T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BÜYÜK VERİ İŞLEMEDE TAM METİN ARAMA TEKNOLOJİLERİNİN UYGULANMASI VE KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Ayşenur DENİZ

Bilgisayar Mühendisliği Ana Bilim Dalı

Tez Danışmanı: Dr. Öğretim Üyesi Ahmet Arif AYDIN

TEMMUZ 2023

T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BÜYÜK VERİ İŞLEMEDE TAM METİN ARAMA TEKNOLOJİLERİNİN UYGULANMASI VE KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Ayşenur DENİZ
(36203619011)

Bilgisayar Mühendisliği Ana Bilim Dalı

Tez Danışmanı: Dr. Öğretim Üyesi Ahmet Arif AYDIN

TEMMUZ 2023

TEŞEKKÜR VE ÖN SÖZ

Bu tez çalışmasının her aşamasında bilgisini, tecrübesini ve desteklerini esirgemeden beni her konuda yönlendiren ve cesaretlendiren saygıdeğer danışman hocam Dr. Öğretim Üyesi Ahmet Arif AYDIN'a,

Hayatımın her anında olduğu gibi bu çalışmamda da yanımda olan, desteklerini hiçbir zaman eksik etmeyen değerli AİLEME,

Yorulduğumda, öğrenme arzumu ve hedeflerimi bana hatırlatan değerli DOSTLARIMA

teşekkür ederim.

ONUR SÖZÜ

Yüksek lisans tezi olarak sunduğum “Büyük Veri İşlemede Tam Metin Arama Teknolojilerinin Uygulanması ve Karşılaştırılması” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Ayşenur DENİZ

İÇİNDEKİLER

ÇİZELGELER DİZİNİ

ŞEKİLLER DİZİNİ

SEMBOLLER VE KISALTMALAR

API : Application Programming Interface / Uygulama Programlama Arayüzü

CPU : Central Process Unit / Merkezi İşlemci Birimi

JSON : JavaScript Object Notation

NDJSON : Newline Delimited JSON

M1 : Makine 1

M2 : Makine 2

M3 : Makine 3

V1 : Veri Seti 1

V2 : Veri Seti 2

V3 : Veri Seti 3

ÖZET

Yüksek Lisans Tezi

BÜYÜK VERİ İŞLEMEDE TAM METİN ARAMA TEKNOLOJİLERİNİN UYGULANMASI VE KARŞILAŞTIRILMASI

AYŞENUR DENİZ

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı

43 + VI sayfa

2023

Danışman: Dr. Öğretim Üyesi Ahmet Arif AYDIN

Günümüzde verilerin boyutları ciddi bir hızla artarak devam etmektedir. Büyük veri setlerinde yapılan işlemler verinin boyutu arttıkça bazı zorluklara sebep olmaktadır. Örneğin, bir veri seti içerisinde arama yapmak temel işlemlerden biridir ve veri miktarı arttıkça çeşitli zorlukları açığa çıkarmaktadır. Bu tez çalışmasında, arama işlemlerindeki zorluklardan yola çıkılarak tam metin arama metodu üzerine araştırma yapılmaktadır. Tam metin arama, indekslenmiş veriler üzerinden arama işleminin gerçekleştirildiği bir yöntemdir. Bu yöntem, büyük bir veri setinde veriye daha hızlı erişim ve etkili arama gerçekleştirebilme noktasında avantaj sağlamaktadır. Bu çalışmada, tam metin aramada kullanılan popüler teknolojilerden Apache Solr ve Elasticsearch için indeksleme ve arama performansının bir karşılaştırılması yapılmıştır. Öncelikle, üç farklı veri seti ve üç farklı makine kullanılarak her teknoloji için indeksleme süreleri alınıp karşılaştırılmıştır. Daha sonra, indeksleme performansı en iyi olan makinede, 10 sorgu kullanılarak her iki teknoloji için arama süreleri incelenmiştir. Sonuçlar değerlendirildiğinde Apache Solr hem indekslemede hem de aramada daha iyi performans göstermiştir. Bu yüzden, bu çalışma için geliştirilen web uygulaması Apache Solr üzerine inşa edilmiştir. Uygulama kısmında, Web of Science platformunda yer alan Engineering, Computing & Technology koleksiyonundaki 1,655 derginin çeşitli bilgilerinin toplandığı özgün bir veri seti oluşturulmuş ve kullanılmıştır. Bu uygulama sayesinde, araştırmacılar çalışmalarını yayınlamak için amaçlarına uygun dergileri listeleyebilmektedir.

Anahtar Kelimeler: Apache Lucene, Apache Solr, Büyük Veri, Elasticsearch, Tam Metin Arama, Veri İşleme

ABSTRACT

Master Thesis

APPLICATION AND COMPARISON OF FULL TEXT SEARCH TECHNOLOGIES FOR BIG DATA PROCESSING

AYŞENUR DENİZ

Inonu University
Graduate School of Nature and Applied Sciences
Department of Computer Engineering

43 + VI pages

2023

Supervisor: Assist. Prof. Ahmet Arif AYDIN

Today, the size of the data continues to increase at a serious pace. Operations performed on large datasets cause some difficulties as the size of the data increases. For example, searching within a dataset is one of the basic operations, and as the amount of data increases, it reveals various difficulties. In this thesis, a research is accomplished on the full-text search method based on the difficulties in the search processes. Full-text search is a method in which the search is performed on indexed data. This method provides an advantage in terms of faster access to data and effective search in a large dataset. In this study, a comparison of indexing and search performance was made for Apache Solr and Elasticsearch, which are popular full-text search technologies. First, indexing times for each technology were taken and compared using three different datasets and three different machines. Then, search times for both technologies were examined using 10 queries on the machine with the best indexing performance. Considering the results, Apache Solr performed better in both indexing and searching. Therefore, the web application developed for this study is built on Apache Solr. In the application part, a unique dataset was created and used, in which various information was collected from 1,655 journals in the Engineering, Computing & Technology collection on the Web of Science platform. Thanks to this application, researchers could list the journals suitable for their purpose to publish their work.

Keywords: Apache Lucene, Apache Solr, Big Data, Elasticsearch, Full-Text Search, Data Processing

1. GİRİŞ

1.1 Tezin Amacı

Teknolojinin ilerlemesiyle hemen hemen her alanda verinin oluşturulması artarak devam etmektedir. Her yerde rahatça erişilebilir olan ve sıklıkla kullanılan internet, bu artışın en belirgin sebep ve araçlarından biridir. Çeşitli kaynaklar, hizmetler, yazılım araçları ve donanım aygıtları, çeşitli biçimlerde, boyutlarda ve hızlarda büyük miktarlarda veri üretmektedir. Exabyte ve zettabyte'lara ulaşan veriler "Büyük Veri" olarak adlandırılmaktadır (Halevi ve Moed, 2012). Büyük veri, bir kerede işlenemeyen ve karmaşık işleme araçları, teknolojileri ve yöntemleri gerektiren büyük miktarda veri olarak tanımlanabilir. Birçok kaynaktan çok sayıda ve farklı veri türlerinde günlük veriler üretilmektedir (Domo Company, 2022). Bu durum büyük verinin önemini arttırmaktadır. Çünkü veriler, kurumların iş piyasasında rekabetçi kalabilmeleri için gizli öngörüler içermektedir. Büyük veri beş V ile tanımlanmaktadır: *hacim/volume* (veri miktarı), *hız/velocity* (gelen verinin hızı), *doğruluk/variety* (güvenilirlik), *çeşitlilik/veracity* (çeşitli türler ve farklı biçimler) ve *değer/value* (faydalı bilgi) (Lashkaripour, 2020). Bu özelliklerin her biri, veri yoğunluğu olan sistemlerin geliştiricileri için çeşitli zorluklar ortaya çıkarmaktadır. Bu zorlukların üstesinden gelmek, alan gereksinimlerini ve kullanıcı ihtiyaçlarını karşılamak amacıyla verinin akışı, depolanması ve analitiği için çeşitli araçlar geliştirilmiştir (Rao ve diğ, 2018).

Büyük veri çağında, büyük miktarlardaki verilerden makul bir süre içinde faydalı bilgiler ortaya çıkarmak oldukça önemlidir. Kaldı ki, tek bir yöntem ile bütün işlemler yapılamaz. Farklı talepleri, veri türleri, zaman kısıtlamaları ve öncelikleri olan verinin yoğun olduğu sistemler geliştirilirken donanımda mevcut kaynakların tümü, yazılım geliştiricilerin işlemlerini hızlı bir şekilde gerçekleştirmek için kullanacağı teknolojiler ve yöntemler üzerinde bir etkiye sahiptir. Bu nedenle, bir veriyi işlemek için uygun teknolojiyi seçme görevi, tüm veri işleme sürecini doğrudan etkilediğinden önem arz etmektedir. Ayrıca arama, kullanıcının taleplerini karışılaması noktasında önemlidir (Barrenechea ve diğ, 2017). Öte yandan, geleneksel yöntemlerle hızlı bir şekilde sonuç almak, büyük veri için çok zordur. Bu nedenle, büyük veri üzerinde analiz yapmak için yeni teknolojilere ihtiyaç vardır. Çeşitli veri işleme araçları yerleşik arama özellikleri sunmaktadır. Programlama dilleri kullanılarak yazılan kodlar vasıtasıyla gerçekleştirilen aramaların yanı sıra Apache

Lucene, Apache Solr ve Elasticsearch gibi tam metin arama kütüphaneleri veya teknolojileri de bu amaca yönelik kullanılmaktadır. Tam metin arama teknolojileri günümüzde arama motorları, e-ticaret uygulamaları, eğitim platformları (Y. Aldailamy ve diğ, 2018), sosyal ağ platformları, mobil bankacılık uygulamaları, bilgi alma uygulamaları (Wang ve diğ, 2022), video akış hizmetleri, veri yoğunluklu sistemler (Anderson ve diğ, 2015), akıllı şehir ve IoT uygulamaları (Bellini ve diğ, 2019) dahil olmak üzere birçok alanda kullanılmaktadır. Oldukça önemli bir yere sahip olan arama motorlarının, arama işlemleri için uygun teknolojilerin seçilmesi aşamasına ışık tutması açısından, bu tez çalışmasında iki popüler tam metin arama teknolojisi olan Apache Solr ve Elasticsearch karşılaştırılmaktadır. Uygulama aşamasında, iki arama motorundan biri kullanılarak araştırmacılar için web tabanlı bir dergi arama motorunun geliştirilmesi amaçlanmaktadır.

1.2 Tezin Akışı

Bu çalışmanın devamında, ikinci bölümde, literatürde bulunan bu tez çalışması ile ilgili mevcut yayınlar incelenmektedir. Üçüncü bölümde, tam metin arama teknolojilerinden Apache Lucene, Apache Solr ve Elasticsearch için detaylı bir anlatım ve karşılaştırma mevcuttur. Akabinde her iki teknolojinin özellik karşılaştırması yapılmaktadır. Üçüncü bölümün üçüncü alt başlığında ise testlerde kullanılan veri setlerinden, sorgulardan ve yapılandırma ortamlarından bahsedilmektedir. Dördüncü bölümde, deneysel sonuçlar indeksleme ve arama performansları açısından değerlendirilmektedir. Beşinci bölümde, web tabanlı uygulamanın mimarisi ve arayüzü anlatılmaktadır. Sonuçlar ve öneriler ise altıncı bölümde yer almaktadır.

2. LİTERATÜR ARAŞTIRMASI

Bu bölümde Apache Solr ve Elasticsearch teknolojilerini karşılaştıran makaleler incelenmektedir.

Oussous ve Benjelloun (2022), tam metin arama için yapılan araştırmaların ayrıntılı bir analizini çalışmalarında sunmaktadır. Arama motorlarının, özellikle Solr ve Elasticsearch'ün kapsamlı bir karşılaştırması yapılmaktadır. Teknolojilerin kullanım durumları, dizin oluşturma performansları, arama süreleri, parçalama ve yeniden dengeleme, veri görselleştirme ve veri kaynakları gibi çeşitli faktörler göz önünde bulundurularak mevcut arama ve dizin oluşturma teknolojileri analiz edilmektedir. Bu çalışmada yazarlar, tam metin arama konusunda önceki araştırmaları incelemekte ve kapsamlı bir teknik karşılaştırma sunmaktadır.

Elasticsearch vs. Solr Performance: Round 2 (2015), Apache Solr ve Elasticsearch için sorgulama ve indeksleme hızlarını karşılaştırmaktadır. Teknolojiler kullanım kolaylığı ve zorlukları, yapılandırma biçimleri ve mimarileri açısından analiz edilmektedir. Bu tez çalışmasına benzer şekilde, her iki teknoloji için indeksleme ve arama hızı karşılaştırılmaktadır.

Luburić ve Ivanovic (2016), Apache Solr ve Elasticsearch arasındaki ortak özellikleri ve farklılıkları karşılaştırarak incelemektedir. Yazarlar, bu iki teknolojinin karşılaştırılmasına dayanan diğer yayınlanmış çalışmaları inceleyerek ayrıntılı ve kapsamlı bir inceleme de sunmaktadır. Tez çalışmasına benzer şekilde, indeksleme ve arama performansı her iki teknoloji açısından test edilmektedir.

Kılıç ve Karabey (2016), Apache Lucene, Apache Solr ve Elasticsearch'ün çalışma ilkelerinden bahsetmektedir. Ardından Solr ve Elasticsearch teknolojileri, tam metin arama, gelişmiş filtreleme, Rest API ve içerik ekleme gibi yeteneklerine göre incelenmektedir. Diğer bir deyişle, Solr ve Elasticsearch etkinlik, kullanılabilirlik, hız ve güvenlik açısından karşılaştırılmakta ve her iki arama motorunun avantajları ve dezavantajları sunulmaktadır. Son olarak, her iki teknoloji için güvenlik yapılandırmaları ve kontrolleri incelenmektedir. Bu tez çalışması ile benzer yönü, her iki teknoloji için teknik bir karşılaştırma sunmasıdır.

Hansen ve diğ. (2018), Solr ve Elasticsearch'ü, tam metin arama sürecindeki bellek ve zaman tüketimi, işlevselliği ve indeksleme verimliliği bağlamında karşılaştırmaktadır.

Yazarlar, bir dizi deneyin sonuçlarını analiz ederek, Elasticsearch'ün indeks boyutu ve indeksleme süresi açısından Solr'a göre üstün olduğunu, ayrıca Apache Solr arama motorunun büyük ölçekli veri setlerinde daha iyi performans gösterdiğini belirtmektedir. Bu çalışma, literatür içerisinde metodoloji ve test etme açısından bu tez çalışmasına en çok benzeyen yayındır. Farklı olarak bu tezde, Apache Solr ve Elasticsearch teknolojilerinin en yeni sürümleri ve yetenekleri temel alınmaktadır.

