Düzce Üniversitesi
Bilim ve Teknoloji Dergisi

Derleme Makalesi

Web Tarayıcılarında Tohum URL Seçimi ve Performans Analizi: Kapsamlı Bir İnceleme

Zülfü ALANOĞLU [a,*], M. Ali AKCAYOL [b]

[a] *Bilgisayar Teknolojileri Bölümü, Antakya MYO, Hatay Mustafa Kemal Üniversitesi, Hatay, TÜRKİYE*
[b] *Bilgisayar Müh. Bölümü, Mühendislik Fakültesi, Gazi Üniversitesi, Ankara, TÜRKİYE*
** Sorumlu yazarın e-posta adresi: zalanoglu@gmail.com*
DOI:10.29130/dubited.1097123

ÖZ

Web, İnternet üzerinde yayınlanan çeşitli türden bilgilerin bulunduğu bir veri deposudur. Bu bilgileri üzerinde bulunduran ve birbirlerine köprülerle bağlı olan yapılara web sayfaları denir. Web tarayıcıları, web sayfaları üzerindeki köprüleri kullanarak Web'i tarayan ve sayfaları indiren programlardır. Bir arama motorunun performansı da web tarayıcısının performansına bağlıdır. Web tarayıcılarının performans metrikleri, kapsamı ve tohum URL seçim yöntemleri performansı etkileyen en önemli faktörlerdir. Bu çalışmada, genel, odaklanmış, artırılmış, gizli, mobil ve dağıtılmış olmak üzere altı kategoride sınıflandırdığımız web tarayıcılarının performansları, kapsamları ve tohum URL kullanım yöntemleri hakkında kapsamlı bir inceleme ve analiz yapılmıştır. Ayrıca her bir tarayıcının çeşitli çalışmalarda yapılmış performans ölçütleri karşılaştırılmıştır.

Anahtar Kelimeler: *Web tarayıcıları, Web sayfaları, Kapsam genişletme, Tohum URL*

Seed URL Selection and Performance Analysis in Web Crawlers: A Comprehensive Review

ABSTRACT

Web is a data repository where various types of information posted on the internet are found. Structures that contain this information and are connected to each other by hyperlinks are called web pages. Web crawlers are programs that browse the web and download pages using hyperlinks on web pages. The performance of a search engine also depends on the performance of the web crawler. Performance metrics, scope, and seed URL selection methods of the web browsers are the most important factors affecting performance. In this study, a comprehensive review and analysis of the performances, scopes and seed URL usage methods of the web crawlers, classified in six categories as general, focused, incremental, hidden, mobile and distributed, was carried out. In addition, the performance criteria of each crawlers in various studies were compared.

Keywords: *Web crawlers, Web pages, Scope expansion, Seed URL*

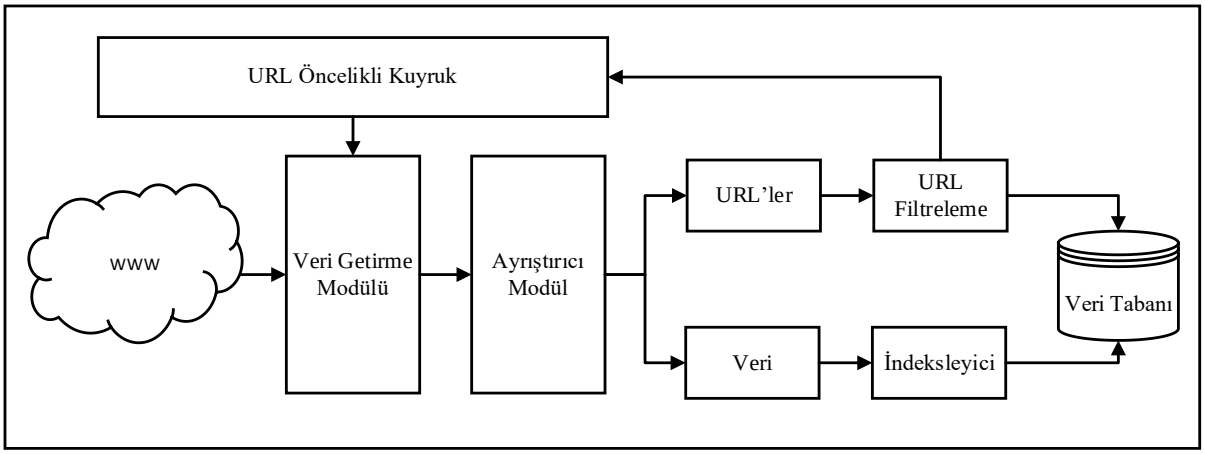
Geliş: 04/04/2022, Düzeltme: 24/05/2022, Kabul: 15/11/2022

I. GİRİŞ

Web sayfalarının sayısı ve boyutu her geçen gün artmaktadır. Web sayfalarını arama ve indeksleme işlemleri, arama motorları tarafından yapılmaktadır. Web 'in devasa büyüklükte bir veri kaynağı olması nedeniyle, arama motorları, tüm web sitelerini kapsayan bir tarama yapmakta zorlanmaktadır. Günümüzde tüm Web'i tarayan ve indeksleyen bir tarayıcı yoktur. Hedefi tüm Web olan tarayıcılar, genel tarayıcı olarak adlandırılır. Kullanıcıların isteklerine uygun verileri elde etmek için genel tarayıcılar yetersiz kalmıştır. Bu nedenle çeşitli tarayıcı türleri geliştirilmiştir.

Bir web tarayıcısının en önemli özelliği, kullanıcı isteklerine bağlı olarak, en geniş kapsamda ve mümkün olan en çok ve farklı sayfaya en kısa sürede ulaşmaktır. Fakat Web ortamında tarama yapılırken, bağlantılar takip edildiğinden, hiçbir web sayfası ile bağlantısı olmayan, farklı (javascript, flash vb) yapılar ile tasarlanmış vb. nedenler ile ulaşılamayan web sayfalarının bulunup taranması çok zordur. Web ortamındaki web sitelerinin sayısı üstel bir şekilde arttığından dolayı etkili ve verimli bir tarama için tarama stratejisinin önemi büyüktür.

Geleneksel bir tarayıcı, başlangıçta tohum tekdüzen kaynak buluculardan (Uniform Resource Locator -URL) başlar ve arama algoritmalarını kullanarak yeni URL'leri keşfeder. Bulunan URL'ler tekrarlanma ve standartlara uyma gibi özelliklerine göre filtrelenerek öncelikli kuyruğa eklenir. Sırası gelen URL taranarak öncelikli kuyrukta URL kalmayana kadar bu işlem devam eder. Öncelikli kuyruğa eklenen URL'ler farklı yöntemler ile sıralanır ve farklı tarama yöntemleri ile yeni URL'leri keşfedebilir. Bütün tarayıcılar tohum URL'leri kullanarak taramaya başlar. Bundan dolayı tohum URL'lerin kalitesi taramanın başarısını da belirler. Şekil 1'de geleneksel bir tarayıcının mimarisi gösterilmiştir.



Şekil 1. *Geleneksel Web Tarayıcı Mimarisi.*

Bu kapsamlı derleme çalışmasının geri kalanı şu şekilde düzenlenmiştir. Bölüm 2'de çalışmanın amacı ve motivasyonu ile ilgili detaylı açıklamalar yapılmıştır. Bölüm 3'de web tarayıcılarında taramaya başlamak için kullanılan tohum URL seçim metotları ile ilgili çalışmalar incelenmiştir. Bölüm 4'de web tarayıcıların kapsamı ile ilgili yapılan çalışmalar incelenmiş ve tarayıcı türüne göre kapsam genişletmeleri analiz edilmiştir. Bölüm 5'de tüm web tarayıcı çeşitleri ayrı ayrı ele alınmış, yapılan çalışmalar detaylı bir şekilde incelenmiş ve karşılaştırmalı analizleri yapılmıştır. Buna ek olarak, genel web tarayıcıları bölümünde, açık kaynak web tarayıcıları, geliştirildikleri programlama dilleri, uyumlu işletim sistemleri, kapsamları bakımından tarayıcı türleri ve lisans ölçütlerine göre karşılaştırılmıştır. Odaklanmış web tarayıcıları, hasat oranları, kesinlik/hassasiyet, ilişkili sayfa oranı ve toplam taranan sayfa sayıları ölçütlerine göre karşılaştırılmıştır. Artımlı web tarayıcıları, başarı oranları, tekrar ziyaret sıklığı, tarayıcı türleri ve sayfa sayısı/belge seti ölçütlerine göre karşılaştırılmıştır. Gizli web tarayıcıları, kapsama oranı, kullanılan sorgu sayısı ve kapsam ölçütlerine göre karşılaştırılmıştır. Mobil web tarayıcıları, başarı oranları, mobil ajan sayısı, taranan URL sayısı ve ağ trafiğinde azalma yüzdesi

ölçütlerine göre karşılaştırılmıştır. Son olarak dağıtılmış web tarayıcıları, tarama zamanı, taranan sayfa sayısı/boyutu, odaklanmış tarayıcının türü ve dağıtım sayısına göre karşılaştırılmıştır. Son olarak bölüm 6' da yapılan kapsamlı literatür çalışmaları ile ilgili sonuçlar verilmiş ve gelecekteki çalışmalar ile ilgili öneriler sunulmuştur.

