kuzey Atlantik Anlaşması ile kurulmuştur. 2024 yılı itibariyle bu ittifaka 31 ülke üyedir. NATO'nun amacı, insanları korumak, ortaklıkları geliştirmek, yeni tehditlerle savaşmak, barış ve istikrarı korumak olarak ifade edilebilir (NTV, 2023).

Dünyadaki en önemli ve kapsamlı oluşumlardan biri olan NATO siber güvenlik konusunda da gerekli çalışmaları yaparak Müşterek Siber Savunma Mükemmeliyet Merkezi'ni 14 Mayıs 2008 tarihinde kurmuştur. CCDCOE'nin amacı oldukça geniş kapsamlı ve stratejik bir rolü üstlenmektedir. 2008 yılında Estonya'nın başkenti Tallinn'de kurulan merkez, bilgi güvenliği konusunda NATO ülkelerini koruma ve savunma görevini üstlenmiştir. Merkezin yönetimine "Bilgi Güvenliği Baş Sorumlusu" başkanlık etmekte olup, NATO merkezlerini öncelikli olarak korumak, siber olaylara müdahale etmek ve saldırıları önlemek temel amaçlar arasında yer almaktadır. Ayrıca, üye ülkeler arasında bilgi paylaşımı, iş birliği ve ortak tatbikatlar düzenlemek de merkezin stratejik hedefleri arasında bulunmaktadır. Bu kapsamda, 2016 yılında Belçika'nın Mons şehrinde Siber Uzay Operasyon Merkezi'nin kurulması,

NATO'nun siber savunma kapasitelerini güçlendirmeye yönelik bir adım olarak öne çıkmaktadır. NATO CCDCOE, siber güvenlik alanında müttefikler arasında koordinasyonu ve dayanışmayı sağlayarak, siber tehditlere karşı etkili bir savunma mekanizması oluşturmayı amaçlamaktadır (CCDCOE, 2023).

2004 yılında NATO'ya katılan Estonya, bu süreçte ittifaka bir siber savunma merkezi kurulması önerisini ortaya atmıştır. Ancak, 2007 yılında Estonya'ya yönelik siyasi amaçlı ilk siber saldırılar, bölgesel ve uluslararası düzeyde büyük bir endişe yaratmış ve siber güvenlik konusundaki hassasiyeti artırmıştır. Bu olaylar, siber saldırıların sadece teknik bir mesele olmanın ötesinde, siyasi ve stratejik bir boyuta da sahip olduğunu göstermiştir (Dilek & Talih, 2023). Bu deneyimler, siber güvenlik konusundaki stratejik yaklaşımların geliştirilmesine ve güvenlik tedbirlerinin artırılmasına katkıda bulunmuştur.

CCDCOE tarafından yürütülen faaliyetler, müttefik ülkeler arasında siber güvenliği güçlendirmeyi hedefleyen önemli bir rolü yansıtmaktadır. Merkez, üye ülkelerin gerçek zamanlı izlenmesi sorumluluğunu taşımaz ve bu görevleri üye devletlere bırakır. Ancak, siber tehditlere karşı koordinasyonu sağlamak ve kolektif savunma çabalarını desteklemek amacıyla kolaylaştırıcı bir rol üstlenir. Merkez, üye devletlerin talepleri doğrultusunda yardım sağlama ve siber savunma kapasitelerini güçlendirmek üzere iş birliği yapma yeteneğine sahiptir. Bilgi paylaşımı, iş birliği ve ortak tatbikatlar düzenleme konusundaki faaliyetleri ile CCDCOE, müttefik ülkeler arasında siber güvenlik konusunda bilinçlenmeyi artırmak ve etkili bir kolektif savunma mekanizması oluşturmak adına önemli bir rol oynamaktadır (CCDCOE, 2023).

CCDOCOE ve ENISA amaç ve hedefleri olaraktan bir birilerine benzemekte ve dijitalleşen dünyada siber güvenlik konusunda örnek olabilecek çalışmalar, mevzuatlar, geleceğe yönelik olaraktan çalışmalar yaptığı görülmektedir.

TÜRK DEVLETLERİ TEŞKİLATI KOLEKTİF SİBER SAVUNMA GÜCÜ

Türk Devletleri Teşkilatı ve Siber Güvenlik

Türk Devletleri Teşkilatı, Türk Devletleri arasında kapsamlı iş birliğini teşvik etmek için uluslararası bir örgüt olarak 2009 yılında kurulmuştur. Teşkilatın kurucu üyeleri Azerbaycan, Kazakistan, Kırgızistan ve Türkiye`dir. Ekim 2019'da Bakü'de gerçekleştirilen 7. Zirve sırasında Özbekistan Teşkilata tam üye olarak katılmıştır. Macaristan ise Eylül 2018'de Kırgızistan'ın Cholpon-Ata şehrinde düzenlenen 6. Zirve sırasında, Türkmenistan Kasım 2021'de İstanbul'da düzenlenen 8. Zirvede, Kuzey Kıbrıs Türk Cumhuriyeti Semerkant'ta düzenlenen 9. Zirvede ve Ekonomik İş birliği Teşkilatı (EİT) 2023 yılında Astana'da düzenlenen 10. Zirve sırasında Teşkilat nezdinde gözlemci statüsü kazanmıştır. Nahcivan Anlaşmanın önsözünde üye devletler, Birleşmiş Milletler Anlaşması'nın amaç ve ilkelerine bağlılıklarını teyit ederek, TDT'nin genel amacını, Türk Devletleri arasında kapsamlı iş birliğini derinleştirmek, bölgesel ve küresel barış ile istikrara katkıda bulunmak olarak tanımlamışlardır. Üye ülkeler ayrıca, demokrasi, insan haklarına saygı, hukukun üstünlüğü ve iyi yönetişim gibi temel ilkelere bağlılıklarını ifade etmişlerdir. TDT kapsamındaki iş birliği, üye ülkeler arasındaki ortak tarih, kültür, kimlik ve Türk dili konuşan halkların dil birliğinden kaynaklanan özel dayanışma temelinde inşa edilmektedir (Türk Devletleri Teşkilatı, 2024). 2024 Ocak ayı itibariyle TDT asıl ve gözlemci üye ülkeler Şekil 4'te sunulmuştur.

Dijitalleşen dünyamızda siber güvenlik tanım, kavram ve konuya bakış açıları kökten değişmiş, siber tehdit ve saldırılar bireysellikten devletler arası düzeye çıkmıştır. Uzaydaki siber güç dengeleri, güvenlik stratejilerini ve ulusal kabiliyetlerin geliştirilmesini zorunlu kılmaktadır. Günümüzde ülkelerin fiziksel sınırlarını korumak kadar, ülkenin verisini ve dijital altyapısını korumak kaçınılmaz olmuştur. Siber güvenlik, devlet güvenliğiyle eşdeğer öneme sahip bir konu haline gelmiştir.

Bu çalışmanın ilgili bölümünde dünyada yaşananlar, jeopolitik olaylarla, siber olaylar arasında korelasyon olduğu verilerle detaylı

bir şekilde açıklanmıştır. Dünyada ortak siber güvenlik yapıları olarak ENISA ve CCDCOE'nin ortak siber güvenlik yapıları detaylı incelenmiştir. Bu bölümde de belirtildiği üzere AB'de de NATO'da da bazı ülkeler siber güvenlik saldırılarına maruz kalmış ve bu saldırılar ENISA ve CCDCOE kuruluşunu hızlandırmıştır. Bu iki organizasyonun ortak siber güvenlik yapılarında ortak amaç üye devletleri arasında siber güvenlik iş birliğini artırmayı, kabiliyetleri güçlendirmeyi ve dirençli bir siber güvenlik ekosistemi oluşturmayı amaçlamışlardır. Yine saldırıya uğrayan ülkelere yardım sağlama, üye ülkelerin siber olaylara müdahale ekipleri ile iş birliği yapma ve ulusal kapasitelerin geliştirilmesine destek olma gibi görevlerle hareket etmekteler ve kolektif siber güvenliği artırmak adına bilgi paylaşımı, eğitim, iş birliği ve ortak tatbikatlar düzenlemektedirler.

Dünyada artan siber hareketlilik konusunda Türkiye'ye yönelik olaraktan önemli siber olayların yaşandığı görülmüştür. Bu saldırılardan banklalar, Telekomünikasyon şirketleri, sektör bazı şirketler etkilenmiştir (TRT Haber, 2019), (BBC News Türkçe, 2015). Benzer şekilde TDT üye ülkelere yönelik siber olayların olduğu geçmiş yıllarda gözlenmiştir. TDT üye ülkelere yönelik birçok siber saldırılırsa bilinse de uluslararası yayınlara düşen şu örnekleri verebiliriz (The Hacker News, 2012; The Hacker News, 2012; The Hacker News, 2023; The Hacker News, 2012; The Hacker News, 2012; The Hacker News, 2023; The Hacker News, 2022; The Hacker News, 2019).

Tüm bu gelişmeleri dikkate aldığımızda, dijitalleşen ve kutuplaşan dünyada bireysellikten devletler arası düzeye çıkan siber güvenlikte Türk Devletleri Teşkilatı üye devletler olarak ortak hareket etmek bir zorunluluk ve çok kritik bir stratejik iş birliği gereksinimi ortaya çıkmıştır. Ayrıca, TDT üye devletlerine genel olarak bakıldığında bazı devletlerin teknolojik altyapı, insan kaynağı, mevzuat ve siber güvenlik altyapılarının iyileştirilmesinin kaçınılmaz olduğu söylenebilir. Türk dünyasının büyük düşünürü İsmail Gaspıralı bundan yaklaşık 110

Şekil 4. TDT üye ve gözlemci devletler

yıl önce «Dilde, işte, fikirde birlik» sözleriyle tüm Türk halklarını birlik ve dayanışmaya çağırmıştır. Türk topluluklarının gelişmesi için yol çizen, basın ve eğitim çalışmalarıyla iz bırakan Kırım Tatar Türk'ü İsmail Gaspıralı fikirleriyle Türk dünyasını etkilemeye devam etmektedir (Anadolu Ajansı, 2020).

Geçen 110 yılda dünyamızda çok büyük değişimler olmuş, sanayi devrimi, gelişen teknolojiler, Endüstri 4.0 ile dijitalleşen altyapıları, siber uzaydaki gelişmeler ve önümüzdeki yıllar yapay zekâ destekli siber saldırıları düşündüğümüzde "teknoloji ve siber güvenlik" her şeyin merkezinde olacağı söylenebilir. Bu bakışla Türk düşünürü İsmail Gaspıralı'nın sözüne atıfla "Dilde, fikirde, işte, **teknolojide** ve **güvenlikte** birlik" için iş birliği ve ortak çalışmalar yapılabilir. "Birlikte daha güvendeyiz" bakışıyla bu yapının kurulmasının Türk dünyası için tarihi bir sorumluluk olduğunu düşünülebilir. Bu kapsamda TDT üye devletleri için «TDT Kolektif Siber Güvenlik Gücü» kurulması önerilmektedir.

TDT Kolektif Siber Güvenlik Gücünün Gerekliliği

TDT üye ülkeler arasında ulusal güvenliğin ve kritik yapıların korunması, ulusal siber güvenlik olgunluk seviyesinin güçlendirilmesi, ulusal güvenlik politika ve stratejilerinin desteklenmesi, devletler, kurumlar ve sektörler arasında köprüler kurulması, uluslararası iş birliğinin teşvik edilmesi ve kapasite geliştirme konusunda iş birliği yapılması kaçınılmazdır.

TDT üye devletleri arasında siber güvenlik iş birliğini artırmak, kabiliyetleri güçlendirmek, güçlü bir siber güvenlik ekosistemi oluşturmak önem arz etmektedir. TDT üye ülkelerimizden birine olası siber saldırı olduğunda yardım sağlamak, üye ülkelerin siber olaylara müdahale ekipleri ile iş birliği yapmak ve ulusal kapasitelerin geliştirilmesine destek olmanın yanında kolektif siber güvenliği artırmak adına bilgi paylaşımı, eğitim, iş birliği ve ortak tatbikatlar düzenlenmesi ülkelerimizi siber saldırılarının korumanın yanında daha güvende hissettirecektir.

TDT Kolektif Siber Güvenlik Gücünün Amacı ve Hedefi

İstihbarat paylaşımı, kolektif hareket, önleme, zarar toplama, etkin caydırmayı amaçlayan TDT Kolektif Siber Güvenlik Gücünün hedeflerini kısaca aşağıda özetlenebilir:

- Gizliliği korumak ve teşvik etmek,
- Kapasite geliştirme konusunda iş birliği yapmak,
- Uluslararası kuruluşlar ve kilit oyuncular/ ortaklar ile stratejik ortaklıklar kurmak,
- Ulusal siber güvenlik olgunluk seviyesinin güçlendirilmesi için ilgili ulusal politikaların, stratejilerin oluşturulmasını desteklemek,
- Kamu sektörü, özel sektör, akademik kurumlar arasında köprüler kurmak,
- Belirli siber istişareler için ikili veya ortak komisyonlar oluşturmak,
- Eğitim & ARGE faaliyetleri ile sürekli iyileştirmelerde bulunmak,
- Ekosisteme özgü zorlukları periyodik olarak ele almak,
- En iyi uygulamaları geliştirmek ve teşvik etmek.

TDT Kolektif Siber Güvenlik Gücünün Organizasyon Yapısı:

Bir oluşumun başarılı olabilmesi için; planlama, örgütlenme, yönetme, denetim gibi unsurları yürüten yönetim organizasyon yapısının oluşturulması son derece önemlidir. Bu kapsamda TDT Kolektif Siber Güvenlik Gücünün ilk oluşumu aşamasında yönetim komitesi ve alt komiteler Şekil 5 ve 6 da sunulmuştur.

Şekil 5'te görüldüğü üzere TDT Kolektif Siber Güvenlik Gücü iş sürekliliği ve kriz yönetim, yasal ve uyum, teknoloji ve inovasyon, bilgi güvenliği ve yönetim komitesi olmak üzere dört komiteden oluşması önerilmektedir. Gelişen teknolojiler ve günün ihtiyaçlarına göre yönetim komitesinin sayısı artırılabilir.

Şekil 6'te görüldüğü üzere TDT Kolektif Siber Güvenlik Gücü, İş Sürekliliği ve Kriz Yönetimi, Program Yönetişimi, Operasyonel Model Tasarımı, Eğitim & ARGE, Denetim & Risk & Yasal ve Uyum, Siber Güvenlik Yönetimi Ekibi olmak üzere altı ekipten oluşması önerilmektedir.

TDT Kolektif Siber Güvenlik Gücünün Faaliyet Alanları

TDT Kolektif Siber Güvenlik Gücü, kritik altyapının korunması, ulusal güvenliğin korunması, vatandaşların mahremiyetinin ve verilerinin korunması, ekonomik büyümenin desteklenmesi, uluslararası iş birliğinin teşvik edilmesi gibi konularda faaliyet gösterebilir.

▪ *Kritik altyapının korunması:* Siber güvenlik tehditleri elektrik şebekeleri, ulaşım sistemleri ve finans kurumları gibi kritik altyapılar için önemli bir risk oluşturmaktadır. Kritik altyapıya yönelik bir siber saldırı, temel hizmetleri aksatarak ve yaygın ekonomik hasara yol açarak yıkıcı sonuçlar doğurabilir. Ülkeler ve ittifak güçleri siber güvenlik yapıları kurarak kritik altyapılarını siber saldırılardan koruyabilir ve temel hizmetlerin devamlılığını sağlayabilirler.

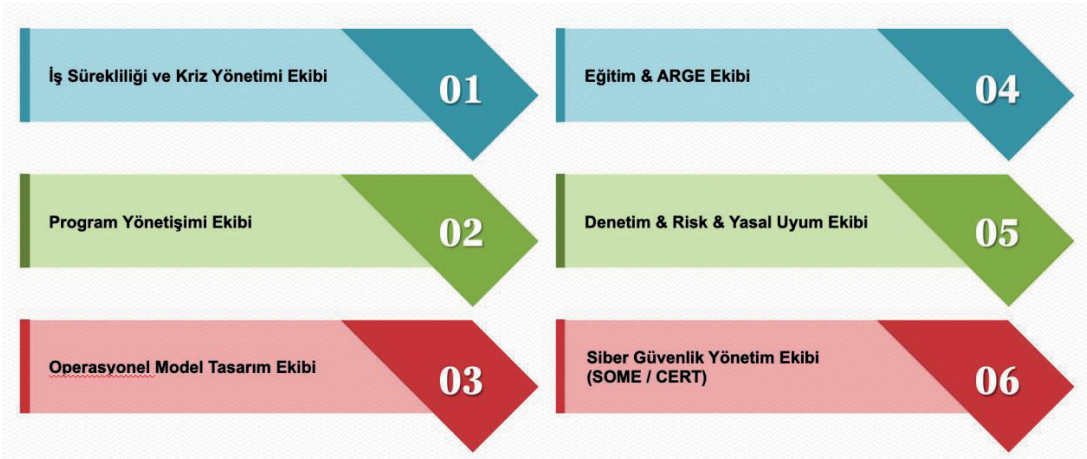
▪ *Ulusal güvenliğin korunması:* Siber saldırılar hassas bilgileri çalmak, askeri operasyonları sekteye uğratmak ve hatta halk arasında nifak tohumları ekmek için kullanılabilir. Ülkeler ve ittifak güçleri siber güvenlik yapıları kurarak ulusal güvenliklerini siber saldırılara karşı koruyabilir ve düşmanlarına karşı stratejik bir avantaj sağlayabilirler.

▪ *Vatandaşların mahremiyetinin ve verilerinin korunması:* Siber suçlular, kişisel bilgileri ve finansal verileri çalmak için bireyleri giderek daha fazla hedef almaktadır. Ülkeler ve ittifak güçleri siber güvenlik yapıları kurarak vatandaşlarının gizliliğini ve verilerini siber saldırılardan koruyabilir ve vatandaşlarının çevrimiçi hizmetleri kullanırken kendilerini güvende hissetmelerini sağlayabilir.

▪ *Ekonomik büyümenin desteklenmesi:* Siber güvenlik günümüzün dijital ekonomisinde ekonomik büyüme için vazgeçilmezdir. Her büyüklükteki işletme faaliyetlerini sürdürmek için internete güvenmektedir. Siber saldırılar

Şekil 5. TDT Kolektif Siber Güvenlik Gücü Yönetim Komitesi

Şekil 6. Kolektif Siber Güvenlik Gücü Çalışma Ekipleri



şirketlerin operasyonları aksatabilir, itibara zarar verebilir ve mali kayıplara yol açabilir. Ülkeler ve ittifak güçleri siber güvenlik yapıları kurarak işletmelerin gelişmesi için daha güvenli ve istikrarlı bir ortam yaratabilir. Bu durum ekonomik büyüme ve istihdam yaratılmasına katkı sağlar.

▪ *Uluslararası iş birliğinin teşvik edilmesi:* Siber güvenlik, etkin bir şekilde ele alınması için uluslararası iş birliği gerektiren küresel bir sorundur. Ülkeler ve ittifak güçleri, siber güvenlik yapıları kurarak bilgi paylaşmak, tehdit istihbaratı geliştirmek ve siber saldırılara karşı müdahalelerini koordine etmek için birlikte çalışabilirler. Bu iş birliği genel siber saldırı riskini azaltmaya ve küresel toplumu daha güvenli hale getirmeye yardımcı olabilir.

TDT Kolektif Siber Güvenlik Gücünün Kritik Başarı Faktörleri

TDT Kolektif Siber Güvenlik Gücünün başarı kriterlerini yönetim, mevzuat ve finansal olmak üzere üç başlıkta incelenebilir.

▪ *Yönetim Yapısı:* Önerilen yapının TDT çatısı altında kurulması ve projeye tam destek verilmesi, siber güvenlik stratejisinin başarılı bir şekilde uygulanması açısından kritik bir rol oynar. Bu destek, projenin sürdürülebilirliğini sağlamak, hedeflere ulaşmak ve ulusal siber güvenlik kapasitesini güçlendirmek adına önemlidir. Sahiplik olmadan, her bir kurumun kendi önceliklerine odaklanması, projenin bütünlüğünü ve etkisini azaltabilir. Önerilen yapının başkanının, belirlenen komitelerden seçilen uzman bir ekip olması, projenin yönetiminde etkinlik ve odaklanma sağlayacaktır. Özellikle Yönetim Komitesinde, bu yapının baş yöneticisinin ve her ülkeyi temsilen sabit üyelerin olması önem arz etmektedir. Yine kurucu ekibin hedeflenen kurulum süresi boyunca sabit kalması projenin başarısı açısında önemli olacaktır. Bu komitelerde yer alacak üyelerin tek görevlerinin siber güvenlik işine odaklanması, uzmanlık ve derinlemesine bilgi sağlayarak projenin başarısına katkıda bulunacaktır. Bu sayede, hızla evrilen siber tehditlere karşı etkili bir mücadele stratejisi oluşturmak mümkün olacaktır. Projenin başarılı olabilmesi için sadık siber güvenlik uzmanlarının yetiştirilmesi ve korunması önemlidir. Bu uzmanlar, projenin teknik gereksinimlerini karşılamak ve güvenlik önlemlerini güncel tutmak adına kritik bir rol oynarlar. Ayrıca, personel sirkülasyonlarına karşı alternatif çözümler belirlenmesi, bilgi birikiminin korunmasını sağlayarak projenin uzun vadeli etkinliğini artırabilir.

▪ *Ülke Mevzuatları:* Önerilen yapının başarılı bir şekilde kurulabilmesi için, ülkeler arasındaki siber güvenlik mevzuatlarının uyumlu hale getirilmesi kritik bir öneme sahiptir. Bu uyum, siber güvenlikle ilgili iş birliğini güçlendirmek, veri koruma standartlarını belirlemek ve projenin etkili bir şekilde yönetilmesini sağlamak için gereklidir. Ayrıca, genel veri koruma yönetmeliği (General Data Protection Regulation- GDPR) gibi mevzuatlar göz önünde bulundurularak toplanacak veriye dair eksiksiz ve hatasız bir mevzuat oluşturulmalıdır. Bu mevzuat, kullanıcıların gizliliğini koruma, veri güvenliğini sağlama ve siber saldırılara karşı koruma sağlamak üzere tasarlanmalıdır. Önerilen yapının amacına uygun olarak işlevselliğini yerine getirebilmesi için, ülkeler arasında siber istihbarat amaçlı veri paylaşımı büyük bir önem taşır. Bu, hızlı ve etkili bir şekilde siber tehditlere karşı mücadele edebilmek adına kritiktir. Bu bağlamda, ülke ülke siber istihbarat verilerinin paylaşılabilmesi için mevzuat düzenlemelerinin yapılması ve değişikliklerin devlet nezdinde desteklenmesi gerekmektedir. Veri paylaşımını destekleyen bir mevzuat çerçevesi, siber güvenlikle ilgili bilgilerin güvenli bir şekilde paylaşılmasını sağlayarak projenin etkinliğini artırabilir ve ülkeler arası iş birliğini güçlendirebilir. Bu noktada, siber güvenlik mevzuatının oluşturulması ve düzenlenmesi sürecinde paydaşların, uzmanların ve sivil toplum kuruluşlarının katılımı önemlidir. Şeffaf, güçlü ve katılımcı bir süreç, mevzuatın etkili bir şekilde uygulanmasını ve siber güvenlik alanında başarılı bir yapı oluşturulmasını destekleyecektir.

▪ *Bütçe Oluşturulması:* Önerilen yapının kurulması için, gerekli bileşenlerin alınması, ortamların kurulması, yönetilmesi, maliyetlerin karşılanması, kaynakların temin edilmesi ve belirlenen şartların yerine getirilmesi için yatırım mekanizmasının oluşturulması gerekmektedir.

SONUÇ VE DEĞERLENDİRME

TDT Kolektif Siber Güvenlik Gücünün kuruluşu, günümüzde karşılaşılan karmaşık ve artan siber tehditlerle başa çıkabilmek ve ulusal güvenliği sağlamak adına ülkeler arasında güçlü bir iş birliği ve koordinasyonun gerekliliğini vurgulamaktadır. Bu ortak yapı, siber tehditlere etkin bir şekilde karşı koymayı amaçlayarak bilgi paylaşımını artırır ve ortak savunma stratejileri geliştirir. Koordineli bir şekilde hareket etmek, siber saldırılara hızlı ve etkili yanıtlar verme kapasitesini artırır. Bu yapı, ülkelerin siber güvenlik kapasitelerini güçlendirerek, ulusal sistemlerini daha dirençli hale getirmeyi hedefler.