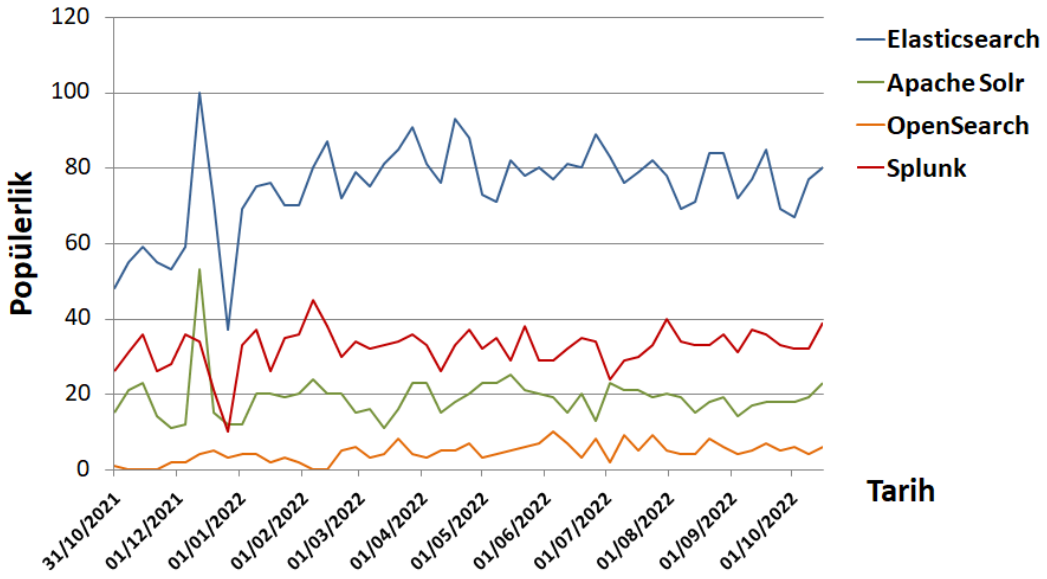
Voit ve diğ. (2017), tam metin aramada sıklıkla duyulan Apache Solr, Elasticsearch, Sphinx ve Xapian teknolojilerini, indeksleme, arama ve çeşitli teknik özellikler açısından karşılaştırmaktadır. Ancak karşılaştırmanın yapıldığı ve detaylandırıldığı çalışma, yayın içinde bahsedilen konferans çalışması D.S. (2016)'dir ve Rusça yazılmıştır. Bu nedenle derinlemesine incelenememiştir.

Yurtsever ve diğ. (2022), resim metinleri üzerinde arama yapan bir uygulama geliştirilmektedir. Deneylerde resim metinlerinden hızlı ve etkili bir arama için amaca en uygun teknoloji aranmaktadır. Bu nedenle Apache Solr ve Elasticsearch teknolojileri arasında bir karşılaştırma yapılmaktadır. Bu tez çalışmasına benzer şekilde yazarlar, iki teknolojinin arama sürelerini karşılaştırmaktadır.

Gonçalves ve Sunye (2020), DSpace veri havuzu platformunu kullanarak Apache Solr ve Elasticsearch için bir kıyaslama sağlamaktadır. İndeksleme süresi, kullanılan RAM boyutu ve oluşturulan indeks boyutu açısından bu iki teknolojinin avantajları ve dezavantajları karşılaştırılmaktadır.

3. TAM METİN ARAMA (FULL TEXT SEARCH)

Tam metin arama, büyük veri kümeleri için yaygın olarak kullanılan bir arama yöntemidir. Web arama motorları, kurumsal arama siteleri ve çeşitli veri yoğunluklu sistemler için oldukça önemlidir. Tam metin aramada iki ana adım bulunmaktadır: indeksleme ve arama. Başlangıçta veri seti indekslenmektedir, daha sonra oluşturulan indekslere göre çeşitli arama istekleri gerçekleştirilmektedir. Tam metin arama uygulamaları çoğunlukla ters dizin (inverted index) yapısını kullanmaktadır. Şekil 3.1, DB-Engines (DB-Engines, 2022) tarafından oluşturulan arama motorları sıralamasındaki ilk dört teknolojinin (Apache Solr, Elasticsearch, OpenSearch ve Splunk) popülaritesini göstermektedir. Grafik, Google Trends (Google Trends, 2022)'ten alınan verilerle oluşturulmuştur. Dört teknoloji de tam metin arama yöntemine dayanmaktadır. Elasticsearch ilk sırada yer alırken, Apache Solr en çok kullanılan üçüncü arama motorudur.



Şekil 3.1 : Son bir yılda tam metin arama araçlarının popülaritesi.

3.1 Teknolojiler

Bu bölümde Apache Lucene, Apache Solr ve Elasticsearch teknolojileri hakkında ayrıntılı bilgiler verilmektedir.

3.1.1 Apache Lucene

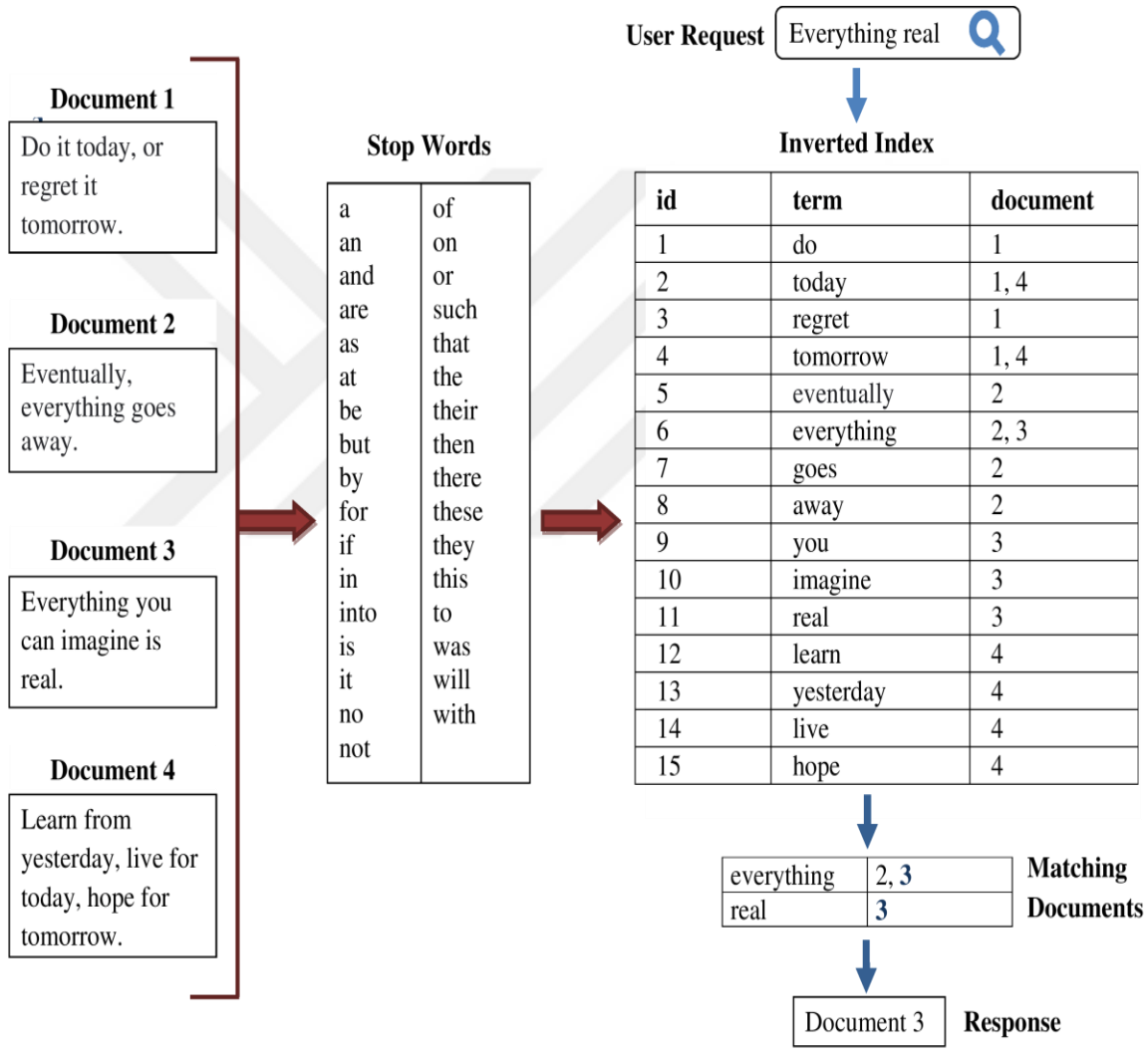
Apache Lucene, Apache Yazılım Vakfı (Apache Software Foundation) tarafından desteklenen Java tabanlı, açık kaynaklı bir arama kitaplığıdır. Güçlü bir indeksleme ve arama işlevselliği sunmaktadır. Bu nedenle tam metin arama açısından öne çıkan bir kütüphanedir (Apache Lucene, 2022). Ayrıca Apache Lucene, Python (PyLucene), .Net (DotLucene) ve C (CLucene) gibi birçok programlama dilleri üzerinde kullanılabilmektedir. PyLucene (Lokoč ve diğ, 2021) ve DotLucene (Lakhara ve Mishra, 2017) uygulamalarda kullanılan birkaç örnektir. Apache Solr ve Elasticsearch arama motorlarının temelini oluşturan Apache Lucene kütüphanesi, Java geliştiricileri tarafından kolaylıkla kullanabilmektedir. Bir arayüze sahip olmaması kullanıcılara bazı zorluklar çıkarırken, bu noktada Apache Lucene kütüphanesinin özelliklerini bir arayüz ile sunan Apache Solr ve Elasticsearch teknolojileri ön plana çıkmaktadır.

Lucene tam metin arama yönteminin üç temel modülü (analiz, indeksleme ve arama) bulunmaktadır (Gao, 2012). Daha detaylı incelendiğinde, Lucene API için paketler ve bu paketlerin temel görevleri (*org.apache.lucene*) şu şekildedir:

- *.analysis*: Dokümanları parçalara bölerek kullanılma sıklığı yüksek olan kelimeleri (stop words) ayırmaktadır. İki alt modüle sahiptir: StandardAnalyzer ve SimpleAnalyzer.
- *.codes*: Ters çevrilmiş indeks yapısının kodlanmasını ve kodun çözümlenebilmesini sağlayan soyut bir sınıftır.
- *.document*: Kayıtların ve bölümlerin yönetiminden sorumludur.
- *.index*: İndeksleme, kayıt ekleme ve silme işlemlerini içerir. Bilgi alma sistemlerinin çekirdeğidir.
- *.search*: Arama sorgularının yapıldığı ve arama sonuçlarını barındıran pakettir.
- *.store*: Kalıcı verileri depolamak için tanımlanan soyut bir sınıftır.
- *.util*: Veri yapılarını ve birim sınıflarını içermektedir.

Büyük veride ölçeklenebilirlik oldukça önemlidir (Oussous ve diğ, 2018). Apache Lucene kullanılırken sistem gereksinimleri, işlenecek belge ve isabet sayısı, belgelerin boyutu gibi

özellikler ölçeklenebilmektedir, bu açıdan verimli bir indeksleme sağlamaktadır. Örneğin, yalnızca 1 MB önbellek ile daha az RAM kullanır ve oluşturulan dizin, mevcut verilerin yaklaşık üçte biri kadardır. Bu bağlamda, indeksleme işlemi önemli ölçüde optimize edilmektedir. Apache Lucene, arama motorlarına hızlı ve etkili erişim sağlayan ters çevrilmiş bir dizin (inverted index) mekanizması sağlamaktadır (bkz. Şekil 3.2). Dizinleme başka bir deyişle indeksleme, yapılandırılmış veriler (structured data) için gerçekleştirilmektedir.



Şekil 3.2 : Tersine çevrilmiş indeksin mekanizması (Deniz ve diğ, 2023).

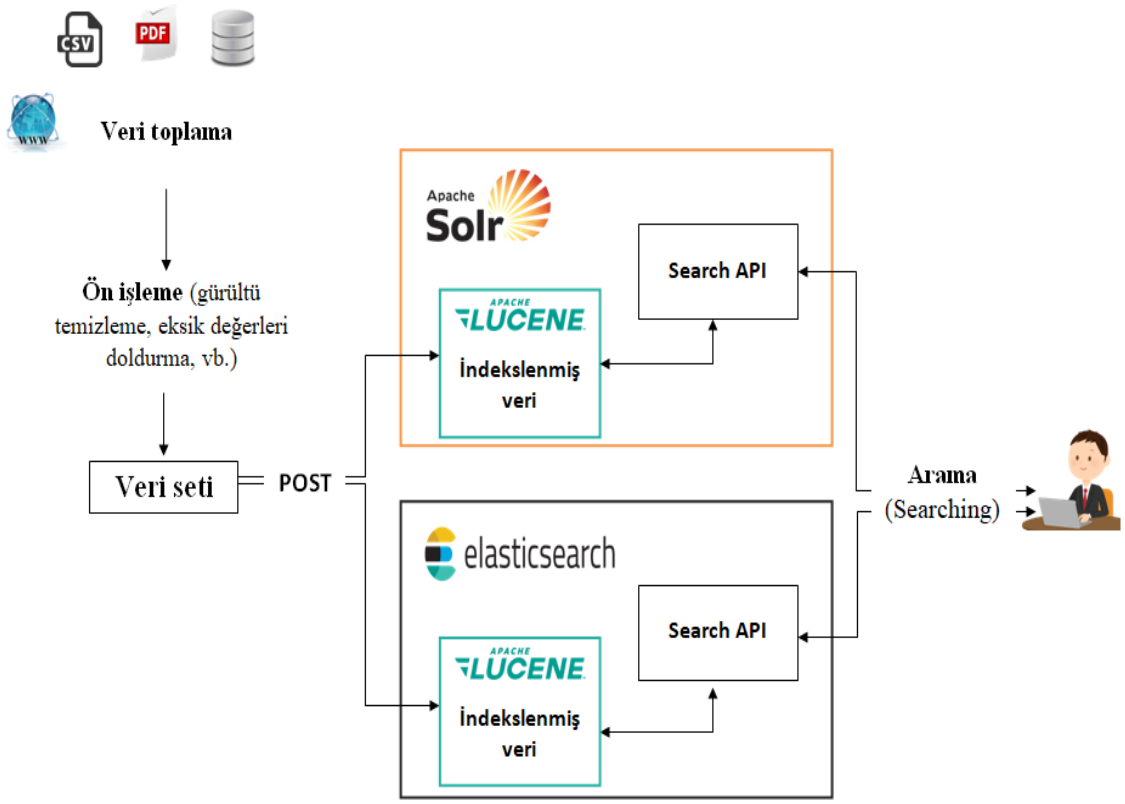
Şekil 3.2'de ters indeks mekanizmasının bir örneği gösterilmektedir. Belgeler (document), dizine eklenecek veri kümesini göstermektedir. Diğer bir deyişle belgeler, ilişkisel bir veri tabanındaki her satır olarak düşünülebilir. Durdurma sözcükleri (stop words) kümesindeki ifadeler göz ardı edilir. Her bir terim için ilişkili belgeler, tersine çevrilmiş dizin yapısına

eklenir ve arama işlemleri bu dizin kullanılarak gerçekleştirilir. Örneğin Şekil 3.2'deki gibi "Everything real" ifadesi arandığında, ters dizinde aranan her kelime için eşleşen dokümanlar bulunmaktadır. Bulunan belgelerde, her ifade için ortak olanlar filtrelenerek bir yanıt döndürülmektedir. Sonuç olarak, belge 3 (document 3) uygun cevap (response) olarak kullanıcıya sunulmaktadır.

Apache Lucene kitaplığının arama yetenekleri incelendiğinde, neden Apache Solr ve Elasticsearch gibi iki popüler teknolojinin temeli olduğu açıkça görülmektedir. Apache Lucene doğru, güçlü ve verimli bir arama sağlamaktadır ve öne çıkan en önemli özelliklerinden biri, iyi sonuçların döndürülmesi ilkesine göre sıralı arama yapmasıdır (Apache Lucene, 2022). Öne çıkan diğer bazı özellikleri aşağıdaki gibi örneklendirilebilir:

- Arama, tümcecik ve yakınlık sorgulama,
- Unvan, isim, yaş gibi herhangi bir alanda arama yapabilme,
- Herhangi bir alana göre sıralama yeteneği,
- Esnek vurgulama, birleştirme ve sonuç gruplandırma.