II. ÇALIŞMANIN AMACI ve MOTİVASYONU

Web tarayıcıları Web' de bulunan verileri belirli bir amaç doğrultusunda alan ve indeksleyen arama motorlarının en önemli parçasıdır. Çalışmamızda web tarayıcılarının çalışmaya başlaması için gerekli olan tohum URL setleri, kapsamları ve tarama stratejileri detaylı bir şekilde incelenmiş ve çeşitli eksiklikler tespit edilmiştir. Web taramada kullanılan çeşitli yöntemler ve teknolojiler incelenmiş ve önceki çalışmalarda yapılan deneylerin karşılaştırmalı analizleri yapılmıştır. Ayrıca her bir tarayıcı türüne göre kullanılan performans ölçütleri ve başarı oranları karşılaştırılmıştır.

Web tarayıcıları ile ilgili güncel çalışmaları da kapsayan detaylı çalışmaların az olması motive edici bir faktördür. Web tarayıcılarının tohum URL seçimleri, çeşitleri ve performans ölçütleri ile ilgili tüm veri tabanı incelenmiş, analiz edilmiş ve gelecekteki çalışmalar için araştırma alanları bildirilmiştir.

III. TOHUM URL SEÇİMİ

Tohum URL seçimi, taramanın kapsamını ve koleksiyonun içeriğini belirleyen en temel adımdır. Bu nedenle, tohum URL'ler, geniş kapsam sağlamak için dikkatli bir şekilde seçilmelidir. Web 'in önemli bir bölümünü keşfeden nispeten küçük bir tohum URL kümesi çıkarmak mümkündür. Rastgele bir küme seçmek, Web 'in önemli bir bölümünü keşfedilmemiş bırakabilir [1]. Tohum URL'ler manuel, yarı otomatik ve otomatik olmak üzere farklı metotlar ile seçilmektedir.

Kleinberg kapsamı hızlı bir şekilde genişletmek amacıyla kaliteli tohum URL'leri seçen, web sayfalarını merkez ve otorite olarak gruplayan HITS (Hyperlink-Induced Topic Search) algoritmasını önermiştir. Önerilen algoritma sadece web sayfaları arasındaki bağlantıları dikkate almıştır. Özellikle çok sayıda dış bağlantı sağlayan merkez web sayfalarının iyi bir tohum URL olduğu belirtilmiştir. Önerilen yöntemde yalnızca web sayfasının içerdiği köprüler dikkate alınmış ve köprülerin işaret ettiği sayfaların durumu dikkate alınmamıştır. Yapılan çalışmaya göre, tohum URL seçiminde merkez web sayfalarının seçimi kapsamın genişlemesini sağlamaktadır [2].

Zheng ve arkadaşları, tohum URL seçiminde ilk çalışmalardan biri olan grafik tabanlı bir yaklaşım önermişlerdir. Tohum URL seçimi için rastgele, en yüksek PageRank değeri ve en çok dış bağlantıya sahip k sayfaya dayalı tohum seçim stratejileri kullanmışlardır. Farklı tohum URL'lerin "iyi" ya da "kötü" olarak adlandırılan koleksiyonlar ile sonuçlanabileceğini belirterek, tohum URL seçiminin önemini vurgulamışlardır. Kullanılan tohum seçim stratejilerinin etkinliğini, 100 gerçek web sitesinden alınan veriler üzerindeki deneysel sonuçlar ile kanıtlamışlardır. Çalışmada web tarayıcısının mevcut tohum URL'leri nasıl kullanabileceğini belirtilmiş, ancak tohum URL'lerin nasıl tanımlanması gerektiği ile ilgili bilgi verilmemiştir [3].

Dmitriev, kaliteli tohum URL'leri belirlemek amacı ile tohum URL seçim sürecini ele almış ve sunucu bilgisayar tabanlı bir mekanizma önermiştir. Çalışmada bir sunucuya ait bir URL'i tohum olarak kullanma kararını verirken, sunucunun kalitesini, önemini ve potansiyel verimini dikkate almışlardır. Sunucu seçimi bölgeye göre yapılmış ve sunuculara bir puan atanmıştır. Sunuculara atanan puanlar belirlenirken; sayfadaki diğer sayfaları işaret eden URL sayısı, sunucudaki spam olasılığı ve geçmiş tarama istatistiklerini kullanarak hesaplanan, ana bilgisayarın verimliliği gibi özellikleri dikkate almışlardır. Sonuç olarak tohum seçim sürecini iyileştirip daha verimli bir tarama gerçekleştirmişlerdir [4].

Daneshpajouh ve arkadaşları, etkili bir tohum URL seti oluşturmak amacı ile yeni ve hızlı bir algoritma sunmuşlardır. Çalışmada, HITS algoritmasına dayalı tohum URL kümesi oluşturmak için

çalışma süresi O(n) olan yeni bir algoritma sunulmuştur. Düşük çalışma süresi sunulan algoritmayı benzersiz kılmıştır. Çalışmada, PageRank, Trawling, HITS ve ağ akış tabanlı keşif algoritmaları üzerinde çalışılmıştır. Yaptıkları deneylerde, web tarayıcısı önerilen yöntem ile tanımlanan tohum URL'leri taramaya başlarsa, rastgele tohum kümesi başlatmaktan daha az yinelemede ve farklı topluluklardan daha yüksek PageRank değerine sahip daha fazla sayfa tarandığını göstermişlerdir [5].

Sharma ve Bhagat, en etkili tohum URL seçim stratejisini belirlemek amacı ile önemli ve faydalı sayfaları dikkate alan, URL' in köprü yapısı ile ilgilenen, minimum bir 'k' URL seti oluşturan ve sonrasında yönlendirilmiş grafik için kapsanan grafiği keşfeden BUDG (Base URL's Set for Directed Graph) isimli bir çerçeve önermişlerdir. Çalışmada Tohum URL seçiminde (1) rastgele seçim, (2) bir tepeye gelen 0 ayrıt sayısı (indegree), (3) bir tepeden çıkan en yüksek aygıt sayısı (outdegree) ve (4) en yüksek PageRank değerine göre 4 farklı tohum URL setinin karşılaştırmalı deneyleri yapılmıştır. Deneysel sonuçlara göre çerçeve farklı alanlar için düzgün çalışmakla birlikte, genel olarak sonuçlar net bir kazanan göstermemektedir. Sonuçlar arasında tüm stratejiler arasında en yüksek dış bağlantıya dayalı yaklaşımın en iyi olduğu belirtilmiştir [6].

Ayrıca tohum URL seçiminde en çok kullanılan yöntemler; manuel seçim [7-9], DMOZ ve curlie.org [10, 11] gibi açık kaynak dizinlerinden yapılan seçim ve Twitter [12, 13] gibi sosyal medyadaki kullanıcıların paylaştıkları URL'ler üzerinden seçimlerdir. Bunlara ek özellikle odaklı tarayıcılarda Google ve Yahoo gibi arama motorları ile yapılan aramalarda, ortaya çıkan URL'leri, tohum URL olarak seçen çalışmalarda mevcuttur [14-17].

Yapılan çalışmalar incelendiğinde tohum URL seçiminin geliştirilen tarayıcının türüne bağlı olduğu görülmüştür. Ontoloji tabanlı odaklanmış tarayıcılar kullanıldığında, tohum URL seçimleri alan uzmanları tarafından ya da açık kaynak dizinlerden konuyla ilgili olan URL'ler arasından seçilmektedir. Genel tarayıcılarda ise hedef tüm Web olduğundan ontolojiden bağımsız her konuda ve her bölgede tarama yapılmalıdır. Bunu için ise en iyi yöntemlerden biri bölgelerde en çok ziyaret edilen, en iyi içeriğe sahip, en çok dış bağlantısı olan vs. gibi özelliklere göre sıralanmış en iyi web sayfaların seçilmesinin yararlı olacağı söylenebilir.

IV. KAPSAM GENİŞLETME

Kapsam, web tarayıcılarının kullanıcı isteklerine uygun verilerin tarama oranını belirten bir performans ölçüsüdür. Arama motorlarının en zorlandıkları konu, istenilen bilgilerin tamamına nasıl ulaşılacağı ile ilgilidir. Web tarayıcılarının türüne bağlı olarak kapsam, sadece URL'ler takip edilerek, bağlantı metinleri analiz edilerek ya da sayfadaki veriler analiz edilerek genişletilebilir. Kapsam, tarayıcı çeşidine göre farklılık gösterir. Genel tarayıcılarda kapsam tüm Web iken, odaklı tarayıcılarda belli bir konu ya da belirli bir etki alanı olabilir. Bunun dışında artırılmış, gizli, mobil ve dağıtılmış web tarayıcıları, genel ve odaklanmış tarayıcıların özelliklerini taşırlar.

Lee ve arkadaşları, kapsamı genişletmek ve tekrardan korunmak amacıyla, tek sunucu kullanarak milyarlarca sayfayı indirebilen ve performansı modelleyen IRLbot adını verdikleri bir web tarayıcısını tanıtmışlardır. Verimli ve büyük ölçekli bir tarayıcı oluşturmadaki engelleri aşmak için sadece BFS tarama sırasını değiştirmiş ve düşük maliyetli disk tabanlı veri yapıları tasarlamışlardır. IRLbot, her etki alanına, algılanan itibarına dayalı olarak belirli bir değer vermekte ve her zaman aralığında yalnızca belirli bir değeri karşılayan URL'leri taramaktadır. Önerilen yöntemin temel farkı, bekleyen kuyruğunu sıralamamasıdır. Bunun yerine URL'leri karıştıran doğrusal taramalar kullanarak, değerlerini bir dizi öncelik sırasına ve başarısız olanları ayrı bir biriktirme dosyasına aktaran, doğrusal taramalar kullanarak çalışmasıdır. Yöntem, tarama boyutu arttıkça iyi bir ölçeklenebilirliğe sahiptir. Performansı, birikmiş bağlantıların miktarından etkilenmemektedir [18].