TDT Kolektif Siber Güvenlik Gücünün kurulmasıyla birlikte, üye ülkeler arasında siber güvenlik konusunda iş birliği ve koordinasyonun güçlenmesi, bölgesel ve küresel düzeyde güvenliği artırır. Ayrıca, ortak bir siber güvenlik politikası oluşturarak uyum ve standartlar bütünlüğünü sağlar. Bilgi ve deneyim paylaşımının artması, ülkelerin siber tehditlere karşı daha hazır ve bilgili olmalarını sağlar. Ortak tehditlere karşı kolektif önlemler alabilme yeteneği, üye ülkelerin daha etkili bir şekilde güvenliklerini sağlamalarına olanak tanır. Bu yapı aynı zamanda dijital ekonominin gelişimine destek olacak standartların belirlenmesine katkı sağlar, bireylerin ve kurumların dijital verilerinin güvenliği ve gizliliği korunur.

Uluslararası Siber Güvenlik Yapılarına katılmamanın beraberinde getirdiği riskler, modern dünyanın karmaşık siber tehdit ortamında bir ülkeyi önemli zorluklarla karşı karşıya bırakabilir. Savunma zayıflığı, ulusal bilgi ve veri güvenliğini tehdit edebilir, ekonomik ilişkileri ve dış politika stratejilerini olumsuz etkileyebilir. Ayrıca, gelişen siber tehditlere hazırlıksız olma olasılığı artar ve ülkeyi beklenmedik siber saldırılara karşı savunmasız bırakabilir. Bu bağlamda, uluslararası siber güvenlik yapılarına katılmak, bir ülkenin siber güvenlik kapasitesini artırarak, küresel düzeyde daha güvenli ve dirençli bir dijital ortam oluşturmasına katkı sağlayabilir.

TDT Kolektif Siber Güvenlik Gücünün kuruluşu, kutuplaşan dünya düzeni, artan siber saldırılar, uzaydaki değişen siber güç dengeleri gibi faktörlerle şekillenen modern dünyanın gerçeklerine uygun olarak ortak bir siber güvenlik teşkilatı kurma ihtiyacını vurgular. Bu oluşum, üye ülkeler arasında güç birliği oluşturarak siber tehditlere karşı etkili bir savunma sağlamayı amaçlar. Aynı zamanda, bireysellikten devletler arası düzeye çıkan siber tehditlerle başa çıkabilmek adına ulusal güvenlik stratejilerini ve kabiliyetlerini revize etmeyi hedefler.

TDT Kolektif Siber Güvenlik Gücünün kuruluşuyla birlikte üye ülkeler, ortak bir siber güvenlik teşkilatının avantajlarından faydalanarak siber tehditlere etkili bir mücadele verme kapasitesini artırır. Bu çaba, ülkeler arasında güç birliği, caydırıcılık, siber tehditlere ortak müdahale, istihbarat paylaşımı, kolektif hareket, önleme ve toparlama stratejilerinin birlikte oluşturulması gibi hedefleri içerir. Bu sayede, ülkelerin verileri ve dijital altyapıları daha güvenli bir şekilde korunurken, bölgesel dayanışma ve siber güvenlikte liderlik güçlenir.

TDT Kolektif Siber Güvenlik Gücünün kuruluşu, üye ülkeler için sağlayacağı katma değerli hizmetlerle dijital güvenliğin güçlendirilmesine yönelik kapsamlı bir strateji sunar. Ortak siber savunma gücü oluşturulması, TDT üye ülkelerini siber tehditlere karşı güç birliği içinde ortak müdahale etme imkanıyla donatır. Bu yapı, siber saldırılardan korunmak adına kapsamlı bir ekosistem oluşturulmasına katkı sağlar. Aynı zamanda, olası finansal kayıpların engellenmesi amacıyla zamanında siber saldırıları fark etme ve önleme yetenekleri güçlendirilir.

Bilgi ve altyapı eşitlenmesi, TDT üye ülkelerinin kendi içlerindeki siber güvenlik açıklarını merkezi bir düzeyde koruma altına almasını sağlar. Bu sayede, ülkeler siber saldırı sonrasında en az hasarla toparlanma yeteneklerini artırır ve dijital güvenliklerini daha etkin bir şekilde yönetir. Ayrıca, teknoloji ve güvenlik altyapılarında standardizasyon sağlanarak ülkeler arasında siber güvenlikte bir uyum ve iş birliği ortamı oluşturulur.

TDT Kolektif Siber Güvenlik Gücünün kuruluşu, sadece TDT üye ülkelerine değil, tüm dünyada siber saldırılara karşı etkili önlemler

alınmasına katkı sağlayacak ve dijital güvenliği güçlendirecektir. Bu yapı, bölgesel güvenliği artırmanın yanı sıra benzer siber güvenlik oluşumlarına örnek teşkil edebilir. Bölgesel güvenlik için sağladığı avantajlar, TDT'nin dijital dünyadaki liderliğini pekiştirir ve küresel düzeyde siber güvenliği artırmaya yönelik bir çabanın önemli bir parçası olur.

Yukarıda belirtilen unsurların hepsi bir araya geldiğinde, TDT Kolektif Siber Güvenlik Gücünün kurulması, üye ülkelerin siber güvenlik kapasitelerini güçlendirerek, siber tehditlere karşı daha etkin bir savunma mekanizması oluşturmayı amaçlamaktadır. Bu çabaların sonucunda, dijital dünya daha güvenli bir geleceğe doğru adım atmayı hedeflemekte ve ulusal güvenliği güçlendirmektedir.

Kaynakça

Çıtak, E. (2021). Siber Terörizm: Potansiyelin Gerçekçi Tehdidi. *Turkuaz Uluslararası Sosyo-Ekonomik Stratejik Araştırmalar Dergisi*, , Cilt 3, Sayı 1, Sayfalar 1 - 16.

Acar, H., & Pekcandanoğlu, M. (2020). Rusya'nın Siber Güvenlik ve Siber Espiyonaj Politikalarının Analizi. *Türkiye Rusya Araştırmaları Dergisi*, , Sayı 3, Sayfalar 167 - 189.

Ada, M., & Çakır, H. (2017). Kuzey Atlantik Antlaşma Örgütü'nün (NATO) Siber Güvenlik Stratejisinin İncelenmesi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, Cilt 5, Sayı 2, Sayfalar 632 - 656.

Ajansı, A. (2020, 09 24). *Türk dünyasının büyük düşünce adamı: Gaspıralı İsmail*. Ocak 2024 tarihinde https://www.aa.com.tr/tr/portre/turk-dunyasinin-buyuk-dusunce-adami-gaspirali-ismail/1982859 adresinden alındı

Altın, O. (2023). AB'nin Siber Güvenlik Alanındaki Politikalarının ve Uygulamalarının Etkinliği: Bir Siber Güvenlik Temsilcisi Olarak AB'nin Yeterliliği. *Çankırı Karatekin Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*,, Cilt 13, Sayı 2, Sayfalar 482 - 507.

Anadolu Ajansı. (2018, Mart 14). *İngiltere ilke Rusya arasında casus krici tırmanıyor*. Aralık 2023 tarihinde https://www.aa.com.tr/tr/dunya/ingiltere-ile-rusya-arasinda-casus-krizi-tirmaniyor-/1088291 adresinden alındı

Anadolu Ajansı. (2020, 09 23). *Türk dünyasının büyük düşünce adamı: Gaspıralı İsmail*. Ocak 2024 tarihinde https://www.aa.com.tr/tr/portre/turk-dunyasinin-buyuk-dusunce-adami-gaspirali-ismail/1982859 adresinden alındı

Atakan, M. (2021). Siber Güvenlik Ve Covid 19 Salgının Uzaktan Denetim Üzerinde Etkileri. *Denetişim*, , Cilt 0, Sayı 22, Sayfalar 27 - 39.

BBC News Türkçe. (2015, 12 24). Ocak 2024 tarihinde Türkiye'ye siber saldırının 10 günü: Ne oldu?: https://www.bbc.com/turkce/haberler/2015/12/151224_siber_saldiri_arslan adresinden alındı

CCDCOE. (2023). *About us*. Ocak 2024 tarihinde https://ccdcoe.org/about-us/ adresinden alındı

Dilek, E., & Talih, Ö. (2023). Estonya 2007 siber saldırılarının incelenmesi ve ülkelerin ulusal siber güvenlik politikalarına etkileri. *Bilgi yönetimi dergisi, 6*(2), 332 - 347.

Dolma, Ö. (2023). Siber Güvenlik İhbarcılarının Korunması Açısından ABD ve AB Yaklaşımlarının Karşılaştırılması. *Pamukkale Üniversitesi İşletme Araştırmaları Dergisi*,, Cilt 10, Sayı 2, Sayfalar 615 - 631.

Eldem, T. (2021). Uluslararası Siber Güvenlik Normları ve Sorumlu Siber Egemenlik. *İstanbul Hukuk Mecmuası*, Cilt 79, Sayı 1, Sayfalar 345 - 376.

ENISA. (2024). *Stracture and organization*. Ocak 2024 tarihinde https://www.enisa.europa.eu/about-enisa/structure-organization adresinden alındı

Göçoğlu, V., & Aydın, M. D. (2019). Siber Güvenlik Politikası: Abd, Rusya Ve Çin Üzerine Karşılaştırmalı Bir Analiz. *Güvenlik Bilimleri Dergisi*, , Cilt 8, Sayı 2, Sayfalar 229 - 252.

Gündüz, M. Z., & Daş, R. (2022). Kişisel Siber Güvenlik Yaklaşımlarının Değerlendirilmesi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*,, *13*(3), 429-438.

Gündoğdu, S. (2023). Uluslararası Politikada Bir Etki Aracı Olarak Siber Güvenlik Ve Türkiye'nin Siber Güvenlik Politikası Uygulaması: Ulusal Siber Olaylara Müdahale Merkezi (Usom). *Fırat Üniversitesi Sosyal Bilimler Dergisi*,, Cilt 33, Sayı 3, Sayfalar 1325 - 1337.

Güngöe, U., & Güney, O. (2017). Uluslararası İlişkilerde Güvenliğin Dönüşümü Çerçevesinde Bilgi Güvenliği Ve Siber Savaş. *Karadeniz Araştırmaları*, Cilt 14, Sayı 55, Sayfalar 131 - 146.

Güntay, V. (2018). Siber Güvenliğin Uluslararası Politikada Etki Aracına Dönüşmesi ve Uluslararası Aktörler. *Güvenlik Stratejileri Dergisi*, Cilt 14, Sayı 27, Sayfalar 79 - 111.

Ünal, E., Kanat, S., & Gürkaynak, M. (2023). Hibrit Tehditler Ve Avrupa Birliği . *Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, Cilt 1, Sayı 45, Sayfalar 391 - 412.

Köker, A. E. (2022). Avrupa Birliği'nin Gelişen ve

Değişen Tehdit Algısı: Siber Güvenlik. *EURO Politika,* , Sayı 14, Sayfalar 48 - 77.

Köksal, F. (2020). Avrupa Birliği'nin Siber Güvenlik Politikası: Kurumsalcılık mı Tutarlılık mı? *Güvenlik Stratejileri Dergisi*, Cilt 16, Sayı 35, Sayfalar 635 - 674.

Kaptan, S. (1991). *Bilimsel araştırma ve istatistik teknikleri.* Ankara: Tekışık Web Ofset Tesisleri.

Karasoy, H. A., & Babaoğlu, P. (2021). Türkiye'de Siber Güvenlik: Yasal Ve Kurumsal Altyapı. *Yasama Dergisi*, Sayı 44, Saylar 123-155.

Kişman, Z. A., & Güleç, Ö. (2021). Uluslararası İlişkiler Açısından Siber Güvenlik ve NATO'nun Siber Güvenlik Stratejileri. *Akademik Açı*, Cilt 1, Sayı 1, Sayfalar 127 - 154.

Kurnaz, S., & Önen, S. (2019). Avrupa Birliğine Uyum Sürecinde Türkiye'nin Siber Güvenlik Stratejileri. *International Journal of Politics and Security*, Cilt 1, Sayı 2, Sayfalar 82-13.

Milli Savunma Bakanlığı. (2018). *Zeytin dalı harekatı*. Aralık 2023 tarihinde https://www.msb.gov.tr/ZeytinDaliHarekati adresinden alındı

Nezgitli, S., & Benzer, R. (2020). Avrupa Birliği Siber Güvenlik Kanunu. *Journal of Information Systems and Management Research*, Cilt 2, Sayı 1.

NTV . (2018, Nisan 11). *Turmp:Hazır ol Rusya, füzelere gelecek*. Aralık 2023 tarihinde https://www.ntv.com.tr/dunya/trump-hazir-ol-rusya-fuzeler-gelecek,IiARdXk_xEq7aq0I8tgplw adresinden alındı

NTV. (2023, 4 4). *NATO nedir, ne demek? NATO ne zaman kuruldu?* Ocak 2024 tarihinde https://www.ntv.com.tr/dunya/nato-nedir-ne-demek-nato-ne-zaman-kuruldu,kZ0uYGG3GEG9NYJ3rz4A1Q# adresinden alındı

Paltacı, B. M. (2022). Ukrayna-Rusya Savaşı Bağlamında Siber Güvenlik Ekosistemi'Nde Yaşanan Gelişmeler Ve Değerlendirmeler. *Orta Doğu ve Orta Asya-Kafkaslar Araştırma ve Uygulama Merkezi Dergisi*, Cilt 2, Sayı 2, Sayfalar 1 - 19.

Renda, K. K. (2022). Avrupa Siber Güvenlik Politikasının Gelişimi: Eşgüdümcü Rol'den Siber Güce? *Ankara Avrupa Çalışmaları Dergisi,* , Cilt 21, Sayı 2, Sayfalar 469 - 495.

Türk Devletleri Teşkilatı. (2024). *Türk Devletleri Teşkilatı*. Ocak 2024 tarihinde https://www.turkicstates.org/tr/turk-konseyi-hakkinda#:~:text=T%C3%BCrk%20Devletleri%20Te%C5%9Fkilat%C4%B1%20(eski%20ad%C4%B1yla,%C3%B6rg%C3%BCt%20olarak%202009%20y%C4%B1l%C4%B1nda%20kurulmu%C5%9Ftur. adresinden alındı

Türk Dil Kurumu. (2024, Ocak 02). *Güncel Türkçe sözlük*. Türk Dil Kurumu Sözlükler: https://sozluk.gov.tr/ adresinden alındı

Taş, E. (2017, Eylül 9). *Siber Guvenlik 2030.* Ocak 2024 tarihinde DigitalAge TechSummit: https://digitalagesummit.com/en/schedule/vivamus-vitae-quam-dui/ adresinden alındı

The Hacker News. (2012, 3 29). *Apple Azerbaijan got hacked by Team Nuts*. Ocak 2024 tarihinde https://thehackernews.com/2012/03/apple-azerbaijan-got-hacked-by-team.html adresinden alındı

The Hacker News. (2012, 2 23). *Azerbaijan Arrests Iranian terror group, Iranian Hackers hit Azerbaijan Sites*. Ocak 2024 tarihinde https://thehackernews.com/2012/02/azerbaijan-arrests-iranian-terror-group.html adresinden alındı

The Hacker News. (2012, 1 30). *Embassy of Kazakhstan hacked by Anonymous Supporters*. 1 2024 tarihinde https://thehackernews.com/2012/01/embassy-of-kazakhstan-hacked-by.html adresinden alındı

The Hacker News. (2012, 2 24). *Iran Cyber Army in Action, Azerbaijani TV Down !* Ocak 2024 tarihinde https://thehackernews.com/2012/02/iran-cyber-army-in-action-azerbaijani.html adresinden alındı

The Hacker News. (2019, 8 21). *Russian Hacking Group Targeting Banks Worldwide With Evolving Tactics*. 1 2024 tarihinde https://thehackernews.com/2019/08/silence-apt-russian-hackers.html adresinden alındı

The Hacker News. (2022, 6 17). *Researchers Uncover 'Hermit' Android Spyware Used in Kazakhstan, Syria, and Italy*. 1 2024 tarihinde https://thehackernews.com/2022/06/researchers-uncover-hermit-android.html?m=1 adresinden alındı

The Hacker News. (2023, 1 12). *Chinese Hackers Using SugarGh0st RAT to Target South Korea and Uzbekistan*. 1 2024 tarihinde https://thehackernews.com/2023/12/chinese-hackers-using-sugargh0st-rat-to.html adresinden alındı

The Hacker News. (2023, 10 19). *Operation Rusty Flag: Azerbaijan Targeted in New Rust-Based Malware Campaign*. 1 2024 tarihinde https://thehackernews.com/2023/09/operation-rusty-flag-azerbaijan.html adresinden alındı

TRT Haber. (2019, 10 28). Ocak 2024 tarihinde Türkiye'ye yönelik siber saldırılar bertaraf edildi: https://www.trthaber.com/haber/turkiye/turkiyeye-yonelik-siber-saldirilar-bertaraf-edildi-437841.html adresinden alındı

Yüksek Seçim Kurumu. (2019). *Cumhurbaşkanı seçimi ve 27. dönem milletvekili genel seçimi*. Aralık 2023 tarihinde https://www.ysk.gov.tr/tr/24-haziran-2018-secimleri/77536 adresinden alındı

Yılmaz, O. (2018). Küreselleşme Sürecinde Dönüşen Güvenlik Algısı Ve Siber Güvenlik. *Cyberpolitik Journal*, Cilt 2, Sayı 4, Sayfalar 22 - 43.

"This page is left blank for typesetting"

Bu sayfa dizgiden dolayı boş bırakılmıştır

Journal of Quantum Technologies and Informatics Research
JQTAIR

KUANTUM TEKNOLOJİLERİ VE ENFORMATİK ARAŞTIRMALARI
Volume / Cilt: 2, Issue / Sayı: 1, 2024, pp. 15-29
E - ISSN: 3023-4735
URL: https://journals.gen.tr/index.php/jqtair/
DOI: https://doi.org/10.5281/zenodo.10102956

Received / Geliş: 11.01.2024
Acccepted / Kabul: 14.01.2024

RESEARCH ARTICLE / ARAŞTIRMA MAKALESİ

Otomatik Birim Test Oluşturmak İçin Opcode Ayrıştırma Yaklaşımının Geliştirilmesi

Enhancing The Opcode Parsing Approach for Automated Unit Test Generation

Sevdanur Genç

Kastamonu Üniversitesi, Taşköprü Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Türkiye, e-mail: sgenc@kastamonu.edu.tr

Öz

Yazılım geliştirme süreçlerinde doğruluk ve güvenilirlik, birim testlerin etkin bir şekilde oluşturulmasıyla doğrudan ilişkilidir. Bu bağlamda, birim test üretimi ve yazılım testi süreçleri, geliştiricilerin ve yazılım mühendislerinin önemli bir odak noktası haline gelmiştir. Bu çalışma, otomatik birim test oluşturma sürecindeki gelişmeler üzerinde durarak, geliştirilen Opcode ayrıştırma yönteminin Java Agent teknolojisiyle entegrasyonunu incelemekte ve bu entegrasyonun yazılım testi alanındaki potansiyel etkilerini değerlendirmektedir. Java bytecode seviyesindeki opcode'ları analiz ederek, Java Agent'ların dinamik kod manipülasyonu kabiliyetini kullanarak, otomatik test senaryolarının oluşturulması hedeflenmiştir. Bu yöntem, yazılım geliştirme süreçlerinde test kapsamını artırarak, kod kalitesini iyileştirmeyi ve yazılımın doğruluğunu sağlamayı amaçlamaktadır. Ayrıca, çalışma, Java Agent teknolojisinin opcode analiziyle birleştirilmesinin, otomatik birim test oluşturma sürecindeki etkisini deneysel verilerle destekleyerek, bu teknik entegrasyonun pratik uygulanabilirliğini değerlendirmektedir. Elde edilen sonuçlar, Java Agent'ların opcode analiziyle birleşerek otomatik test oluşturma sürecindeki potansiyelini vurgulayarak, yazılım mühendisliği alanına önemli bir katkı sağlamaktadır.

Anahtar Kelimeler: Opcode Ayrıştırma Yöntemi, Birim Test Üretimi, Bytecode, Java Agent, Yazılım Testi.

Abstract

In software development processes, accuracy and reliability are directly associated with the effective creation of unit tests. In this context, the production of unit tests and software testing processes have become a significant focal point for developers and software engineers. This study examines the developments in automated unit test generation processes, focusing on the integration of the developed Opcode parsing method with Java Agent technology, and evaluates the potential impacts of this integration in the field of software testing. By analyzing opcodes at the Java bytecode level and leveraging the dynamic code manipulation capabilities of Java Agents, the aim is to generate automated test scenarios. This method aims to enhance test coverage in software development

Citation/Atıf: GENÇ, S. (2024). Otomatik Birim Test Oluşturmak İçin Opcode Ayrıştırma Yaklaşımının Geliştirilmesi. *Kuantum Teknolojileri ve Enformatik Araştırmaları*. 2(1): 15-29, DOI: 10.5281/zenodo.10102956

Corresponding Author/ Sorumlu Yazar:
Sevdanur Genç
E-mail: sgenc@kastamonu.edu.tr

processes, improve code quality, and ensure software accuracy. Additionally, the study assesses the practical applicability of this technical integration by supporting it with experimental data, highlighting the impact of merging Java Agent technology with opcode analysis in the process of automated unit test generation. The results underscore the potential of Java Agents combined with opcode analysis in automating the test generation process, offering a significant contribution to the field of software engineering.

Keywords*:* The Opcode Parsing Method, Unit Test Generation, Bytecode, Java Agent, Software Testing.

1. GİRİŞ

Sun Microsystems tarafından geliştirilen ve 1995 yılında piyasaya sürülen Java programlama dilinin ana amacı, taşınabilir, öğrenmesi kolay, genel amaçlı, platformdan bağımsız ve nesne odaklı bir programlama dilinin oluşturulmasıydı. Java derleyicisi, kaynak kodunu platformdan bağımsız bir ara dil olan Java bytecode'a dönüştürür. Bu kod daha sonra her platformda Java sanal makinesi üzerinden işlenir ve çalıştırılır. Java ajanları, JVM tarafından çalışma zamanında yürütülen uygulama mantığını esnek bir şekilde değiştirmek için kullanılır. Bir Java Agent, özel olarak tasarlanmış bir jar dosyasıdır. Bu dosya, mevcut JVM'de yüklenmiş olan bytecode'ları değiştirmek için Instrumentation API'yi kullanır. Java Agent araçlarının en önemli özelliği, çalışma zamanında sınıfları yeniden tanımlayabilme veya değiştirebilme yetenekleridir. Metot gövdelerini sabit ve değişken özelliklerini yeniden tanımlayarak değiştirebilirler. Ayrıca, metotların nesne veya kalıtım özelliklerini değiştirebilirler.

Yazılım kalitesi ve üretkenlik ihtiyaçları yazılım geliştirme sürecinde bir arada değerlendirilir. Bu nedenle, akıcı bir algoritma ve güçlü risk yönetimi gibi faktörler mevcut yazılımın bu kriterlere uygunluğunu test etmek için gereklidir. Bu faktörleri yerine getirmek için test alanında ciddi sorumluluklar bulunmaktadır. Hızlı testler ve sonuçların yüksek doğruluğu, yazılım geliştirme sürecinde fark yaratan faktörlerdir. Bu faktörü gerçekleştirmek için yazılım test otomasyonu gereklidir. Yazılım projelerindeki odak noktası genellikle yazılım test süreçleridir. Başarılı bir test sürecinin sonunda, hataları en az olan ve yüksek doğrulukta bir yazılım elde edilir. Türkiye'deki yazılım kalitesi üzerine yapılan çalışmalar, test odaklı yazılım geliştirme süreci ve test araçlarına olan ihtiyacın ülkemizde artmakta olduğunu göstermektedir [1]. Test odaklı yazılım geliştirmede hedef, gerekli işi yapacak kod yazmadan önce, öncelikle test edilebilir bir kod yazmak ve bu koda ait senaryoyu oluşturmaktır. Farklı yazılım geliştirme prensipleri bu test edilebilir kodu tasarlar; eğer yüksek doğrulukta bir sonuç alınırsa, yazılım başarılı bir şekilde testi geçmiş olur. Test sonuçları başarısız olursa, başa dönülür, kod incelenir ve sorun düzeltilmeye çalışılır. Yazılım projelerinde, her birim (sınıf ve metot) hatasız çalıştığını kanıtlamak adına birim testleri denilen testler yazılır. Birim testleri yazılım geliştirme sürecini kolaylaştırır, hızlandırır ve her sınıf ve metodun doğru şekilde çalıştığından emin olunmasını sağlar.