Şekil 3.3, Apache Lucene kütüphanesinin Apache Solr ve Elasticsearch teknolojileri ile arasındaki ilişkiyi göstermektedir. Bu şekil hem genel yapıyı hem de bu tez özelindeki bağlantıyı temsil etmektedir. Kullanılacak veri seti, veri toplama aşamasında oluşturulmaktadır ve indekslenmeden önce, kullanım amacına uygun olarak gürültü temizleme, kayıp değerleri doldurma gibi ön işleme tabi tutulmaktadır. Daha sonra, her iki teknoloji kendi altyapısında bulunan Apache Lucene kütüphanesini kullanarak sunucularında verileri indeksleyerek depolamaktadır. Son olarak, kullanıcılardan gelen farklı arama istekleri için indekslenen veriler üzerinden arama API'leri kullanılarak kullanıcıya bir yanıt verilmektedir.



Şekil 3.3 : Apache Lucene ve ilgili teknolojiler.

3.1.2 Apache Solr

Apache Solr, Apache Lucene üzerine kurulmuş açık kaynaklı bir tam metin arama motorudur. Büyük veri kümelerinde yüksek performanslı aramalar yapmak için tasarlanmıştır. En büyük avantajlarından biri, kullanıcı dostu bir arayüz sağlamasıdır. Apache Solr, metin tabanlı yapılandırılmış veriler üzerinde çalışır. Veriler temel olarak JSON olarak indekslenir, ancak CSV ve XML gibi diğer biçimleri de desteklemektedir. Apache Solr'da veri kümeleri üç farklı yol ile indekslenebilir: Solr arayüzüden, curl ile komut satırından veya bir API aracılığıyla.

Apache Solr, çok alanlı ve yönlü arama ile karmaşık sorgulara hızla yanıt verebilmektedir. Ayrıca sonuçları daraltan ve birleştiren güçlü matematiksel ifadelere sahiptir. En önemli özelliği, büyük hacimli ve veri yoğunluklu uygulamalarda etkin bir şekilde kullanılabilmesidir. Aynı zamanda, birden çok sunucuda dağıtılmış bir sistemde çalışabilme imkanı sağlamaktadır. Günümüzde Macy's, eBay ve Zappos gibi Apache Solr kullanan siteler, bu teknolojinin yüksek hacimli ve yoğun veri kullanan uygulamalardaki birkaç örneğidir (Recources Apache Solr, 2022).

Şekil 3.4'te, responseHeader (sorgunun temel parametreleri) ve response (yanıtın içerikleri) anahtarlarını içeren örnek bir yanıt, başka bir deyişle anahtar/değer (key/value) biçimindeki JSON dosyası verilmektedir. ResponseHeader anahtarı, sorgu parametrelerini (params), gerçekleştirilen işlem süresini (QTime) ve işlemin hatasız yapılıp yapılmadığını (status) bildirmektedir. Ek olarak, yanıt (response) anahtarı, arama sonrasında bulunan belgelerin sayısını (numFound) ve içeriğini (docs) bulundurmaktadır.


```
{
  "responseHeader":{
    "status":0,
    "QTime":29,
    "params":{
      "q":"*:*",
      "rows":"1"}},
  "response":{"numFound":64295,"start":0,"numFoundExact":true,"docs":[
      {
        "App":["10 Best Foods for You"],
        "Translated_Review":["nan"],
        "Sentiment":["nan"],
        "Sentiment_Polarity":["nan"],
        "Sentiment_Subjectivity":["nan"],
        "id":"68dfde9f-f28b-434c-ace8-58a673e7c276",
        "_version_":1745872260939382784}]
  }}
```


Şekil 3.4 : Apache Solr'dan bir yanıt (response) örneği.

3.1.3 Elasticsearch

Elasticsearch, Apache Solr gibi Apache Lucene kütüphanesi üzerine inşa edilmiştir. Açık kaynaklı, JSON tabanlı bir arama ve analiz motorudur. Elasticsearch, metinsel, niceliksel, coğrafi, yapılandırılmış ve yapılandırılmamış verileri depolamak için kullanılanılan bir veri tabanı olarak da kullanılabilmektedir. Kapsamlı REST API'ler, dağıtılmış bir mimari, gerçek zamanlı işlemler, verimlilik, esneklik ve ölçeklenebilirlik açısından çeşitli avantajlar sunmaktadır. Elasticsearch, Logstash ve Kibana uygulamaları ile entegreli bir şekilde arama, analiz ve görselleştirme gerçekleştirmektedir. Ek olarak, Elasticsearch, güvenlik analitiği ve altyapı izleme gibi zamana duyarlı kullanım durumları için de tercih edilmektedir (Elastic Installation and Upgrade Guide [8.4], 2022). Mozilla, Foursquare, GitHub gibi büyük projelerde içerik arama, veri analizi ve sorgular için Elasticsearch kullanılmaktadır.

Şekil 3.5'te, arama işlemi için harcanan zaman (took), zaman aşım durumu (timed_out), bulunan belgelerin miktarı (value) ve içeriği (hits) gibi çeşitli bilgileri kapsayan örnek bir Elasticsearch yanıtı gösterilmektedir.


```
{
  "took" : 37,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "apps_reviews",
        "_id" : "GPyMsYMB9ZTBF8qgpnd6",
        "_score" : 1.0,
        "_source" : {
          "App" : "10 Best Foods for You",
          "Sentiment_Polarity" : "nan",
          "Sentiment_Subjectivity" : "nan",
          "Translated_Review" : "nan",
          "Sentiment" : "nan"
        }
      }
    ]
  }
}
```


Şekil 3.5 : Elasticsearch'ten bir yanıt örneği.

Elasticsearch'te dosyalar Kibana uygulaması üzerinden ya da API aracılığı ile yüklenebilmektedir. Ek olarak komut isteminden de yüklenebilir ama bunun için veri setinin Elasicsearch'e özgü bir formata dönüştürülmesi gerekir. Bu format, 3.3.2 alt başlığında detaylı bir şekilde anlatılmaktadır.

3.2 Teknik Karşılaştırma

Bu bölümde Apache Solr ve Elasticsearch'ün özellikleri açıklanmaktadır. Her iki sistem tarafından şu terminoloji kullanılmaktadır: field (alan), value (değer), document (belge), node (düğüm), core/indice (çekirdek/indeks ya da dizin), collection (koleksiyon), cluster (küme) ve docs/hits (belgeler/isabetler).

Field, verilerin nasıl tanımlandığını gösterirken *value*, bu tanıma karşılık gelen değerleri belirtmektedir. Bu iki terimi field-value çifti olarak da tanımlamak mümkündür. *Core*, Apache Solr'daki mantıksal bir dizini temsil etmektedir. Başka bir deyişle, bir Solr sunucusu örneğidir. Apache Solr'daki core terimi, Elasticsearch'teki *indice* terimine karşılık gelmektedir. Bir veya daha fazla alana (fields) sahip olan belge (document), dizin birimi olarak kabul edilir. *Collection,* bir veya daha fazla belgeden oluşmaktadır ve her koleksiyon, tek bir mantıksal dizin oluşturan parçalara veya kopyalara sahiptir. Ek olarak, her koleksiyon farklı ve esnek ayarlara ve şema tasarımlarına sahip olabilmektedir. *Node*, fiziksel bir sistem veya sunucu üzerinde çalışan tek bir Apache Solr veya Elasticsearch örneğidir. Aynı zamanda c*luster,* birden çok düğüm içeren yapıları göstermektedir. Apache Solr'da bir sorgu sonucu *docs* parametresinde listelenirken, Elasticsearch'te sonuçlar *hits* parametresinde listelenmektedir.

Çizelge 3.1, Apache Solr ve Elasticsearch'ün genel özelliklerinin bir karşılaştırmasını sunmaktadır. Karşılaştırma şu özelliklere dayanmaktadır: yayın yılı, arama motorunun geliştirilmesinde kullanılan kütüphane, geliştiren yayın kuruluşu, bu çalışma esnasında var olan sürüm, erişim protokolleri, sunucuya veri yükleme araçları, sunucuda desteklenen veri formatları, desteklenen programlama dilleri ve teknolojilerin kullanılabileceği işletim sistemleri. Çizelge incelendiğinde, Elasticsearch'ün ApacheSolr'dan daha fazla programlama dilinde kullanılabildiği görülmektedir. Apache Solr'ın kendi arayüzüne sahip olması kullanıcıya bir avantaj sağlarken, Elasticsearch teknolojisi Kibana, Marvel gibi üçüncü parti uygulamalar ile bir arayüz sunmaktadır. Bu çalışmada, Apache Solr'un 9.0.0 ve Elasticsearch'ün 8.4.2 sürümleri kullanılmaktadır. Apache Solr 9.0.0 versiyonu

minimum Java 11, Elasticsearch'ün 8.4.2 sürümü ise minimum Java 17 gerektirmektedir. Bu tez çalışmasında her iki teknoloji için Java 17 kullanılmaktadır.

Çizelge 3.1 : Apache Solr ve Elasticsearch teknolojilerinin özellik karşılaştırması.

	Apache Solr	**Elasticsearch**
Yayın yılı	2004	2010
Altyapıda kullanılan kütüphane	Apache Lucene	Apache Lucene
Geliştirici	Apache Software Foundation	Elastic
Lisans	Açık kaynak	Açık kaynak ya da kurumsal
Mevcut sürüm	9.0.0	8.4.2
Web arayüzü	Kendi sisteminde mevcut	Kibana, Marvel gibi uygulamalar aracılığıyla
Erişim protokolleri	REST API (http kullanılarak)	REST API (http kullanılarak)
Veri yükleme araçları	Data Import Handler (DIH), ApacheTika (PDF, Word, vb.)	Kibana (JSON, CSV, NDJSON)
Desteklenen veri formatları	CSV, XML, JSON	JSON
İstemci kütüphaneler	Java, Python, Ruby, PHP, C# / .NET, Scala, Perl, JavaScript / JSON, Node.js, Clojure, Go, Rust, R, C++, Lua	B4J, C++, Clojure, ColdFusion (CFML), Erlang, Go, Haskell, Java, JavaScript, Kotlin, Lua, .NET, Perl, PHP, Python, R, Ruby, Rust, Scala, Smalltalk, Swift, Vert.x
Desteklenen işletim sistemleri	Java Sanal Makinesi (Java Virtual Machine) içeren bütün işletim sistemleri	Java Sanal Makinesi (Java Virtual Machine) içeren bütün işletim sistemleri

3.3 Uygulama Ortamı

Bu bölümde testlerde kullanılan cihazların donanım özellikleri, karşılaştırılacak arama motorlarının mevcut konfigürasyonları, kullanılan veri setleri ve arama işlemi için oluşturulan sorgular ele alınmaktadır.

3.3.1 Donanım ve konfigürasyon

RAM boyutu, işlemci türü, sabit disk kapasitesi, CPU çekirdek sayısı ve işletim sistemi farklı olan üç bilgisayarda indeksleme performansları test edilmiştir. Deneylerde kullanılan makinelerin özellikleri Çizelge 3.2'de verilmektedir.

Çizelge 3.2 : Testlerde kullanılan makinelerin özellikleri.

Parametre	Makine 1 (M1)	Makine 2 (M2)	Makine 3 (M3)
RAM boyutu	32 GB	16 GB	12 GB
İşlemci türü	Intel(R) Core (TM) i7-12700H 2.50 GHz	Intel(R) Core (TM) i7-9750H 2.60 GHz	Intel(R) Core (TM) i5-4210M 2.60 GHz
Sabit disk kapasitesi	1 TB M.2 3.0 SSD	256 GB M.2 3.0 SSD	256 GB SATA 3.0 SSD
CPU çekirdek sayısı	14 çekirdek	6 çekirdek	2 çekirdek
İşletim sistemi	Win 11	Win 11	Win 10

Apache Solr'ın temelde üç yapılandırma dosyası bulunmaktadır:

- *solrconfig.xml*: Apache Solr'a yapılan istekleri işleyen dosyadır. Örneğin, sorgu sonuçlarını döndürme, veri tabanı bağlantısı, dizine belge ekleme gibi işlemler için bu dosyanın yapılandırılması gerekmektedir.
- *schema.xml* (ya da *managed-schema.xml*): Apache Solr'ın verileri indeksleyip kullanabilmesi için veri setindeki veri türlerinin önceden belirtilmesi gerekmektedir. Bu tanımlamaların yapıldığı dosya *schema.xml*'dir. Bunun yanı sıra tanımlamalar için Schema API de kullanılabilir. Schema API, verilerin türlerini tahmin ederek xml dosyasına otomatik tanımlamalar yapar. Schema API ile çalışıldığında, dizinde *schema.xml* dosyası yerine yönetilen şema olduğunu belirten *managed-schema.xml* dosyası yer alır.
- *core.properties*: Core bilgileri bulunmaktadır.

Aynı şekilde, Elasticsearch için de üç konfigürasyon dosyası mevcuttur:

- *elasticsearch.yml*: Mevcut konfigürasyonu güncellemek, yeni parametreler ve nitelikler eklemek için yapılandırılan dosyadır. Başka bir deyişle küme veya düğüm için temel ayarlar burada yapılmaktadır.
- *jvm.options*: Java sanal makinesi ile ilgili bütün ayarlar bu dosya üzerinden işlenmektedir.
- *log4j2.properties*: Elasticsearch'ün günlük kayıt ayarlarını içermektedir.

Apache Solr ve Elasticsearch iki şekilde yapılandırılabilmektedir: tek düğüm (single node) ve çoklu düğüm (multi node). Bu çalışmada, her iki teknoloji için tek düğüme göre yapılandırma yapılmıştır. Yukarıda Apache Solr ve Elasticsearch için bahsedilen konfigürasyon dosyaları incelendiğinde bazı temel özelliklerin varsayılan değerleri ve tanımları şu şekildedir:

- Apache Solr arama sorgularında *rows* parametresi değiştirilmediği sürece sadece 10 sonuç arar ve bunları ekrana yazdırır.
- Elasticsearch'te *size*, Apache Solr'daki rows parametresine karşılık gelir ve varsayılan değeri 10,000'dir. Ancak ekrana yalnızca ilk 10 sonucu yazdırır.
- Apache Solr'daki RAM arabellek boyutu 100 MB'tır ve arabelleğe alınabilecek maksimum belge sayısı 1,000'dir.
- Elasticsearch için minimum arabellek boyutu 48 MB'tır.
- Önbellek boyutu, Elasticsearch'ün varsayılan ayarlarında sınırsız olarak tanımlanmıştır.

Adil bir karşılaştırma yapmak amacıyla her iki teknoloji için de varsayılan yapılandırma ayarları (güvenlik ayarları hariç) kullanılmıştır. Ek olarak, her iki teknolojinin bu çalışmada kullanılan sürümlerinde, Apache Solr'un güvenlik ayarları başlangıçta devre dışıdır, ancak Elasticsearch'ün varsayılan olarak etkindir. Bu nedenle testlerden önce Elasticsearch için, elasticsearch.yml dosyasındaki **xpack.security.enabled**, **xpack.security.enrollment.enabled** ve **xpack.security.http.ssl** özellikleri false olarak güncellenmiştir. Böylece her iki teknoloji için güvenlik ayarları devre dışı bırakılmıştır.

Apache Solr ve Elasticsearch teknolojileri arasında detaylı bir karşılaştırma yapmak için bu çalışmada üç farklı veri seti kullanılmaktadır. Çizelge 3.3'te, veri setlerinin boyutları ve içerikleri hakkında ayrıntılı bilgiler sunulmaktadır.

Çizelge 3.3 : Veri setlerinin özellikleri.