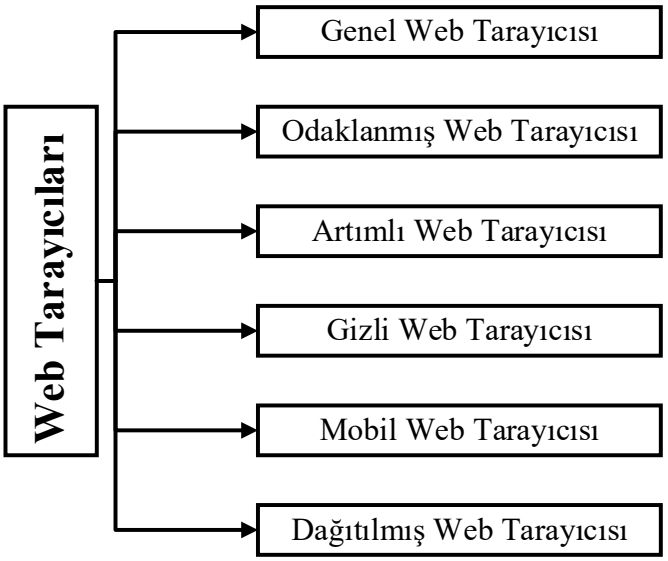
Baker ve Akcayol, kapsamı hızlı bir şekilde genişletmek amacı ile öncelik kuyruğu kullanarak bir web crawler algoritması geliştirmişlerdir. Geliştirilen algoritmaya göre, web sayfası içindeki URL'ler,

InterDomain ve IntraDomain olmak üzere iki kategoriye ayrılmıştır. Daha sonra InterDomainler için 2/3, IntraDomainler için ise 1/3 değerleri verilerek, bulunan URL'ler ile öncelikli kuyruk yapısı oluşturulmuştur. Bu sayede, InterDomain URL'ler yüksek öncelikli değere sahip olup, IntraDomain URL'lere göre öncelikli taranırlar. Bunun ana sebebi domain içerisindeki bağlantı döngülerini önlemektir. Tarama işleminin başlangıcından itibaren ilk 60 dakika içerisinde öncelikli kuyrukta 300.000' den fazla URL olduğu görüşmüştür. Ayrıca önerilen algoritma, URL frontier içerisindeki URL'leri tararken uzun bekleme durumlarını önlemek ve frontierin hafızasını yeni gelen yüksek öncelikli URL'lerde kullanmak için bir zamanlama mekanizması kullanmaktadır. Bu mekanizma kuyruk içerisindeki maksimum bekleme süresini aşan URL'leri kuyruktan çıkarmaktadır. Yapılan deneysel sonuçlara göre, önerilen algoritma iyi tarama performansı vermektedir [19].

Thangaraj ve Sivagaminathan, kapsamı tüm web olan bir tarayıcı tasarlamış ve geniş çapta bir arama yapmak için genişlik öncelikli arama algoritmasını geliştirmişlerdir. Yapılan çalışmaya göre genişlik öncelikli aramada en çok kullanılan yöntemlerden biri olan sabit seviyedeki derinlik yerine olasılık tabanlı dinamik rastgele derinlik ele alınmıştır. Önerilen tarayıcı daha fazla arama ve çıkarma için istenen belgeleri almak üzere, sınırsız genişlik ve rastgele derinlikte gezinir. Yazarlara göre mevcut genişlik öncelikli arama, bir Poisson olasılık dağılımını veya 0 ila 35 arasında değişen optimize edilmiş bir rastgele sayı döndürmek için olasılık kütle fonksiyonunu (Probability Mass Function-PMF) çağırarak geliştirilebilir. Mevcut tarayıcıyı geliştirmek amacı ile tarayıcıya bir PMF eklenmiştir. PMF ayrı bir rastgele değişkenin, bir değere tam olarak eşit olma olasılığını veren bir fonksiyondur. Web 'i taramak sonsuz değerlere yol açtığı ve hiç bitmeyebileceği için PMF 0 ile 35 arasında değişen rasgele derinlik üretmektedir. Bu, tarama sırasında ziyaret edilen her sayfadan indirilebilir sayfaları en üst düzeye çıkarmak amacıyla yapılır. Yapılan deneysel sonuçlara göre algoritma, 0 ila 35 arasında bir derinlik aralığı seçmiştir ve mevcut genişlik taramadan daha başarılı olduğu kanıtlanmıştır [20].

V. WEB TARAYICILARININ SINIFLANDIRILMASI

Web tarayıcıları kullanım amacı, kapsam, tarama politikası ve kullanılan stratejilere bağlı olarak çeşitli araştırmacılar tarafından farklı yöntemler ile sınıflandırılmıştır. Şekil 2'de web tarayıcılarının kapsamlı bir sınıflandırılması yapılmıştır.



***Şekil 1**. Web Tarayıcılarının Sınıflandırılması.*

A. GENEL WEB TARAYICILARI

Genel web tarayıcıları, kapsamı tüm Web olan tarayıcılardır. Belirli bir tohum URL seti ile taramaya başlar ve genişlik öncelikli arama ile taramaya devam eder. Google, Bing, Yandex vb. gibi ticari

tarayıcıların çoğu genel tarayıcılardır. Ticari olarak kullanıldıkları için mimari ve algoritmaları ile ilgili araştırmalar sınırlıdır.

Heydon ve Najork çalışmalarında, Mercator ismini verdikleri ölçeklenebilir ve genişletilebilir genel bir web tarayıcısını geliştirmişlerdir. Çalışmada, ölçeklenebilir bir tarayıcının ana bileşenleri açıklanmış ve tasarım alternatifleri tartışılmıştır. Yazarlar, Mercator 'u 8 günlük bir çalışma süresince, Google ve İnternet arşiv tarayıcısı ile karşılaştırmışlardır. Tarayıcının kapsamı, simülasyon çalışmalarında sınırlı tutulmuş fakat tüm Web olacak şekilde genişletilebileceği belirtilmiştir [21].

Günümüzde en popüler genel web tarayıcısı olan Google'ın kurucularından Page ve arkadaşları, web sayfalarının göreceli önemini ölçmek ve Web 'in grafiğine dayalı olarak her web sayfası için bir sıralama hesaplama yöntemi olan PageRank 'ı önermişlerdir. PageRank' in arama, göz atma ve trafik tahmininde bulunan uygulamaları vardır. Günümüzde kapsam olarak en iyi sonuç veren ve en çok kullanılan arama motoru olan Google, bu algoritma temel alınarak geliştirilmiştir [22].

Ticari olarak kullanılan tarayıcıların dışında açık kaynak kodlu tarayıcılar da geliştirilmiştir. Tablo 1'de Github' ın yıldız referans sistemine göre tercih edilen 15 açık kaynak tarayıcının programlama dili, uyumlu işletim sistemi, tarayıcı türü ve lisans bilgileri gösterilmiştir.

Tablo 1. *Açık Kaynak Tarayıcılar.*

Açık Kaynak Tarayıcı	Programlama Dili	İşletim Sistemi	Tarayıcı Türü	Lisans
Scrapy	Python	Çapraz Platform	Genel	BSD
Apache Nutch	Java	Çapraz Platform	Dağıtılmış	Apache
Heritrix	Java	Linux	Dağıtılmış	Apache
Crawl4J	Java	Çapraz Platform	Genel	Apache
PySpider	Python	Windows	Genel	Apache
Cola	Python	Çapraz Platform	Dağıtılmış	---
BUbiNG	Java	Linux	Dağıtılmış	Apache
WebMagic	Java	Çapraz Platform	Dağıtılmış	Apache
Portia	Python	Çapraz Platform	Dağıtılmış	BSD
Gnu Wget	C	Çapraz Platform	Genel	GNU
WebSphinix	Java	Çapraz Platform	Genel	GNU
WebCollector	Java	Çapraz Platform	Dağıtılmış	GPL
Node-Crawler	JavaScript	Windows	Genel	MIT
HAWK	C#	Windows	Genel	Apache
Goutte	PHP	Çapraz Platform	Genel	MIT

Tablo 1'de görüldüğü üzere geliştirilen açık kaynak tarayıcılarda Java ve Python programlama dilleri tercih edilmiş, büyük çoğunluğu platformdan bağımsız çapraz platformda inşa edilmiş ve en çok lisanslama Apache tarafından gerçekleştirilmiştir.

B. ODAKLANMIŞ WEB TARAYICILARI

Odaklanmış web tarayıcıları, belirli bir konu ile ilgili arama yapan tarayıcılardır. Bu tarayıcı, genel tarayıcılar gibi tüm bağlantılar yerine, kullanıcıların tercihine göre belirli bir alanı ya da konuyu taramak için kullanılır.

İlk odaklanmış web tarayıcısı Chakrabarti ve arkadaşları tarafından geliştirilmiştir. Önerilen odaklanmış web tarayıcısı belirlenen bir dizi konu ile ilgili Web araması gerçekleştirmektedir. Yazarlara göre odaklı tarayıcı, uzmanlığı örneklerden öğrenen ve Web'i keşfeden bir sistemdir [23]. Odaklı tarayıcılar, Web 'in alakasız bölgelerinden kaçınarak konu ile ilgili en alakalı olabilecek sayfaları bulur. [24]'de yazarlar beş kategoride sınıflandırılan odaklanmış tarayıcı yaklaşımlarının bir incelemesini sunmuştur. Bu yaklaşımlar öncelik tabanlı tarayıcı, yapı tabanlı tarayıcı, öğrenme tabanlı

tarayıcı, bağlam tabanlı tarayıcı ve diğer odaklı tarayıcılardır. [25]'da ise yazarlar odaklanmış tarayıcıları klasik, semantik ve öğrenen odaklanmış tarayıcılar olmak üzere üç gruba ayırmıştır. İlgi alanına ve aranan kelimelere göre istenilen bilgileri hızlı ve doğru bir şekilde getirdiği için odaklı tarama önemli bir araştırma alanı olmuştur. [26]'de odaklı tarayıcıların aranan konu ile ilgili tahminlerini iyileştirmek için önlemler geliştirilmiş ve [27]'de odaklı taramada kullanılan yöntemler tartışılmıştır.