Java platformunda en yaygın kullanılan iki temel test çerçevesi bulunmaktadır: JUnit ve TestNG [2]. Her ikisi de karmaşık test durumlarında istenilen kod parçaları üzerinde test yapmaya imkan tanıyacak kadar güçlüdür. JUnit, yinelenebilir testler yazmak ve çalıştırmak için açık kaynak kodlu bir framework'tür. Çeşitli test durumlarıyla test verilerini çalıştırarak programdan beklenen sonuçları test eder. TestNG, JUnit'den daha işlevseldir ve kullanımı daha kolaydır. TestNG, test thread'lerinde test durumlarını paralel olarak çalıştırmayı destekler. Ayrıca esnek test yapılandırmaları, detaylı hata mesajlarının analizi, gelişmiş arşivleme ve editörler için eklenti desteği gibi birçok özelliğe sahiptir.

Geliştirici tüm bu birim testlerini otomatik olmayan sistemlerle yani elle yazarak kodlar. Testleri otomatikleştirmek, zamanı kaliteli bir şekilde kullanmayı ve iş trafiğini hızlandırmayı önerir. Bu nedenle, test araçlarının hızı ve doğruluğu önemlidir. Örneğin, anahtar kelime tabanlı kodlar önemli üstünlüklere sahiptir.

Bu yaklaşımda, test edilen yazılımın boyutu önemlidir, test sayısı değil. Bu, kod bakım maliyetini büyük ölçüde azaltır ve otomatik testlerin uygulanmasını hızlandırır. Aynı zamanda, test otomasyonunda başarı elde etmek için kodların ve verilerin yeniden kullanılabilir olması gereklidir. Bu, tekrarlanan görevleri ortadan kaldırır ve yeni testlerin uygulanmasını hızlandırır. Bununla birlikte, otomatik testler, elle yazılan testlere kıyasla yazım hatalarının da önlenmesine yardımcı olabilir [3].

Bu araştırma, çalışma zamanında birim testlerin otomatik olarak oluşturulabilmesi için bir çözüm olarak opcode ayrıştırma yöntemini önermektedir. Bu yazılım, bir masaüstü uygulaması olarak geliştirilmiş olup belirtilen Java sınıflarında birim test dönüşümleri gerçekleştirebilmektedir. Tüm bu dönüşümler, Java Agent ve bytecode temellidir. Çalışılacak örnek Java sınıflarının nesneleri, metotları ve değişkenleri ile ilgili tüm bilgi çalışma zamanında veriye dönüştürülmüştür. Bu dönüşüm sırasında bilgi hem veritabanına kaydedilmiş hem de kaydedilen verilerle birlikte bir şablon motoru aracılığıyla otomatik birim testleri haline getirtilmiştir.

Bu çalışmada, otomatik birim testi üretimi için çeşitli yaklaşımlar sunulmuştur. Çalışmanın literatüre katkıları şunlardır:

1. Bytecode dönüşümleri sırasında çalışacak olan bir opcode ayrıştırma yöntemi, Java string fonksiyonlarından yararlanarak geliştirilmiştir. Bu yöntemle, her bir opcode'a karşı nesneler, değişkenler ve giriş-çıkış parametreleri gibi değerler ayrılmış ve bu veriler JSON formatında listelenmiştir. Geliştirilen opcode ayrıştırma yöntemi, açık kaynak kodlu bir yaklaşım olduğu için gelecekte farklı ihtiyaçlara göre geliştirilmeye açıktır. Literatürdeki çalışmalarda, Bytecode API'nin sınırlı hazır fonksiyonları kullanılmış ve yine bunlarla birlikte farklı yaklaşımlar sunulmuştur.

2. Bir Java sınıfındaki her bir nesne, NoSql veritabanı koleksiyonları kullanılarak JSON formatında saklanır. Bu şekilde, değişkenlerin ve giriş-çıkış parametrelerinin değerleri, oluşturulacak birim testlerde yeniden kullanılabilir veya bu değerlere benzer rastgele değerler atanabilir. Bu veriler aynı zamanda sistemde bir arşiv dosyasında da saklanır. Literatürde, XML ve Oracle gibi farklı veri depolama ortamları kullanılmıştır.

3. JUnit standartlarına göre oluşturulacak her bir birim test için gerekli olan doğrulama yapısı, FTL (FreeMarker template) şablon motoru kullanılarak hazırlanmıştır. FTL şablon motorunun literatürdeki otomatik birim testi üretimi ürünlerinde kullanılmadığı görülmüştür. Bunun yerine farklı yöntemler geliştirilmiştir.

2. İLGİLİ ÇALIŞMALAR

Yazılım test uygulamalarında birim test üretimi üzerine son 20 yılda yayınlanan farklı çalışmalar incelenmiş olup Çizelge 1'de bu çalışmalar özetlenmiştir.

3. MATERYAL VE YÖNTEM

3.1. Otomatik Birim Test

Yazılım test otomasyonlarının mantığı, elle yazılan testleri otomatikleştirmek ve bir araca dönüştürmektir. Otomasyon, test senaryolarının sürekli tekrarlanmasını ifade eder. Bu durumda, test senaryolarını hazırlayan çalışanlar, kodları çalıştırabilen otomatik bir yazılıma ihtiyaç duyar ve bu yazılımlara yazılım test otomasyonu denir. Yazılım test otomasyonunda, test edilebilir her türlü algoritma, metot ve sınıf yapıları bulunmaktadır. Bunları ayrı ayrı test edebilmek için de birim test yapıları ortaya çıkmıştır. Birim testlerde kullanılan test senaryoları öncelikle geliştirilir ve yazılım bu senaryoların sonuçlarına göre kodlanır. Amaç, test sonuçlarında doğruları ve hataları aramaktır. Test kodu geliştiricileri tarafından bulunan tüm hatalar, düzeltilebiliyorsa çalışma zamanında düzeltilir. Aksi takdirde, test sonuçlarının raporlarını ilgili birimlere yönlendirerek yardımcı olurlar. Doğru kabul edilen her bir birimin testi, ürün yazılımına ait kodunun yazılmasıyla devam ettirilir, böylece her modül tamamlandıktan sonra ürün birleştirilir ve tamamlanır.

Bu çalışmada, birim testlerin üretilebilmesi için otomatik yazılım test otomasyonlarında kullanılabilecek bir opcode ayrıştıma yöntemi tasarlanmıştır. Bu yöntemin tasarım aşamasında, java bytecode, java agent ve javassist gibi önemli yapılar kullanılmıştır.

3.2. Java Agent, Java Bytecode ve Javassist

Java Agent: Java Instrumentation API'sını kullanarak JVM üzerinde çalışan uygulamalara müdahale ederek bytecode manipüle edebilen özel bir Java kütüphanesidir. Genellikle bir jar dosyası olarak hazırlanır. Java agent'ı temsil eden sınıflar, Java API kütüphanesinde bulunan diğer sınıflardan farklı değildir. Ancak onları özel kılan şey, Java kodunu JVM'de çalışan diğer herhangi bir uygulamayı engellemesine izin veren belirli bir kuralı takip etmeleridir. Buradaki tek amaç, sadece bytecode'ları sorgulayan veya değiştiren ajanlar oluşturmaktır. Bu güçlü özellik, normalde bir Java programının yapabileceğinden daha fazlasını sağlar. Bir bakıma, bir programa

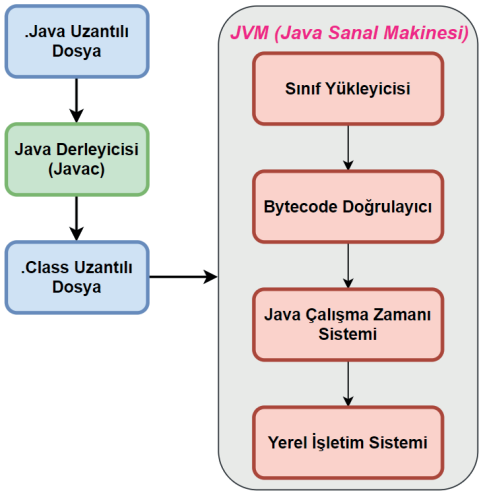
Çizelge-1: Birim test üretimine ilişkin önceki araştırmalar.

Yazar	Çalışma	Bulguları
Csallner ve çalışma arkadaşları [4]	JCrasher adlı otomatik bir test üretim aracı geliştirilmiş. Bu araç, Eclipse IDE ile entegre bir şekilde çalışabilir. Test için JUnit kullanılmıştır.	Test üretimi için verilen örnek Java sınıfının bilgilerini inceledikten sonra rastgele verilerle test edilir. Test aşamasında oluşan hataları da tespit edebilir.
Pacheco ve çalışma arkadaşları [5,6]	JUnit kullanarak Randoop adlı otomatik bir birim testi üretim aracı geliştirilmiş.	Nesne odaklı programlar için geribeslemeli birim testi oluşturabilir ve oluşan hataları yakalayabilir. Rastgele test verileri oluşturmak ve test sonuçlarını birleştirmek için geliştirilmiştir.
Simons ve çalışma arkadaşları [7]	JWalk adlı bir Java birim testi aracı geliştirilmiş.	Örnek bir sınıfın gelişmiş özelliklerini ortaya çıkardıktan sonra birim testlerin sistemli bir şekilde oluşturur. Java test sınıflarının durumu hakkında bilgi sağlayabilir ve JUnit gibi uzmanlaşmış birim testi uygulamalarıyla karşılaştırılmıştır.
Sen [8]	Java ortamında Cute adında bir uygulama geliştirmiş olup, C programlama dilinde yazılmış kodları test etmek etmektedir. Otomatik ve rastgele test mantığını birleştirerek çalışır.	Sembolik kod yürütme kullanarak özel girişler ve kısıtlayıcı çözümleri aşmaya yardımcı olur. Ancak, sistem çağrılarını analiz etme ve doğrusal olmayan tamsayı denklemlerini çözme gibi iyileştirilmesi gereken alanları bulunmaktadır.
Charreteur ve çalışma arkadaşları [9]	Java bytecode programları için otomatik test girdisi elde etmek amacıyla kısıt tabanlı bir akıl yürütme yaklaşımı kullanmışlardır.	Her bir bytecode için geriye doğru arama yapılmasına ve bellek üzerindeki karmaşık kısıtları çözmeye izin veren bir kısıt tabanlı model geliştirilmiştir. JAUT adını verdikleri bu çalışma, Cute, JTEST ve PEX gibi çalışmalar için öncülük etmektedir.
Fraser ve çalışma arkadaşları [10]	EvoSuite adlı bir test oluşturma aracı geliştirilmişlerdir. Java'da yazılmış olan bu test oluşturma aracı geniş özelliklere sahiptir.	Bu araçla gerçekleştirilen tüm testler istenen kriterlerle karşılaştırılabilir. Karşılaştırma sonucunda analiz ve optimizasyon işlemleri gerçekleştirilir.
Sakti ve çalışma arkadaşları [11]	JTExpert adında bir otomatik test oluşturma aracı geliştirilmiş. Java programlama dilinde kullanılabilen bu araç, çalıştırılabilir bir jar dosyasıdır.	JTExpert aracı, test edilen her bir java sınıfı için JUnit formatında otomatik olarak bir test veri paketi oluşturur.
Tanno ve çalışma arkadaşları [12]	CATG adında bir hibrit birim test aracı geliştirilmiş.	Sembolik ve somut girişleri dinamik olarak gerçekleştiren bir kavram olan konkolik testi kullandılar.
Brill ve çalışma arkadaşları [13]	TACKLETEST adında açık kaynaklı bir araç geliştirilmiş. Java uygulamaları için otomatik birim seviyesinde test senaryoları oluşturmak için kullanılır.	Birim testleri için kapsama hedeflerini hesaplama için yeni ve tamamlayıcı bir yöntem uygular.
Higo ve çalışma arkadaşları [14]	Otomatik test oluşturma tekniklerini kullanarak bir veri kümesi oluşturma aracı geliştirilmiş.	Yaklaşık 36 milyon satırlık bir kaynak kodundan fonksiyonel olarak eşdeğer Java metotları veri kümesi oluşturmuşlardır.
Lukasczyk ve çalışma arkadaşları [15]	Dinamik programlama dili olan Python için Pynguin adında otomatik bir birim test oluşturma yazılımı geliştirilmiş.	Yüksek kod kapsamına sahip regresyon testleri üreten genişletilebilir bir Python test oluşturma çerçevesi geliştirdiler.

girilebilir ve bytecode'unu değiştirebilir veya karışıklık çıkarabilir.

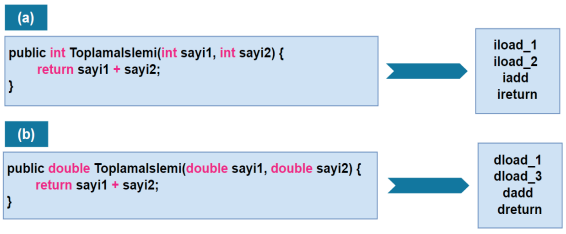
Javassist: Java programları üzerinde dinamik olarak bytecode üretme, değiştirme ve analiz etme işlemlerini sağlayan bir kütüphanedir. Bu kütüphane, bytecode düzeyinde programları manipüle etmek için kullanılır. Java sınıflarını oluşturmak, düzenlemek veya analiz etmek için kullanıcıya geniş olanaklar sunar.

Java Bytecode: C ve C++ derleyicileri assembler ile temsil edildiği gibi, java programları da bytecode ile temsil edilir. Bir java derleyicisinin üreteceği bytecode aslında programın kendisidir. Aynı zamanda bytecode, Java'nın taşınabilirlik ve güvenlik gibi sorunlarına bir çözüm olmak zorundadır. Java derleyicisinin çıktısı yürütülebilir olmadığı için bytecode'a ihtiyaç duyulur. Bytecode'lar JVM tarafından yorumlanır. Bu sayede bytecode'lar çalışma zamanında iyi bir şekilde optimize edilir. Bytecode sayesinde bir java programı birçok farklı ortamda çalıştırılabilir. Şekil 1'de bir bytecode'un çalışma akışı verilmiştir.



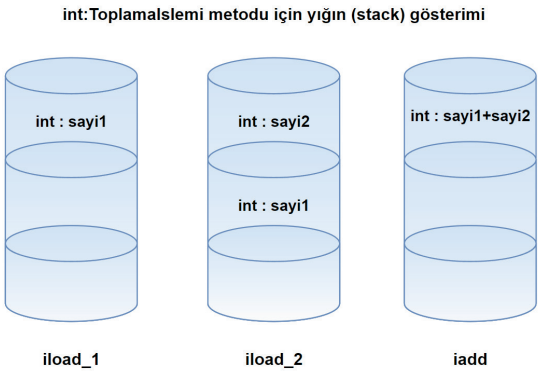
Şekil-1: Java Bytecode'un çalışma akışı

Bir metodun bytecode akışı, JVM için bir dizi talimatı içerir. Her talimat, bir byte opcode'undan ve bu opcode'u takip eden sıfır veya daha fazla operand'dan oluşur. Operand'lar, bir komutun veya talimatın işleme sokulacak olan veri veya hedef değerleridir. Bu değerler, komutun işlevini gerçekleştirmek için kullanılır. Bu bağlamda, bir opcode (bytecode talimatı) belirli bir işlemi temsil ederken, operand'lar bu işlemi tamamlamak için gerekli olan değerleri veya hedefleri içerir. Opcode, gerçekleştirilecek işlemi gösterir. JVM işlem yapmadan önce daha fazla bilgi gerekiyorsa, bu bilgi opcode'u takip eden bir veya daha fazla operanda kodlanır. Her opcode türü bir mnemonic'e sahiptir. Mnemonic, bir bilgisayar programında veya işletim sistemlerinde, spesifik bir işlemi veya komutu temsil etmek için kullanılan sembolik veya hatırlatıcı bir kısaltma yada semboldür. Özellikle bellek adresleri, komutlar veya işlemcilerin komut setleri gibi teknik veya karmaşık konuları temsil etmek için kullanılırlar. Bu bağlamda, bir opcode'un işlevini veya



Şekil-2: Java ile yazılmış parametre alarak toplama işlemini gerçekleştiren bir metodun farklı veri türündeki operand durumları verilmiştir [16]. (a)'da int, (b)'de ise double veri türlerine örnek verilmiştir.

Şekil 2'de java ile yazılmış basit bir metodun almış olduğu farklı veri türündeki parametrelerle kullanımı ve bu kodlara karşılık gelen opcode'lar verilmiştir. Şekil 3'de ise Şekil 2.(a)'da verilen metot yapısının operand yığın gösterimi basit bir şekilde incelenmiştir.



Şekil-3: int veri türü dönüşümlü ToplamaIslemi metodunun yığın üzerinde gösterimi

Bytecode ile ilgili kodların olduğu talimat seti karmaşık olacak şekilde tasarlanmıştır. Tüm talimatlar, tablo oluşturmak için iki kod dışında bayt sınırlarına hizalanmıştır. Opcode'ların toplam sayısı azdır, bu nedenle bytecode'lar

yalnızca bir byte yer kaplar. Böylece, JVM tarafından çalıştırılmadan önce sınıf dosyalarının boyutunu en aza indirmeye yardımcı olur. Ayrıca, JVM uygulamasının boyutunu küçük tutmaya da yardımcı olur. JVM'deki tüm hesaplamalar yığın üzerinde gerçekleştirilir. Bu nedenle, bytecode talimatları öncelikle yığın üzerinde çalışır [17].

Java bytecode, .class uzantılı dosya biçimindeki makine kodudur. Java'da bytecode kullanımı, JVM için komut setidir ve derleyiciye benzer şekilde çalışır. Bytecode'un detaylı incelenmesi, belirli opcode'ların olduğunu ortaya koyar. Bazı opcode'ların önünde şekil 2'de de görüldüğü a veya i gibi harfler bulunur. Örneğin, aload_0 ve iload_2. Bu ön ekler, opcode'un hangi türle çalıştığını temsil eder. a ön eki, opcode'un bir nesne referansını değiştirdiğini ifade eder. i ön eki, opcode'un bir tam sayıyı işlediğini belirtir. Diğer opcode'lar; byte için b, char için c ve double için d olarak kullanılır. Bu ön ekler, işlenen veri türü hakkında bilgi sağlar.

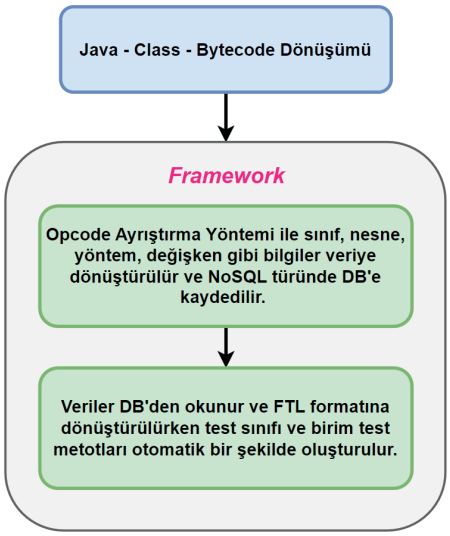
JVM tarafından bytecode'un yürütülmesi için yığın tabanlı bir makine kullanılır. Her bir iş parçacığının bir JVM yığını vardır; bu yığın, verilerini bir çerçevede depolar ve bir çerçeve yığınına dönüştürür. Bir yöntem çağrıldığında her seferinde bir çerçeve yığını oluşturulur ve operand yığını, yerel değişkenlerin bir seti ve mevcut sınıfın çalışma zamanı gibi veriler içerir. Yerel değişkenler seti (dizi), aynı zamanda yerel değişkenlerin değerlerini tutmak için kullanılan yerel değişkenler tablosu olarak da bilinir. Parametreler, dizindeki 0›dan başlayarak saklanır. Yapı bir yapıcı ya da dinamik yöntem içinse, referans 0 pozisyonunda saklanır. Ardından, 1. pozisyon ilk formal parametreyi ve 2. pozisyon ikinci formal parametreyi alır. Bir statik yöntem için, ilk formal yöntem parametresi 0 pozisyonunda saklanır, ikincisi ise 1. pozisyonda. Yerel değişkenler dizisinin boyutu derleme zamanında belirlenir ve yerel değişkenlerin sayısı ve boyutu bir formal prosedür parametine bağlıdır. Operand yığını, değerlerin yerini değiştirmek için LIFO (Son Giren İlk Çıkar) yöntemini kullanır. Belirli opcode talimatları, operand yığınına değerler taşır; diğerleri operatörleri yığından çeker, manipüle eder ve sonucu oluşturur. Operand yığını aynı zamanda yöntemlerin tümünde dönüş değerlerini almak için de kullanılır.

3.3. Opcode Ayrıştırma Yöntemi

Geliştirlen opcode ayrıştırma yöntemi için öncelikle bir Java sınıfı sisteme tanıtılır. Tanıtılan Java sınıfı, bir java agent yardımıyla bytecode dosya formatına dönüştürülür. Bu dönüşüm gerçekleştirilirken; sınıf adı, değişkenler, nesneler, metotlar ve bunların girdi-çıktı parametreleri gibi birçok opcode, java string fonksiyonları kullanılarak satır satır ve kelime kelime ayrıştırılır. Örneğin, invokevirtual, invokestatic, ifge, iflt, ifeq, iinc gibi opcode'lar, bytecode dosyasında java string fonksiyonları kullanılarak kontrol edilir ve karşılık gelen parametreler belirlenir, değişkenlere aktarılır. Aynı zamanda, sınıfta tanımlanan nesne türlerini gösteren opcode'ların verdiği parametreler kontrol edilerek Mock-Stub yöntemlerinin kullanımı için de ayrım yapılabilir. Değişkenlere aktarılan tüm parametreler anlık olarak JSON formatında listelenir ve NoSql veritabanı yapısı kullanılarak veri halinde kaydedilir. Tüm dönüşüm işlemleri tamamlandıktan sonra kaydedilen veriler, FTL şablon motorunun yardımıyla anında birim test formatına otomatik olarak dönüştürülür. FreeMarker Java Template Engine-FTL (Java Template Engine), Apache tarafından üretilen bir şablon motorudur. Şablonlar ve değişen verilere dayalı olarak HTML web sayfaları, e-postalar, yapılandırma dosyaları ve kaynak kodları gibi metin çıktıları üretebilen bir Java kütüphanesidir. Genellikle java gibi genel amaçlı bir programlama diline verileri hazırlamak için kullanılır. Ardından FTL şablonları kullanılarak hazırlanan verileri görüntüler. FTL şablonlarında verilerin "nasıl" sunulacağı ve şablon dışında "hangi" verilerin sunulacağına odaklanır [18]. Böylece, esnek bir framework olarak üretilen opcode ayrıştırma yöntemi sayesinde istenen opcode türü için gerekli kod değişiklikleri yapılabilir. Bu tamamen açık kaynak kodlu olması nedeniyle projeye katkı sağlamaktadır.

Bu çalışmada, çalışma zamanında veri toplayarak otomatik birim testlerin oluşturulmasına imkan sağlayan bir opcode ayrıştırma yöntemi geliştirilmiştir. Söz konusu araç,

Java programlama dili kullanılarak Netbeans ortamında konsol ve masaüstü uygulaması olarak geliştirilmiştir. Bu çalışma, Java programlama dili üzerine kurulu olduğundan, Java tabanlı kodları test etmek için JUnit framework'ü kullanıldı ve birim testlerin oluşturulması bu bağlamda yapıldı. Geliştirilen yöntemin akış şeması Şekil 4'de verilmiştir.