Parametre	**Veri seti 1 (V1)**	**Veri seti 2 (V2)**	**Veri seti 3 (V3)**
Adı	Google Play StoreApps (Google Play Store Apps, 2022)	Web of Science (Kowsari ve diğ, 2018)	Dota 2 Matches (Dota 2 Matches, 2022)
Boyutu	~17 MB	~ 75 MB	~ 300 MB
Veri miktarı (satır bazlı)	64,295	46,985	1,500,000
Veri tipi	Metin ve sayısal değerler	Metin	Sayısal değerler
Alanlar	App, Translated_Review, Sentiment, Sentiment_Polarity, Sentiment_Subjectivity	Y, Y2, Y1, Domain, Area, Keywords, Abstract	match_id, player_slot, buybacks, damage, deaths, gold_delta, xp_end, xp_start

Birinci veri seti, Google Play Store'daki uygulamalarla ilgili kullanıcı görüşlerini içermektedir (bkz. Şekil 3.6). Bu veri setinde iki dosya bulunmaktadır. Bu çalışmada metin yoğunluğu fazla olan dosya kullanılmıştır. Veri kümesinin beş özelliği bulunmaktadır: App (uygulamanın adı), Translated_Review (farklı dillerden çevrilmiş kullanıcı görüşleri), Sentiment (olumlu veya olumsuz görüş), Sentiment_Polarity ve Sentiment_Subjectivity.

```
App,Translated_Review,Sentiment,Sentiment_Polarity,Sentiment_Subjectivity
10 Best Foods for You,nan,nan,nan,nan
10 Best Foods for You,"I like eat delicious food. That's I'm cooking food myself,   case ""10
Best  Foods"" helps lot, also ""Best Before (Shelf   Life)""",Positive,1,0.533333333
```

Şekil 3.6 : Birinci veri setinden bir ekran görüntüsü.

İkinci veri seti, Web of Science platformunda 134 dergi kategorisinde yer alan 46,985 yayın hakkında çeşitli bilgileri bulundurmaktadır. Şekil 3.7'de, ikinci veri setinde bulunan veri çeşitleri şu şekilde ifade edilmektedir: Y1, Y2, Y, Domain (bilgisayar bilimi, elektrik mühendisliği, psikoloji, makine mühendisliği, inşaat mühendisliği, tıp bilimi ve biyokimya başlıklarını içeren birincil alan), Area (alt alan), Keywords (makalelerin anahtar kelimeleri), Abstract (makale özeti).

```
Y1,Y2,Y,Domain,Area,Keywords,Abstract
0,12,12,CS,Symbolic computation, (2+1)-dimensional non-linear optical waves; erbium-doped  optical
fibre; symbolic computation; soliton solution; soliton interaction,"(2 + 1)- dimensional non-linear
optical waves through the coherently excited resonant medium doped with  the erbium atoms can be
described by a (2 + 1)-dimensional non-linear Schrodinger equation coupled with the self-induced
transparency equations. For such a system, via the Hirota method and symbolic computation, linear
forms, one-, two-and N-soliton solutions are obtained. Asymptotic analysis is conducted and suggests
that the interaction between the two solitons is  elastic. Bright solitons are obtained for the fields
E and P, while the dark ones for the  field N, with E as the electric field, P as the polarization in
the resonant medium induced by  the electric field, and N as the population inversion profile of the
dopant atoms. Head-on  interaction between the bidirectional two solitons and overtaking interaction
between the  unidirectional two solitons are seen. Influence of the averaged natural frequency. on the
solitons are studied: (1). can affect the velocities of all the solitons; (2) Amplitudes of  the
solitons for the fields P and N increase with. decreasing, and decrease with. increasing; (3) With.
decreasing, for the fields P and N, one-peak one soliton turns into the two-peak  one, as well as
interaction type changes from the interaction between two one-peak ones to  that between a one-peak
one and a two-peak one; (4) For the field E, influence of. on the  solitons cannot be found. The
results of this paper might be of potential applications in the  design of optical communication
systems which can produce the bright and dark solitons simultaneously."
```

Şekil 3.7 : İkinci veri setinden bir ekran görüntüsü.

Üçüncü veri seti, Opendota tarafından oluşturulan Dota 2 veri dökümünden 50,000 tane sıralanmış merdiven maçı skorunu içermektedir. Veri setinde 19 dosya bulunmaktadır ama bu çalışmada sadece teamfights_players.csv dosyası kullanılmaktadır. Dosyanın sekiz niteliği mevcuttur: match_id (bireysel oyuncu kimlikleri), player_slot (veri kümesindeki diğer dosyalara bağlantı kodu), buybacks, damage, deaths, gold_delta (altın kazanma veya kaybetme durumu), xp_end (sondaki deneyim durumu), xp_start (başlangıçtaki deneyim durumu).

```
match_id,player_slot,buybacks,damage,deaths,gold_delta,xp_end,xp_start
0,0,0,105,0,173,536,314
0,1,0,566,1,0,1583,1418
```

Şekil 3.8 : Üçüncü veri setinden bir ekran görüntüsü.

3.3.2 Sorgular

Bu bölümde, testlerde kullanılan indeksleme ve arama sorguları açıklanmaktadır. Öncelikle indeksleme sorgularında kullanılacak dosya türleri incelendiğinde, Apache Solr için CSV, JSON, XML dosyaları doğrudan kullanılabilirken Elasticsearch için bu formatlar kullanılamamaktadır. Elasticsearch'te JSON yapısına benzeyen yeni satırla ayrılmış JSON

(Newline Delimited JSON ya da NDJSON) yapısı kullanılmaktadır. Komut isteminde, Elasticsearch'ün indeksleme işleminde veri seti dosyasının NDJSON formatında olması gerekmektedir. Apache Solr'da böyle bir kısıtlama olmamak ile birlikte JSON, CSV ve XML dosyaları da komut isteminden dizine eklenebilmektedir. İndeksleme sorgularında, Elasticsearch için NDJSON formatı kullanılırken daha doğru ve adil bir karşılaştırma yapabilmek için Apache Solr'un da bir JSON dosyası kullanması gerektiği düşünülmüştür. Şekil 3.9, Solr'da dizine eklenecek dosyaların genel yapısını sunan iki belgeli örnek bir JSON yapısını göstermektedir.

```
[
    {
        "App": "10 Best Foods for You",
        "Translated_Review": "nan",
        "Sentiment": "nan",
        "Sentiment_Polarity": "nan",
        "Sentiment_Subjectivity": "nan"
    },
    {
        "App": "10 Best Foods for You",
        "Translated_Review": "I like eat delicious food. That's I'm cooking food
myself, case '10 Best Foods' helps lot, also 'Best Before (Shelf Life)'",
        "Sentiment": "Positive",
        "Sentiment_Polarity": "1",
        "Sentiment_Subjectivity": "0.533333333"
    }
]
```

Şekil 3.9 : İki dokümanlı bir JSON dosyası.

Şekil 3.10, Elasticsearch için oluşturulan NDJSON dosyasının iki belgeli örnek bir yapısını göstermektedir. JSON dosyasının aksine içeriğin başında ve sonunda köşeli parantez içermez. Ek olarak, {"index": {…}} yapısı kullanılarak her dokümanın dizin bilgilerini içeren satırlar belgede yer almalıdır. NDJSON, alt satırlar (\n) ile ayrılmış JSON öğelerinden oluşan bir koleksiyondur. NDJSON oluşturmanın dezavantajı maliyet ve zaman gerektirmesidir.

```
{"index": {"_index": "apps_reviews", "_id": "1"}
{"App":"10 Best Foods for You", "Translated_Review":"nan", "Sentiment":"nan",
"Sentiment_Polarity":"nan", "Sentiment_Subjectivity":"nan"}
{"index": {"_index": "apps_reviews", "_id": "2"}
{"App":"10 Best Foods for You", "Translated_Review":"I like eat delicious food. That's I'm
cooking food myself, case '10 Best Foods' helps lot, also 'Best Before (Shelf Life)'",
"Sentiment":"Positive", "Sentiment_Polarity":"1", "Sentiment_Subjectivity":"0.533333333"}
```

Şekil 3.10 : NDJSON için basit bir örnek.

Apache Solr, yerelde 8983 portunda çalışırken Elasticsearch 9200 numaralı bağlantı noktasını kullanmaktadır. Çizelge 3.4'te, iki teknolojinin indeksleme sorgularında kullanılan genel bir sorgu yapısı verilmiştir. Sunucularda oluşturulacak dizin isimleri Apache Solr'da **core_name** alanına, Elasticsearch'te **indice_name** alanına yazılır. Sorgu

yapısında kullanılan core/indice adları, birinci veri seti için apps_reviews, ikinci veri seti için wos_papers ve üçüncü veri seti için teamfights_players şeklindedir. Böylece Apache Solr'da oluşturulan core isimleri ile Elasticsearch'te oluşturulacak indice isimleri her veri seti için aynı olacaktır. Bu isimler kullanılarak Solr ve Elasticsearch için indekslenecek dosyalar oluşturulmuştur. Çizelge 3.4'te görüldüğü gibi Apache Solr için **core_name** hem çekirdek ismi hem de json dosyasının ismini oluşturmaktadır. Elasticsearch için de ndjson dosyası **indice_name** kısmına ek olarak **nd_** belirteci kullanılarak oluşturulmuştur. Örneğin, birinci veri seti için curl sorgusunda core/indice adı apps_reviews olacaktır. Böylece Apache Solr sunucusuna **apps_reviews**.json dosyası, Elasticsearch sunucusuna da **nd_apps_reviews**.json dosyası gönderilecektir. Komut isteminde, indeksleme sorguları POST metodu ile gerçekleştirilmektedir.

Çizelge 3.4 : İndeksleme sorgularının genel yapısı.

Teknoloji	**İndeksleme sorgu**
Apache Solr	curl-H "Content-Type: application/json" -XPOST http://localhost:8983/solr/**core_name**/update -T "C:/solr-9.0.0/example/**core_name**.json"
Elasticsearch	curl -H "Content-Type: application/x-ndjson" -XPOST http://localhost:9200/_bulk --data-binary @C:/elasticsearch-8.4.2/example/nd_**indice_name**.json

Çizelge 3.5'te, Apache Solr ve Elasticsearch teknolojilerindeki arama işlemleri için genel bir sorgu formatı verilmektedir. Sorgu içeriği (**query content**) öğesi, her iki teknoloji için de ortaktır. Elasticsearch, arama sorgularında Kibana Query Language (KQL) formatını da desteklemektedir ancak arama sürelerinde daha doğru ve adil sonuçlar için her iki platformun da desteklediği Lucene sorgulama dili kullanılmıştır. Ek olarak, Elasticsearch sorgularında varsayılan değeri false olan **track_total_hits** parametresi true olarak değiştirilmiştir.

Çizelge 3.5 : Arama sorgularının genel yapısı.

Teknoloji	**Arama sorgusu**
Apache Solr	http://localhost:8983/solr/**core_name**/select?q=**query content**
Elasticsearch	http://localhost:9200/**indice_name**/_search?q=**query content**&track_total_hits=true

Apache Solr bir arama sırasında tüm sonuçları ararken, Elasticsearch belirli bir eşiğe ulaştıktan sonra (varsayılan 10,000) kalan verilere odaklanmaz. Çizelge 3.5'te 10,000'den

fazla sonuç getiren sorgular da bulunmaktadır. Bu yüzden track_total_hits değişkeninin ayarlanması gerekmektedir. Apache Solr'da olduğu gibi Elasticsearch'ün tüm isabetleri bulması, arama sürelerinin karşılaştırılması noktasında oldukça önemlidir.

Çizelge 3.6 : Arama testlerinde kullanılan sorgular.

	Sorgu	**Amaç**	**query content**
V1	S1	Translated_Review değeri'nan' olan dokümanları listeleyiniz.	Translated_Review:nan
	S2	Sentiment_Subjectivity değeri 0.1 ile 0.746 arasında olan dokümanları listeleyiniz.	Sentiment_Subjectivity:[0.1 TO 0.746]
	S3	App alanında 'Food', Sentiment alanında 'Positive' ve Translated_Review alanında 'Full', 'great', 'good' ya da 'enjoy' kelimelerini içeren dokümanları listeleyiniz.	App:*Food* AND Sentiment:Positive AND Translated_Review:*Full* OR Translated_Review:*great* OR Translated_Review:*good* OR Translated_Review:*enjoy*
	S4	Sentiment_Subjectivity değeri 0.79 ile 0.82 arasında ve Sentiment_Polarity değeri de 0.716666667 olan dokümanları listeleyiniz.	Sentiment_Subjectivity:[0.79 TO 0.82] AND Sentiment_Polarity:0.716666667
	S5	Keywords alanında 'Parkinson' değerini içeren dokümanları listeleyiniz.	Keywords:*Parkinson*
V2	S6	Keywords ya da Abstract alanında 'algorithm' kelimesini içeren dokümanları listeleyiniz.	Keywords:*algorithm* OR Abstract:*algorithm*
	S7	Keywords alanında 'analysis' ya da Domain alanında 'CS' ya da Abstract alanında 'system' değerini içeren dokümanları listeleyiniz.	Keywords:analysis OR Domain:*CS* OR Abstract:system
V3	S8	xp_end değeri 32417 olan dokümanları listeleyiniz.	xp_end:32417
	S9	buybacks ve deaths değerleri 1 olan dokümanları listeleyiniz.	buybacks:1 AND deaths:1
	S10	buybacks değeri 0 ve deaths değeri 1 ve damage değeri 0 ya da gold_delta değeri 0 olan dokümanları listeleyiniz.	buybacks:0 AND deaths:1 AND damage:0 OR gold_delta:0

Çizelge 3.6’da, değişen karmaşıklık seviyelerine sahip olan 10 sorgu mevcuttur: S1 ile S4 arası birinci veri seti için, S5 ile S7 arası ikinci veri seti için, S8 ile S10 arası üçüncü veri seti için oluşturulmuştur. Sorgu karmaşıklığı, her sorgu kümesinin (S1-S4, S5-S7 ve S8-S10) kendi içinde artmaktadır. Apache Solr ve Elasticsearch araçları, Apache Lucene üzerine inşa edildiğinden Lucene sorgu yapısını kullanmaktadır. Bu sorgu dilinde AND ve OR operatörleri birkaç alanı birleştirmek için kullanılmaktadır. * simgesi, kelimenin başında veya sonunda bir ifade olabileceğini belirtirken, [sayı1 TO sayı2] ifadesi sayı1 ile sayı2 arasındaki değerleri göstermektedir.

4. DENEYSEL SONUÇLAR VE KARŞILAŞTIRMA

Bu bölümde, Apache Solr ve Elasticsearch teknolojileri için yapılan testlerin sonuçları indeksleme ve arama süreleri açısından karşılaştırılmaktadır.

4.1 İndeksleme

Bu bölümde, farklı donanım özelliklerine sahip üç makine (bkz. Tablo 3.2) için üç farklı veri setindeki indeksleme süreleri incelenmektedir. İlk olarak, mevcut konfigürasyonlardaki varsayılan yığın boyutu için indeksleme süreleri ölçülmüştür. Daha sonra, indekslemede en iyi performansa sahip makinede (M1), farklı dosya boyutları için indeksleme süreleri test edilmiştir. Çizelge 4.1'de, Gigabyte (GB) cinsinden farklı yığın boyutlarına (6, 8, 12, 16, 20 ve 24) göre, yığın boyutunun indekslemeyi nasıl etkilediği hakkında bir karşılaştırma yapılmıştır.