Web' deki konuyla ilgili verileri almak ve yeni URL' leri çıkarmak için anahtar kelime ve ya arama ölçütlerine dayalı anahtar kelime tabanlı yaklaşımlar kullanılmaktadır [28]. Anahtar kelime tabanlı yaklaşımda, anahtar kelime içeren web sayfalarından veriler ve yeni URL' ler çıkarılmakta ve arama ile alakasız sayfalar görmezden gelinmektedir [27]. Kumar ve arkadaşları, anahtar kelime sorgusu tabanlı odaklı tarayıcıyı tartışmışlardır. Sorgu tabanlı tarayıcının harcanan zaman ve hassasiyet açısından önceki genişlik öncelikli tarayıcılardan daha verimli olduğu sonucuna varmışlardır. Çalışmada tohum URL'ler, anahtar veri setleri ile Google uygulama programlama ara yüzleri ve tohum arama ara yüzü kullanılarak elde edilirmiştir [29].

Web' in belirli bir bölümünü almak için kümeleme, sınıflandırma, Bulanık küme, Naive Bayes vb. gibi veri madenciliğini kullanan yaklaşımlar kullanılmaktadır [30-32].Safran ve arkadaşları çalışmalarında, ziyaret edilmeyen URL'lerin alaka düzeyini tahmin etmek için dört alaka özniteliği kullanan, öğrenmeye dayalı yeni bir odaklı tarama yaklaşımı sunmuşlardır. Kullanılan özellikler; sayfada bulunan URL sözcükleri, URL'lerdeki bağlantı metinleri, URL sözcüklerinin bağlı olduğu alan adları işaret eden ana sayfalar ve bu URL'leri çevreleyen metinlerdir. Deneysel çalışmalarında Naïve Bayesian sınıflandırma modeli benimsenmiştir. Tohum URL seçim işlemlerinde, ilk olarak konu kelimelerinden olan "Borsa", Google, Yahoo ve MSN arama motorlarına sorgu olarak gönderilmiştir. Daha sonra aday tohum URL'ler, belirlenen arama motorlarından elde edilen en iyi k-URL'dir Son olarak tohum URL'ler, aday k-URL'lerin üç arama motorunun en az ikisinin sonuç listesinde görünen URL'ler alınarak oluşturulmuştur [33].

Odaklanmış web tarayıcılarının performanslarını iyileştirmek için ontoloji tabanlı yaklaşımlar kullanılmaktadır. Ontoloji, alanın öğeleri arasında var olabilecek kavramları ve ilişkileri tanımlayan bir kavramsallaştırmanın özelliğidir [34]. Bu yaklaşımda ontoloji, bilgi deposunu yapılandırmak ve alakasız veriler filtrelemek amacı ile kullanılmaktadır [35, 36]. Agre ve Dongre, ontoloji tabanlı internet tarayıcısı için yalnızca konu ile alakalı web sitelerini alan, taramada en iyi tahmini kullanan ve tarama performansının iyileştirilmesine yardımcı olan bir odaklanmış tarayıcı kullanmışlardır [37]. Du ve arkadaşları, odaklı web tarayıcının konusu ile ilgili ontolojiden yararlanarak, kullanıcıların ilgi alanlarına dayalı olan tohum URL'leri seçmek için yeni bir yöntem geliştirmişlerdir. Bu çalışmada ontoloji oluşturma yaklaşımının üç adımını tanımlamışlardır. Bunlar çalışmada kavram seçimi, optimize edilmiş kavram kafesinin oluşturulması ve kavram kafesinin kullanıcı-çıkar ontolojisine eşlenmesi sürecidir. Bu kullanıcı ilgi ontolojisi, örtük kavramları ve aralarındaki ilişkiyi tanımlamıştır. Yazarlara göre etkili tohum URL seçimi ile tarayıcı, arama sorgusuyla alakalı daha verimli sonuçlar üretmektedir [38].

Odaklanmış web tarayıcılarında temel amacı arama konusu ile taranan sayfanın birbirleri ile anlamsal olarak benzerliklerinin belirlenmesi olan semantik tabanlı yaklaşımlar da kullanılmaktadır. Semantik tabanlı yaklaşımda anlamsal benzerlik oranını belirlemekteki temel şart içerik analizinin yapılmasıdır [39]. İçerik analizi yapılırken en çok kullanılan algoritmalar PageRank [22], görsel tabanlı sayfa segmentasyonu [40], SiteRank [41] ve densometrik segmentasyon [42] olarak sıralanabilir.

Odaklanmış web tarayıcılarında literatürde en çok kullanılan performans ölçütleri hasat oranı, kesinlik ve ilişki oranıdır. Hasat oranı odaklı tarayıcılarda kullanılan bir performans ölçütü olup ilgili web sayfaları ile taranan web sayfaları arasındaki oranı temsil etmektedir. Kesinlik oranı ya da diğer adıyla hassasiyet oranı, odaklanmış tarayıcılarda verimliliği değerlendirmek için elde edilen verilerin kalitesinin ölçüsüdür. Kesinlik yüksek olması odaklı web tarayıcısının alakasız sayfaları o derecede iyi reddettiğini göstermektedir. Kesinlik temel olarak döndürülen sonuçların toplam sonuçlara oranını göstermektedir. Geri çağırma oranı ise bir arama talebine cevap olarak alınan ilgili sayfaların oranını

ifade etmektedir. Tablo 2'de yapılan çalışmalarda önerilen odaklanmış web tarayıcıların hasat oranı, kesinlik/hassasiyet, sayfaların ontoloji ile ilişki oranları ve taranan toplam sayfa sayıları gösterilmiştir.

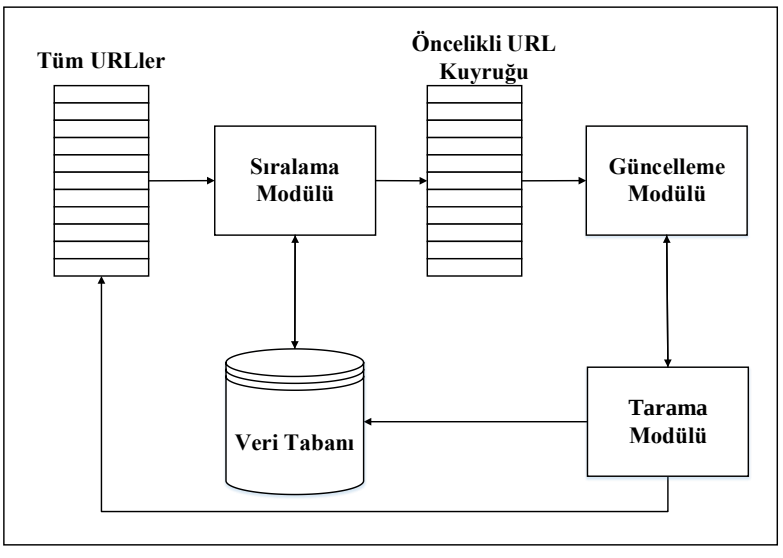
Tablo 2. *Odaklanmış web tarayıcılarının performansları.*

Kaynak	Hasat Oranı	Kesinlik/ Hassasiyet*(%)*	Geri Çağırma Oranı	Taranan Sayfa Sayısı
[43]	0.389	36.00	0.611	5000
[44]	0.411	---	---	1000
[45]	0.810	---	---	5000
[46]	0.850	---	---	5000
[47]	0.500	---	0.600	6500
[48]	---	92.01	0.590	495
[49]	0.890	---	---	1000
[50]	0.500	32.00	---	3200
[51]	0.830	---	---	1200
[26]	---	92.00	0.300	2000
[25]	0.750	---	0.400	10000
[52]	---	---	0.820	400
[53]	0.850	---	---	6000
[54]	0.700	---	0.470	13377

Tablo 2' de görüldüğü gibi odaklı web tarayıcılarında en çok kullanılan performans ölçütü hasat oranıdır. Hasat oranı taranan sayfa sayıları arttıkça genel olarak artmaktadır. Odaklı web tarayıcılarında ontolojinin kapsamı, taramaya başlanılan tohum URL setinin kalitesi ve kullanılan tarama algoritmasının başarısına göre performanslar değişiklik göstermektedir.

C. ARTIMLI WEB TARAYICILARI

Artımlı web tarayıcısı, tarama işlemini her defasında yeniden başlatmak yerine, sayfaları düzenli olarak güncelleyen geleneksel bir tarayıcı türüdür. Artımlı tarayıcı Web'i sürekli tarar ve sayfaları periyodik olarak yeniden ziyaret eder [55]. Artımlı web tarayıcısının temel hedefi yerel koleksiyonu taze tutmak ve yerel koleksiyonun kalitesini artırmaktır [56]. Odaklanmış taramada artımlı web tarayıcılardan toplanan bilgilerin alaka düzeyi diğer tarayıcılara kıyasla az olabilmektedir. Ayrıca artımlı tarayıcılar gizli webi taramada başarısız olurlar. Artımlı web tarayıcısının mimarisi Şekil 3'de gösterilmiştir [57].