Şekil-4: Çalışma Zamanı Verilerini Toplayarak Otomatik Birim Test Oluşturmada Kullanmak için Opcode Ayrıştırma Yönteminin Akış Şeması

Şekil 5'te opcode ayrıştırma yönteminden bir kesit verilmiştir. Burada getstatic ve ifge gibi ilgili opcode'ların dosya içerisinden nasıl okunacağı hakkında bir algoritma kullanılmıştır. Tamamen string fonksiyonlarından yararlanılarak her bir satırdaki kodlar taranmış ve yakalanan ifadeler ilgili MongoDB'ye gönderilmesi için aracı değişkenlere aktarılmıştır. Ayrıca ifge gibi koşul belirten yapıların kullandığı < ve > gibi operatörlerde kontrol edilmiştir.

3.3.1. Opcode Ayrıştırma Yönteminin Tasarımı

Java - Class - Bytecode **Dönüşümleri:**

Geliştirilen ayrıştırma yöntemi kullanıcılar tarafından kolaylık olması amacıyla bir arayüz olarak NetBeans ortamında tasarlanmıştır. Öncelikle bir Java sınıfı bu sisteme yüklenir. Bu Java dosyası ara yüzde bulunan bir buton yardımıyla .class dosyasına dönüştürülür ve geliştirilen yazılımın neredeyse her bölümünde .class dosyası kullanılır. Sınıf dosyasını derlemek için JavaCompiler kütüphanesi kullanılmıştır. Burada birden fazla Java sınıfı ile ilişkili ve bağlantılı olan bir sınıf sisteme tanıtılırsa, dosyanın bulunduğu konumdaki ilgili diğer sınıflarda sistem tarafından otomatik olarak algılanmaktadır. Geliştirilen sistem okuduğu tüm dosyaları Iterable koleksiyon listesinde saklar ve daha sonra bu liste opcode dönüşüm işlemlerinde kullanılır.

• Java dosyasından Sınıf dosyasına dönüşümün ardından, bytecode dönüşüm işlemleri gerçekleştirilir. Bu işlem için javassist kütüphanesindeki ClassPool, CtClass, CtMethod ve InstructionPrinter alt sınıflar kullanılmıştır. Bu alt sınıflardan bahsedecek olursak;

• ClassPool: Yüklenen sınıfları ve bunların metodlarını, değişkenlerini ve diğer özelliklerini yöneten bir veri havuzu olarak işlev görür. Sınıfların ve bunların bileşenlerinin depolanması, erişilmesi ve değiştirilmesi için kullanılır. Genellikle sınıf dosyalarını temsil eder ve bu sınıflar üzerinde yapılan işlemleri sağlar.

```
// _" + linenumber + ""
if (line_.contains(": getstatic #")) {
    objectClassName2 = line_.substring(line_.indexOf("Field") + 6, line_.indexOf("("));
    document.append("$set", new BasicDBObject().append("ObjectInClass_Field", convertToByteCodeField(objectClassName2)));
    searchQuery = new BasicDBObject().append("Execution Id", "'" + executionId + "'");
    table.update(searchQuery, document, true, false);
    document.append("$set", new BasicDBObject().append("ImportField", StringUtils.substringBeforeLast(objectClassName2, ".")));
    searchQuery = new BasicDBObject().append("Execution Id", "'" + executionId + "'");
    table.update(searchQuery, document, true, false);
}

if (line_.contains(": ifge")) {   //value>=0
    ifMap.add(new bcList("ifge", Integer.parseInt(line_.substring(line_.indexOf("ifge") + 5, line_.length()))));
    ifSetVariable = ifGetVariable("<", linenumber);
    writeToDBtoIf("ifge" + linenumber, ifSetVariable);
}
```

Şekil-5: Opcode ayrıştıma yönteminden bir örnek

• CtClass: Javassist kütüphanesinde temsil edilen bir sınıfın yapısını ve özelliklerini içerir. Bu sınıf, yüklü sınıfları temsil eden bir veri yapısıdır. Sınıfların yapısını değiştirmek, özelliklerini incelemek veya düzenlemek için kullanılır.

• CtMethod: Javassist kütüphanesindeki bir sınıfın veya arabirimin bir yöntemini temsil eder. Bu sınıf, yöntemlerin özelliklerini içerir, örneğin yöntem adı, parametreleri, dönüş türü vb. yöntemleri oluşturmak, değiştirmek veya incelemek için kullanılır.

InstructionPrinter: Javassist kütüphanesinde kullanılan bir alt sınıf olup, bytecode veya işlem talimatlarını, metotların içeriğini yazdırmak ve görüntülemek için kullanılır. Bu sınıf, bytecode'larını okunabilir bir formatta görüntülemek için kullanılır ve genellikle bytecode'unu veya metot içeriğini denetlemek ve anlamak için kullanıcıya yardımcı olur.

Özetle, yapılan çalışmada, okunan java sınıf dosyasındaki sınıf ve metot bilgileri classPool adlı sınıf havuzunda tutulur. Daha sonra, classPool'dan InstructionPrinter ile bilgi alınır ve sonuçlar bir çıktı olarak görüntülenir.

Şekil 6›de java sınıfının bir bytecode'a dönüştürülmüş haline ait örnek bir çıktı verilmiştir. Sınıftaki her metot MethodName satırları arasında ayrılır. Her metot 0 numaralı satır ile başlar ve geri kalan diğer metotlardan ayrımı otomatik olarak yapılmış olur. Bu metot, dreturn anahtar kelimesiyle bir değer döndürür ve aynı zamanda lcd2_w anahtar kelimesiyle parametreleri aldığını gösterir.

```
MethodName : InterestRate
0: dload_0
1: ldc2_w #2 = int 100.0
4: ddiv
5: dload_2
6: dmul
7: dreturn
MethodName : InterestRateEx
0: dload_2
1: dconst_1
2: dcmpg
3: ifge 8
6: dconst_1
7: dstore_2
8: dload_0
9: ldc2_w #2 = int 100.0
12: ddiv
13: dload_2
14: dmul
15: dreturn
```

Şekil-6: Bir Bytecode örneği

Bu bytecode'lar, Javassist kütüphanesi tarafından okunur ve ilgili sınıfın dosya adı ve yolu gibi parametrelerde ClassPool, CtClass ve CtMethod gibi yapılar içinde saklanır. Bytecode'a dönüştürülen tüm kod satırları, Netbeans arayüzünün konsol ekranında görüntülenebilir; ayrıca metin belgeleri olarak da çıktı alınabilir. Bu ayrıştırma yöntemi için geliştirilen başka bir sınıf ise FindObjectsInClasses'dir. Bu sınıfın yardımıyla, her bir bytecode, bytecode dosyasının çıktısından satır satır okunur ve bu kodlar NoSQL formatında MongoDB veritabanına depolanır. Bunun için, ConnectionDB olarak adlandırılan bir sınıf framework içerisinde oluşturulmuştur. Bu sınıfta, com.mongodb kütüphanesinin yöntemlerini kullanarak bir koleksiyon oluşturma ve koleksiyondaki verilere erişme gibi özellikler bir araya getirilmiştir. Bu sınıf aynı zamanda tüm JSON formatındaki verilerin bir metin belgesinde arşivlenmesini sağlar.

Burada önemli olan nokta, bilgilerin geri alınması ve veritabanına kaydedilmesinde java string fonksiyonlarından yardım alınmasıdır. Bu, çalışma için geliştirilen opcode ayrıştırma yöntemi yardımcı olmaktadır. Sınıfta kullanılan değişkenler, varsa metot isimleri, aldıkları parametreler ve gönderdikleri sonuçlar, oluşturulan tüm nesneler, kullanılan koşul ve döngü bloklarına ilişkin bilgiler, sırasıyla string fonksiyonlarıyla alınır. Alınan bilgiler, Şekil

7 ve Şekil 8'de görüldüğü gibi MongoDB'de saklanır. Bunun yanı sıra, invokevirtual, getfield, invokestatic, getstatic, ifge, ifle, iflt, ifgt, ifeq, ifne, iinc, if_icmp ve goto gibi en sık kullanılan opcode'larla ilgili bilgiler de kaydedilir. Aynı zamanda, bu bilgiler işlemin yapıldığı gün ve saatine göre sistem dosyaları altında metin belgesi (.txt) biçiminde arşivlenir. Böylece MongoDB tarafından iki koleksiyon kullanılır. Bunlardan ilki, Şekil 7'da görülen byteCoding adlı koleksiyondur.


```
/* 1 */
{
    "_id" : ObjectId("622e1bb20fb2df14e4b6d6ce"),
    "Execution Id" : "'25'",
    "MethodName" : "'InterestRate'",
    "ClassName" : "'CalculateCredit'",
    "Returned" : "'728.28'"
}

/* 2 */
{
    "_id" : ObjectId("622e1bb20fb2df14e4b6d6cf"),
    "Execution Id" : "'26'",
    "MethodName" : "'InterestRateEx'",
    "ifge12Line" : "< 1.0",
    "ClassName" : "'CalculateCredit'",
    "Returned" : "'1936.69'"
}

/* 3 */
{
    "_id" : ObjectId("622e1bb20fb2df14e4b6d6d0"),
    "Execution Id" : "'27'",
    "MethodName" : "'InterestRateFor'",
    "Loop" : "InterestRateFor",
    "ClassName" : "'CalculateCredit'",
    "Returned" : "'475800'"
}

/* 4 */
{
    "_id" : ObjectId("622e1bb20fb2df14e4b6d6d1"),
    "Execution Id" : "'28'",
    "MethodName" : "'InterestRateIF'",
    "ifle41Line" : "> 7.0",
    "ifge45Line" : "< 9.0",
    "iflt52Line" : ">= 3.0",
    "ifgt56Line" : "<= 6.0",
    "ifle63Line" : "> 1.6",
    "ifgt67Line" : "<= 2.95",
    "ClassName" : "'CalculateCredit'",
    "Returned" : "'4.56'"
}
```


Şekil-7: *byteCoding* adlı koleksiyonun yapısı

NoSQL yapısı altında, veri anahtar-değer (key-value) yapısıyla listelenir. Anahtar alanlarının adlandırılması bazı alanlarda yapılacak işlemlere ilişkin olurken, diğerlerinde bytecode terimlerini hatırlatır. Örneğin, MethodReturnType adlı anahtar alanı, methodun gönderdiği sonucu depolar; ifge45Line adlı anahtar alanı ise if bloğunun hangi işleme karşılık geldiğinin değerini saklar. MongoDB'de kullanılan ikinci koleksiyon adı ise kayitlar'dır. Şekil 7'de görüldüğü gibi, sınıflar ve metodlar hakkında tüm bilgiler kaydedilmiştir. Öyle ki, mock uygulanabilecek aday işlemlere ait nesnelerin sınıf ve nesne adları, bu liste içinde mocking anahtar kelimesi altında yer alır.

Veri Aktarımı - Şablon Dönüştürme:

MongoDB'de arşivlenen tüm veriler, veri okuma veya yazma gibi işlemler için otomatik birim test yazılımı ile iletişim kurabilmelidir. Bu nedenle, veritabanı ile yazılım arasında iletişimi sağlamak için bir POJO sınıfı yazılmıştır. Her türlü basit seviyedeki test senaryolarına cevap verebilecek şekilde kurucu ve getter-setter yöntemleri, değişkenleriyle birlikte tanımlanmıştır. POJO sınıfı, geliştirilen ayrıştırma yöntemi içerisinde veri taşıyıcı olarak işlev görür.

Şekil 7 ve Şekil 8'de de görüldüğü üzere veriler, veritabanında iki tabloda yani koleksiyonda depolanmaktadır. Bu koleksiyonlar, birleştirilmiş ve id'ler kullanılarak birbirleriyle ilişkilendirilmiştir. Bu ilişkilendirilmiş koleksiyonlardan veri okunabilmesi için iki farklı sınıf yazılmıştır. Bunlardan ilki, framework için geliştirilmiş GetItFromMongoDB sınıfıdır. Çalışma zamanında okunan koşul ve döngülerin kodları byteCoding koleksiyonunda saklanmıştır. Bu koleksiyonda okunan veriler, birim test koduna dönüştürülmek üzere FTL formatına yönlendirilir ve bu işlem POJO sınıfının yardımıyla gerçekleştirilir. ByteCoding koleksiyonunda 'if' ile başlayan anahtar kelimeler için değişkenler; sınıf adı, metot adı ve dönüş değerleri ile birlikte POJO sınıfına iletilir. 'if' ile başlayan tüm değerler bir metot içerisinde kullanılan şart yapılarının tüm olasılıklarını karşılayacak şekilde opcode ayrıştırma yönteminde alt algoritması geliştirilmiştir.

MongoDB veritabanından veri çekmek amacıyla ReadDataFromDB sınıfı oluşturulmuştur. Bu sınıfta, birleştirme işlemleri için iki koleksiyona bağlanmıştır. İlgili tüm değişkenler hem veritabanından okunur hem de birleştirme işlemleri için güncellenmesi gereken alanlar olup olmadığı kontrol edilir. Tüm değişkenler, çerçevenin ana sınıfına yönlendirilir. Veritabanından alınan tüm veriler ise, taşıyıcı sınıf POJO ile çekilir. Veritabanında depolanan bu verilerle birlikte, birim testleri istenen dosya biçiminde FTL kullanılarak otomatik olarak hazırlanır ve bir çıktıya dönüştürülür. Bu çıktıların içeriği, çalışma zamanında toplanan

verilere dayalı olarak oluşturulan yeni bir sınıfa ait kodlardır.

4. DENEYSEL SONUÇLAR

Opcode Ayrıştırma Yönteminin Uygulanması:

Uygulamanın backend kodları tamamlandıktan sonra, frontend kodları ile arayüzü hazırlandı. Böylece, uygulama sadece bir konsol ortamı olarak değil, aynı zamanda bir masaüstü platformu olarak da geliştirildi. Bunun için UnitTestGeneratorGUI adlı bir sınıf tasarlandı. Bu tasarım için hem İngilizce hem de Türkçe dillerinde arayüzler oluşturuldu. Şekil 9'de verilen ekran görüntüsü tasarlanan bu sınıfa aittir. Sınıf, framework'ün ilgili bölümlerinin çalışması için gerekli hiyerarşiyi kullanır; tüm komut düğmeleri, listeler ve metin kutuları bu hiyerarşideki sınıflara ait işlevleri gerçekleştirir. Deneyler, Windows 10 işletim sistemine sahip, i7 2.50 GHz CPU ve 16.0 GB belleğe sahip bir bilgisayarda JDK 11 çalıştırılarak gerçekleştirildi.

Çalışma kapsamında çeşitli java sınıfları ile ilgili test sınıfları gerçekleştirildi. Geliştirilen yöntem sadece değişkenler, nesneler ve metotlar üzerinde çalışmıyor. Aynı zamanda, koşullu yapılar ve döngüler gibi ek özellikleri de kapsamaktadır. Şekil 9›da da görüldüğü gibi, arayüz iki ana bölümden oluşmaktadır. İlk olarak, test dosyasının hazırlanacağı java dosyasını sisteme yüklemek için dosya seç düğmesi tıklanır. Dosya yüklendikten sonra, ikinci kısımda sekmeler yer almaktadır. Seçilen Java sınıfının içeriği, bir döküm olarak Java Kod Dosyası penceresine gelir. Sınıf kodlarındaki her bir kod bytecode'a dönüştürülecek olsa da, yorum satırlarındaki ifadeler ya da kod cümleleri bytecode'a dönüştürülmez. Bu özellik tüm derleyicilerde bulunmaktadır. Ayrıca şekildeki örnekte, bu sınıfın hem koşul hem de döngü yapıları içerdiği görülmektedir. Bilgi al düğmesine tıklandığında, sınıfın adı ve ait olduğu metodların adları ekranın sağ tarafında listelenmektedir. Bunun için arka panda çalışan GetInfoAboutJavaAndByteCodeFile adlı bir sınıf tasarlandı. Burada öncelikle, dosya yoluyla okunan java dosyası, JavaCompiler kütüphanesi kullanılarak bir sınıf dosyasına dönüştürülmektedir. Bu dönüşümle birlikte ilgili sınıfın bytecode'u da elde edilmektedir. Bu dönüşümler tamamlandıktan sonra ilgili sınıfın adı ve sahip olduğu metotların adları bir dizi değişkeni ile toplanmakta ve bu bilgi ekranın sağ tarafındaki listede gösterilmektedir. Bytecode dosyasını oluştur düğmesine tıklandığında, Opcode ayrıştıma yöntemine ait alt fonksiyon ve özellikler çalışır ve Şekil 10'da gösterilen Bytecode Dosyası sekmesi aktif hale gelir.

Javassist kütüphanesinin yardımıyla, ilgili örnek sınıfın .class dosyası oluşturuldu ve bytecode'a dönüştürüldü. Bytecode'ları, javassist kütüphanesi tarafından belirlenen standarta uygun olarak, metot isimlerine göre ayrılmıştır. TestFile Oluştur düğmesine tıklanıldığında tüm bilgiler MongoDB'ye aktarılır ve FTL dönüşüm işlemleri gerçekleştirilir.

Şekil 11'de üçüncü sekme ile FTL şablonuna ait çıktıların bu ekrana aktarılan test sınıfı ve birim test metotları gösterilmiştir.

Şekil 12'de dördüncü sekme ile MongoDB


```
/* 1 */
{
    "_id" : ObjectId("622e16570fb2df4654f26bbb"),
    "Execution Id" : "'1:1'",
    "Kaynak" : "'public static double org.brutusin.instrumentation.logging.CalculateCredit3.CalculateInterest(double)'",
    "Baslangic Suresi" : "'Sun Mar 13 19:05:43 EET 2022'",
    "Degiskenler" : "'[484.0]'",
    "ClassName" : "'CalculateCredit3'",
    "MethodName" : "'CalculateInterest'",
    "MethodReturnType" : "'double'",
    "MethodParameters" : "'[double arg0]'",
    "ObjectInClass_Field" : "intRate",
    "ObjectClassName_Method" : "InterestRate",
    "Mocking" : "InterestRate.monthlyRate",
    "ImportMethod" : "org.brutusin.instrumentation.logging.InterestRate",
    "ImportField" : "org.brutusin.instrumentation.logging.CalculateCredit3",
    "Toplam Suresi" : "'277 ms'",
    "Returned" : "'5.808'"
}
```


Şekil-8: *kayitlar* isimli koleksiyonun yapısı

- JSon Görünüm dosya formatlarına ulaşılabilmektedir. İki sekme bulunmaktadır. MongoDB yazılımından çekilmiş iki NoSQL formatında koleksiyonlar bulunmaktadır. İlk sekmede; framework tarafından toplanan veriler kaydedildiğinde, ilgili java sınıfının adı ve metod isimleri, aldığı parametreler, nesnelerle ilişkilendirilen değerler, bunlara sahip oldukları sınıf adlarına ait isimler, metodlardan dönen değerler gibi bilgiler içeren kayıtlar koleksiyonu listesidir. Ayrıca, bu listede bir taklit (mocking) olup olmadığı gibi bilgiler de yer alır. İkinci sekmede ise; byteCoding koleksiyonunun verileri listelenmektedir. Java sınıfındaki her metodun döngü ve koşul yapılarının ayrı ayrı analizi yapıldı ve bunlarla ilgili değerlerin özet bilgisi ayrı ayrı toplandı ve kaydedildi.

Literatürde yaygın olarak kullanılan diğer otomatik birim test oluşturma araçlarıyla bu geliştirilen otomatik birim test oluşturma yazılımının ve bu yazılım ile elde edilen çalışma zamanı verilerinin toplanmasıyla gerçekleştirilen işlemlerin karşılaştırması, tartışma, sonuç ve önerilier bölümünde ele alınmıştır.

Bu geliştirilen çalışmanın uygulaması https://github.com/SevdanurGENC/Nano-Automatic-Unit-Test-Generator adresinde yayınlanmıştır.

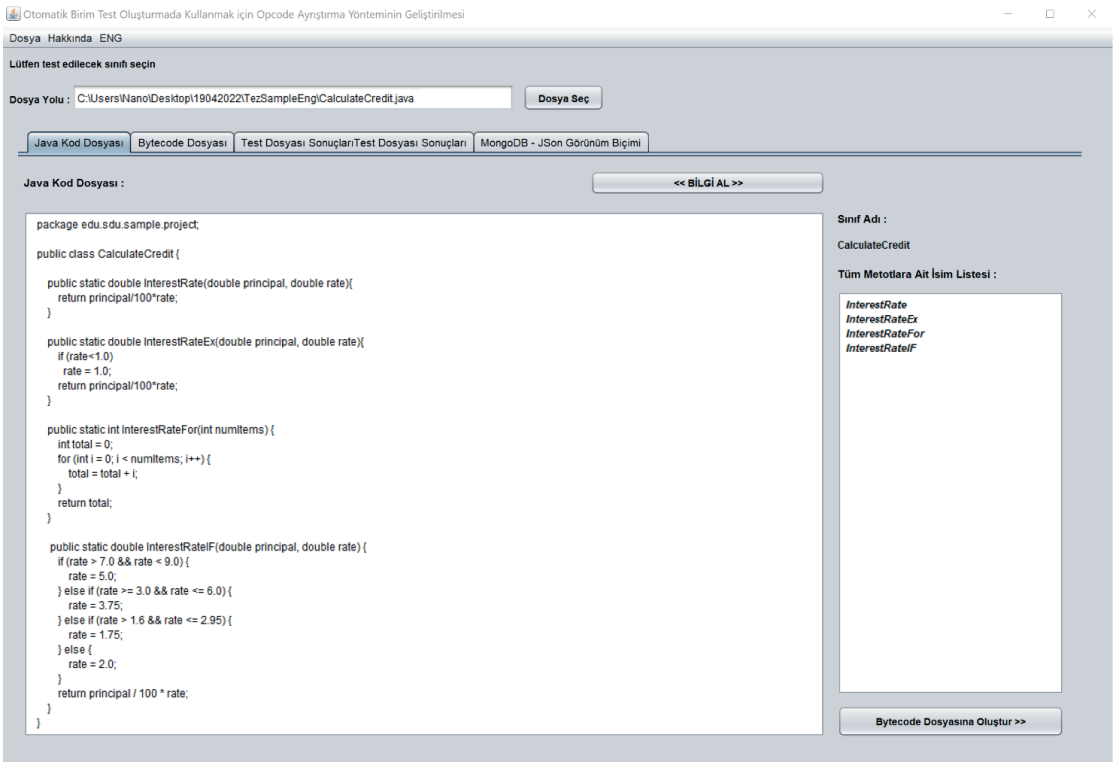
5. TARTIŞMA, SONUÇ VE ÖNERİLER

Bu çalışmada, bir Java Agent yardımıyla çalışma zamanında veri toplayan, bu verileri NoSQL veritabanında depolayan ve bu verileri JTL şablon motorunu kullanarak birim testine dönüştüren bir uygulama geliştirildi. Yapılan çalışmalar ve geliştirilen araç, literatürdeki önceki çalışmalara kıyasla çeşitli açılardan farklı bir yapı kullanmaktadır.