Çizelge 3.4'te gösterildiği gibi, dizin oluşturma süresi, Windows komut satırı aracılığıyla her veri seti için curl istekleri kullanılarak toplanmaktadır. Şekil 4.1'de, komut isteminde yürütülen bir sorgunun çıktısı gösterilmektedir. İndeksleme süresini hesaplamak için Çizelge 3.4'te verilen Apache Solr ve Elasticsearch curl sorgularının başına **timecmd** komutu eklenmektedir. Bu komut curl sorgularının çalışma zamanını hesaplayan bir toplu iş dosyasını (Elasticsearch vs Solr, 2023) ifade etmektedir. Command took, timecmd dosyası tarafından hesaplanan süreyi göstermektedir. Bu çalışmada, 'command took' parametresinin verdiği süreler ele alınmıştır.

```
C:\>timecmd curl -H "Content-Type: application/json" -XPOST http://localhost:8983/
solr/apps_reviews/update  -T "C:/solr-9.0.0/example/apps_reviews.json"
{
  "responseHeader":{
    "status":0,
    "QTime":5030}}
command took 0:0:5.13 (5.13s total)

C:\>
```

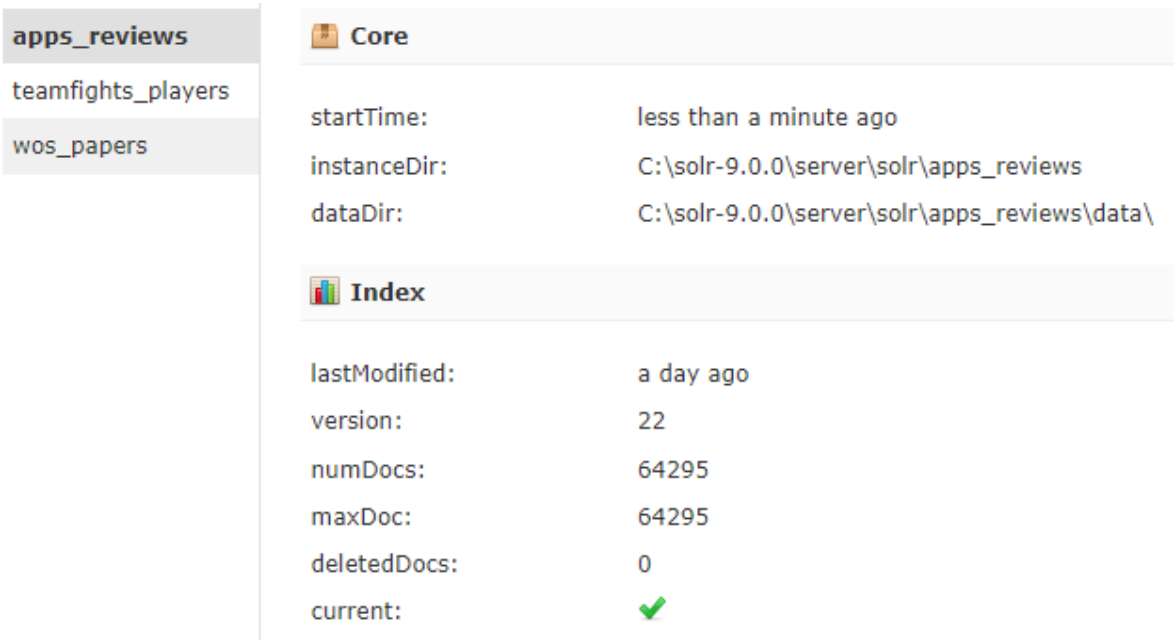
Şekil 4.1 : Komut isteminde gönderilen örnek bir curl sorgusu.

Çizelge 3.4'te verilen indeksleme sorguları çalıştırıldıktan sonra, Elasticsearch'te oluşan dizinler ve detaylı bilgileri (çalışma durumu, kopya sayısı, belge sayısı) Şekil 4.2'de gösterilmektedir. Bu ekran görüntüsü, Kibana uygulaması üzerinden alınmıştır. Aynı şekilde Şekil 4.3'te Apache Solr'da oluşturulan üç core ve ayrıntıları (çekirdeğin

oluşturulduğu ve verilerin bulunduğu dosya yolları, belge sayısı, çekirdeğin aktif durumu, günlük tarihleri) gösterilmektedir.

Name	Health	Status	Primaries	Replicas	Docs count
apps_reviews	yellow	open	1	1	64295
teamfights_players	yellow	open	1	1	1500000
wos_papers	yellow	open	1	1	46985

Şekil 4.2 : Elasticsearch'te oluşturulan indeksler.



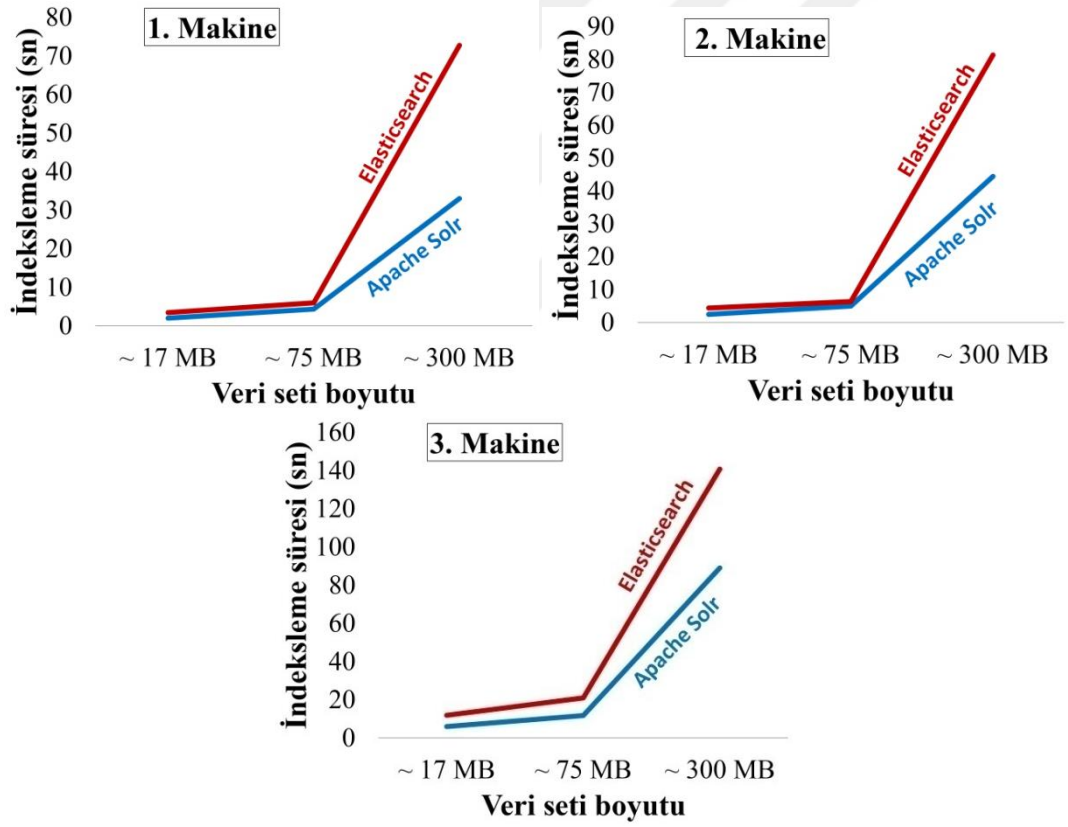
Şekil 4.3 : Apache Solr'da oluşturulan çekirdekler.

Çizelge 4.1, farklı boyutlardaki veri kümelerinin, varsayılan yığın boyutunda Apache Solr ve Elasticsearch sunucularındaki dizin oluşturma sürelerini göstermektedir. Apache Solr'ın yığın boyutunun varsayılan değeri 512 MB'tır, ancak Elasticsearch'ün varsayılan yığın boyutu, çalışmalarda kullanılan makinenin RAM kapasitesinin yarısı kadardır, başka bir deyişle dinamik bir yapıya sahiptir. Süreler hesaplanırken, her indeksleme sorgusu beş kez çalıştırılarak elde edilen verilerin ortalaması çizelgeye eklenmiştir. Her sorgu isteklerinden önce, Apache Solr ve Elasticsearch araçlarındaki indeksler ve veriler silinerek parametreler varsayılana sıfırlanmıştır.

Çizelge 4.1 : Varsayılan yığın boyutu için dizin oluşturma süreleri (sn).

Veri seti boyutu	M1		M2		M3	
	Apache Solr	Elasticsearch	Apache Solr	Elasticsearch	Apache Solr	Elasticsearch
~ 17 MB	2.004	3.445	2.46	4.386	5.968	11.852
~ 75 MB	4.36	6	5.818	6.382	11.758	20.922
~ 300 MB	33.012	72.752	44.426	81.37	89.028	140.788

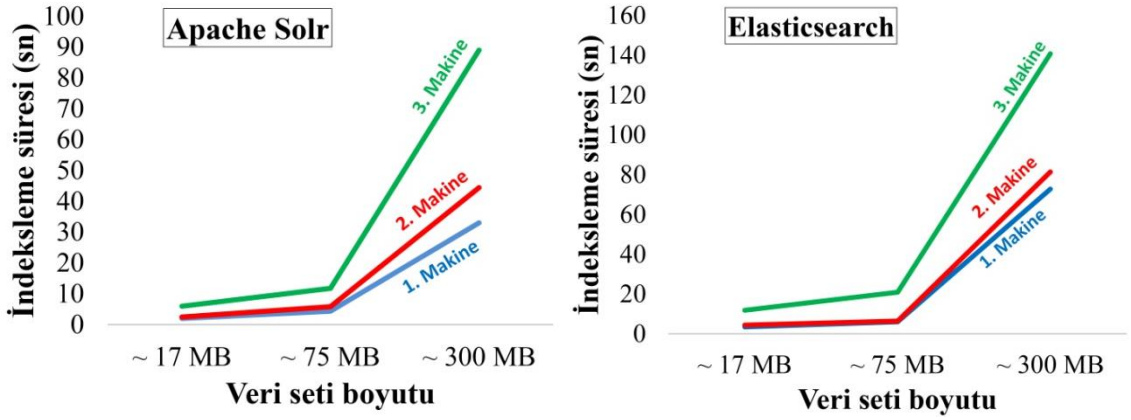
Çizelge 4.1'de listelenen, ortalama indeksleme sürelerinin veri boyutuna bağlı değişimi Şekil 4.4'te resmedilmiştir. Apache Solr, üç veri setinde de indeksleme hızı açısından Elasticsearch'ten daha iyi performans göstermiştir. Şekil 4.4'te görüldüğü gibi, veri kümesinin boyutu arttıkça indeksleme süresi de artmaktadır.



Şekil 4.4 : Farklı makinelere göre indeksleme zamanlarının (sn) karşılaştırılması.

Şekil 4.5, üç farklı makine için alınan indeksleme süreleri ile Apache Solr ve Elasticsearch teknolojilerinin yığın boyutu arasındaki ilişkiyi göstermektedir. M1, daha yüksek sistem kaynaklarına sahip olduğundan dizin oluşturmada diğer iki bilgisayardan daha hızlıdır (bkz. Çizelge 3.2). Bu nedenle, güçlü bilgisayarların kullanılması indeksleme için bir

avantajdır. Ancak, en iyi özelliğe sahip makineler mutlaka en iyi sonuçları verir iddiası her zaman için doğru değildir.



Şekil 4.5 : Apache Solr ve Elasticsearch teknolojilerinin farklı makinelerdeki indeksleme sürelerinin karşılaştırılması.

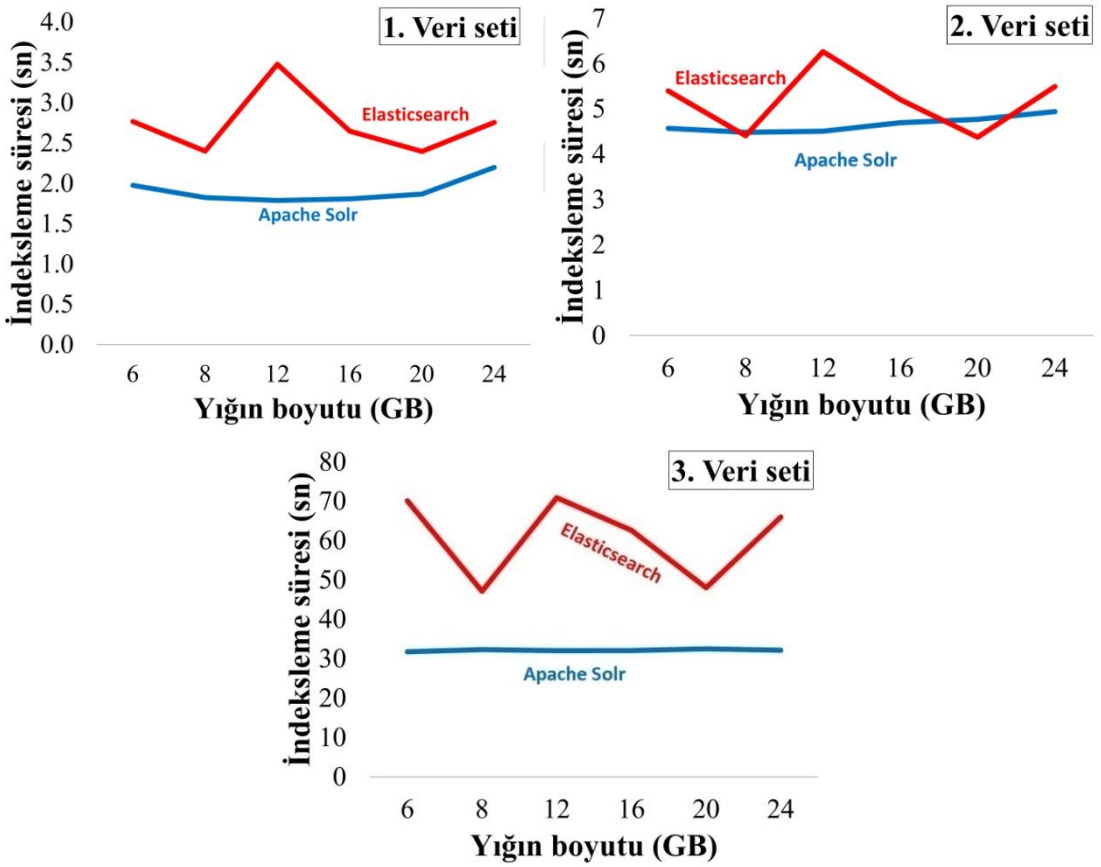
Çizelge 4.2'de, indekslemede en iyi performansı veren M1 kullanılarak test edilen altı farklı yığın boyutu için indeksleme sürelerini göstermektedir. Yığın boyutları rastgele seçilmiştir. Buradaki amaç, yığının boyutuna bağlı olarak Apache Solr ve Elasticsearch araçlarının indeksleme performanslarını karşılaştırmaktır. Bu açıdan, her iki teknoloji için belirlenen yığın boyutları yapılandırıldıktan sonra, üç veri seti için indeksleme süreleri yeniden ölçülmüştür.

Çizelge 4.2 : Yığın boyutuna göre indeksleme sürelerinin (sn) karşılaştırılması.

	Apache Solr			Elasticsearch		
Yığın boyutu	**V1**	**V2**	**V3**	**V1**	**V2**	**V3**
6 GB	1.976	4.574	31.788	2.768	5.398	70.106
8 GB	1.824	4.486	32.32	2.4	4.41	47.148
12 GB	1.788	4.51	32.072	3.478	6.266	70.908
16 GB	1.808	4.698	32.104	2.648	5.208	62.638
20 GB	1.868	4.772	32.528	2.396	4.378	48.038
24 GB	2.198	4.942	32.176	2.756	5.494	65.986

Şekil 4.6'da, iki teknolojinin indeksleme performanslarının farklı yığın boyutularına göre değişimi gösterilmektedir. Apache Solr'ın performansına bakıldığında, Elasticsearch kadar makine özelliklerine bağlı olmadığı görülmektedir. Sonuçlar incelendiğinde, yığın boyutunu artırmak, Apache Solr'ın dizin oluşturma performansını doğrusal bir şekilde

etkilemektedir. Bunun aksine, Elasticsearch teknolojisi için yığın boyutundaki artış indeksleme performansını da arttırmaktadır ama Elasticsearch'ün indeksleme performansı, doğrusal olarak yükseltilmiş yığın boyutuna göre doğrusal bir değişim göstermemektedir. Elasticsearch, 8 GB ve 20 GB yığın kapasitesi için daha iyi sonuçlar vermiştir. Bu durumda, çok büyük yığın boyutlarının her zaman en iyi performansı sağlamayacağı da görülmektedir. Buradan, yığın boyutu için ideal veya sabit bir değerin olmadığı sonucuna da varılabilmektedir. Bu yüzden, geliştiricilerin çalışma ortamlarına ve projelerinin amacına uygun yığın boyutunu denemeleri ve bulmaları gerekmektedir. Bu alanda çalışan uzmanlara göre yığın boyutu toplam belleğin %50'sini geçmemelidir. Ayrıca, Elasticsearch geliştiricileri yığın boyutunun varsayılan değerde tutulmasının performans için daha iyi olacağını belirtmektedir (Elastic Installation and Upgrade Guide [8.4], 2022). Ek olarak, indeksleme sürelerinin ölçümleri sırasında, Elasticsearch komut ekranına dosyadaki verileri yazdırmaktadır, bu durumun indeksleme süresine bir miktar olumsuz etkisi de bulunmaktadır.