Şekil 3. *Artımlı Web Tarayıcı Mimarisi.*

Gupta ve arkadaşlar, artımlı web tarayıcısında kullanmak için geliştirdikleri mekanizma ile her bir URL'e sayfa sıralaması, sayfa yenileme sıklığının hesaplanması ve web sayfasının içeriğinin kullanıcıların ilgi alanlarına uygun olup olmadığı durumuna göre bir öncelik vermektedirler. Tohum URL'ler eğlence, turizm, cep telefonları, spor, mülk olmak üzere beş farklı alandan olup daha az orta düzeyde ve oldukça dinamik web sayfalarının bir karışımından oluşmaktadır. Kapsam olarak da belirtilen beş farklı alan ile ilgili web sayfalarını içermektedir [58].

Pavai ve Geetha, geliştirdikleri artımlı web tarayıcısı algoritması ile bir web sayfasının gelecekteki değişim zamanını tahmin etmeyi amaçlayan veri madenciliğine dayalı yaklaşımı kullanmışlardır. Yüzey web sayfalarını ve derin web veri tabanlarını aşamalı olarak taramak için Bayes olabilirlik tahmincisini kullanan yeni bir istatistiksel yaklaşım sunmuşlardır. Tohum URL'leri almak için CLIA- "Cross Lingual Information Access" projesinin tohum URL'lerini (20 Ağustos 2015'te taranmış) ve sorgu koleksiyonlarını kullandılar. Çalışmanın kapsamı da filmler, iş, kitap ve uçak bileti rezervasyonu olmak üzere dört alan ile sınırlandırılmıştır [59].

Santos ve arkadaşları, sayfa güncelliğine dayalı yaklaşım benimsemiş ve artımlı web tarayıcısının zamanlama mekanizması için yeni bir algoritma geliştirmişlerdir. Geliştirilen algoritma ile web sayfalarını son taranmalarından bu yana değiştirilme olasılıklarına göre sıralamak ve web tarayıcılarının planlayıcıları tarafından kullanılacak puan işlevlerini otomatik olarak oluşturmak için bir genetik programlama çerçevesi sunmuşlardır. Çalışmada tarayıcının tazelik oranına odaklanılmış ve kapsam olarak bir sınırlama verilmediğinden tüm Web hedeflenmiştir [60].

Tan ve Mitra, geliştirdikleri artımlı web tarayıcısında kümeleme yöntemini kullanmışlardır. Mevcut web sayfalarını hem statik hem de dinamik özelliklerini kullanarak 100 kümeye ayırmışlardır. Çalışmaya göre tarayıcı, sayfaların son indirilmelerinden bu yana değişip değişmediğini kontrol etmek için bir kümeden bir web sayfası örneği getirmektedir. Bir kümedeki önemli sayıda sayfa değiştiyse, kümedeki diğer web sayfaları da karşıdan yüklenmektedir. Web sayfalarının kümeleri farklı değişiklik sıklıklarına sahiptir. Çalışma artımlı taramayı aynı web sayfaları ve sınırlı kapsamda gerçekleştirmiştir. Belirlen web sayfalarının dışındaki sayfalar taranmamıştır [61].

Shi ve arkadaşları çalışmalarında, Scrapy adlı tarayıcı mimarisi temelinde artımlı bir web tarayıcısı geliştirmişlerdir. Tohum URL olarak belirledikleri haber sitelerinin bağlantılarını kullanmışlardır. Geliştirilen tarayıcı, web sitesindeki haber bilgilerini başarılı bir şekilde tarayarak artımlı tarama gerçekleştirmiştir. Buna ek olarak tarayıcı, Bloom filtresi ile haber sitesini izlemiş ve güncellenen haberleri veri tabanında saklamıştır. Ancak web tarayıcısının en büyük dezavantajı, kapsamı haber siteleri olduğu için genel tarama yapmamaktadır [62].

Nagar ve Singhal, artımlı bir web tarayıcısının tekrar ziyaret sıklığı problemi üzerinde durmuş ve web sayfalarına ziyaret sıklığını yönetmek için yeni bir yaklaşım önermişlerdir. Çalışmaya göre sunucudaki bir log dosyası veri tabanı, web sayfası kimliğini, sayfadaki URL'leri ve başarılı bağlantı sayılarını kaydetmektedir. Başarılı bağlantı sayısı belli bir değeri geçen URL'lerin sıralaması yükseltilmektedir. Başarılı bağlantı sayısı bir log dosyasında tutulmakta ve bu bilgi, farklı web sayfalarının öncelikli sıralamasını oluşturmada kullanılmaktadır. Bu öncelikli sıralama bilgisi kullanıcıların arama geçmişinden oluşan log dosyası veri tabanına eklenmektedir. Bunun sonucunda Web'den, yüksek dereceli sayfalar, kullanıcıların arama geçmişine göre tarayıcı tarafından indirilerek veri tabanına kaydedilmektedir [63].

Artımlı web tarayıcılarının en önemli özelliği farklı zaman dilimlerinde web sayfalarını tarayıp değişiklikleri tespit etmektir. Artımlı web tarayıcılarında başarı oranı, toplam denemeler arasındaki başarı yüzdesi olarak tanımlanmaktadır. Başarı oranının yüksek olması kapsamın geniş olması ve daha az gereksiz istek içermesi anlamına gelmektedir. Artımlı web tarayıcılarındaki temel prensip ziyaret edilen web sayfalarını güncellenme sıklıklarına bağlı olarak tekrar ziyaret etmektir. Kullanılan tarama algoritmasına göre ziyaret sıklıkları sabit ya da değişken aralıkta olup farklılık göstermektedir. Buradaki en önemli konu sık güncellenen sayfaları nadiren ya da hiç güncellenmeyen sayfalara göre daha kısa süre sonra ziyaret etmektir. Tablo 3'de bazı araştırmacıların yaptıkları çalışmalarda tespit

edilen değişiklikler (Başarı Oranı), sayfaları tekrar ziyaret sıklığı, kullanılan tarayıcı türü ve deneylerde veri setlerinin oluşturduğu sayfa sayıları/belge setleri gösterilmiştir.

***Tablo 3**. Artımlı web tarayıcılarının performansları.*

Kaynak	Başarı Oranı	Ziyaret Sıklığı(Saat)	Tarayıcı Türü	Sayfa Sayısı/ Belge Seti
[58]	%90.0	0.25-0.75	Odaklanmış	56-148 web sitesi
[59]	%83.0	0-720	Odaklanmış	10 M belge seti
[61]	%80.0	168-1344	Genel	210 web sitesi
[64]	%91.6	8760	Odaklanmış	14 web sitesi
[65]	%80.0	2	Odaklanmış	2 web sitesi
[66]	%89.0	2,5	Odaklanmış	718 web sitesi

Artımlı web tarayıcılarında başarı oranlarının %80 ve üzerinde bir değer aldığı, ziyaret sıklıklarının da yapılan çalışmada kullanılan algoritmaya, kapsama ve tarayıcı türüne göre değiştiği görülmüştür. Artımlı web tarayıcıları sayfaları tekrar tekrar ziyaret ettiği için performans ve bant genişliği kullanımı bakımından dezavantajları vardır. Bundan dolayı çalışmalarda bant genişliğini boşa harcamamak için ziyaret sıklıkları sayfalara göre farklılık gösterebilmektedir. Buna ek olarak kapsamı küçültmek ve belirli bir konuda tarama yapmak için çoğunlukla odaklanmış tarayıcı türleri tercih edilmiştir.

D. GİZLİ WEB TARAYICILARI

Web sayfalarında bilgiler yüzeysel ve gizli olmak üzere iki şekilde saklanmaktadır. Yüzeysel webdeki verilere herhangi bir tarayıcı ulaşabilir. Gizli webdeki verilere ise oturum açma formu, arama veya sorgu arabirimleri ile ulaşıldığından, arama motorlarının kapsamı dışındadır. Gizli webdeki web sayfaları, bir veri tabanındaki sorgu arabirimleri aracılığıyla gönderilen sorgulara yanıt olarak birleştirilir. Web ‘de formların, ara yüzlerin vb. arkasına gizlenmiş ve dizine eklenemeyen büyük miktarda bilgi vardır [67]. Bu verilere ulaşan tarayıcılar gizli web tarayıcılarıdır. Literatürde bu tarayıcılar derin web tarayıcıları olarak da anılmaktadır.

Zerfos ve arkadaşları, gizli Web’ deki veriler sorgu ile almak için üç farklı sorgu oluşturma politikası önermişlerdir. Bunlar anahtar kelime listesinden rastgele sorguları seçen bir politika, genel bir metin koleksiyonunda sorguları sıklıklarına göre seçen bir politika ve uyarlanabilir bir şekilde iyi bir sorguyu temel alan bir politikadır. Yazarlar tarafından seçilen dört gerçek gizli web sitesi üzerinde deneysel çalışma yapılmış ve yaklaşık 100 sorgu yaptıktan sonra bir gizli web sitesinin %90'ından fazlasını karşıdan yükleyebilmişlerdir [68].

Kaur ve Geetha, gizli Web’i tarama amacı ile SIMHAR adını verdikleri ağaç tabanlı yaklaşımı benimseyen dağıtılmış bir web tarayıcısı önermiş ve uygulanmışlardır. Scrapy, Framework ve Redis sunucusunu birleştirerek taramayı, uyarlama, ilgili kaynak seçimi ve temel içerik çıkarma olmak üzere üç aşamaya bölmüşlerdir. Tarayıcı, aranabilir formları doğru bir şekilde algılamış ve göndermiştir. Çalışmada önerilen tarayıcı, gizli webden belirli bir konu ile alakalı sonuçları aradığından odaklanmış tarayıcı gibi çalışmaktadır. Tohum URL’ler ise sisteme yazarlar tarafından konu ile ilişkili olarak DMOZ veri setinden seçilerek eklenmiştir. Veri seti 6 farklı alanda 260000’den fazla URL içermektedir [69].