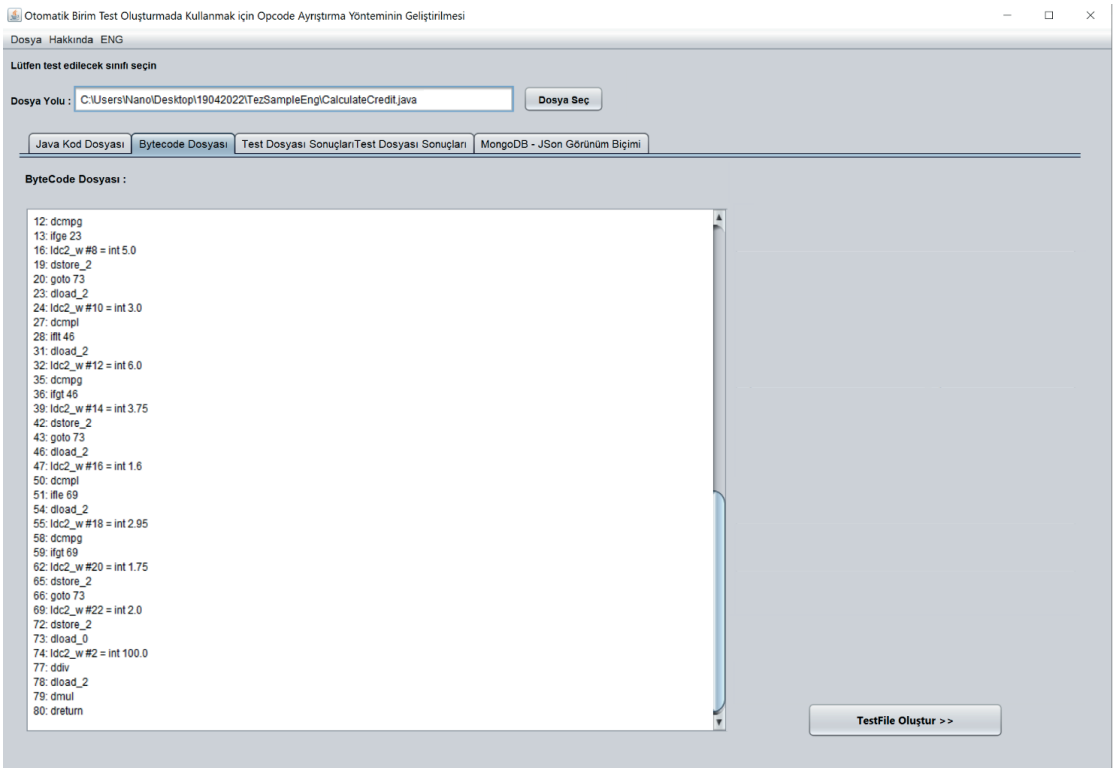
• Opcode ayrıştırma yönteminin üstünlük ve eksikliklerinden bahsedecek olursak;

• Bu çalışmada, önceki literatürdeki bazı çalışmalar test görevlerini otomatize etmek için genetik algoritmalar gibi meta-sezgisel optimizasyon arama tekniklerini kullanmıştır [19]. Ancak bu çalışmada herhangi bir optimizasyon arama veya genetik algoritma yöntemi kullanılmamıştır. Doğrudan, bytecode'ları oluşturulduğu bir java sınıfının opcode'ları, java string metotları kullanılarak ayrıştırılmıştır.

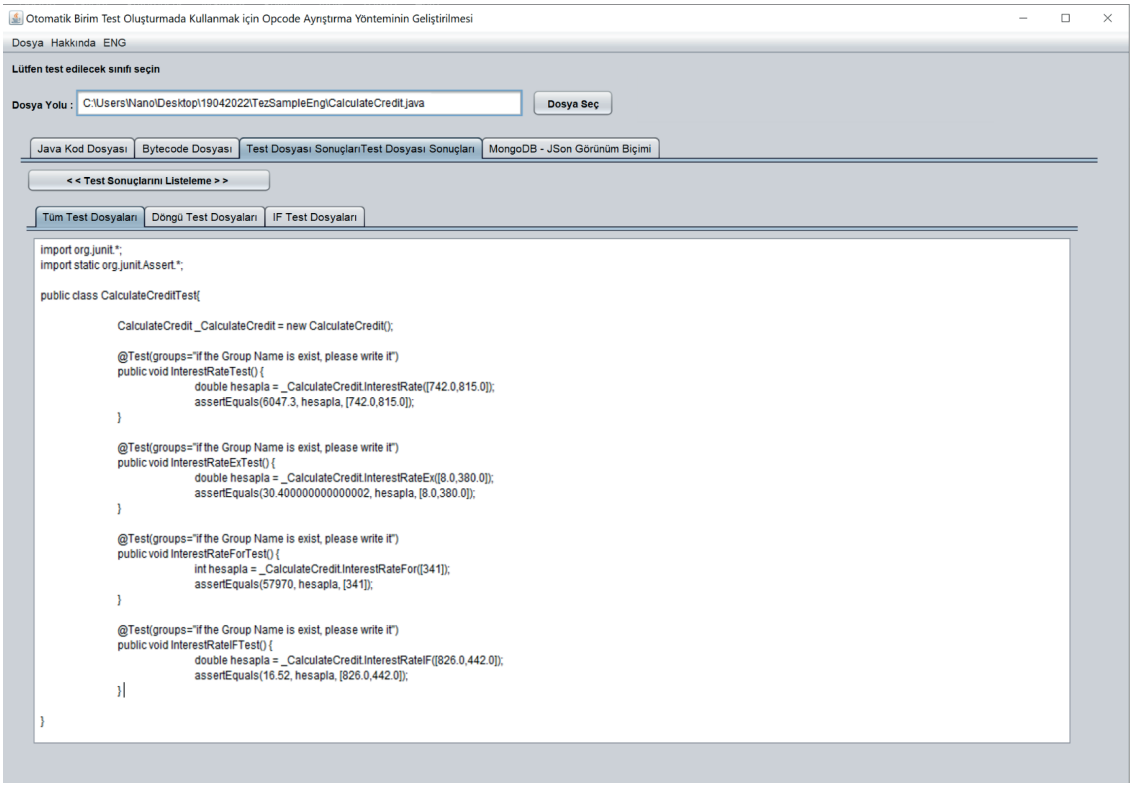
• Bytecode API tarafından sunulan sınırlı ve



Şekil-9: Otomatik birim testi oluşturma Framework'ünün ekran görüntüsü



Şekil-10. Java sınıfının bytecode dönüşümü



Şekil-11: MongoDB'de yer alan *kayitlar* koleksiyonunun JSon formatı

hazır fonksiyonların yanı sıra geliştirilen bu opcode ayrıştırma yöntemiyle, istenen opcode'a karşılık gelen giriş-çıkış parametrelerine bytecode'a dönüştürülen çıktıda erişilebilir. Aynı zamanda, bu geliştirilen yöntem açık kaynak kodlu olduğundan, kullanıcılar her bir fonksiyon içinde kendi ihtiyaçlarına göre istedikleri özellikleri ekleyebilir veya kaldırabilirler.

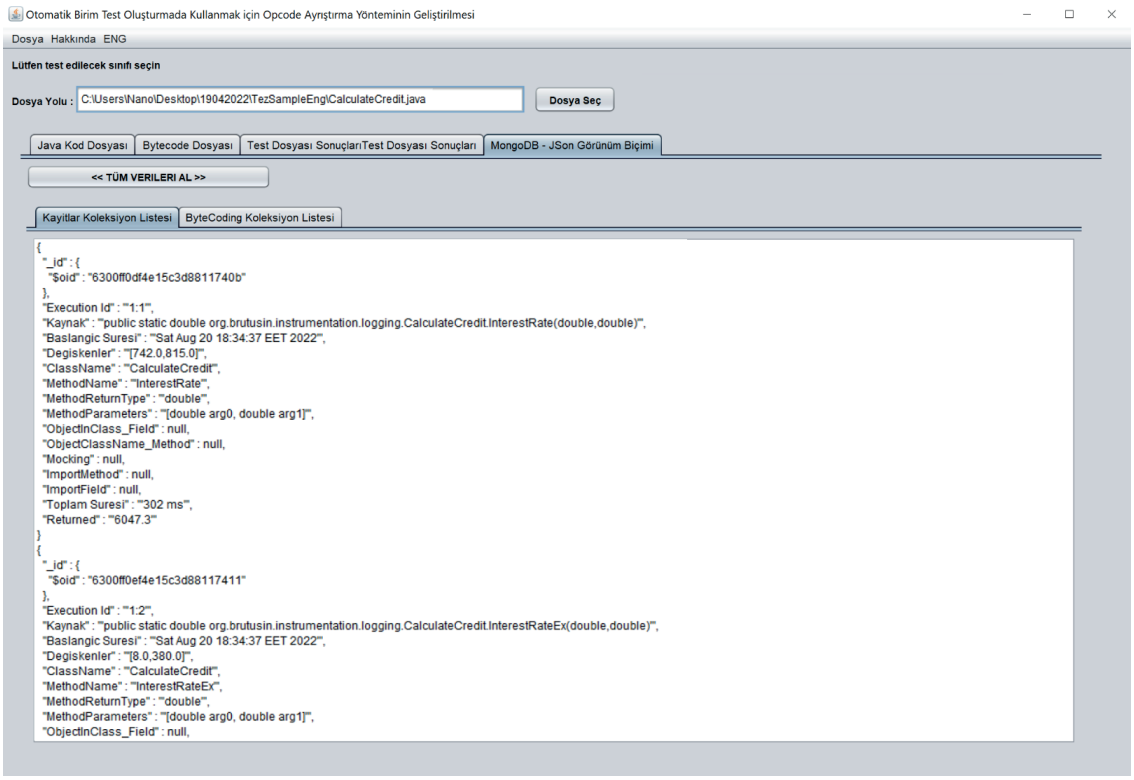
•Bir sınıfta üretilen bir nesneye ait diğer sınıflar aynı yerde olduğu sürece, temel sınıf ile birlikte otomatik olarak işlenebilir. Özellikle mock-stub uygulamaları için düşünüldüğünde, kullanıcıdan ekstra bir eylem gerektirmez ve geliştirilen çerçeve bunu kendisi algılar.

• Bu çalışmada, literatürde Venkatesan ve diğerlerinin çalışmasında olduğu gibi, opcode ayrıştırma yönteminden elde edilen tüm veri türlerini depolamak için esnek bir veri depolama sistemi gerekiyordu [20]. Opcode ayrıştırma yöntemi sonrasında çıktı parametreleri JSON formatında kaydedildiği için, bu veriler herhangi bir NoSQL tabanlı veritabanı yönetim sisteminde kolayca incelenebilmektedir.

• Şimdilik geliştirilen uygulama, belirli bir java sınıfı üzerinde detaylı işlemler yapabilir. Sonraki çalışmalarda geliştirilen bu yöntemle, birbirine bağlı birçok sınıfa ait bir projenin konumundan sonra projenin tüm özelliklerine göre otomatik birim testler oluşturmayı planlanmaktadır.

Test verisi olarak geliştirilen sistemde, int ve double gibi sayısal veri türlerine ait rasgele veriler atanmaktadır. Gelecek çalışmada, tüm veri tiplerini kapsayan rasgele veriler üzerinden otomatik veri seçimi veya hatta içe aktarılabilen örnek bir veri seti planlanmaktadır.

Bu çalışma kapsamında geliştirilen aracın, ulusal yazılım testi alanında da önemli bir yer edineceği düşünülmektedir. Geliştirilen opcode ayrıştırma yöntemi, yazılım testçileri tarafından gerçekleştirilecek birim testlerde etkin bir şekilde kullanılması amaçlanmaktadır. Mevcut formuyla en temel test senaryolarına yanıt verebilen bu araç, bytecode tabanlı bir yapıya sahip olduğundan, çok daha gelişmiş senaryolarda nasıl davranması gerektiği konusunda geliştirilebilecek bir yapıya sahiptir. Bunun en büyük nedenlerinden biri Java'nın açık kaynak kodlu bir sistem olmasıdır. Gelecekte



Şekil-12: MongoDB'de yer alan *kayitlar* koleksiyonunun JSon formatı

ihtiyaç duyulabilecek diğer ilgili bytecode'ları da tercüme edecek olan bu çerçeve için farklı modüller de geliştirilebilir.

Yerli bir ürün olarak hedeflenen otomatik birim test oluşturma yazılımının geliştirilmesiyle, opcode dönüşüm işlemleri gerçekleştirildikten sonra basit bir şekilde otomatik birim testi oluşturulmaktadır.

Bu çalışma şu anda, sınıf içinde oluşturulan nesne bağlantılı diğer sınıfları hariç tutarak, tek bir Java dosyası için çalışmaktadır. Gelecekteki çalışmalarda, önceden tanımlanmış test senaryolarının kullanıcı tarafından seçilmesi ve opcode ayrıştırma yöntemi ile toplanan tüm bilgilerin bu senaryolara direkt otomatik olarak uygulanacağı bir uygulama geliştirilmesi planlanmaktadır. Ayrıca bu ikinci aşamada, framework bütünlüğü olarak birden fazla Java dosyası ekledikten sonra, belirtilen test senaryolarına göre olası otomatik birim testlerinin üretilmesi de hedeflenmektedir. Ek olarak, SF110 sınıf örnekleri gibi en son gelişmelerin takip edildiği örneklerle karşılaştırma yapılacaktır [21]. Kod satırları ve sistemdeki kullanılan bellek süreleri hakkında kıyaslamalar yapılarak analizler hakkında bilgi verilmesi hedeflenmektedir.

Kaynakça

Damar M, Özdağoğlu G, Özdağoğlu A, Eylül Üniversitesi Rektörlüğü D. Küresel Ölçekte Yazılım Kalitesi ve Standartları: Sektörel ve Bilimsel Perspektiften Literatürdeki Eğilimler. DergiparkOrgTr 2018;6:325–48. https://doi.org/10.17093/alphanumeric.404102.

Felice S. JUnit Vs TestNG: Differences Between JUnit and TestNG | BrowserStack n.d. https://www.browserstack.com/guide/junit-vs-testng (accessed January 29, 2023).

Fewster M, Graham D. Software test automation 1999.

Csallner C, Smaragdakis Y. JCrasher: an automatic robustness tester for Java. Softw Pract Exp 2004;34:1025–50. https://doi.org/10.1002/SPE.602.

Pacheco C, Ernst MD. Randoop: Feedback-directed random testing for Java. Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA 2007:815–6. https://doi.org/10.1145/1297846.1297902.

Pacheco C, Lahiri SK, Ernst MD, Ball T. Feedback-directed random test generation. Proceedings - International Conference on Software Engineering 2007:75–84. https://doi.org/10.1109/ICSE.2007.37.

Simons AJH. JWalk: A tool for lazy, systematic testing of java classes by design introspection and user interaction. Automated Software Engineering 2007;14:369–418. https://doi.org/10.1007/S10515-007-0015-3/METRICS.

Sen K. Cute: A concolic unit testing engine for c and java 2007. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Sen%2C+K.+%282007%29.+Cute+%3AA+concolic+unit+testing+engine+for+c+and+java.+Web%2C+%5Bcited+at+p.39%2C+44%5D.+Retrieved+from++http%3A%2F%2Fosl.cs.uiuc.edu%2F%CB%9Cksen%2Fcute%2F&btnG= (accessed January 29, 2023).

Charreteur F, International AG-2010 I 21st, 2010 undefined. Constraint-based test input generation for java bytecode. IeeexploreIeeeOrg n.d.

Fraser G, symposium AA-P of the 19th AS, 2011 undefined. Evosuite: automatic test suite generation for object-oriented software. DlAcmOrg 2011:416–9. https://doi.org/10.1145/2025113.2025179.

Sakti A, … GP-IT on, 2014 undefined. Instance generator and problem representation to improve object oriented code coverage. IeeexploreIeeeOrg n.d.

Tanno H, Zhang X, … TH-2015 I 37th I, 2015 undefined. TesMa and CATG: automated test generation tools for models of enterprise applications. IeeexploreIeeeOrg n.d.

Tzoref-Brill R, Sinha S, … AN-… IC on, 2022 undefined. TackleTest: A Tool for Amplifying Test Generation via Type-Based Combinatorial Coverage. IeeexploreIeeeOrg n.d.

Higo Y, Matsumoto S, … SK-P of the 19th, 2022 undefined. Constructing dataset of functionally equivalent Java methods using automated test generation techniques. DlAcmOrg 2022:2022. https://doi.org/10.1145/3524842.3528015.

Lukasczyk S, International GF-P of the A 44th, 2022 undefined. Pynguin: Automated unit test generation for python. DlAcmOrg 2022. https://doi.org/10.1145/3510454.

Wan B, Dong S, Zhou J, Qian Y. SJBCD: A Java Code Clone Detection Method Based on Bytecode Using Siamese Neural Network. Applied Sciences 2023, Vol 13, Page 9580 2023;13:9580. https://doi.org/10.3390/APP13179580.

Venners B. Bytecode basics : A first look at the bytecodes of the Java virtual machine. 1996. https://www.

infoworld.com/article/2077233/bytecode-basics.html (accessed January 29, 2023).

FreeMarker Java Template Engine n.d. https://freemarker.apache.org/ (accessed January 29, 2023).

on PM-2011 IFIC, 2011 undefined. Search-based software testing: Past, present and future. IeeexploreIeeeOrg n.d.

Venkatesan P, Rozario RG, Fiaidhi J. Junit framework for unit testing. pdf 2020.

Fraser G, and AA-AT on SE, 2014 undefined. A large-scale evaluation of automated unit test generation using evosuite. DlAcmOrg n.d. https://doi.org/10.1145/0000000.0000000.

"This page is left blank for typesetting"

Bu sayfa dizgiden dolayı boş bırakılmıştır

Journal of Quantum Technologies and Informatics Research
JQTAIR

KUANTUM TEKNOLOJİLERİ VE ENFORMATİK ARAŞTIRMALARI
Volume / Cilt: 2, Issue / Sayı: 1, 2024, pp. 31-43
E - ISSN: 3023-4735
URL: https://journals.gen.tr/index.php/jqtair/
DOI: https://doi.org/10.5281/zenodo.10102956

Received / Gelis: 05.01.2024
Acccepted / Kabul: 22.02.2024

RESEARCH ARTICLE / ARAŞTIRMA MAKALESİ

Öğrenen Makineler ve Fasiyes Ayrımı; İlk Sonuçlar

Machines Learning and Facies Discrimination; Prelimineary Results

Ayetullah Ercel[1] **Emin Uğur Ulugergerli**[2]

[1] Çanakkale Onsekizmart Üniversitesi, Mühendislik Fakültesi, Jeofizik Mühendisliği Bölümü, Türkiye, e-mail: ayetullahercel@gmail.com
[2] Çanakkale Onsekizmart Üniversitesi, Mühendislik Fakültesi, Jeofizik Mühendisliği Bölümü, Türkiye, e-mail: emin@comu.edu.tr

Öz

Makine öğrenmesi uygulamalası kuyu loglarından jeolojik istifin ayırtlanması için kullanılmıştır. Kuyu logu verilerinden fasiyesleri tahmin etmek için makine öğrenmesi yöntemi sınıflayıcılarından biri olan değişim arttırıcı türev (gradient boosting) algoritmasından yararlanılarak ağaç tabanlı (tree-based) bir eğitim modeli geliştirilmiştir. Tahmin başarı oranını arttırmak için veri topluluğu üzerinde iyileştirmeler yapılmıştır. Deneme veri topluluğu olarak, Society of exploration geophysics tarafından makine öğrenmesi için önerilen, Kansas (ABD) eyaletindeki kuyu verileri kullanılmıştır. Çalışmada makine öğrenmesi yöntemi olarak değişim arttırıcı türev ile sınıflama algoritması ile tekil deneme kuyusu üzerinde % 57, komşu fasiyes bilgisi ile %88 oranında doğruluk ile elde edilen tahmin sonuçları elde edilmiştir.

Anahtar Kelimeler: makine öğrenmesi, gradient boosting, kuyu logları

Abstract

Machine learning applications have been used to distinguish the geological sequence from well logs. A tree-based training model was developed using the gradient boosting algorithm to predict facies from the well log data set. Testing data were, obtained from Kansas state (USA), recommended for machine learning studies by the Society of exploration geophysics. In the study, the gradient boosting algorithm predicted results with 57% accuracy on the single trial well and with 88% accuracy with using well log information from neighbourhood.

Keywords: machine learning, gradient boosting, well logs

Citation/Atıf: ERCEL, A. & ULUGERGERLİ, E.U. (2024). Öğrenen Makineler ve Fasiyes Ayrımı; İlk Sonuçlar. *Kuantum Teknolojileri ve Enformatik Araştırmaları*. 2(1): 31-43, DOI: 10.5281/zenodo.10102956

Corresponding Author/ Sorumlu Yazar:
Ayetullah Ercel
E-mail: ayetullahercel@gmail.com

GİRİŞ

Yeraltı kaynaklarının aranmasında kuyu logları yaygın olarak kullanılmaktadır (örn. Anderson 2001, Pekiner 2002). Yüksek ayrımlılık gücüne karşın karmaşık jeolojik istiflerde hatalı yorumların yapılması en önemli sorunlardan biridir (örn. Dubois vd. 2007). Bu soruna bir çözüm olarak insan tabanlı ve eldeki var olan veriler ile edinilecek bilgi birikiminin kullanımı en yaygın yoldur (URL1). Bu çalışmada, günümüzde her alanda olduğu gibi yeraltının araştırılmasında da geniş bir kullanım alanı bulan yapay zeka uygulamaları (YZU) kuyu loglarından jeolojik istifin ayırtlanması için kullanılmıştır (örn., Ahmadi vd. 2013).

Jeolojik tanımlamada yaygın olarak kullanılan fasiyes terimi, belirgin bir jeolojik süreci ve birikme ortamını ve koşullarını yansıtan tortul birimlerin ve özelliklerinin genel tanımı olarak kullanılagelmektedir. Fasiyeslerin birbirleriyle olan ilişkilerinde kademeli geçişler izlenebilir ve özellikleri birbirine oldukça yakın komşu fasiyesler oluşabilir. Bu ve benzeri sorunlar litolojik olarak benzer fasiyeslerin sınırlarının tanımlanmasında zorluk oluşturmaktadır (Dubois vd. 2007)

Çalışmada kuyu logu verisinden yola çıkarak fasiyesleri tahmin etmek için makina öğrenmesi algoritması geliştirilmiştir. Algoritmada Python programlama dili için geliştirilen "scikit-learn-SCL" açık kaynak makina öğrenmesi kütüphanesi (Pedregosa vd. 2011) kullanılarak "gradient boosting" (GB) algoritması ile bir sınıflayıcı karar ağacı modeli oluşturulmuş ve tahmin başarı oranını geliştirmek için veri topluluğu üzerinde iyileştirmeler yapılmıştır.

Veri olarak, Society of exploration geophysics (SEG) tarafından YZU için önerilen, Kansas (ABD) eyaletinde elde edilmiş halka açık kuyu verileri kullanılmıştır. Anılan veriler hem SEG tarafından düzenlenen YZU çalıştaylarında hemde Kansas Üniversitesinde Sinir Ağları ve Bulanık Sistemler (URL2) üzerinde yapılan çalışmalarda kullanılmıştır (Bohling ve Dubois 2003, Dubois vd. 2007).

İzleyen bölümlerde kuyu logları kısaca tanıtılmış, YZU nun makine öğrenmesi uygulamalarından biri olan GB ile kuyu logu verilerinin bilgisayar tarafından tanımlanması, sınıflayıcı algoritması ve verilerin sınıflandırılması açıklanmıştır.

Kuyu logları

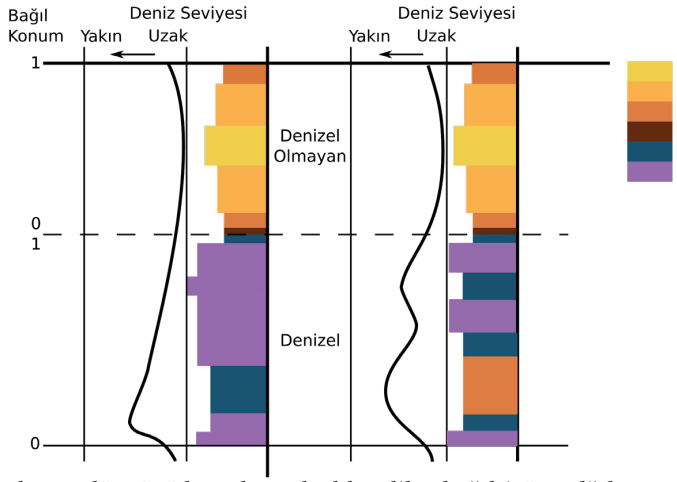
Kuyu logları, jeolojik birimlerin fiziksel özelliklerini doğrudan birim içinde yapılan ölçümler ile belirlemeye çalışan bir jeofizik yöntemdir (örn. Darling 2005). Bu çalışmada beş farklı log verisi ele alınmıştır; gamma ışını (Gamma Ray - GR), özdirenç (Resistivity), fotoelektrik etki (Photoelectric effect - PE), netron-yoğunluk (Neutron - Density - ND) gözeneklilik farkı (Porosity Difference) logları kullanılmıştır (örn. Glover 2000). Bu loglara dahil edilmek üzere iki farklı jeolojik tanımlayıcı koşul (constraint) işlemlere dahil edilmiştir; denizel veya karasal ortam belirleyici (nonmarine-marine indicator - NM_M) ve göreceli konum (relative position- RELPOS).