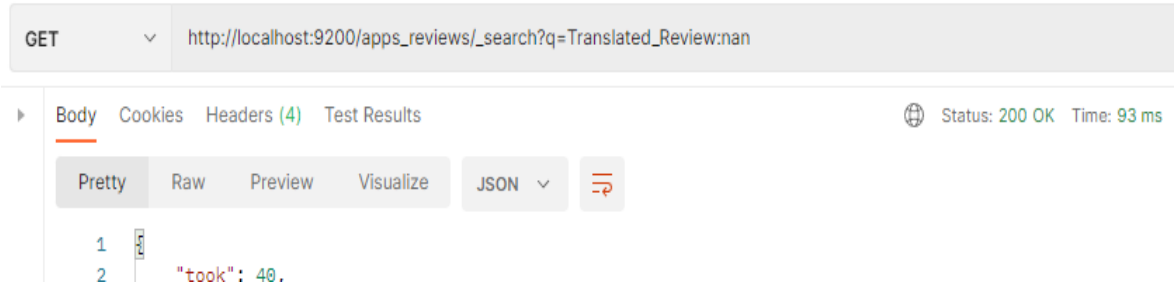


Şekil 4.6 : Makinelerin farklı yığın boyutlarına göre indeksleme sürelerindeki değişim.

Yapılan ölçümlerin sonuçlarına göre Apache Solr, indeksleme açısından Elasticsearch'ten daha iyi performans göstermektedir. Ayrıca indeksleme söz konusu olduğunda, Elasticsearch için donanım kaynaklarının daha önemli olduğu söylenebilir.

4.2 Arama

Bu çalışmada, arama sürelerini test etmek için uygulama programlama arayüzü oluşturmayı ve test etmeyi sağlayan API platformu Postman kullanılmıştır. Sorgular (bkz. Çizelge 3.5) bir istek olarak gönderilip yanıtlar Postman uygulaması üzerinden alınmaktadır. Her sorgu on kez çalıştırılıp tablolara elde edilen sürelerin ortalama değerleri eklenmiştir. Şekil 4.7, Postman uygulamasının arayüzünü ve sunucuya gönderilen bir arama isteğini göstermektedir. URL istekleri GET yöntemi kullanılarak gerçekleştirilmiştir. Arama süreleri, ekran görüntüsünün sağ tarafında bulunan süre (Time) paramatresi kullanılarak toplanmıştır.



Şekil 4.7 : Postman uygulamasının arayüzü.

Çizelge 4.3, on defa çalıştırılan sorguların ortalama arama sürelerini ve kullanıcıya cevap olarak döndürülen kayıt sayılarını göstermektedir. Bir sorgu için sözde kod aşağıdaki gibidir:

1. Sorgu dizesi oluştur (kullanıcı isteklerine göre)
2. Postman aracılığıyla GET komutunu kullanarak sorgu talebi gönder
3. Arama süresini (Time) al
4. Sonuçları kaydet

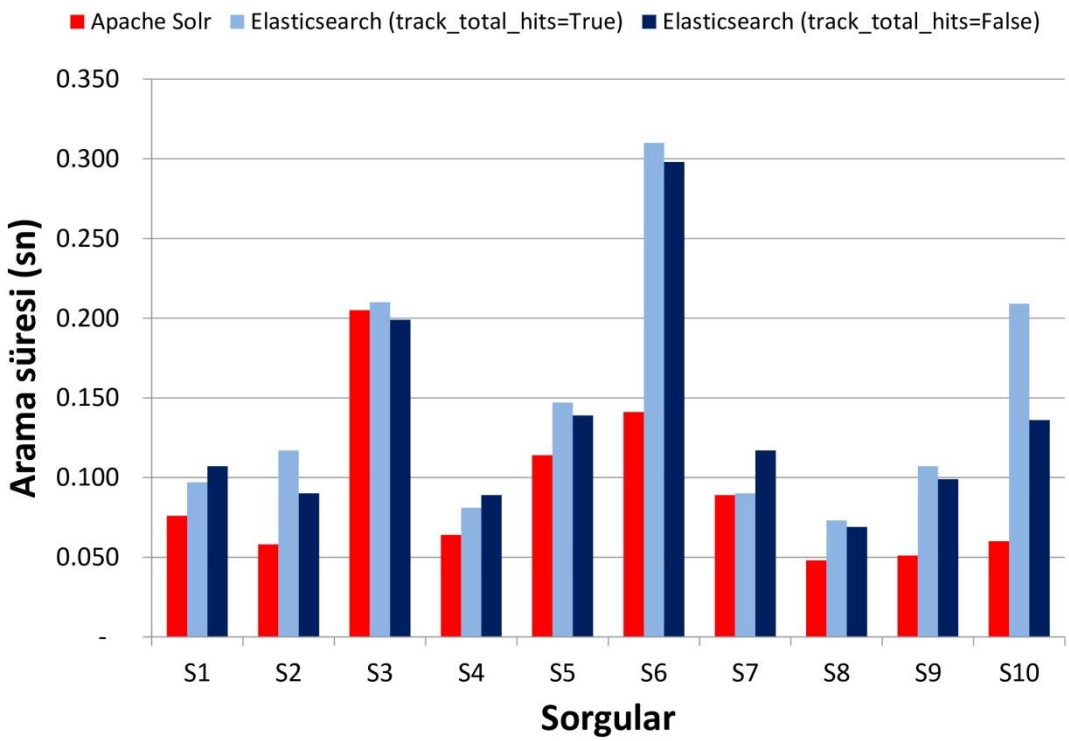
Elasticsearch'te **track_total_hits** parametresinin varsayılan değerinin değiştirilmesi, arama süresinde farklılıkları ortaya çıkarmaktadır. Çizelge 4.3'te bulunan alan türü sütunu, sorguda kullanılan alanların (dergi ismi, web site gibi) veri türünü belirtmektedir. Arama sırasında alınan en yüksek veri sayısı S10 için 55,263 değeridir. Bu nedenle, toplam isabet sayısı, tüm sorguları kapsayacak şekilde 60,000 olarak ayarlanmıştır. Bu çalışmada, döndürülen doküman sayısı için 60,000 eşiği belirlense de bu miktar, başka çalışmalarda

ya da projelerde kullanılan veri kümesine ve beklenen sonuçlara göre değişiklik gösterecektir.

Çizelge 4.3 : Arama sürelerinin karşılaştırılması (sn).

Sorgu	Apache Solr	**Elasticsearch /** track_total_hits		Alan türü	Döndürülen doküman sayısı
		= True	= False		
S1	0.076	0.097	0.107	Metin	26,863
S2	0.058	0.117	0.09	Sayısal	26,866
S3	0.205	0.21	0.199	Metin	6
S4	0.064	0.081	0.089	Sayısal	6
S5	0.114	0.147	0.139	Metin	294
S6	0.141	0.31	0.298	Metin	3,955
S7	0.089	0.09	0.117	Metin	14,926
S8	0.048	0.073	0.069	Sayısal	237
S9	0.051	0.107	0.099	Sayısal	26,089
S10	0.06	0.209	0.136	Sayısal	55,263

Çizelge 4.3'te, her iki teknoloji için toplam isabetlerin belirlendiğini ancak arama sürelerine yalnızca 10 değerin ekrana yazdırılma süresinin dahil edildiği sonuçlar eklenmiştir. Sonuçlar incelendiğinde, S3 ve S7 sorguları her iki teknoloji için yaklaşık olarak aynı arama süresine sahipken, S6 ve S10 sorguları arasında anlamlı bir fark vardır. Genel olarak bu çalışmada kullanılan veri kümelerine, yapılan sorgulara ve alınan sürelere bakıldığında, Apache Solr karmaşık arama sorgularında Elasticsearch'ten daha iyi performans göstermektedir. Çizelge 4.3 detaylı incelendiğinde, S1 ve S2 için döndürülen sonuçların sayısı S3 için döndürülen sonuçların sayısından daha fazladır. Buna rağmen, S1 ve S2 sorgularının arama hızı S3'ten daha iyidir. S3'ün daha karmaşık bir sorgu olması (bkz. Çizelge 3.6) her iki teknoloji için arama süresinin artmasına neden olmuştur. Ek olarak, alınan veri boyutu ne kadar büyük olursa arama süresi de doğru orantılı olarak artmıştır. Başka bir durum, S3 ve S4 için alınan doküman sayısı 6'dır. Her iki teknoloji için de S4, S3'ten daha iyi bir arama süresine sahiptir. Çizelge 3.6'da verilen bilgiler doğrultusunda her iki sorguda aynı veri seti üzerinde yapılmaktadır, ancak S3'teki sorguda arama için kullanılan parametrelerin içeriği metin verisi iken, S4'teki sorguda kullanılan alanlar sayısal verilerden oluşmaktadır. Dolayısıyla, aramanın gerçekleştirildiği alanın veri tipi ve miktarı da arama süresini önemli ölçüde etkilemektedir. Şekil 4.8, Çizelge 4.3'te sunulan verilerin grafiksel gösterimini sağlamaktadır.



Şekil 4.8 : Arama sürelerinin (sn) karşılaştırılması.

Bu çalışmada Elasticsearch için track_total_hits parametresinin arama süresine olan etkisi de araştırılmıştır. Şekil 4.8 incelendiğinde, track_total_hits parametresinin false olması durumunda S1, S4 ve S7 sorguları hariç genel olarak arama sürelerinde bir azalma meydana gelmiştir. S1, S4 ve S7 sorguları için daha ayrıntılı bir analiz yapılması daha iyi sonuçlar alınmasını sağlayabilir. Ek olarak, bu çalışmada her bir sorgunun ortalama süresini almak için yapılan test sayısını artırmak, bu üç sorgunun arama süresini iyileştirebilir. Sonuç olarak, Elasticsearch'ün track_total_hits parametresinin true veya false olması da arama hızını önemli ölçüde etkilemektedir. Özetlemek gerekirse, Apache Solr genel olarak arama süreleri açısından Elasticsearch'ten daha iyi performans göstermiştir.

5. UYGULAMA

Bu bölümde, tez kapsamında geliştirilen web uygulaması açıklanmaktadır. Uygulamanın amacı, araştırmacılar tarafından yazılan makalelere uygun dergiyi bulmak için bir dergi arama motoru geliştirmektir. Bu bağlamda, bölüm 3.2.2'de anlatılan veri setlerinden farklı olarak bir veri kümesi oluşturulmuştur. Uygulamada kullanılan veri seti (Deniz ve Aydin, 2022), Web of Science platformunun Engineering, Computing & Technology bölümünde yer alan 1,655 derginin çeşitli bilgilerini içermektedir. Veri seti manuel olarak oluşturulmuştur. Şekil 5.1'de veri setinden örnek bir JSON dosyası gösterilmektedir. Veri setinin 9 niteliği bulunmaktadır: Journal Name (dergi ismi), Publisher (yayıncı), ISSN / eISSN (elektronik numaralar), Web of Science Core Collection (Web of Science'da bulunduğu koleksiyon), Additional Web of Science Indexes (Web of Science'da eklendiği dizinler), Journal Website, Publication Frequency (yayınlanma sıklığı), Aims and Scope (amaç ve kapsam), Indexing and Abstracting (eklendiği dizinler).


```
[
   {
       "Journal Name": "GEOMECHANICS AND ENGINEERING",
       "Publisher": "TECHNO-PRESS , PO BOX 33, YUSEONG, DAEJEON, SOUTH KOREA, 305-600",
       "ISSN / eISSN": "2005-307X / 2092-6219",
       "Web of Science Core Collection": "Science Citation Index Expanded",
       "Additional Web of Science Indexes": "Current Contents Engineering, Computing & Technology |
Essential Science Indicators",
       "Journal Website": "http://www.techno-press.org/?journal=gae&subpage=5",
       "Publication Frequency": "Monthly",
       "Aims and Scope": "The Geomechanics and Engineering aims at opening an easy access to the
valuable source of information and providing an excellent publication channel for the global
community of researchers in the geomechanics and its applications.\nTypical subjects covered by the
journal include:\n Analytical, computational, and experimental multiscale and interaction mechanics\n
Computational and Theoretical Geomechnics\n Foundations\n Tunneling\n Earth Structures\n Site
Characterization\n Soil-Structure Interactions",
       "Indexing and Abstracting": "SCOPUS,  SCIE Web of Science"
   }
]
```

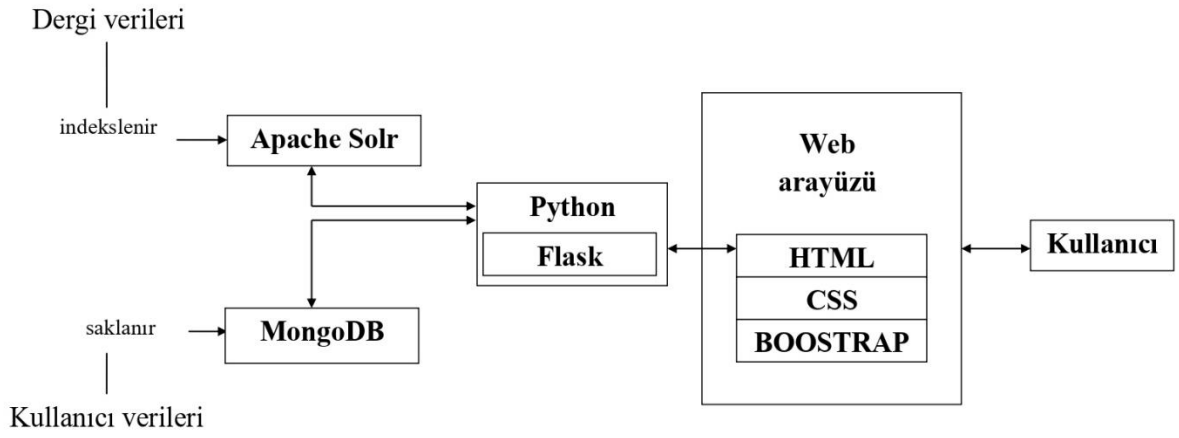

Şekil 5.1 : Veri seti (Deniz ve Aydin, 2022) için örnek bir JSON görüntüsü.

Uygulamada kullanılacak veri setinin boyutu (~ 4 MB) ve yapılan testlerin sonuçları temel alındığında, geliştirilen dergi arama motorunda Apache Solr teknolojisi kullanılmıştır. Bölümün devamında yazılım mimarisi ve web arayüzü detaylı bir şekilde anlatılmaktadır.