Gupta ve Bhatia çalışmalarında, HiCrawl adını verdikleri alana özgü yaklaşımı benimseyen ve tıp alanındaki siteleri taramayı hedefleyen odaklanmış gizli bir web tarayıcısını önermişlerdir. Önerilen tarayıcı beş temel bölüme sahiptir. Bunlar indirici, web sayfası çözümleyicisi, form çözümleyicisi, etki alanına özgü bir veri havuzu kullanan form işlemcisi ve etki alanına özgü sınıflandırma hiyerarşisidir. HiCrawl tohum URL seti, tıp alanına ait web sayfa URL’lerinden oluşmaktadır. Bu tohum URL seti, Google, Yahoo vb. gibi herhangi bir güvenilir web arama motorunun dizinlerinden elde edilmiş ve bir FIFO kuyruğu olan tohum URL sınırında saklanmaktadır [70].

Bhatia ve arkadaşları çalışmalarında, AKSHR adını verdikleri etki alanına özgü gizli web tarayıcısını önerdiler. Önerilen tarayıcının gizli web veri tabanlarını taramak için arama ara yüzlerinin otomatik olarak indirilmesi, etki alanına özgü ara yüz yaklaşımı kullanılarak arama ara yüzü öğeleri arasındaki anlamsal eşlemelerin tanımlanması ve arama ara yüzlerini otomatik doldurma yeteneği olduğu belirtilmiştir. Önerilen çalışmanın deneysel değerlendirmesi için kitaplar, havayolu, elektronik, emlak ve filmler olmak üzere beş alan dikkate alınmış ve arama ara yüzü tarayıcısına her etki alanı için bir tohum URL sağlanmıştır. Önerilen tarayıcının ortalama tarama hassasiyeti %76,9 ile %93,1 arasında değiştiği görülmüştür. Genel hassasiyet ortalaması, geri çağırma ortalaması ve genel ortalama F-ölçüsünün sırasıyla %85.23, %86.32 ve %85.44 olduğu ve AKSHR' nin hem tutarlı hem de verimli olduğunu belirtilmiştir [71].

Gizli web tarayıcılarında performans ölçütü olarak genel tarayıcılarda kullanılan ölçeklenebilirlik, tazelik ve benzeri ölçütler kullanılır. Bunların yanında hedef gizli web sitesinde bulunan toplam sayfa sayısının, getirilen sayfa sayısına oranı olan hasat oranı da önemli bir ölçüttür. [72]'de yazarlara göre tarayıcı tarafından taranan toplam sayfa sayısı Pc, toplanan derin web veri tabanının alana özgü toplam sayfa sayısı Nf ve tarayıcının arama alanındaki alan özgü aranabilir sayfaların toplam sayısı Ncf ise hasat ve kapsama oranı denklem 1 ve 2 'deki gibi ifade edilebilir.

$$Hasat\ Oranı = \frac{N_{cf}}{P_c} \quad (1)$$

$$Kapsama\ Oranı = \frac{N_{cf}}{N_f} \quad (2)$$

Burada hasat oranı ve kapsama oranı Ncf arttığında artmaktadır, Pc ve Nf arttığında ise azalmaktadır. Webde bulunan kaliteli verilerin büyük çoğunluğu çeşitli kurumlar ve özel firmalar tarafından eklenmektedir. Bundan dolayı gizli web tarayıcılarının kapsamı yüzey web tarayıcılarına göre daha geniştir ve bu da gizli web tarayıcılarının önemini arttırmaktadır. Gizli web tarayıcıları, verileri form doldurma, üye girişi ve sorgulamalar sonucu almaktadır. Gizli web tarayıcıları verilere direk ulaşamaması beraberinde birçok dezavantajı getirmektedir. Bunlar ölçeklenemez olması, ağ üzerinde ağır yük oluşturması, yarı otomatik olması ve belirli bir formatta arama yapamamasıdır. Tablo 4'de önceki çalışmalarda elde edilen veriler sunulmuştur.

Tablo 4. *Gizli web tarayıcılarının performansları.*

Kaynak	Kapsama Oranı	Sorgu sayısı	Kapsam (web sitesi)
[73]	%72.0	200	5
[72]	%65.1	---	50
[68]	%85.0	3500	154
[70]	%87.5	16	1
[74]	%70.0	6	1
[75]	%96.0	2	1308
[76]	%92.5	2	68
[77]	%90.0	487	2
[78]	%73.7	---	1
[79]	%90.0	53	100

Tablo 4'de görüldüğü gibi gizli webin ölçeklenemez olmasından dolayı çalışmalarda kapsam olarak belirli sayıda web siteleri kullanılmıştır. Tarama yapılan gizli web sitesinin yapısına ve giriş için istenen işlemlere göre sorgu sayısı değişiklik göstermektedir.

E. MOBİL WEB TARAYICILAR

Mobil web tarayıcılarında sayfa seçim ve filtreleme işlemleri, arama motoru tarafından değil sunucu

tarafından gerçekleştirilir. Mobil taramada verilerin koda taşınması yerine kodun verilere taşınması söz konusudur. Bu durum geleneksel web tarayıcılarının neden olduğu ağ yükünü azaltır [67]. Web taraması bağlamında bir tarayıcının, veri kaynağına yani web sunucusuna geçiş yapabilme yeteneği mobilite olarak adlandırılmaktadır [74].

Kumar ve Bhatia yaptıkları çalışmalarında, gizli webden veri almak için mobil taramanın çeşitli avantajlarını kullanan bir yaklaşım önermişlerdir. Tarama sırasında geleneksel olarak verileri koda taşımak yerine, kodu verilere taşıma kavramını kullanmışlardır. Önerilen yaklaşım, yinelemeli olarak sorguları çalıştırarak yerel veri tabanından verileri alır. Sorgular, minimum sayıda sorguda maksimum veri kümesini kapsayacak şekilde potansiyel anahtar kelimeler kullanılarak oluşturulmuştur. Çalışmada kapsam olarak Hindistan dışında yaşayan Hint kökenli akademisyenlerin verilerini toplamak olduğu için odaklı bir tarama gerçekleştirilmiştir. Buna ek olarak tohum URL olarak yazarlar tarafından seçilmiş sitelerden veriler alınmıştır [74].

Anbukodi ve Manickam yaptıkları çalışmada, genel tarayıcıların ağ bant genişliği kullanımı, CPU döngülerinin ve belleğinin kullanımı, işletim sistemlerinin sınırları vb. nedenler ile uzak sunucuda yüke sebep olan durumların ortadan kaldırılması için mobil tarayıcı mimarisi önermişlerdir. Önerilen mobil taramanın avantajı, tarama işlevselliğini tüm Web üzerinde dağıtmamıza izin vermesi ve son taramadan bu yana değiştirilmemiş sayfaları filtreleyebilmesidir. Yerelleştirilmiş veri erişimi uzak sayfa seçimi, filtrelenmesi ve sıkıştırılması gibi özelliklere sahiptir. Bu tür taramada sanal bir sistem üzerinde denemeler yapıldığından kapsam bir veya birkaç sunucu iken tohum URL olarak da yazarların belirlediği birkaç web sitesi kullanılmıştır [80].

Nath ve Bal, mobil web tarayıcılarının etkili bir şekilde çözebildiği Web 'in katlanarak büyümesi ve buna bağlı olarak trafik ve bant genişliğinin tüketimi sorununu ele almışlardır. Çalışmalarında mobil tarayıcıya dayalı verimli bir indeksleme sistemi sunarak bant genişliği tüketim sorununa çözüm önerileri sunmuşlardır. Yazarlar tarafından geliştirilen tarayıcı, uzak sayfayı ziyaret ederek son taramadan sonra değişiklik için sayfayı analiz eder ve sadece değiştirilmiş olanları döndürür. Yazarlar tarafından belirlenen 100 web sayfasında 30 gün boyunca denemeler yapılmıştır. Önerilen mobil tarayıcı sistemi, geleneksel tarayıcı sistemiyle sayfa değiştirme davranışı, ağdaki yük ve bant genişliği olmak üzere üç parametre ile karşılaştırılmıştır. Deneysel sonuçlar, önerilen mobil tarayıcının geleneksel tarayıcı sistemlerinden daha verimli olduğunu göstermiştir [81].

Mobil tarayıcıların temel özelliği, sayfaların değişip değişmediğini sunucu üzerinde kontrol ederek sadece değişen sayfaların alınmasını sağlamaktır. Bu sayede bant genişliğinden ve işlemci performansından kazanç sağlanır. Tablo 5'de çeşitli çalışmalarda değişen sayfaların alınıp değişmeyen sayfaların alınmaması nedeniyle elde edilen başarı oranı, mobil ajan sayısı, deneylerde taranan URL sayısı ve ağ trafiğinde azalma yüzdesi (ATAY) gösterilmiştir.