Elde edilen verilerin yüzey jeofizik yöntemlerden farkı özdirenç, hız vb. fiziksel büyüklükleri doğrudan birim içinde ölçülmesidir. Ancak, elde edilen veriler algılayıcıların (probe) boyutu ile jeolojik birimin kalınlık ilişkisine bağlı olarak gerçek değerinden saparlar. Bu aşamada gerçek fiziksel değerler elde etmek için abaklar (URL1) veya modelleme çalışmaları (örn. Anderson 2001, Ulugergerli 2017) yapılmalıdır. Her iki yaklaşımda kendi içinde sorunlar barındırır. Bunun başlıca nedenleri jeolojik birimlerin tekdüze olmaması kimyasal içeriğinin yer bağımlı olması vb. nedenler sayılabilir. Makine öğrenmesinin log verilerine uygulanması bu soruna çözüm için farklı bir yaklaşım olarak önerilmektedir.

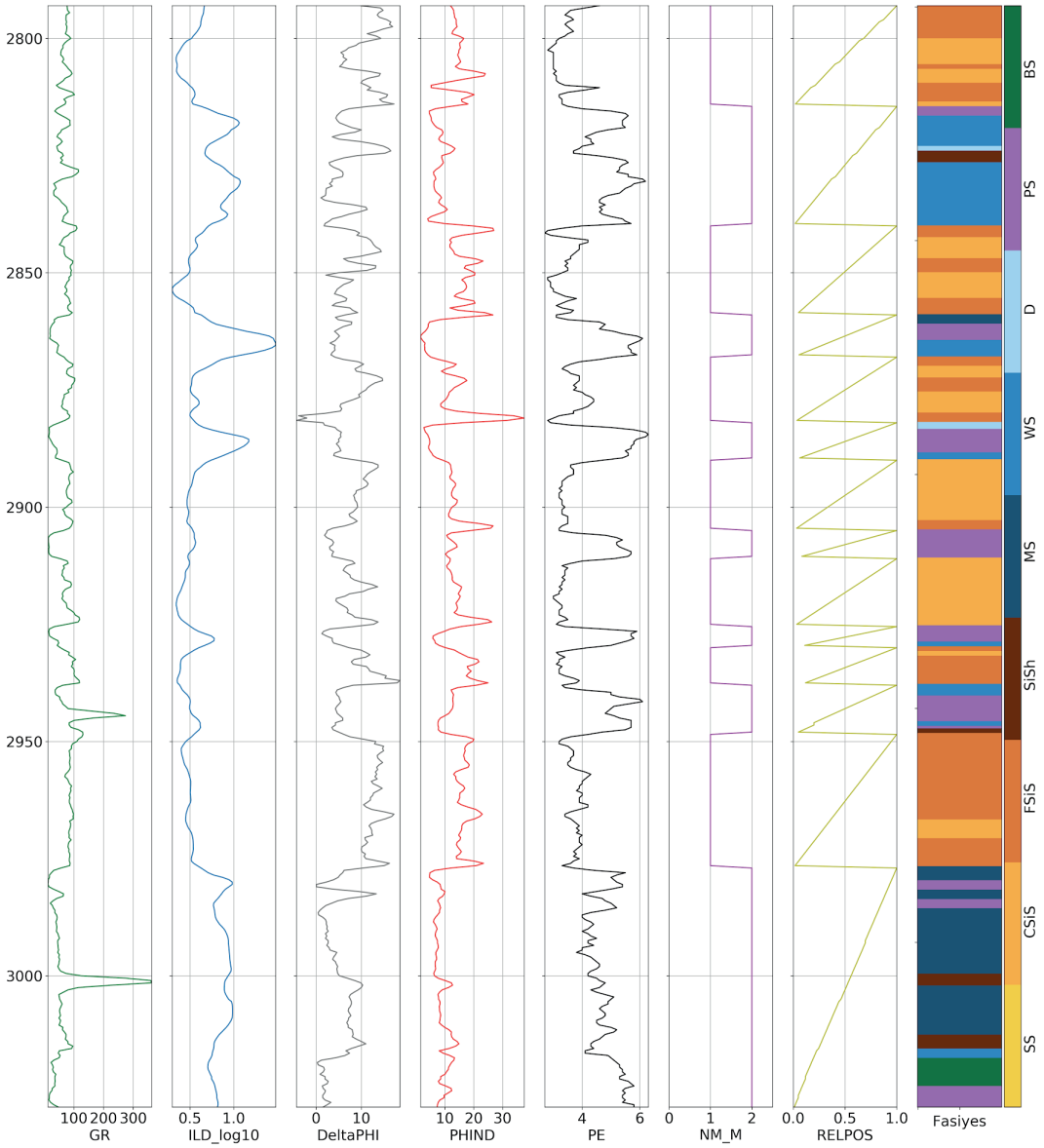
Log verisi stratigrafik olarakta bir anlam taşıdığından bu veri ile ortamın tanımlamasını fiziksel olarak yapmak olasıdır. Bu nedenle, kuyu loglarına göre ayrımlama, kuyu verisinin ön işlemesinin önemli bir adımıdır. Log eğrilerinden elde edilen sonuçların karşılaştırılması, önerilen modelin kuyunun bulunduğu bölgesindeki birimlere uyumu doğruluğunun değerlendirilmesine olanak sağlar.

Öğrenen makinelerin katkısı

Fasiyeslerin sınıflandırılması sedimanların ölçülen belirli fiziksel veya kimyasal özelliklerine



Şekil 1. Sedimanter kayaç döngüsü baz alınarak elde edilen bağıl (göreceli) konum ve ortam belirleyicisinin fasiyeslere göre etkisinin eğri ile ifadesi. (Dubois vd 2007'ten değiştirilerek alınmıştır.)



Şekil 2. Kuyu log verisi örneği

göre bir gruba atanması işlemidir. Sınıflandırma için kuyudan çıkarılan karotların analizlerinden yararlanılır, fakat maliyetinden dolayı yaygın değildir (örn. Bestagini vd. 2017).

Buna karşın dolaylı bir yaklaşım olan kuyu logları da fasiyeslerin sınıflama işlemlerinde kullanılabilir. Ancak log verilerinin geleneksel yöntemler yardımı ile kullanımı oldukça zahmetli ve zaman gerektiren bir süreçtir. Bu nedenle, çeşitli alternatif yaklaşımlar önerilmiştir. İlk çalışmalarda istatistiksel yöntemler geliştirilmiştir (Wolf vd. 1982, Busch vd. 1987). Daha sonra Wolf vd. (1982) ile Busch vd. (1987) sınıflandırma işlemi için yapay sinir ağlarının kullanılmasını önermiştir (Baldwin vd. 1990, Rogers vd. 1992).

Günümüzde özellikle son birkaç yılda bilgisayarların hesaplama güçlerinin artması ve depolama alanlarının genişlemesine koşut olarak gelişmekte olan büyük veri (big data) olgusu ile, makina öğrenmesi teknikleri farklı alanlarda ele alınmaya başlamıştır. Makina öğrenmesi teknikleri günümüzde araştırma grupları tarafından da irdelenmeye başlanmıştır (Smith ve Treitel 2010, Zhang vd. 2014, Zhao vd. 2015, Kobrunov ve Priezhev 2016).

Bu bağlamda Hall (2016) tarafından makina öğrenmesi teknikleri ile basit bir fasiyes sınıflama eğitimi sunmuştur. Eğitimde küçük bir veri topluluğu olarak Kansas'ın güneyindeki Hugoton gaz alanında bulunan 10 farklı kuyudan 7 log bilgisi kullanılmıştır. Bu veri topluluğundaki loglar yorumlanarak ve karotiyer bilgisi kullanılarak fasiyes sınıflaması yapılmıştır (Dubois vd. 2007). Bu veri topluluğundan yola çıkarak fasiyes bilgisi bilinmediği kabul edilen bir deneme kuyu verisinin sonuçları tahmin edilmiştir.

Makina öğrenimi teknikleri veri topluluğunun niteliklerine göre danışmalı öğrenme (supervised learning - SL), danışmasız öğrenme (unsupervised learning -USL) ve türevi olan yöntemlerden oluşur. SL, önceden gözlemlenmiş ve sonuçları bilinen verileri kullanarak bu sınıflamayı tekrardan üretebilecek model parametrelerinin belirlenmesini tanımlar. USL ise giriş çıkış ilişkisini ele almadan verinin kendi içindeki örüntüyü kullanarak tanımlama işlemidir (Nizam ve Akın 2014). Bu çalışma için SL yaklaşımı kullanılmış ve $f_{d,w}$ girdisi ve $y_{d,w}$ çıktısı eşleştirilerek öğrenme (eğitim) sağlanmıştır. Eğitimi izleyen aşamada, eğitilen modelden yola çıkarak bilinmeyen verinin sınıflandırma işlemini algoritmaya yaptırılmaktadır (örn. Alpaydın 2009).

Bu amaçla, kontrol edilmiş ve sınıflanmış veri topluluğundan yola çıkarak özellik vektörünü $f_{d,w}$

$$f(x)_{d,w} = [f_{d,w}^{GR}, f_{d,w}^{OZD}, f_{d,w}^{FE}, f_{d,w}^{Nfark}, f_{d,w}^{Nort}, f_{d,w}^{NM}, f_{d,w}^{GK}] \quad (1)$$

olarak tanımlayabiliriz. Önceki çalışmalarda (Hall 2016, Bohling ve Dubois, 2003, Dubois vd. 2007) sunulan aynı düzenlemeyi varsayarak, ile işaretlenen her bir kuyu için derinlikte yedi farklı sayıl log değeri bulunmaktadır. Bunlar;

- Gamma ray $(f_{d,w}^{GR})$ doğal oluşum radyoaktivitesini,
- Özdirenç $(f_{d,w}^{OZD}$ yerin elektrik akımı karşı gösterdiği direncini
- Fotoelektrik etki $(f_{d,w}^{FE})$ ışık ışınları ile aydınlatılan fasiyeslerin elektron emisyonunu,
- Netron yoğunluk- porozite farklı $(f_{d,w}^{Nfark})$, ortalama netron-yoğunluk porozite : $(f_{d,w}^{Nort})$ fasiyes yoğunluğu ile ilişkisini sunan loglardır. Bunlara ek olarak;
- Karasal /Denizel belirleyici $(f_{d,w}^{NM})$,
- Göreceli konum $(f_{d,w}^{GK})$

veri incelemesi veri topluluğuna dahil edilmiştir. Fasiyes sınıfları (etiketleri)$(y_{d,w})$

$$y_{d,w}^{\square} \in \{SS, CSiS, FSiS, SiSh, MS, WS, D, PS, BS\}$$

Kümesi ile tanımlanmıştır. SL' de $f(x)_{d,w}$ girdisi ile çıktısını birbirine eşleyen bir işlev (sınıflayıcı) aşağıdaki gibi bir görselleştirilebilir.

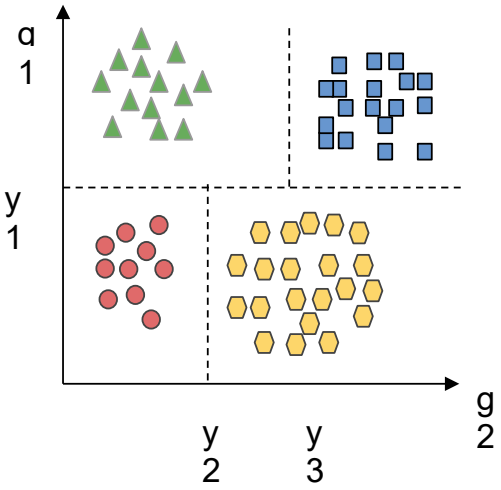
Verilerin Sınıflanması

Eğitimde kullanılacak verilerin sınıflandırılması amacıyla için birçok farklı algoritma geliştirilmiştir (örn. Bishop 2006). Çalışmada kullanılan veri topluluğunun yedi özellikten oluşması ve sınırlı miktarda eğitim verisi olduğunu göz önüne alınarak sınıflandırma işlemi için basit bir karar ağacı topluluğu algoritması kullanılmıştır (Breiman vd. 1984, Friedman 2000). İzleyen bölümde yöntem

hakkında kısa bir bilgi verilecektir. Konunun ayrıntılı açıklaması için Bishop (2006) ve Alpaydın (2009) çalışmalarına bakılabilir.

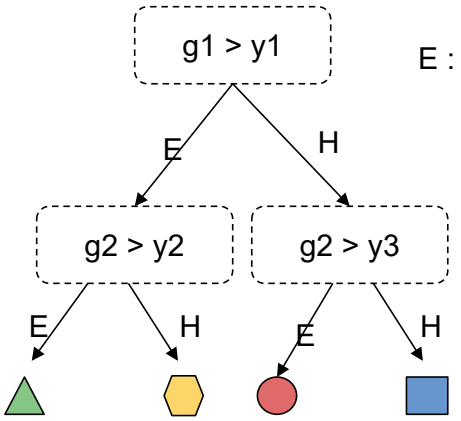
Karar ağaçları verilerin sınıflandırılması amacıyla kullanılan başlıca tekniklerden biridir. Sınıflandırma işlemini veriden elde ettiği kuralları öğrenerek yapar. Bir ağaç yapısını andıran bu yöntemde ağaç düğümleri özellik vektörünün elemanlarını belirtirken, ağacın dalları ise bu elemanlara yapılacak işlemleri göstermektedir (Mitchell 1997, Drucker 1996).

Karar ağacı sınıflayıcısı izleyen örnekle açıklanabilir. Eşitlik 1 de verilen özellik vektörü İki sayıl özellik barındırsın, $f = [f(g1), f(g2)]$ sınıf kümesi, y ise dört farklı fasiyes (sınıf, etiket) içersin.



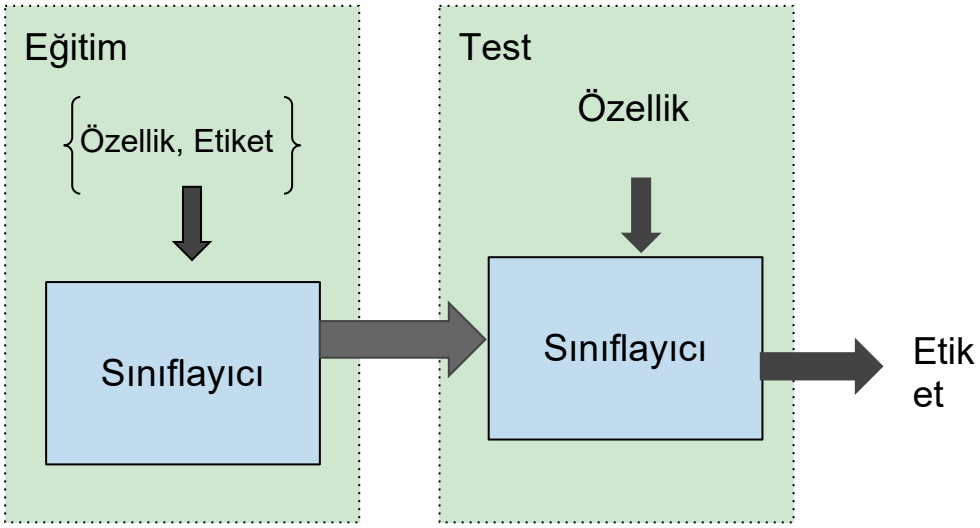
Şekil 4. Özellik vektörünün farklı renkler ve şekil ile tanımlanmış sınıflar için dağılımı

Bu örnekte, (g1, g2) her biri farklı bir sınıfa ait özelliği içeren dört bölüme kolayca ayrılabilir. Eğitim sırasında, karar ağacı sınıflandırıcısı, farklı sınıflara ait özellikleri ayırmak için verileri bölerek eşik değerlerini (örn., y1, y2 ve y3) öğrenir (belirler). Bu işlem uygun bir şekilde tanımlanmış bazı amaç fonksiyonlarını en küçükleyerek yapılır (Rokach ve Maimon 2005). Sınıflayıcı modeli bilinmeyen veriye uygulandığında, veriyi sınıflandırmak için, g1 ve g2 bileşenlerini öğrenilen üç eşikle karşılaştırmak yeterlidir. Bu adım, mantıksal olarak bir ağaç diyagramı olarak gösterilebilir (Şekil 5). Bu nedenle bu tür sınıflandırıcılar karar ağacı olarak bilinir (Breiman vd. 1984).



Şekil 5. Karar ağacı diyagramı

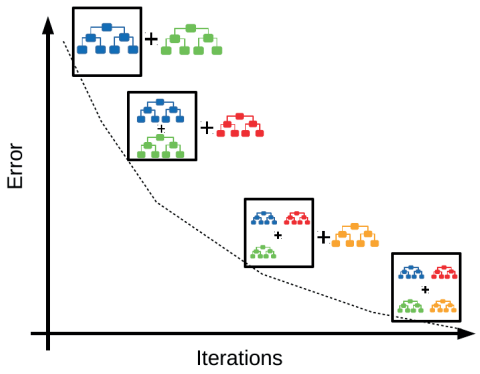
Uygulamada sık rastlanılan sorunlardan biri, algoritmanın eğitim setini "ezberleme" eğilimi göstermesidir. Bunun sonucunda elde



Şekil 3. Önerilen algoritmanın genel hatları, eğitim ve deneme aşamaları.

edilen eşik değerleri başka veri setlerine uyum gösteremezler (Ho, 1998). Bu etkiyi önlemek için, birkaç küçük ağaçtan elde edilen sonuçların birleştirilmesi yönünde çeşitli yaklaşımlar geliştirilmiştir (Ho 1995, 1998, Friedman 2000).

Schapire (1990) tarafından önerilen değişim arttırıcı (boosting) algoritma 2000'li yıllara kadar gelişim göstermiştir (Friedman 2001, Friedman vd. 2000, Freund vd. 1996). Anılan algoritmada amaç zayıf temel öğreniciler ile güçlüleri bir araya getirip güçlü bir sınıflandırıcı oluşturmaktır. Bu amaç için geliştirilmiş birçok algoritma olmakla birlikte anılan çalışma Friedman (2001) tarafından geliştirilen değişim arttırıcı türev sınıflayıcı (Gradient Boosting Classifier – GBC) yaklaşımına dayalıdır ve gözlemleri farklı alt kümelere bölerek çalışır. Her alt kümeden, türev tabanlı hatayı en küçükleme (gradient descent-GD) üzerine kurulu bir algoritmayı çözerek yeni bir karar ağacı oluşturmak için sınırlı oranda özellik seçilir (Friedman, 2001). GBC yöntemi özelliği gereği önceki (doğru olmayan) eğitim sonuçlarından yola çıkarak farklı bir model oluşturur. Her yeni model bir önceki modellerin genel hatasını en küçüklemeye çalışır. Yeni bir özellik alan f vektörünün sınıflandırılması gerektiğinde, ilk önce oluşturulan her farklı ağaçta denenir. Her ağaç bir aday sınıfı sağlar. Tüm ağaçlardan elde edilen sonuçlar daha sonra tek bir kararda birleştirilir (y^t). Böylece karar ağaçları öğrenme tekniği ile eğitilen sistem için bir tahmin modeli üretilmiş olur. Sisteme eklenen her yeni modelin genel hatayı nasıl düşürdüğü Şekil 6'de gösterilmiştir (Kalaycı, 2018).



Şekil 6. Sistemin model eklenerek en küçüklemesi (Kalaycı 2018'dan değiştirilerek alınmıştır.)

Bu bölümden ilk olarak Friedman (2001) tarafından ortaya atılmış öğrenme algoritması GBC'nin basitleştirilmiş mantığını sunacağız. Algoritmaların ve özelliklerinin matematiksel çıkarımı bu makalede ele alınmayacaktır.

SL yönteminde, eğitim seti için f özellik vektörüne karşılık gelen etiket değerleri olmak üzere veri topluluğu uzunluğu için amaç $\hat{f}(x)$ fonksiyonunu bir kayıp (loss) fonksiyonu $L(y, f(x))$ ile $\hat{f}(x)$ fonksiyonu için

$$\hat{f}(x) = min \sum_{xy}^{\square} [L(y, f(x))] \quad (2)$$

biçiminde bir yaklaşımda bulunmaktır. Burada $[L(y, f(x))]$ türevlenebilir bir kayıp fonksiyonunu tanımlar. Friedman (2001) tarafından geliştirilmiş bu yöntem en küçükleme için kayıp fonksiyonu olarak genellikle Eşitlik 3'de bulunan karesel hata ortalamasını kullanır.

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x)_i^t)^2 \quad (3)$$

Burada n veri topluluğu büyüklüğü, y_i $i.$ hedef değeri, $f(x)_i{}^t$ tahmin değeri, olmak üzere $L(y_j, f_j{}^t)$ kayıp fonksiyonunu tanımlar. tane veri için iyileştirilmiş tahmin modeli,

$$\hat{f} = min\{\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))\} \quad (4)$$

şeklinde elde edilir. GB nin amacı, değerlendirilen kayıp fonksiyonu $u^{[k]}$ yı $u^{[k-1]}$ in negatif gradyan vektörüne

$$u_{i,k} = -[\frac{-\partial L(y_i, f(x_i))}{\partial f(x_i)}]_{f(x)=f_{k-1}(x)} \quad (5)$$

şeklinde kestirmektedir. GB algoritması tahmini zor verilere odaklanır ve yineleme yoluyla yeterince eğitilemeyen model parametrelerinin kestirilebilirliğini arttırır (Mayr vd. 2014).

Karışıklık Matrisi ve Başarı ölçütü

Sınıflandırma yöntemlerinden elde edilen modellerin başarılarını değerlendirmek için doğruluk (Accuracy), duyarlılık (Recall), kesinlik (Precision) ve F1-ölçütleri kullanılır. Başarı, doğru ve yanlış sınıflara atanan örneklerin sayı nicelikleri ile ilgilidir.

Sınıflandırıcının tahmin yeteneğini değerlendirmek amacıyla hedef değişken için tahmin değeri ile gerçek değeri karşılaştırılır. Ulaşılan sonuçların başarı bilgileri karışıklık matrisi ile ifade edilir. Bu matriste satırlar değerlendirilen verideki örneklere ait gerçek sayıları, kolonlar ise tahmin değerlerini ifade eder (Tablo 2).

2x2 örnek olarak verilen matriste DN, YP, YN ve DP modelin gözlem değerlerini belirtmektedir. DN ve DP köşegeni doğru tahmin değerlerini, YP ve YN köşegeni yanlış tahmin değerlerini göstermektedir.

Model başarısının ölçülmesinde kullanılan doğruluk oranı, doğru sınıflandırılmış örneklerin toplam örnek sayısına oranı ile elde edilir.

$$Doğruluk = \frac{DP+DN}{DP+DN+YP+YN} \quad (6)$$

Kesinlik değeri, doğru olumlu (pozitif) olarak sınıflandırılan gözlem sayısının, tahmin değerleri olumlu sınıfı olan tüm gözlemlere oranı şeklinde hesaplanır.

Tablo 1. GBC algoritmasi (Friedman (2000) den değiştirilerek alınmıştır).

Algoritma: Türev Takviyeli Sınıflayıcı Algoritması

Girdiler

Girdi verisi $(f(x),y)_{i=1}^{n}$
Yineleme sayısı M
kayıp fonksiyonu $L(y,f)$
Bir dizi temel öğrenici $h_1(x_1),..h_p(x_p)$ özelleştirilir.