5.1 Yazılım Mimarisi

Uygulama iki modülden oluşmaktadır: arama modülü ve kullanıcı modülü. Arama modülündeki veriler Apache Solr sunucusunda indekslenerek depolanmaktadır. Kullanıcı modülündeki bilgiler ise tam metin aramayı destekleyen MongoDB veri tabanı kullanılarak

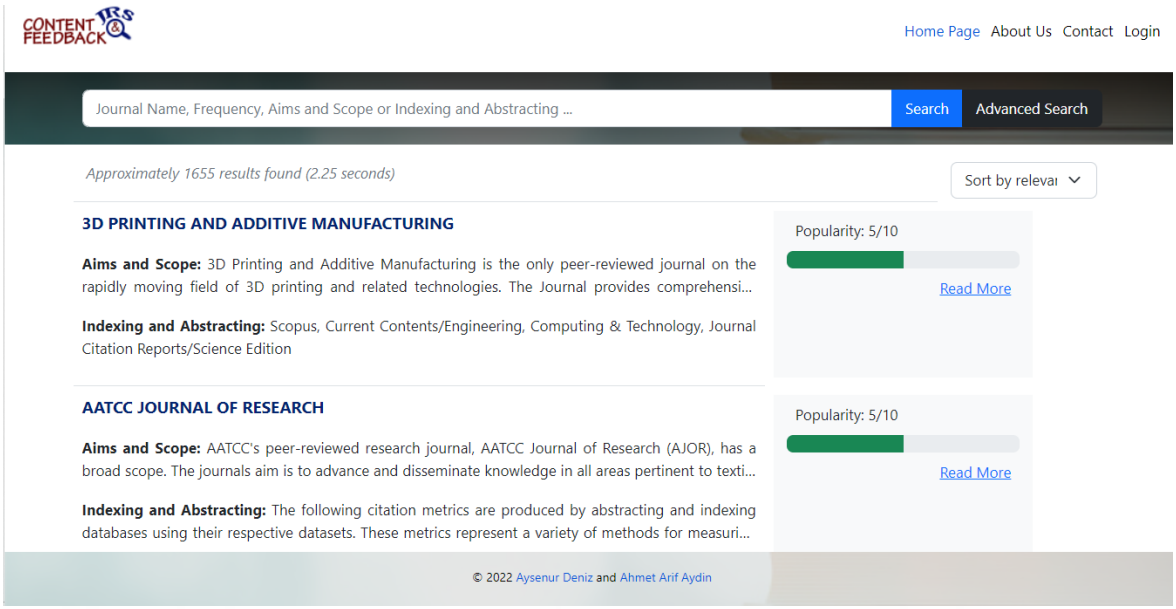
saklanmaktadır. Kullanıcı işlemleri metin ağırlıklı olmadığından MongoDB kullanılarak iş yükünün Apache Solr'dan alınması amaçlanmıştır. Şekil 5.2'de uygulamanın yazılım mimarisi resmedilmiştir. Programlama dili olarak Python, web arayüz ile depolanan veriler arasındaki işlemler için Flask frameworkü, arayüz tasarımı için de HTML, CSS ve Boostrap kullanılmıştır.



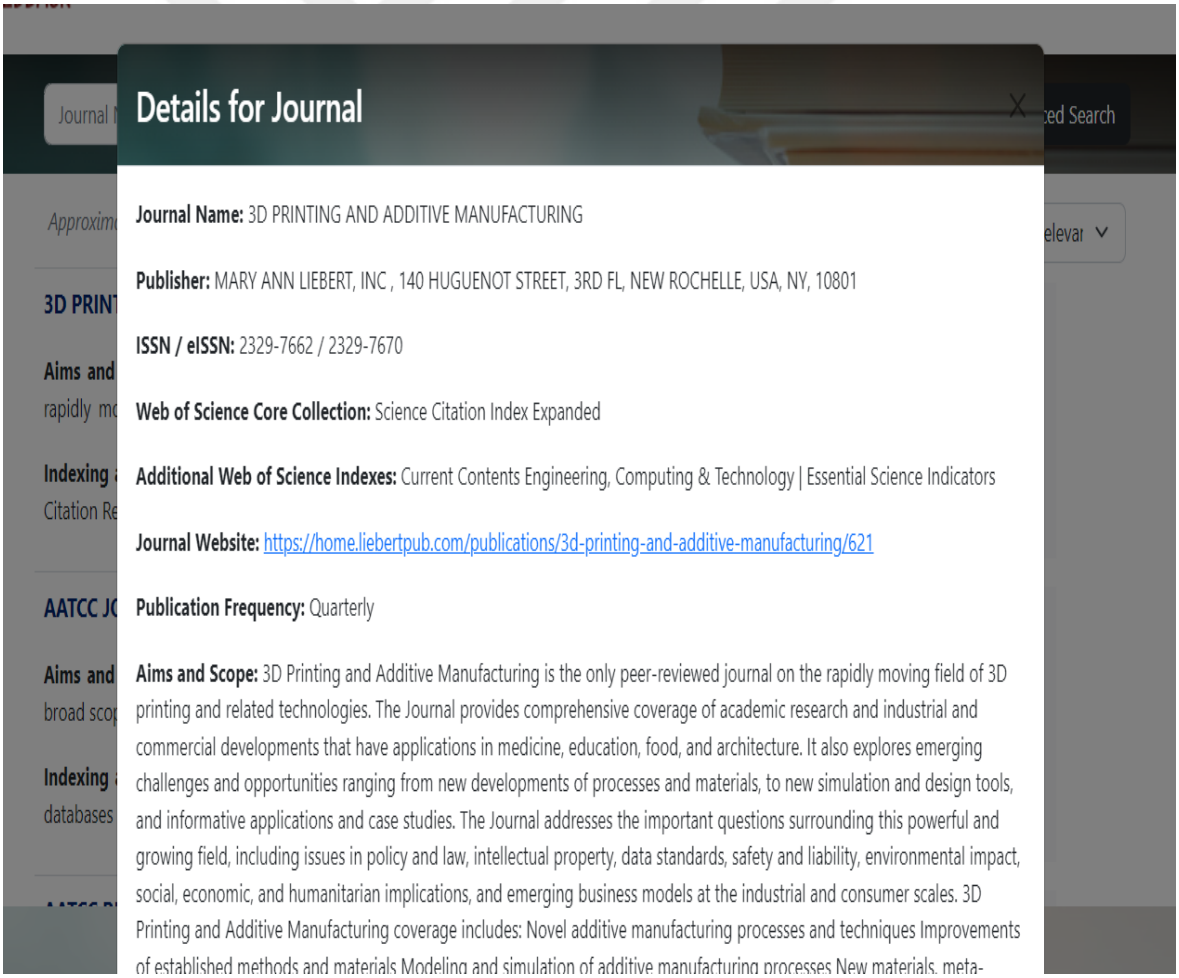
Şekil 5.2 : Web uygulamasının yazılım mimarisi.

5.2 Web Arayüzü

Şekil 5.3'te arayüzün ana sayfası bulunmaktadır. Sayfanın sağ üstünde bir menü (ana sayfa, hakkımızda, iletişim ve giriş) yer almaktadır. Gövde kısmında, öncelikle arama işlemi için bir form grubu bulunmaktadır. Kullanıcı bu formu kullanarak istediği keyword yani anahtar kelime için dergi arayabilmektedir. Arama işleminden sonra dergiler çeşitli bilgiler ile birlikte listelenmektedir. Dergilerin isimleri, amaçları ve kapsamı, indekslendiği dizinler ve kullanıcıların puanlarının ortalamasından oluşan popülerlik skoru da gövde kısmında yer almaktadır. Dergiler için detaylı bilgiye 'read more' linkine tıklayarak ulaşılabilmektedir. Şekil 5.4'te dergi ayrıntılarının bir örneği verilmektedir. Bu kısımda veri setinde bulunan bilgiler mevcuttur. Dergilerin web siteleri de bir link vasıtası ile kullanıcıya sunulmuştur. Ek olarak kullanıcılar, bu açılır pencere üzerinden dergi ile ilgili görüşlerini paylaşabilecekleri yorum form grubuna (bkz. Şekil 5.7) sahiptir.

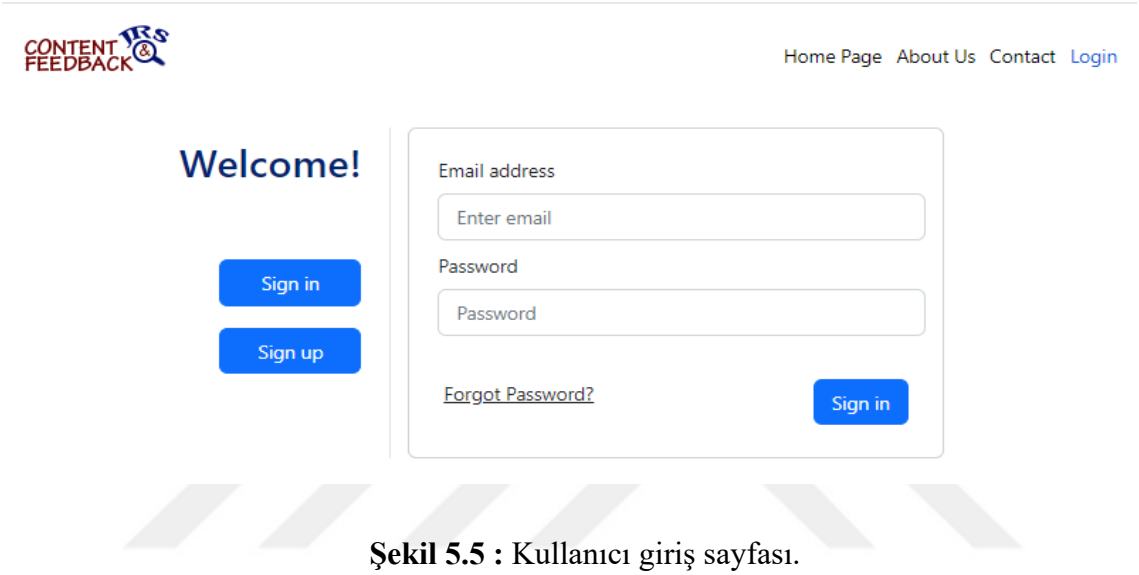


Şekil 5.3 : Web arayüzünde oluşturulan ana sayfa.

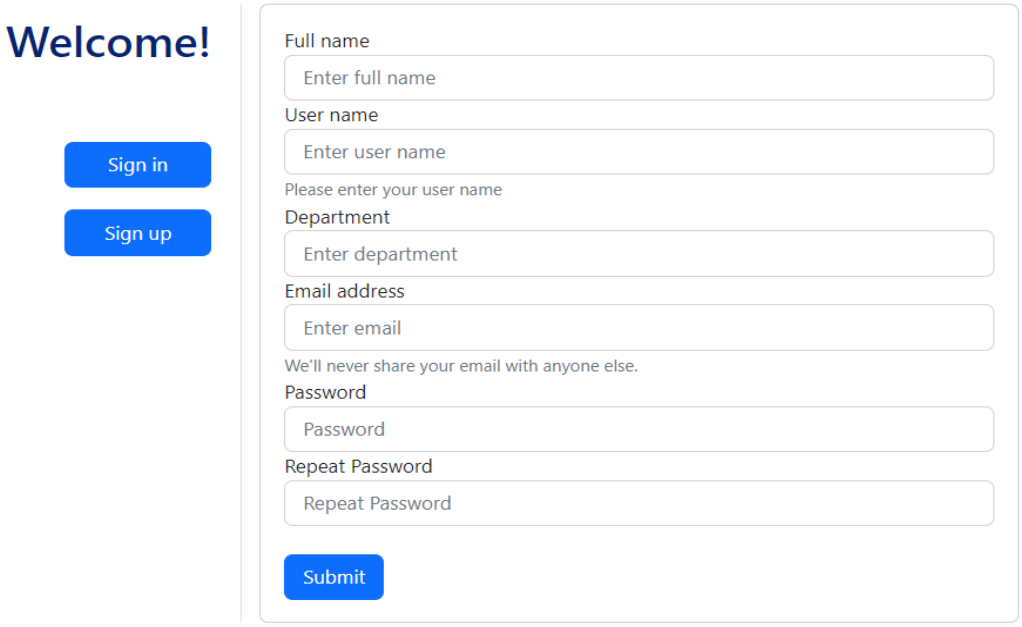


Şekil 5.4 : Dergi detaylarının sunulduğu açılır pencere.

Kullanıcılar, Şekil 5.5'te gösterilen giriş sayfasından kayıt (sign up) ya da giriş (sign in) işlemi yapabilmektedir. Kullanıcıların giriş yapabilmeleri için önceden kayıt işlemi gerçekleştirmiş olmaları gerekmektedir. Kayıt için Şekil 5.6'da gösterilen bilgiler istenmektedir. Kayıt işleminden sonra kullanıcılar giriş yapıp profillerini görüntüleyebilir, dergilere yorum bırakabilir. Giriş yapılmadan yorum yapılamamaktadır (bkz. Şekil 5.8). Yorumlar aracılığı ile kullanıcılar, oluşturulan dergi arama motoru üzerinden diğer araştırmacıların deneyimlerine de ulaşma imkanı bulmaktadırlar.

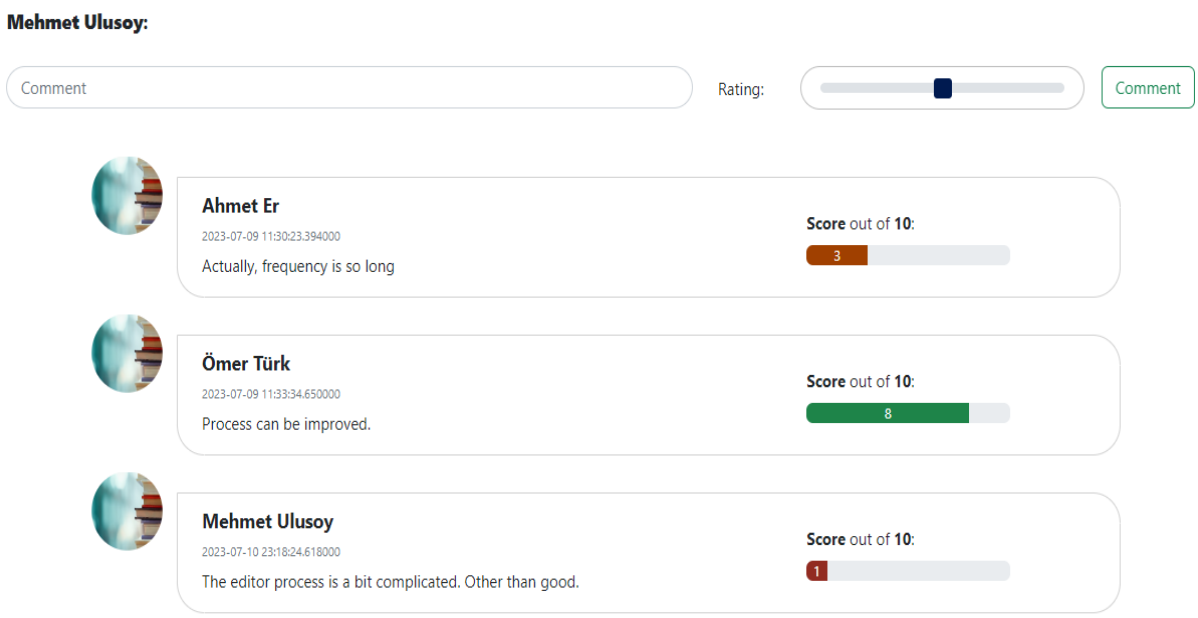


Şekil 5.5 : Kullanıcı giriş sayfası.

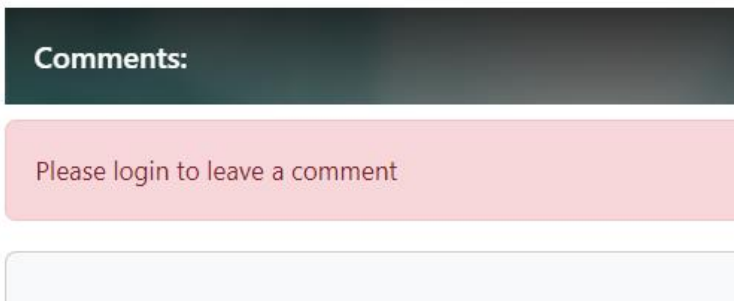


Şekil 5.6 : Kullanıcı kayıt sayfası.