Tablo 5. *Mobil web tarayıcılarının performansları.*

Kaynak	Başarı Oranı	Mobil Ajan Sayısı	Taranan URL sayısı	ATAY
[80]	%48	2	100	%43.71
[82]	%52	15	168000	---
[81]	%37	2	100	%30.40
[83]	%93	1	15000	---
[84]	%95	5	---	---

Tablo 5'de de görüldüğü gibi mobil web tarayıcılarında performans ölçütü olarak kullanılan başarı oranı, taranan toplam URL sayısı içinde mobil ajanlar tarafından değişiklik tespit edilip geri gönderilen URL sayısının yüzdesini temsil etmektedir. ATAY başarı oranına paralel olarak artıp azalmaktadır. Mobil ajanlar sunucu bilgisayarlarda barındırıldıkları için sunucu sayısının artmasına bağlı olarak mobil ajan sayısının da artmaktadır.

F. DAĞITILMIŞ WEB TARAYICILARI

Dağıtılmış web tarayıcıları coğrafi olarak farklı konumlar üzerinde çalışan tarayıcılardır. Farklı konumlarda çalışması ağ trafiğini azalttığı, yüksek ölçeklenebilirlik ve yükü dağıtma eğiliminden dolayı verimliliği arttırır. Bu avantajlarının aksine verileri yönetme ve gizli ağın taranması konularında yetersizdir [85]. Dağıtılmış web tarayıcısı, geleneksel tarayıcılar temel alınarak geliştirildiğinden dolayı çalışma prensibi ve temel yapısı geleneksel web tarayıcısına benzemektedir [86].

Cai ve Zhang çalışmalarında, dinamik web sayfasının ayrıştırma zorluğu nedeniyle eksik bilgi alınması ve tarayıcı verimliliği gibi sorunlarını çözmek için Dis-Dyn adını verdikleri dağıtılmış web tarayıcısını önermişlerdir. Yazarlar geliştirdikleri Dis-Dyn tarayıcısında, dinamik sayfaları ayrıştırmak için HtmlUnit'i kullanmış ve tarayıcının verimliliğini artıran dağıtım özelliğini gerçekleştirmek için Redis ve ZMQ' yu (Zero Message Queue) seçmişlerdir. Tohum URL olarak sadece bir adet dinamik film sitesi üzerinde denemeler yapılmış ve kapsamda bununla sınırlı kalmıştır [87].

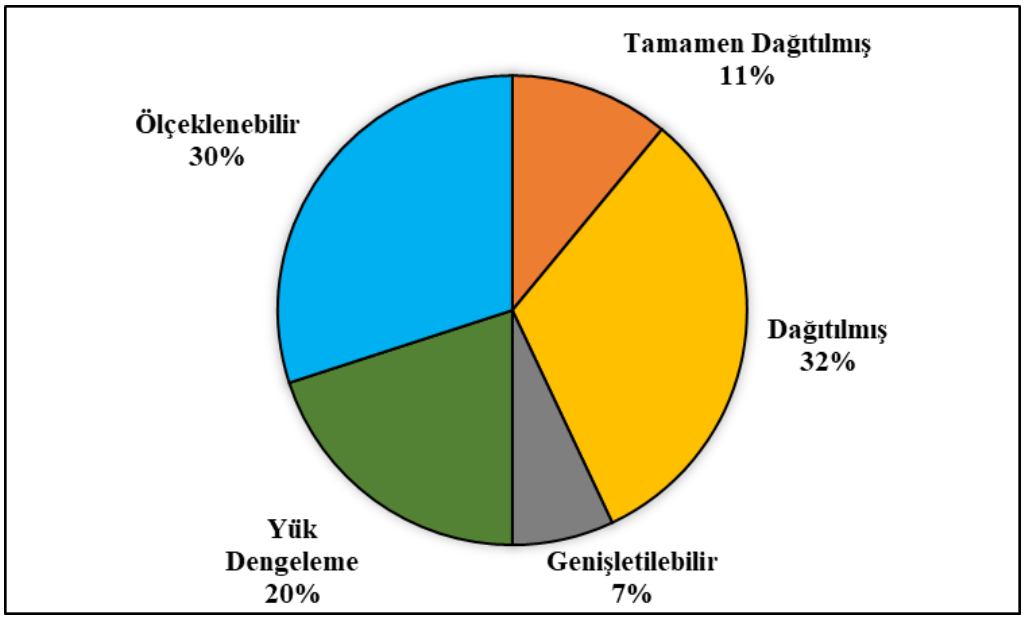
Yu ve arkadaşları çalışmalarında, Hadoop platformunda çalışan dağıtılmış bir web tarayıcı modeli önermişlerdir. Önerilen tarayıcı, geleneksel dağıtık tarama sisteminde var olan uzantı ve yük dengeleme problemini çözmeyi amaçlayan, bulut bilişime dayalı, dağıtık bir web tarayıcısıdır. Yazarlar tarafından seçilen tohum URL'ler ile başlayıp tüm Web 'i hedef alan genel bir tarama yapılmıştır. Tarama sonuçları Nutch ile kıyaslanmış ve önerilen tarayıcının performansının daha iyi olduğu belirtilmiştir [88].

Le Quoc ve arkadaşları, coğrafi olarak dağıtılmış ve harita azaltma tabanlı bir tarayıcı olan UniCrawl'ı sunmuşlardır. UniCrawl, coğrafi olarak dağıtılmış birkaç siteyi düzenlemektedir. Her site bağımsız bir tarayıcı çalıştırmakta ve Web içeriğini almak ve ayrıştırmak için çeşitli teknikler kullanmaktadır. UniCrawl, taranan etki alanını siteler arasında böler ve siteler arası iletişim maliyetini en aza indirirken depolama ve bilgi işlem kaynaklarını birleştirir. Belirli bir bölgeye yayılmış ve yazarlar tarafından belirlenmiş üç web sitesi üzerinde yapılan deneylerde önerilen tarayıcının ağ tüketimi açısından %93,6'lık bir performans artışı ve 1,75'lik bir hızlanma faktörü gösterdiği belirtilmiştir [89].

Pu çalışmasında, dağıtılmış web tarayıcılarının etkin bir şekilde kullandığı Hadoop teknolojisine dayalı bir web tarayıcı sistemi önermiştir. Bu dağıtılmış web tarayıcısının temel özellikleri yapılandırılabilir, verimli ve ölçeklenebilir olmasıdır. Tohum URL seti olarak 20 haber web sitesi URL'i kullanılmıştır. Çalışmada test ortamının ağ bant genişliği saniyede 2 MB olduğu belirtilmiş ve URL bağlantısının tarama derinliği 2 olarak ayarlanmıştır. Tarayıcı yapılan 3 farklı denemede yaklaşık 11000 web sayfasını indirmeyi başarmıştır. Önerilen tarayıcının gerçek ortamdaki kapsamı tüm Web 'dir [90].

Liu ve Jin yaptıkları çalışmalarında, ChainMR Crawler adlı dağıtık bir dikey tarayıcı önermektedir. Önerilen tarayıcı, veri madenciliği tabanlı bir tarayıcı olup URL yönetim modülü, indirme modülü, HTML bölme modülü ve depolama modülünden oluşmaktadır. Her modül URL'i yönetmek için Redis bellek veri tabanını kullanmış ve yinelenen URL'leri filtrelemek için BloomFilter algoritması benimsenmiştir. Nutch ile yapılan karşılaştırmalı taramalar sonucunda ChainMR Crawler' ın verimliliğinin daha iyi olduğu kanıtlanmış ve çeşitli ihtiyaçlara genişletilebilir olduğu belirtilmiştir [91].

Dağıtılmış web tarayıcılarında performans ölçütleri ölçeklenebilirlik, hataya karşı dayanıklılık, dağıtım derecesi, koordinasyon, bölümleme teknikleri, kapsam, örtüşme ve iletişim yüküdür [92]. Şekil 4'de genel olarak dağıtılmış web tarayıcısında bulunan farklı özelliklerinin payı gösterilmiştir [93].



Şekil 4. *Dağıtılmış web tarayıcılarında farklı özelliklerinin payı.*

Dağıtılmış web tarayıcısının performansı dağıtılmıştık derecesi, tarama süresi, bant genişliği ve kullanılan sistemin özelliklerine göre farklılık gösterilmiştir. Tablo 6'da yapılmış çalışmalarda tarama zamanı, taranan sayfa sayısı veya boyutu, dağıtılmış tarayıcı yürü ve dağıtım adetleri özelliklerinin karşılaştırılması sunulmuştur. Dağıtılmış web tarayıcılarında taranan sayfa sayısı/boyutu tarama zamanı ve dağıtım adedine paralel olarak arttığı görülmüştür. Ayrıca dağıtılmış genel web tarayıcılarının dağıtılmış odaklanmış web tarayıcılarına göre daha fazla sayfa taradığı görülmüştür.