Algoritma

1. Modeli bir tahmini $\hat{f}_0$ ile başlat
2. Döngü $k = 1,\ldots,n$ yap
3. Negatif gradyeni hesapla $u_k(x)$
   $u_{i,k} = -[\frac{-\partial L(y_i,f(x_i))}{\partial f(x_i)}]_{f(x)=f_{k-1}(x)}$
   $i = 1,\ldots n$
4. Her bir temel öğrenici $h_j$ye u_k negatif gradyan vektörü uyarlanır.
   $u_k \rightarrow^{temelöğrenici} \rightarrow h_{k,j}(x_j)$❑ $\quad j = 1,..,p$
5. en iyi azalan türev değerini ρ_t için :
   $\rho_t = min_\rho \sum_{i=1}^{n} L[y_i, \hat{f}_{k-1}(x_i) + \rho h_{k,j}(x_i)]$
6. kestirim fonksiyonunu güncelle
   $\hat{f}_k \leftarrow \hat{f}_{k-1} + sl.\rho_k h_{k,j}(x_j)$ Burada sl (küçük adım uzunluğu) öğrenme oranıdır. 0<sl<=1
7. Döngü sonu

Tablo 2. karışıklık matrisi

	Tahmin edilen olumsuz	Tahmin edilen olumlu
Gerçek olumsuz	Doğru olumsuz (DN)	Yanlış olumlu (YP)
Gerçek olumlu	Yanlış Olumsuz (YN)	Doğru olumlu (DP)

$$Kesinlik = \frac{DP}{DP+YP} \quad (7)$$

Duyarlılık, doğru sınıflandırılmış olumlu örnek sayısının toplam olumlu örneklere oranı olarak elde edilir.

$$Duyarlılık = \frac{DP}{DP+YN} \quad (8)$$

F1-ölçütü, kesinlik ve duyarlılık değerlerinin harmonik ortalamasıyla elde edilir. 0 ile 1 arasında olan bu ölçüt değeri 1'e yaklaşması modelin başarısını gösterir.

$$F1 = 2 \times \frac{Kesinlik \times Duyarlılık}{Kesinlik + Duyarlılık} \quad (9)$$

Uygulama

Kullandığımız veri topluluğu (Hall 2016), fasiyes tipleri etiketlenmiş 10 farklı kuyudan (4149 örnek) elde edilen yedi farklı log kaydından oluşmaktadır (Şekil 7). Birçok makine öğrenme problemi için bu, küçük bir eğitim seti olarak kabul edilebilir.

Sınıflandırma yöntemlerinin veriye doğrudan uygulanması yerine, verinin önceden çeşitli yöntemlerle sayısal değere dönüşümü ile daha fazla başarı sağlandığı bilinmektedir. Özellik vektörünün elemanları arasında büyük ölçek farkı varsa, hepsinin aynı ölçeğe getirilmesi hem öğrenmeyi hemde öğrenme başarısını arttırır. İzleyen bölümlerde veri topluluğuna sınıflandırma algoritmalarını uygulamadan önce ölçeklendirme işlemi yapılmıştır. Ölçekleme için SCL kütüphanesindeki "StandardScaler" ölçekleyicisini kullanılmıştır (Pedregosa vd. 2011).

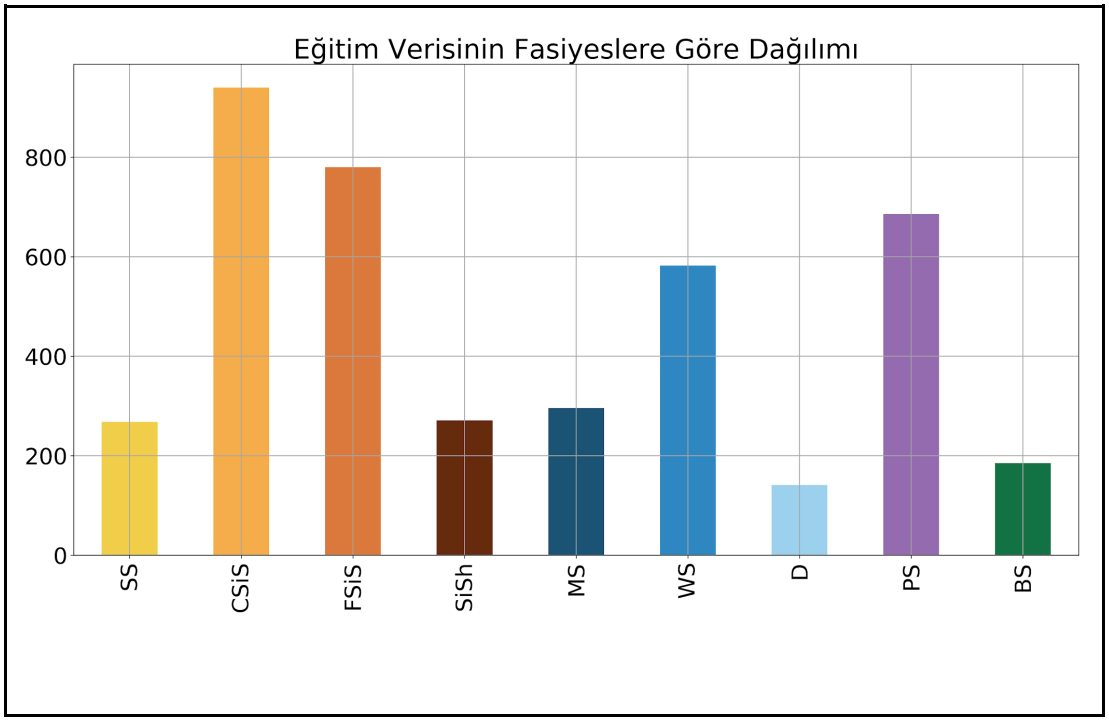
Tabloda 3 çalışmaya konu olan fasiyeslerin kısaltılmış etiketleri ve komşuları tanımlanmıştır. Fasiyes ile ilgili bilgiler karotiyer ile örnek alınarak ve sondaj sırasında yüzeye ulaşan kırıntılardan elde edilmiştir. Üç fasiyes karasal kökenli (SS, CSiS, FSiS) ve altı tanesi denizel kökenli (SiSh, MS, WS, D, PS ve BS). Fasiyelerin birbirlerine yakınlığı karışmalarına neden olur. Fasiyesler ve bunların yakın komşuları, Tablo 3 de kısaltmalar ve tutarlı renklerle belirtilmiş ve işaretlenmiştir.

Fasiyesler içinde kireçtaşı türü olan "bafflestone" (BS) kısıtlı sayıda yer almaktadır. Bu nedenle fasiyesin sınıflama yetkinliğini, diğer bir deyişle tahmin edilebilirliğini, arttırmak için eldeki veri setine ek olarak, tamamı BS taşından oluşan bir kuyu daha eklenmiştir (Dubois et al. 2007).

PE log değeri veri topluluğundaki 4149 örnekten sadece 3232 örnekte bulunmaktadır. Bu problemi ortadan kaldırmak için eksik kalan PE değerleri için sıfır değeri atanmıştır.

Komşu katmanlardaki fasiyeslerin ilişkili olduğunu göz önünde bulundurarak, eğitim sonrası sınıflandırmayı iyileştirmek için izleyen işlem yapılmıştır; Bütün fasiyesin tek bir birimden oluştuğu varsayılırsa, buna bağlı olarak farklı tek bir fasiyes bu hakim fasiyesin içinde yer alırsa bunun tek sebebinin yanlış sınıflandırma hatasından kaynaklandığı şeklinde yorumlanabilir. Bu tür hataları gidermek için, elde edilen sınıf tahminlerine () bir medyan süzgeç uygulanabilir. Bu süzgeç, ara katmanlardaki fasiyesleri en fazla sayıdaki komşu fasiyesle değiştirir.

Giriş verisi olarak veri topluluğunun %25i doğrulama verisi %75i eğitim verisi olarak seçilmiş ve çok-sınıflı (9 farklı fasiyes) bir problemle başa çıkmak için bir seçim stratejisi olan bire-bir (One-vs-One) işlemi uygulanmıştır (Bishop, 2006). İlk adım olarak deneme yanılma yolu ile GBC algoritmasının en iyi çözün sağladığı, yineleme sayısı, öğrenme oranı, özellik sayısı vb parametreleri tanımnıştır. Eğitim sonucunda, eğitimin başarısı önceki bölümde tanımlanan başarı ölçütleri çerçevesinde değerlendirilmiştir. Şekil 7a'da, önerilen yöntemle elde edilen karışıklık matrisini göstermektedir. Köşegenleri baskın bir dizey iyi bir sınıflayıcı sonucuna işaret etmektedir. Daha sonra, eğitimli modeli test etmek için fasiyes bilgisi içermeyen iki kuyu seti (474'ün STUART ve 356'nın CRAWFORD'a ait 830 örnek) elde edilen modelin başarısını gözlemlemek için kullanılmıştır. Son olarak, Hall (2016) tarafından önerildiği gibi, algoritmayı f-ölçüsü açısından değerlendirilmiştir.



Şekil 7. Veri topluluğunun fasiyeslere göre dağılımı

Tablo 3. Kuyu log analizlerinden elde edilen fasiyesler komşuluk ilişkileri ve veride karşılık geldikleri etiket ve renk tanımlamaları.

Fasiyes Renk Temsili	Fasiyes No.	Etiket	Fasiyes	Komşu fasiyesler	
	1	SS	Nonmarine sandstone (Tr) Karasal kumtaşı	2	Denizel olmayan
	2	CSiS	Nonmarine coarse siltstone İri taneli silttaşı	1,3	
	3	FSiS	Nonmarine fine siltstone Karasal taneli ince silttaşı	2	
	4	SiSh	Marine siltstone and shale Denizel silt taşı ve şist	5	Denizel
	5	MS	Mudstone (limestone) Çamurtaşı	4,6	
	6	WS	Wackestone (limestone) Kireçtaşı	5,7	
	7	D	Dolomite Dolomit	6,8	
	8	PS	Packstone-grainstone (limestone) Taneli Kireçtaşı	6,7,9	
	9	BS	Phylloid-algal bafflestone (limestone) Algil Engeltaşı	7,8	

BULGULAR VE TARTIŞMA

Bazı fasiyelerin neden diğerlerinden daha iyi belirlendiğini açıklamak için, Şekil 7b'de her bir kuyuda hangi birimden toplam kaç adet gözlemin bulunduğuna ilişkin bir tablo verilmiştir. Burada her bir fasiyes için farklı sayıda örneğin olduğu görülmektedir, bu durum sınıflandırma problemini dengesiz hale getirir. Örneğin, çamurtaşı (MS), veri topluluğunda iri taneli silttaşına (CSiS) nazaran daha az örnek vardır, bu nedenle, sınıflandırıcının bu fasiyesi yüksek doğrulukla tahmin etmesi beklenemez. Bu durumun aksine, denizel olmayan iri taneli silttaşı (CSiS) örneği daha yaygındır. Bundan dolayı en iyi tanımlanan fasiyelerden biridir.

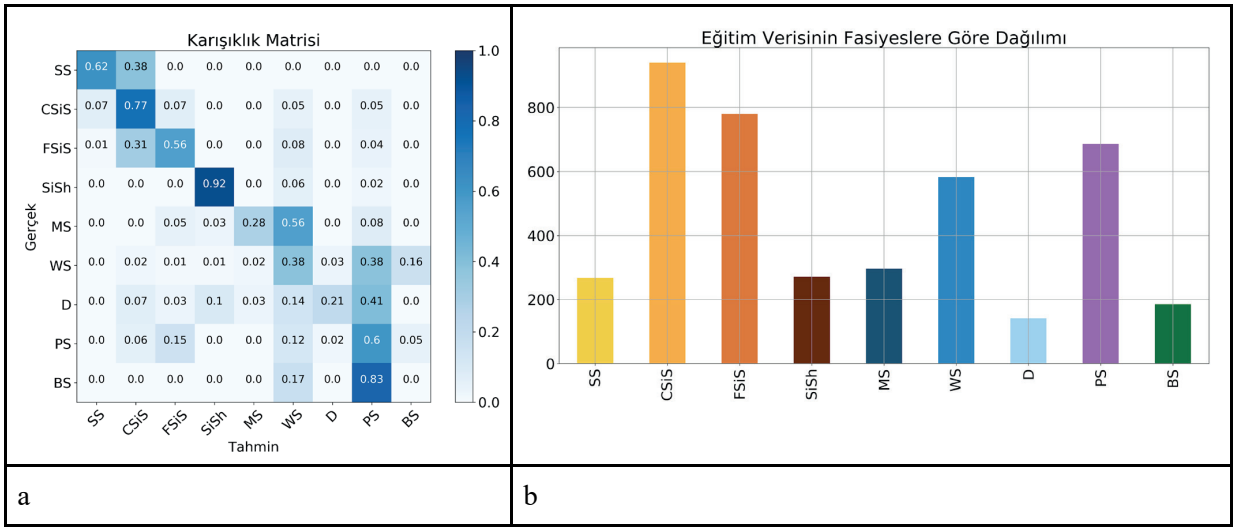
Buna göre, fasiyeslerin doğruluk oranları Şekil 7a'de görüldüğü üzere, bazı fasiyesleri sınıflandırmanın diğerlerinden daha kolay olduğunu görülmüştür. Örnek olarak, Karasal iri silt taşının (CSiS) %77'si doğru şekilde tespit edilmiştir. Bunun aksine, çamurtaşı (MS) ile (WS) birbirine yakın değerler ürettiğinden (0.28, 0.38) doğru tahmin edilebilmeleri zorlaşmıştır.

Fasiyes sınıflandırmasına önerilen yaklaşımın uygulanması, denizel ve denizel olmayan fasiyes için önemli sonuçlar göstermiştir. Jeolojik olarak kısıtlayıcı bir girdinin eklenmesi, modelin tahmin başarısını artırmakla birlikte alana özgü heterojenliği ve değişkenliği tespit etmekte etkili olduğu gözlenmiştir. Deneysel sonuçlar, denizel ve denizel olmayan bölgeleri NM_M eğrisi ile doğrulanan bir şekilde sınıflandırıldığını göstermektedir. (Şekil 8a)

Şekil 8. a) Eğitim modeli ile fasiyes bilgisi tespit edilen kuyunun sınıflandırma sonucu. b) Karşılaştırılan bir diğer algoritma sonucu

Hall (2017), sınıf dengesizliği nedeniyle tahmin sonucu 0,16 başarı değeri civarında olduğunu vurgulamıştır. Algoritmamız, 0.57 f başarı değerine ulaşmaktadır. Fasiyeslerin komşuluk ilişkileri baz alınarak değerlendirildiği takdirde bu sonuç 0,87 başarı değerine kadar yükselmektedir.

Kullandığımız veri topluluğu sınırlı sayıda log verisi içermektedir. Veri topluluğunda yeterince temsil edilmeyen fasiyesler için daha düşük doğrulukta tahminler elde edilmiştir. Gelecekteki çalışmalarda modelin (algoritmanın) doğruluk oranını iyileştirmek amacıyla ek log verisi (yoğunluk, Vs, Vp, vb.) veri topluluğuna dahil edilebilir. Örneğin Vp/Vs oranı doğrudan poisson oranı ile ilişkidir ve bu da fasiyes biriminin yanal ve eksenel gerilmesini tanımlar (Gercek, 2007). Bu bilgi de fasiyes sınıflamasında bir başka belirgin özellik olarak kullanılabilir. İlave jeolojik özelliklerin (örn. her fasiyenin mineralojisi) log verileri ile birlikte kullanımının, karotlu kuyuların fasiyes sınıflandırmasında önemli bir gelişmeye yol açacağı öngörülmektedir.

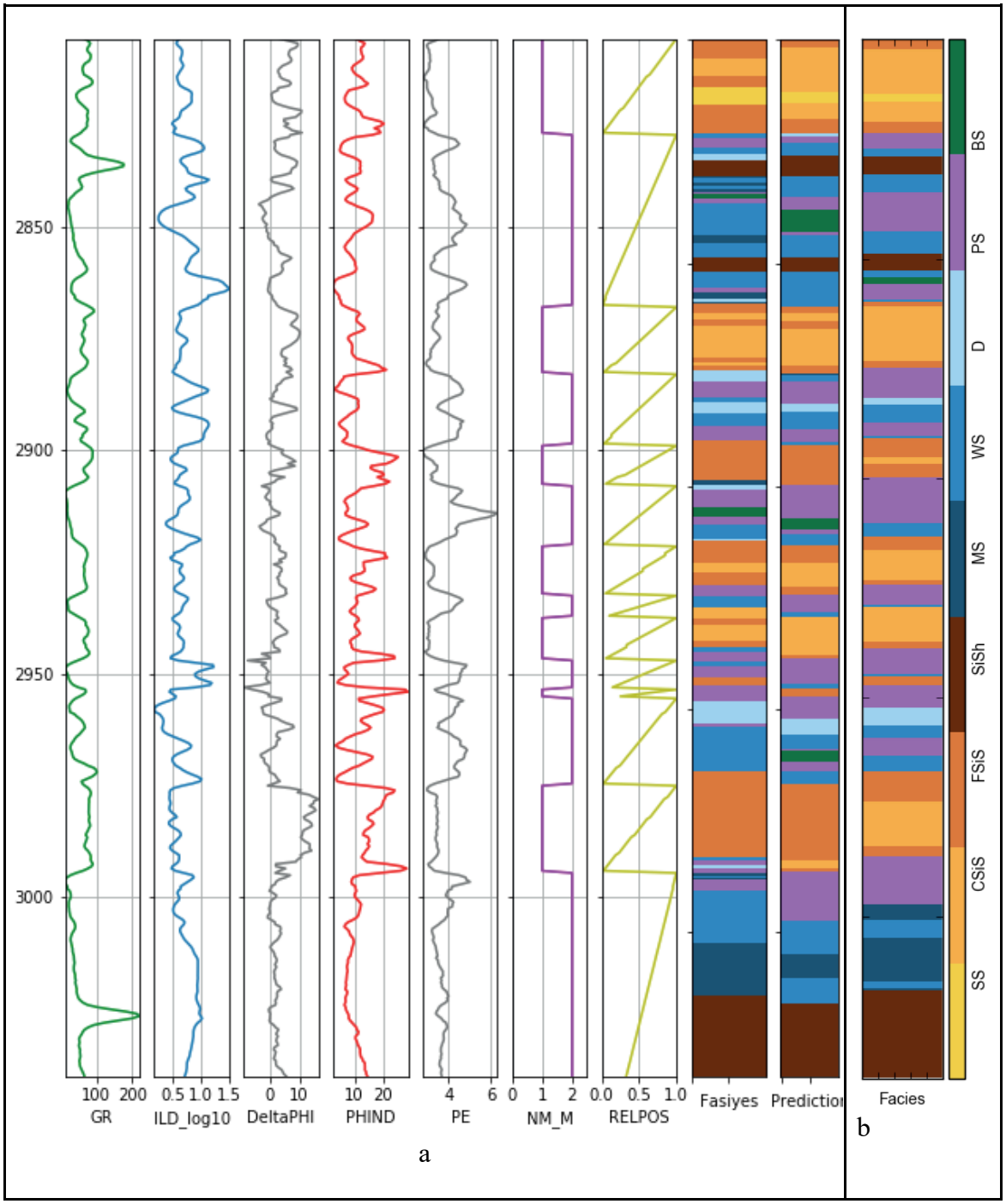


Şekil 7. a) Test verisi için elde edilen karışıklık matrisi b) Eğitim verisinin fasiyeslere göre dağılımı

SONUÇLAR

Bu çalışmada fasiyes sınıflandırma problemine makine öğrenmesi ile bir yaklaşım önerilmiştir. Önerilen algoritma, değişim arttırıcı türev sınıflamasına dayanmaktadır. GBC algoritması ile elde edilen tahmin sonuçları, yorumlanmış log verileri ile benzer olduğu görülmektedir.

Sınıflandırıcı model yaklaşımı başarı gösterip, algoritmanın kendisine öğretilen yorumları kısmi bir şekilde öğrendiğini ve uyguladığını göstermektedir. On kuyudan oluşan veri topluluğu, geliştirilen stratejisinin olumlu bir etkisi olarak, önerilen yaklaşımı doğrulamaktadır. Ayrıca fasiyes bilgisi içermeyen kuyu verisinden elde edilen sonuçlar, makine öğrenmesi algoritmasının yeni veriler için genelleme yapma yeteneğini göstermektedir.



Şekil 8. a) Eğitim modeli ile fasiyes bilgisi tespit edilen kuyunun sınıflandırma sonucu.
b) Karşılaştırılan bir diğer algoritma sonucu

GBC algoritması karmaşık jeolojik yapıların litolojik olarak tanımlanması için uygundur. Önerilen yaklaşım, kuyuların analizinin bir ön aşaması olarak düşünülebilir. Genel olarak özetlemek gerekirse, makine öğrenmesine dayalı olarak önerilen modelin değerlendirme sonuçları, gelecekteki araştırmalar ve fasiyes tanımlamaları için yararlı olacaktır.

Kaynaklar

Ahmadi, A., M., Zendehboudi, S., Lohi, A., Elkamel, A., & Chatzis, I. (2013). Reservoir permeability prediction by neural networks combined with hybrid genetic algorithm and particle swarm optimization. Geophysical Prospecting, 61(3), 582-598.

Alpaydin, E. (2009). Introduction to machine learning. MIT press. https://www.cmpe.boun.edu.tr/~ethem/i2ml/i2ml-figs.pdf (ET Ocak 2024)

Anderson, B. I. (2001). Modeling and Inversion Methods for the Interpretation of Resistivity Logging Tool Response, Delft University Press.

Bailly J. S., Amaud M. and Puech C., (2007). Boosting: a classification method for remote sensing, Int. J. Remote Sensing, 28(7), 1687-1710

Bestagini, P., Lipari, V., & Tubaro, S. (2017). A machine learning approach to facies classification using well logs. In SEG Technical Program Expanded Abstracts 2017 (pp. 2137-2142). Society of Exploration Geophysicists.

Bishop, C. M., (2006). Pattern recognition and machine learning (information science and statistics): Springer-Verlag New York, Inc.

Bohling, G.C., & Dubois, M.K. (2003). An Integrated Application of Neural Network and Markov Chain Techniques to Prediction of Lithofacies from Well Logs (Kansas Geological Survey Open File Report 2003-50).

Breiman, L., J. Friedman, R. Olshen, and C. Stone, (1984). Classification and Regression Trees: Wadsworth and Brooks.

Dubois, M. K., Bohling, G. C., & Chakrabarti, S. (2007). Comparison of four approaches to a rock facies classification problem. *Computers & Geosciences*, *33*(5), 599-617.

Darling, T. (2005). Well logging and formation evaluation. Elsevier.

Drucker, H., & Cortes, C. (1996). Boosting decision trees. In 'Advances in neural information processing systems', pp. 479-485.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. International Conference on Machine Learning, Vol. 96, pp. 148-156.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189-1232.

Gercek, H., (2007). Poisson's ratio values for rocks: International Journal of Rock Mechanics and Mining Sciences, 44, 1-13.

Glover, P. W. (2000). Petrophysics. *University of Aberdeen, UK*.

Hall, B., (2016). Facies classification using machine learning. Lead. Edge 35, 906–909.

Hall, M., and B. Hall, (2017). Distributed collaborative prediction: Results of the machine learning contest: The Leading Edge, 36, 267–269.

Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd international conference on document analysis and recognition,* Vol. 1, pp. 278-282.

Ho, T. K., (1998). The random subspace method for constructing decision forests: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 20, 832–844.

Kalayci, S. (2018). Makine öğrenmesi yöntemleri ile kredi risk analizi. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi.

Mayr, A., Binder, H., Gefeller, O., Schid, M., (2014). The evolution of Boosting algorithms from Machine Learning to statistical modelling. Methods of Information Medicine, 53(6):419-27.

Mitchell, T.M. (1997). Machine learning. McGraw-Hill

Nizam, H., Akın, S., (2014). Sosyal Medyada Makine Öğrenmesi ile Duygu Analizinde Dengeli ve Dengesiz Veri Setlerinin Performanslarının Karşılaştırılması. XIX. Türkiye'de İnternet Konferansı Bildirileri. Yaşar Üniversitesi, İzmir. 129-136.

Pekiner, Y. (2002). Kuyu Logları Tekniğiyle Yeraltının Keşfi, Seçkin Yayıncılık.

Rokach, L., and O. Maimon, (2005). Top-down induction of decision trees classifiers - a survey: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) (TSMCC), 35, 476–487.

Schapire, R. E. (1999). A brief introduction to boosting. In IJCAI (Vol. 99, pp. 1401-1406).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D.,Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

Ulugergerli, E. U. (2011). Two dimensional combined

inversion of short- and long-normal dc resistivity well log data. Journal of Applied Geophysics, 73 (2011) 130–138.