Şekil 5.7'de görüldüğü gibi, bir metin girişi ve bir skor giriş çubuğu bulunan form grubu ile kullanıcılar yorum yapabilmektedir. Kullanıcı yorumunu kaydetmek için, her iki alanı da doldurması gerekmektedir. İlgili form doldurulduktan sonra 'comment' butonu ile kullanıcı yorum yapma işlemini tamamlamış olmaktadır.



Şekil 5.7 : Kullanıcı yorumlarının yapıldığı sayfa.



Şekil 5.8 : Yorum yapmak için kullanıcıya verilen giriş yap uyarısı.

6. SONUÇ VE ÖNERİLER

Bu araştırmada, Apache Solr ve Elasticsearch teknolojileri için çeşitli karşılaştırmalar yapılmıştır. Teknik özellikler açısından her iki teknoloji de önemlidir ve tam metin arama yöntemi açısından popüler arama motorları arasında yer almaktadır. Apache Solr ve Elasticsearch'ün indeksleme ve arama süreleri, Bölüm 4'te açıklanan sorgular kullanılarak toplanmış ve karşılaştırılmıştır. Sonuçlara göre indekslemede Apache Solr, Elasticsearch'ten daha hızlıdır. Ayrıca Elasticsearch için kullanılan sistemin donanımsal özellikleri önemlidir ve bu durum Apache Solr'a göre dezavantaj olarak değerlendirilebilir. Arama sürelerine bakıldığında indekslemede olduğu gibi Apache Solr, Elasticsearch'ten daha iyi bir performans göstermektedir. Sürelerdeki farklılıklara rağmen, her iki teknolojinin de Apache Lucene kütüphanesi üzerine oluşturulmuş olması, bazı güçlü özellikleri de paylaştıklarını göstermektedir.

Apache Solr, Apache Software Foundation tarafından alanında uzman kişilerden oluşan bir ekip ile geliştirilmiş bir projedir. Bu geliştirme sürecinde Apache Solr'a faceting arama, özellik filtreleme, gerçek zamanlı analitik gibi özellikler kazandırılmıştır. Apache Solr, kullanıcı geri bildirimlerine dayalı proje geliştirme konusunda da kendisinden altı yıl sonra yayınlanan Elasticsearch'ten daha avantajlıdır. Çünkü aradaki altı yıl projeyi daha olgun bir aşamaya taşıma fırsatı sağlamıştır. Günümüzde her iki teknolojinin de belli bir seviyeye geldiği görülmektedir. Bu çalışma, Apache Solr'un kullanım ve arama işlevselliği açısından daha kullanıcı odaklı olduğunu göstermektedir. Çünkü kullanıcıya daha sade ve anlaşılır bir yapı sunulmaktadır. Dahası, bu tez çalışmasında yapılan testlerde, Apache Solr hem indeksleme hem de arama konusunda daha iyi performans göstermiştir. Donanım özellikleri Elasticsearch için önemli olsa da Apache Solr sabit donanım özellikleriyle daha iyi performans göstermiştir. Bu duruma göre, Apache Solr'ın daha kararlı bir yapıya sahip olduğu söylenebilir.

Çizelge 6.1, bu tez çalışması ile benzer olan yayınların karşılaştırma parametreleri ve görüşleri yer almaktadır. Yurtsever ve diğ. (2022) ve D.S. (2016), bu çalışmada olduğu gibi indeksleme performansı açısından Solr'ı daha iyi bulurken, Elasticsearch vs. Solr Performance: Round 2 (2015), Hansen ve diğ. (2018) ile Gonçalves ve Sunye (2020), Elasticsearch'ün indeksleme performansı açısından daha iyi olduğunu belirtmektedir. Solr Performance: Round 2 (2015), arama süreleri göz önüne alındığında, Apache Solr'ın

saniyedeki sorgu sayısı açısından daha iyi olduğunu belirtmektedir. Öte yandan, Hansen ve diğ. (2018) ile D.S. (2016), Elasticsearch'ün aramada iyi olduğunu savunmaktadır.

Literatürde bulunan benzer çalışmalar ile daha doğru bir karşılaştırma yapmak için, mevcut yayınlarda kullanılan veri setlerinin bu tez çalışmasında kullanılması amaçlanmıştır. Ancak bazı veri setlerine ulaşılamamış, bazıları ise bu çalışma için uygun olmamıştır. Bu nedenle sayısal değerler birebir karşılaştırılamamıştır. Ancak gelecekte yapılacak çalışmalarda yazarlar, bu yayında kullanılan veri setlerini kullanabilirler.

Çizelge 6.1 : Önceki çalışmaların karşılaştırılması.

Makale	Karşılaştırma Kriterleri	Karşılaştırılan Teknolojiler	Görüşler
Yurtsever ve diğ. (2022)	Doküman ekleme süreleri	Apache Solr, Elasticsearch	Apache Solr daha hızlıdır.
Gonçalves ve Sunye (2020)	İndeksleme süresi, RAM kullanımı, indeksin kapladığı disk boyutu	Apache Solr, Elasticsearch	Genel olarak Elasticsearh, Apache Solr'dan daha iyi performans göstermektedir.
Hansen ve diğ. (2018)	İndeksleme ve arama performansları, sanal makine kullanım durumu	Apache Solr, Elasticsearch	Elasticsearch, indeks boyutu ve indeksleme süresi için daha iyidir. Arama için, ilk çalıştırmada Elasticsearch iyidir, ikinci çalıştırmada Solr daha iyidir. Elasticsearch daha fazla sanal bellek kullanır.
D.S. (2016)	İndeksleme ve arama hızları	Apache Solr, Elasticsearch, Sphinx, Xapian	Elasticsearch aramada iyi iken Apache Solr indekslemede iyidir.
Elasticsearch vs. Solr Performance : Round 2 (2015)	İndeksleme ile arama hızı, indeksleme yükü ile arama süresi, saniye başı sorgu süreleri (QPS)	Apache Solr, Elasticsearch	İndeksleme süresi ile ilgili olarak, Elasticsearch küçük verilerde iyidir, Solr ise büyük verilerde daha iyidir. Elasticsearch, indeksleme yüküyle test etmek için iyidir. Solr, QPS testinde iyidir.

Sonuç olarak, bu çalışmada büyük miktarda veriyi işlemek ve analiz etmek için kullanılan tam metin arama yöntemleri hakkında kapsamlı bir rapor sunulmuştur. Başlangıçta, Apache Solr ve Elasticsearch teknolojilerinin özellikleri ve teknik karşılaştırması derinlemesine incelenmiştir. İkinci olarak, indeksleme süreleri ve üç ayrı veri setinde dikkatlice tasarlanmış sorgulara dayalı olarak elde edilen arama süreleri dikkate alınarak, her iki sistemin kapsamlı ve adil bir karşılaştırması yapılmıştır. Ayrıca, Apache Solr ve

Elasticsearch teknolojilerinin nasıl kullanılacağına dair benzerlikleri ve farklılıkları sunmak için bu çalışma ile alakalı ayrıntılı bir mevcut yayın araştırması yapılmıştır. Sonuçlarımıza göre, genel olarak Apache Solr, Elasticsearch'ten daha iyi performans göstermektedir. Düşük donanım kaynaklarına sahip bilgisayarlarla Apache Solr kullanmanının avantaj olduğu görülmektedir. Bu çalışma, araştırmacılara tam metin arama yöntemleri hakkında arka plan bilgisi sunmaktadır, aynı zamanda bulgular, erişilebilir donanım kaynakları, veri türü ve veri boyutu açısından tam metin arama faaliyetleri için en uygun alternatiflerin seçilmesine yardımcı olmaktadır.

KAYNAKLAR

Anderson, K. M., Aydin, A. A., Barrenechea, M., Cardenas, A., Hakeem, M., & Jambi, S. (2015). Design challenges/solutions for environments supporting the analysis of social media data in crisis informatics research. *2015 48th Hawaii International Conference on System Sciences*, *2015-March*, 163–172. https://doi.org/10.1109/HICSS.2015.29

***Apache Lucene*. (2022).** Retrieved November 11, 2022, from https://lucene.apache.org/

Barrenechea, M., Jambi, S., Aydin, A. A., Hakeem, M., & Anderson, K. M. (2017). Getting the query right for crisis informatics design issues for web-based analysis environments. *Journal of Web Engineering*, *16*(5), 399–432. https://journals.riverpublishers.com/index.php/JWE/article/view/3269/2153

Bellini, P., Bugli, F., Nesi, P., Pantaleo, G., Paolucci, M., & Zaza, I. (2019). Data flow management and visual analytic for big data smart city/IOT. *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019*, 1529–1536. https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00276

***DB-Engines*. (2022).** Retrieved June 19, 2023, from https://db-engines.com/en/

Deniz, A. & Aydin, A.A. (2022). Web of Science Dataset (Engineering, Computing & Technology Journals). *Mendeley Data,* V2. Retrieved June 19, 2023, from https://doi.org/10.17632/syzcbykjw3.2

Deniz, A., Elömer, M. M. & Aydın, A. A. (2023). A comparison of Apache Solr and Elasticsearch technologies in support of large-scale data analysis. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi, 13*(2), 386-404. https://doi.org/10.17714/gumusfenbil.1213317

Domo Company. (2022). *Data Never Sleeps 9.0*. Retrieved November 11, 2022, from https://www.domo.com/learn/infographic/data-never-sleeps-9

***Dota 2 Matches*. (2022).** Retrieved November 11, 2022, from https://www.kaggle.com/datasets/devinanzelmo/dota-2-matches

D.S., S. (2016). A quick search on the projects with a high loads and a large amount of data. *Modern Technologies: Current Issues, Achievements and Innovations — Collection of Articles III International Scientific Conference / under the General Editorship of G. Yu Gulyaev — Penza MCNS « Science and Education »*, 23–32.

***Elastic Installation and Upgrade Guide [8.4]*. (2022).** Retrieved November 11, 2022, from https://www.elastic.co/guide/en/elastic-stack/8.4/index.html

***Elasticsearch vs Solr*. (2023).** Retrieved June 19, 2023, from https://github.com/aysenurdeniz/elasticsearch-vs-solr

***Elasticsearch vs. Solr Performance: Round 2*. (2015).** Retrieved November 11, 2022, from https://www.flax.co.uk/blog/2015/12/02/elasticsearch-vs-solr-performance-round-2/

Gao, Rujia. (2012). Application of Full Text Search Engine Based on Lucene. *Advances in Internet of Things*. 02. 0-0. 10.4236/ait.2012.24013.

Gonçalves, A. A. S., & Sunye, M. S. (2020). Comparison of search servers for use with digital repositories. *ICEIS 2020 - Proceedings of the 22nd International Conference on Enterprise Information Systems*, *1*, 256–260. https://doi.org/10.5220/0009577102560260

***Google Play Store Apps*. (2022).** Retrieved November 11, 2022, from https://www.kaggle.com/datasets/lava18/google-play-store-apps

***Google Trends*. (2022).** Retrieved November 11, 2022, from https://trends.google.com/trends/

Halevi, G., & Moed, H. (2012). The evolution of big data as a research and scientific topic: overview of the literature. *Research Trends*, *30*(36), 3–6.

Hansen, J., Porter, K., Shalaginov, A., & Franke, K. (2018). Comparing open source search engine functionality, efficiency and effectiveness with respect to digital forensic search. *NISK 2018 - 11th Norwegian Information Security Conference,* 108-121.

Kılıç, U., & Karabey, I. (2016). Comparison of solr and elasticsearch among popular full-text search engines and their security analysis. *UBMK'16 - International Conference on Computer Science and Engineering, 2016 October*. https://doi.org/10.13140/RG.2.2.24563.32803

Kowsari, K., Brown, D., Heidarysafa, M., Meimandi, K. J., Gerber, M., & Barnes, L. (2018). Web of science dataset. *Mendeley Data,* V6. Retrieved November 11, 2022, from https://doi.org/10.17632/9RW3VKCFY4.6

Lakhara, S., & Mishra, N. (2017). Desktop full-text searching based on Lucene: a review. *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 2434–2438. https://doi.org/10.1109/ICPCSI.2017.8392154

Lashkaripour, Z. (2020). The era of big data: a thorough inspection in the building blocks of future generation data management. *International Journal of Scientific and Technology Research*, *9*, 321–330.

Lokoč, J., Veselý, P., Mejzlík, F., Kovalčík, G., Souček, T., Rossetto, L., Schoeffmann, K., Bailer, W., Gurrin, C., Sauter, L., Song, J., Vrochidis, S., Wu, J., & Jónsson, B. Þór. (2021). Is the reign of interactive search eternal? findings from the video browser showdown 2020. *ACM Transactions on Multimedia Computing, Communications, and Applications*, *17*(3), 1–26. https://doi.org/10.1145/3445031

Luburić, N., & Ivanovic, D. (2016). Comparing Apache Solr and Elasticsearch search servers. *6th International Conference on Information Society and Technology – ICIST 2016*. http://www.eventiotic.com/eventiotic/files/Papers/URL/icist2016_54.pdf

Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, *30*(4), 431-448.

Oussous, A., & Benjelloun, F. (2022). A comparative study of different search and indexing tools for big data. *Jordanian Journal of Computers and Information Technology*, *8*(1), 1. https://doi.org/10.5455/jjcit.71-1637097759

Rao, T. R., Mitra, P., Bhatt, R., & Goswami, A. (2018). The big data system, components, tools, and technologies: a survey. *Knowledge and Information Systems*, *60*(3), 1165. https://doi.org/10.1007/s10115-018-1248-0

***Resources Apache Solr*. (2022).** Retrieved November 11, 2022, from https://solr.apache.org/resources.html.

Voit, A., Stankus, A., Magomedov, S., & Ivanova, I. (2017). Big data processing for full-text search and visualization with elasticsearch. *IJACSA - International Journal of Advanced Computer Science and Applications*, *8*(12). www.ijacsa.thesai.org

Wang, J.-F., Wang, X.-F., & Li, H. (2022). Design of multimedia distance teaching auxiliary system based on MOOC platform. *ICMTMA 2022 - 14th International Conference on Measuring Technology and Mechatronics Automation,* 1179–1186. https://doi.org/10.1109/ICMTMA54903.2022.00237

Y. Aldailamy, A., Abdul Hamid, N. A. W., & Abdulkarem, M. (2018). Distributed indexing: performance analysis of Solr, Terrier and Katta information retrievals. *Malaysian Journal of Computer Science*, 87–104. https://doi.org/10.22452/mjcs.sp2018no1.7

Yurtsever, M. M. E., Özcan, M., Taruz, Z., Eken, S., & Sayar, A. (2022). Figure search by text in large scale digital document collections. *Concurrency and Computation: Practice and Experience*, *34*(1). https://doi.org/10.1002/CPE.6529

ÖZ GEÇMİŞ

Ad Soyad **:** Ayşenur DENİZ

ÖĞRENİM DURUMU:

- **Lisans** **:** 2018 - 2021, İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü
- **Yüksek Lisans** **:** 2021 - Devam etmekte, İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN ÇALIŞMALAR

- **Deniz, A., Elömer, M. M, & Aydin, A.A. (2023).** A comparison of Apache Solr and Elasticsearch technologies in support of large-scale data analysis. Gümüşhane Üniversitesi Fen Bilimleri Dergisi, 2023.
- **Deniz, A. & Aydin, A.A. (2022).** Web of Science Dataset (Engineering, Computing & Technology Journals). Mendeley Data, V2.