URL1: https://www.spec2000.net/ (E.T., Aralık 2019)

URL2: Resources for EECS 833, Neural Networks and Fuzzy Systems http://www.people.ku.edu/~gbohling/EECS833/ (E.T., Aralık 2019)

"This page is left blank for typesetting"

Bu sayfa dizgiden dolayı boş bırakılmıştır

Journal of Quantum Technologies and Informatics Research
JQTAIR

KUANTUM TEKNOLOJİLERİ VE ENFORMATİK ARAŞTIRMALARI
Volume / Cilt: 1, Issue / Sayı: 2, 2024, pp. 45-49
E - ISSN: 3023-4735
URL: https://journals.gen.tr/index.php/jqtair/
DOI: https://doi.org/10.5281/zenodo.10102956

Received / Geliş: 28.02.2024
Acccepted / Kabul: 28.02.2024

RESEARCH ARTICLE / ARAŞTIRMA MAKALESİ

Kuantum Fourier Dönüşümünün Yüksek Boyutta Cirq Kullanarak Uygulanması

Implementation of High Dimension Quantum Fourier Transform via Cirq

Osman Semi Ceylan[1] **Cumali Yaşar**[2]

[1] Çanakkale Onsekiz Mart Üniversitesi, Bilgisayar Mühendisliği, Türkiye, e-mail: osmansemi.ceylan@comu.edu.tr
[2] Çanakkale Onsekiz Mart Üniversitesi, Eğitim Fakültesi, Türkiye, e-mail: cumali.yasar@comu.edu.tr

Öz

Kuantum Fourier Dönüşümü (QFT), kuantum hesaplamada çeşitli kuantum algoritmaları için çok önemli bir işlemdir. Bu çalışma, devre tabanlı simülasyon yazılımında yüksek boyutta QFT algoritmasının bir uygulamasını sunmaktadır. İlk olarak QFT algoritması, kuantum sistemlerde matematiksel olarak incelenmektedir. Ardından, kuantum devreler üzerinde yüksek boyutlu QFT algoritması verilmektedir. Son olarak, bu tekniklerin Cirq kuantum devre simülasyon yazılımı üzerinde pratik bir uygulamasını sunuyoruz. Çalışmamız, kuantum devre simülasyonlarında yüksek boyutlu QFT algoritmasının uygulanmasına yönelik değerli bilgiler ve pratik yönergeler sağlar. Böylelikle kuantum yüksek boyuta daha fazla ilgi çekeceği düşünülmektedir.

Anahtar Kelimeler: Kuantum Fourier Dönüşümü, Devre Simülasyonu, Yüksek Boyutta Kuantum Algoritmalar

Abstract

Quantum Fourier Transform (QFT) is a crucial operation for various quantum algorithms in quantum computing. This work presents an implementation of the high-dimensional QFT algorithm in circuit-based simulation software. Firstly, the QFT algorithm is examined mathematically in quantum systems. Then, the high-dimensional QFT algorithm on quantum circuits is given. Finally, we present a practical implementation of these techniques on the Cirq quantum circuit simulation software. Our work provides valuable information and practical guidelines for applying the high-dimensional QFT algorithm in quantum circuit simulations. Thus, it is thought that quantum will attract more attention to the higher dimension.

Keywords: Quantum Fourier Transform, Circuit Simulation, High Dimension Quantum Algorithms

Citation/Atıf: CEYLAN, O. S. & YAŞAR, C. (2024). Kuantum Fourier Dönüşümünün Yüksek Boyutta Cirq Kullanarak Uygulanması. *Kuantum Teknolojileri ve Enformatik Araştırmaları*. 2(1):45-49, DOI: 10.5281/zenodo.10102956

Corresponding Author/ Sorumlu Yazar:
Osman Semi Ceylan
E-mail: osmansemi.ceylan@comu.edu.tr

1. GİRİŞ

Kuantum Fourier Dönüşümü (QFT)[1], kuantum hesaplama alanında en değerli işlemlerden biridir. Geleneksel formunda QFT, klasik durumların üst üste binmesini temsil eden bir kuantum durumunu zarif bir şekilde frekans alanı karşılığına dönüştürür. Bu dönüşüm, Shor'un tam sayıları asal çarpanlara ayırma algoritması[2] gibi çeşitli kuantum algoritmalarının elde ettiği hızlanmanın temelini oluşturuyor; bunun özellikle kriptografi alanı için önemli etkileri var[3].

QFT algoritmasının geleneksel iki seviyeli sistemlerdeki etkisi derin olsa da daha yüksek boyutlara yayılması büyüleyici ve nispeten keşfedilmemiş bir sınır sunuyor. Yüksek boyutlu QFT algoritması yalnızca kuantum mekaniğinin matematiksel çerçevesini genişletmekle kalmıyor, aynı zamanda kuantum bilgi işlemeyi ilerletmek için zengin fırsatlar da sunuyor.

Yüksek boyutlu QFT algoritmasının önemli bir özelliği, karmaşık kuantum durumlarını ikili veya kübit temsillerinin ötesinde manipüle etme ve analiz etme yetenekleridir. Yüksek boyutlu QFT algoritması, ikiden büyük boyutlara sahip küditler (kuantum rakamları) üzerinde çalışarak, bilgilerin daha kompakt ve anlamlı bir şekilde temsil edilmesini sağlar. Bu, kuantum sistemlerinde gürültünün etkilerini azaltmak için gerekli olan daha verimli kuantum algoritmalarına ve gelişmiş kuantum hata düzeltme kodlarına yol açabilir[4]. Bu nedenle bu çalışmada QFT algoritması Cirq[5] simülasyon yazılımı üzerinde uygulanmasını inceledik.

Bu makale 4 bölümde hazırlanmıştır. İkinci bölümde kuantum yüksek boyut cebiri anlatılmaktadır. Üçüncü bölümde yüksek boyutta QFT ve devresi verilmektedir. Dördüncü bölümde QFT algoritmasının Cirq yazılımı üzerinde uygulaması bir örnek ile gösterilmiştir. Son bölümde ise bulgularımızdan varılan sonuçlara yer verilmiştir.

2. KUANTUM YÜKSEK BOYUT

Küditler veya kuantum rakamlar, d-boyutlu karmaşık Hilbert uzayındaki $(\mathfrak{H})$ bir kuantum durumunu matematiksel olarak ifade eder. Bu uzaydaki küditlerin ortonormal Fock tabanları eşitlik (1) ile verilmiştir.

$$_d|f\rangle \in \{|0\rangle, |1\rangle, |2\rangle, \dots, |d-1\rangle\} \quad (1)$$

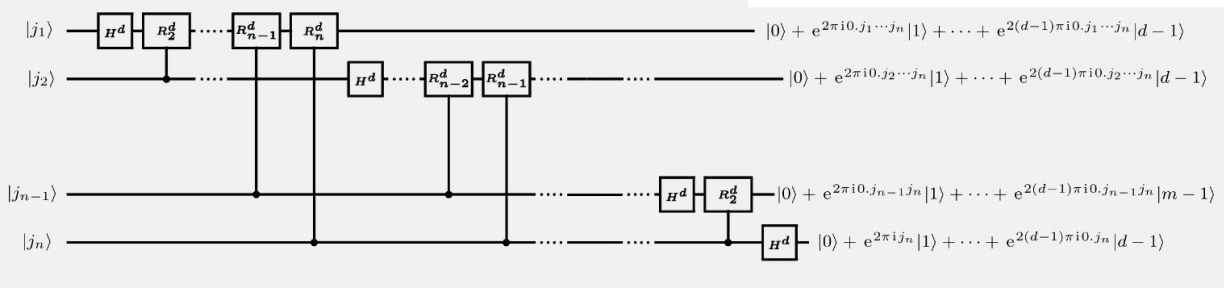
Herhangi bir d -boyuta nicemlenmiş bir kuantum durum ise eşitlik (2)'deki gibi tanımlanmaktadır.

$$_d|\psi\rangle = \sum_{k=0}^{d-1} \alpha_k |k\rangle \in \mathbb{C}^d \quad (2)$$

Vektörel formda olan bir $_d|\psi\rangle$ kuantum durumun özdeğerleri α_k olarak verilir. Geleneksel iki seviyeli sistemlerde olduğu gibi yüksek boyutta da kuantum durum bir Gaussian dağılımı olmak zorundadır.

$$\sum_{k}^{d-1} \alpha_k \alpha_k^* = 1 \quad (3)$$

Herhangi bir d-boyutta tanımlanan küdit bir kuantum durumu dönüştürmek için kullanılan dönüşüm matrisleri $(d^n \times d^n)$ boyutlarında olmalıdır. Herhangi bir d boyutun Weyl dönüşümleri[6] olarak da bilinen temel dönüşüm matrisleri denklem (4) ile hesaplanmaktadır.



Şekil-1: QFT algoritmasının tasarımsal kuantum devresi[8].

$$ {}_d\mathrm{U}_{pq} = \sum_{k=0}^{d-1} e^{\frac{2\pi i}{d}kp} |(k+q) \bmod d\rangle\langle k|, \quad p,q \in \{0,1,\dots,d-1\} \tag{4}$$

Denklem (4) ile verilen dönüşümler kullanılarak yüksek boyutta kuantum hesaplamada iki seviyeli sistemlerde bulunan dönüşümlerin benzerleri elde edilebilir. Bir örnek olarak kapısının denklem (5) ile iki boyutta ve denklem (6) ile 3 boyutta matrisleri verilmektedir.

$$\mathrm{X} = {}_2\mathrm{U}_{01} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{5}$$

$${}_3\mathrm{U}_{01} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad {}_3\mathrm{U}_{02} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \tag{6}$$

Denklem (6) ile verilen dönüşümlerin 3 boyut kuantum özvektörlere uygulanması denklem (7) ile gerçekleştirilir.

$$\begin{aligned} {}_3\mathrm{U}_{01}|0\rangle &= |0 \oplus 1\rangle = |(0+1)\mathrm{mod}(3)\rangle = |1\rangle \\ {}_3\mathrm{U}_{01}|1\rangle &= |0 \oplus 1\rangle = |(1+1)\mathrm{mod}(3)\rangle = |2\rangle \\ {}_3\mathrm{U}_{01}|2\rangle &= |0 \oplus 1\rangle = |(2+1)\mathrm{mod}(3)\rangle = |0\rangle \end{aligned} \tag{7}$$

Benzer olarak d-boyutta QFT içinde gereken tek küdit kuantum faz dönüşüm kapısının matrisi de eşitlik (8)'de verilmektedir.

$$diag\left({}_d\mathrm{RZ}_k\right) = \left(1, \omega_k, \omega_k^2, \dots, \omega_k^{d-1}\right), \qquad \omega = e^{\frac{2\pi i}{d^k}} \tag{8}$$

3. Yüksek Boyut QFT

Yüksek boyutlarda QFT, iki boyutta olduğu gibi tam süperpozisyon durumunda faz dönüşümleri uygular. Bu nedenle genel Fourier dönüşümü formülü ile herhangi bir d-boyutunda hesaplanabilirler. Bu çalışmada da kullanılan Kuantum Genel Fourier Dönüşümü (QGFT)[7] algoritmasının n küditlik bir $|j\rangle$ durumu üzerindeki etkisinin d-boyutlu formülü aşağıdaki biçimdedir[8]:

$$\mathrm{F}(d,N)|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i}{N}jk} |k_d\rangle, \qquad N = 2^n \tag{9}$$

Ayrıca denklem (9) kullanılarak -boyutta tam süperpozisyon üreten Hadamard kapısının matrisi $n = 1$ alınarak eşitlik (10)'daki gibi hesaplanabilir[8].

$${}_d\mathrm{H}|j\rangle = F(d,d)|j\rangle = \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} \omega^{jk} |k\rangle \tag{10}$$

Herhangi bir -boyutta QFT uygulanması için denklem (8) ile verilen faz kapısı ve denklem (10) ile verilen Hadamard kapısı kullanılmaktadır.

Bir d-boyutta QFT gerçekleştiren kuantum devre Şekil-1 ile verilmiştir. Devre üzerinde j ile dizinlenmiş n adet küditten oluşmaktadır. Şekilden görüleceği üzere QFT algoritması d-boyut ${}_d\mathrm{H}$ ve belirli bir desen ile devreye yerleştirilmiş kontrollü ${}_d\mathrm{RZ}_k$ kapılarından oluşmaktadır.

İki seviyeli sistemlerin aksine yüksek boyutlu hesaplamada çok değerli kontrol kapıları (MVCG) bulunmaktadır. Bu kapılar yalnız değil, istenilen diğer özvektörlere koşullu kapıların uygulanmasını sağlamaktadır. Herhangi bir seçilen d-boyut hesaplamada koşullu bir kapı için koşul değerleri kümesi ${}_d\mathrm{G} \subset \{0,1,\dots,d-1\}$ gibi boş olmayan bir altküme olmalıdır. QFT algoritması yüksek boyutta gerçekleştirilmesine bu çeşit kontrollü kapılar kullanılmaktadır.

4. QFT Algoritmasının Cirq Uygulaması

Cirq üzerinde yüksek boyutta kapı tabanlı kuantum hesaplama simülasyonları elverişli bir platformdur. Ayrıca yüksek boyutta kodlama yapabilmek için gerekli bazı araçları da hazır sunmaktadır.

4.1. Yüksek Boyutta Matrislerin Hesaplanması

QFT algoritması için Hadamard ve koşullu faz kapıların Cirq üzerinde tanımlanması gerekmektedir. Bir kuantum kapının tanımlanması için d-boyutlu kapının matrisi hesaplanmalıdır. Hadamard için denklem (10) ile verilen eşitlik kullanılabilir. Aşağıda $d = 3$ için Hadamard matrisi gösterilmektedir.

$$ {}_3\mathrm{H} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 1 & 1 \\ 1 & e^{\frac{2\pi i}{3}} & e^{\frac{4\pi i}{3}} \\ 1 & e^{\frac{4\pi i}{3}} & e^{\frac{8\pi i}{3}} \end{pmatrix} \quad (11) $$

Benzer şekilde $d = 3$ için ${}_3\mathrm{RZ}_1$, ${}_3\mathrm{RZ}_2$, ${}_3\mathrm{RZ}_3$ için faz kapıları ise aşağıdaki biçimde matrisler olmaktadır.

$$ {}_3\mathrm{RZ}_1 = {}_3\mathrm{Z} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{\frac{2\pi i}{3}} & 0 \\ 0 & 0 & e^{\frac{4\pi i}{3}} \end{pmatrix} \quad (12) $$

$$ {}_3\mathrm{RZ}_2 = {}_3\mathrm{S} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{\frac{2\pi i}{9}} & 0 \\ 0 & 0 & e^{\frac{4\pi i}{9}} \end{pmatrix} \quad (13) $$

$$ {}_3\mathrm{RZ}_3 = {}_3\mathrm{T} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{\frac{2\pi i}{27}} & 0 \\ 0 & 0 & e^{\frac{4\pi i}{27}} \end{pmatrix} \quad (14) $$

4.2. Yüksek Boyutta Kapıların Tanımlanması

QFT algoritmasında kullanılan bu matrislerin kapı olarak Cirq yazılımı üzerinde tanıtılması gerekmektedir. Cirq üzerinde matris hazırlamak için Şekil-2'de verilen kod ile yazılımsal sınıf oluşturulmalıdır.

```
class Gate(cirq.Gate):
    def __init__(self, unitary_matrix: dimension: int, diag_info):
        self.unitary_matrix = unitary_matrix
        self.dimension = dimension
        self.qubit_count = int(qcount)
        self.gate_name = diag_info
        super().__init__()

    def qid_shape(self) -> typing.Tuple[int]:
        return (self.dimension, ) * self.qubit_count

    def num_qubits(self) -> int:
        return self.qubit_count

    def unitary(self) -> numpy.ndarray:
        return self.unitary_matrix

    def circuit_diagram_info(self, args) -> cirq.CircuitDiagramInfo:
        return self.gate_name

    def __str__(self) -> str:
        return f'{self.unitary_matrix}'

    @property
    def transform_matrix(self) -> numpy.ndarray:
        return self.unitary_matrix
```

Şekil-2: Cirq üzerinde tanımlı kapı nesnesi üreten sınıf.

Şekil-2 ile verilen yazılımsal sınıf, matrisi boyut miktarı ve kapının ismini parametre olarak almakta ve Cirq üzerinde devrelerde kullanılması için hazırlamaktadır.

4.3. QFT Devresi Oluşturma

QFT algoritmasının Cirq üzerinde uygulanması için öncelikle kuantum devre hazırlanmalıdır. Sonrasında QFT için gerekli kapılar aşağıdaki biçimde verilen sınıf kullanılarak kodlanmalıdır.

```
circuit = cirq.Circuit()
qudits = cirq.LineQid.range(3, dimension=3)

hadamard_gate = Gate(general_Hadamard(dim=3), dimension=3,
diag_info='H')
swap_gate = Gate(swap(dim=3), dimension=3, diag_info=('x', 'x'))
z_gate = cirq.ZPowGate(exponent=1/numpy.power(2, 0),
dimension=3)
s_gate = cirq.ZPowGate(exponent=1/numpy.power(2, 1),
dimension=3)
t_gate = cirq.ZPowGate(exponent=1/numpy.power(2, 2),
dimension=3)
```

Şekil-3: Cirq üzerinde devre ile QFT için gereken kapıların tanımlanmasını içeren kod.

Şekil-3 ile örnek olarak boyut sayısı $d = 3$, küdit sayısı alınarak oluşturulan kuantum devresi ve QFT uygulanması için gereken Hadamard ve faz kapıları tanımlamalarını içerir.

```
z_gate = z_gate.controlled(num_controls=1, control_values=((1, 2 ), ),
control_qid_shape=(3, ))
```

Şekil-4: Cirq üzerinde koşullu kapı oluşturan kod.

Bu faz kapıların QFT algoritmasında kullanması için koşullu yapılması gerekmektedir. Çok değerli koşullu kapılar Şekil-4 ile verilen kod ile eklenmektedir.

Elde edilen kapılar Cirq üzerinde devreye eklenerek QFT algoritması tamamlanır. Şekil-5a ile hazırlanan QFT algoritmasının boyut sayısı $d = 3$ ve küdit sayısı $\boldsymbol{n = 3}$ olduğu durumda Cirq üzerinde hazırlanmış devresi verilmektedir. Şekilden görüleceği üzere ilk adımda SWAP kapısı sonra tüm küditlere Hadamard kapısı ve sıralı biçimde S ve T devreye koşullu olarak eklenmiştir. En son olarak yüksek boyutta ölçüm kapısı eklenmiştir. QFT devresinde çok değerli koşullu kapıların koşul değerleri hepsi için (1,2) olarak belirlenmiştir.

4.4. QFT Devresi Simülasyonu

Bu bölümde oluşturulan d-boyut QFT algoritması devresi örnekte verildiği biçimde simülasyon sonuçları verilmektedir. Denklem (10) ile verilen eşitlik hesaplandığında ölçüm sonuçlarında eşit olasılıkta $d^n = 3^3 = 27$ adet durum vektörü beklenmektedir.

Yapılan simülasyon sonucunda Şekil-5c ile görüleceği üzere beklenilen sonuçlar elde edilmiştir. Yapılan simülasyonda ölçülen özvektörler $|000\rangle, |001\rangle, |002\rangle, |100\rangle, \ldots, |222\rangle$ olmak üzere toplam 27 adet ve eşit olasılıkta oldukları bulunmuştur.

Bu çalışmada yapılan bir diğer simülasyonda ise QFT ve ters QFT algoritması peşi sıra uygulanarak elde edilen sonuçlar incelenmiştir. Şekil-5b'den görülebileceği üzere kuantum durum başlangıç durumuna olan $|0\rangle$ durumuna geri dönmüştür.

5. SONUÇ

Yüksek boyutlu kuantum hesaplama, daha az küdit ile daha fazla bilgi depolayabilmesi, devre karmaşıklığını azaltması, algoritmaların verimliliğini arttırabilmesi ve gürültüye karşı hassasiyeti azaltabilmesi gibi çeşitli avantajları sayesinde günümüzde giderek daha fazla ilgi görüyor. Bu nedenle bu çalışmada yapılan QFT algoritmasının yüksek boyutta Cirq yazılımında kodlanabileceği alanın literatürüne yaptığı katkı bakımından yüksek boyuta ilgiyi arttıracağı düşünülmektedir.

KAYNAKÇA

Nielsen, Michael A., Isaac L. Chuang. *Quantum computation and quantum information*, Cambridge university press, 2010. pp.

Shor, Peter W. *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM review 41.2 (1999): 303-332.

Stetsyuk, P. I. *Theory and software implementations of Shor'sr-algorithms*, Cybernetics and Systems Analysis 53.5 (2017): 692-703.

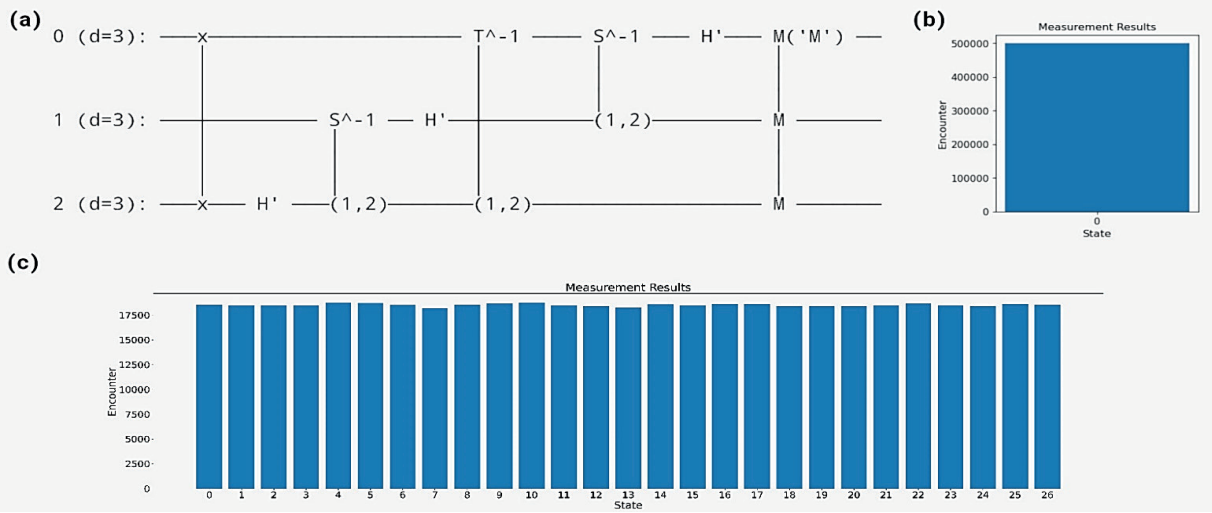
Cozzolino, D., Da Lio, B., Bacco, D., & Oxenløwe, L. K. *High-dimensional quantum communication: benefits, progress, and future challenges*, Advanced Quantum Technologies, 2(12), 1900038.

Cirq Developers. *Cirq*. See full list of authors on Github: https://github .com/quantumlib/Cirq/graphs/-contributors

Patera, J., and H. Zassenhaus. *The Pauli matrices in n dimensions and finest gradings of simple Lie algebras of type A n− 1*, Journal of Mathematical Physics 29.3 (1988): 665-673.

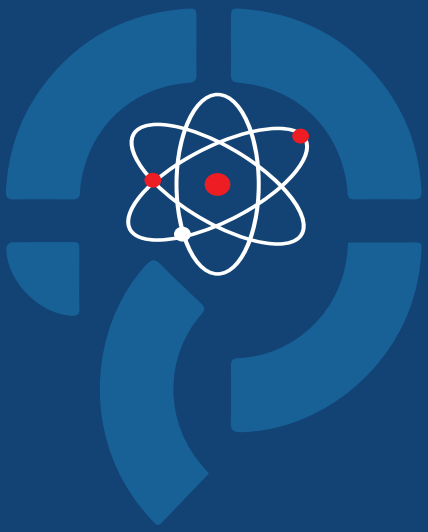
Kitaev, Yu A. *Quantum measurements and the Abelian Stabilizer Problem*, Quanth-ph/9511026 (1995).

Cao, Ye, et al. *Quantum fourier transform and phase estimation in qudit system*, Communications in Theoretical Physics 55.5 (2011): 790.



Şekil-5: (a) QFT algoritmasının Cirq üzerinde devre modelinin çıktısı. (b) QFT ve ardından gelen ters QFT algoritmasını içeren kuantum devrenin simülasyon sonucu. (c) QFT algoritmasının kuantum devre simülasyonu sonucu.

Journal of Quantum Technologies and Informatics Research



Journal of Quantum Technologies
and Informatics Research

JQTAIR

HOLISTENCE
publications