Tablo 6. *Dağıtılmış web tarayıcıların performansları.*

Kaynak	Tarama Zamanı (Dakika)	Taranan Sayfa Sayısı/Boyutu	Tarayıcı Türü	Dağıtım Sayısı(Adet)
[88]	---	26136	Genel	10
[94]	6.3	300	Genel	5
[95]	60.0	---	Genel	4
[96]	32.0	6000	Genel	>100
[90]	37.2	81003	Odaklanmış	4
[97]	43200.0	23501715	Odaklanmış	300
[91]	30.0	4500	Odaklanmış	6
[98]	35.0	96532	Odaklanmış	4
[99]	600.0	7771402	Genel	6
[100]	1.0	560	Odaklanmış	16
[101]	420.0	20000	Odaklanmış	4
[102]	1648.2	725.46 MB	Genel	2

Tablo 7 'de genel, odaklanmış, artımlı, gizli, mobil ve dağıtılmış web tarayıcılarının kapsamlı karşılaştırılması sunulmuş ve bu tarayıcılar, kapsamları, kullandıkları algoritmalar, sayfa seçim yöntemleri, tohum URL kaynağı, örnek tarayıcılar ve ölçeklenebilirlik özelliklerine göre karşılaştırılmıştır. Kapsam bakımından incelendiğinde genel web tarayıcılarının kapsamı tüm Web, odaklı web tarayıcıların kapsamı belirli bir konu ya da alan, mobil tarayıcılarının kapsamı belirli web sayfaları olup diğer tarayıcılar ise çalışma alanına göre genel ve odaklı tarayıcılar ile aynı kapsamı hedeflemektedir. Kapsamı tüm Web olan tarayıcılar genel olarak genişlik öncelikli arama algoritmasını, kapsamı Web 'in bir bölümü ya da belirli bir konu olan tarayıcılar ise derinlik öncelikli arama algoritmasını birincil olarak kullanmaktadır. Genel olarak web tarayıcılarında tarama işlemi tohum URL'lerden başlamaktadır. Odaklanmış ve gizli web tarayıcılarında hedef sayfalar belli

olduğundan konu, anahtar kelime ve form doldurma yöntemleri ile tohum URL'leri seçip aramalarını gerçekleştirmektedirler. Web tarayıcılarının tohum URL kaynakları türlerine, kapsamlarına ve araştırma alanlarına göre manuel, yarı otomatik veya otomatik olarak seçilebilmektedir. Literatürde yapılan çalışmalarda ve ticari olarak geliştirilen arama motorlarında farklı özellikte web tarayıcıları geliştirilmiştir. InfoSpiders, OntoCrawler vb. bazı web tarayıcıları belirli bir türe sahip iken SIMHAR, SmartCrawler vb. bazı tarayıcılar ise birden farklı türde tarayıcı özelliği göstermektedir. Son olarak genel, artımlı ve gizli web tarayıcılarının kapsamları önceden bilinmediğinden ölçeklenebilirlik özellikleri yoktur, bunların dışındaki tarayıcılar ise belirli bir kapsam, anahtar kelime, sorgu sonucunda tarandığından ölçeklenebilir özelliklere sahiptir.

Tablo 7. *Web tarayıcılarının karşılaştırılması.*

Tarayıcı Türü	Kapsam	Kullanılan Algoritmalar	Sayfa Seçimi	Tohum URL Kaynağı	Örnek Tarayıcı	Ölçeklene-bilirlik
Genel Web Tarayıcı	Tüm Web	Genişlik Öncelikli Arama, Page Rank, HITS	Tohum URL	Page Rank ve HITS değeri en yüksek sayfalar	Google, Mercator, Yahoo, Bing, Yandex vs. Açık Kaynak Tarayıcılar	Hayır
Odaklanmış Web Tarayıcı	Webin konu ya da alan tabanlı belli bölümü	Derinlik Öncelikli Arama, Köpek Balığı Arama, Vektör Uzay Modeli, En İyi İlk, En İyi N İlk, Page Rank HITS	Konu veya Anahtar kelime	Kelime veri setlerinin arama sonuçları DMOZ veri seti, Belli konudaki web sayfaları	InfoSpiders, OntoCrawler, LSCrawler,	Evet
Artımlı Web Tarayıcıları	Tüm Web, Webin konu ya da alan tabanlı belli bölümü	Genişlik Öncelikli Arama, Derinlik Öncelikli Arama, En iyi ilk arama, Page Rank, HITS	Tohum URL, Öncelik sırasına bağlı URL	Page Rank ve HITS değeri en yüksek sayfalar Belirli koleksiyonlar	SmartCrawler RCrawler WebMiner	Hayır
Gizli Web Tarayıcıları	Belirli web sayfaları	Uyarlanabilir algoritma, Genel-frekans algoritması, rastgele16K, rastgele-1M	Anahtar Kelime sorgusu veya form doldurma	Kelime tabanlı sorgulama, Form tabanlı sorgulama, özellik ve etiket çıkarmaya dayalı sorgulama, DMOZ veri seti, arama motorları.	SIMHAR SmartCrawler HiCrawl AKSHR	Hayır
Mobil Web Tarayıcılar	Belirli Web sayfaları	Genişlik Öncelikli Arama, Derinlik Öncelikli Arama	Tohum URL	Kelime tabanlı sorgulama		Evet
Dağıtılmış Web Tarayıcıları	Tüm Web, Webin konu ya da alan tabanlı belli bölümü	Genişlik Öncelikli Arama	Tohum URL	Tohum URL Seti	UbiCrawler SIMHAR Dis-Dyn UniCrawl Nutch	Evet

VI. WEB TARAYICILARI İÇİN GÜNCEL TEKNOLOJİLERİN ANALİZİ

Yeni web teknolojilerinin ortaya çıkması, sosyal medya ağlarının yaygınlaşması ve nesnelerin interneti ile üretilen verilerin artması ile insanlığın büyük veri çağına girdiği kabul edilmektedir. Büyük veri hacim, hız, çeşitlilik, değer ve doğruluk gibi özelliklere sahip bir veri kümesidir [103]. Büyük verinin ölçeklenebilirliği, karmaşıklığı ve büyüme hızı nedeni ile klasik ilişkisel veri tabanlarında depolanması, işlenmesi ve analiz edilmesi zor bir süreçtir [104]. Web' de bulunan heterojen büyük veriler yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış olmak üzere üç farklı biçimde bulunmaktadır. Yapılandırılmış veriler önceden tanımlanmış bir veri modeline bağlı, saklanabilen, erişilebilen ve işlenebilen her türlü veri olarak adlandırılmaktadır [105]. İlişkisel veri tabanlarında saklanan veriler yapılandırılmış veriye örnek olarak gösterilebilir. Bunun aksine, yapılandırılmamış

veriler bilinmeyen yapı ve forma sahip olan ve önceden tanımlanmış bir veri modeline sahip olmayan verilerdir [106]. Metin dosyası, video, resim vb. veriler ve bunların birleşimini içeren heterojen veri kaynakları yapılandırılmamış veri olarak gösterilmektedir. Yarı yapılandırılmış veri ise her iki veri türünü de içeren bir yapıya sahiptir. Yarı yapılandırılmış veriler açık bir veri modeline sahip olmayan ve eksik olabilen düzensiz verilerdir. HTML web sayfaları, RSS verileri ve XML dosyaları yarı yapılandırılmış veri olarak temsil edilmektedir [105, 107]. Web ortamında bulunan büyük verinin toplanması ve indekslenmesi için temel araç web tarayıcılarıdır. Web tarayıcıları Web ortamında bulunan metinlerin yanı sıra resim, video, doküman vb. gibi heterojen verileri de toplamaktadırlar. Web tarayıcıları ile toplanan veriler yapılandırılmış formda ilişkisel veri tabanlarında saklanabilmektedir Yarı yapılandırılmış ve yapılandırılmamış verilerin son derece yoğun bilgi işlem ve depolama sorunları nedeni ile ilişkisel veri tabanları yetersiz kalmaktadır. Bu sorunlarla başa çıkmak için literatürde NoSQL veri tabanları ve bulut tabanlı mimariler kullanılmaktadır [108].

Web günlükleri ve sosyal medya platformları tarafından oluşturulan Web verileri geleneksel web sayfası içeriklerinden farklılık göstermektedir. Bu platformlarda zamansal olarak sürekli bir veri akışı gerçekleştirilmektedir. Bu akış verisini elde etmek için web tarayıcısının düşük gecikme süresi, yüksek veri kalitesi, uygun ağ nezaketi ve yüksek ölçeklenebilirlik gereksinimi karşılaması gerekmektedir [109]. Twitter, Flickr, Youtube vb. gibi birçok sosyal medya platformları API' leri aracılığı ile kullanıcılar ve oluşturulan içerikler ile ilgili yapılandırılmış verilere erişim izni sağlamaktadır. Bunun yanı sıra belirli bir kullanıcı topluluğunun Twitter verilerini toplayan ve analiz eden [110], Twitter akışlarında konu algılamayı ele alan [111] ve birbirine bağlı Web ve Sosyal Web verilerini entegre bir şekilde toplayan [112] çalışmalarda yapılmıştır.

Web tarayıcılarını güncel teknolojiler ile birleştiren çalışmalarda yapılmıştır. Blok zinciri (Blockchain) Nakamoto tarafından 21. Yüzyılın başlarında önerilmiştir [113]. Blok zinciri, büyük miktarda veriyi merkezi olmayan bir şekilde düzenleyebilen, mutabakat sürecini basitleştiren, veri güvenliğini sağlayan ve bilgi paylaşımını etkin bir şekilde kullanan dağıtılmış bir defter teknolojisidir. Wang ve arkadaşları blok zinciri tekniklerini kullanarak web sunucularının iş yükünü hafifletmek ve belirli kurallara göre veri toplamaya izin vermek için blok zinciri tabanlı bir odaklanmış web tarayıcısı geliştirmişlerdir [114]. Görüntü işleme konusunda Web ortamından görüntüleri toplayıp görüntü veri tabanı oluşturan web tarayıcıları geliştirilmiştir. Kalmukov ve Valova, içerik tabanlı görüntü alma tekniklerini, yöntemlerini ve algoritmaları test etmek için kullanacakları veri setini oluşturan web tarayıcısının mimarisini önermişlerdir [115]. Ali ve arkadaşları, Hadoop YARN kullanarak büyük ölçekli görüntü veri kümesi oluşturma amacı ile Web' i sistematik bir şekilde tarayan bir tarayıcı sistemi geliştirmişlerdir. Geliştirilen tarayıcı küçük resim ve simgeler gibi görüntüleri azaltmak için bazı teknikler kullanmakta ve açık kaynaklı web tarayıcısı olan Apache Hadoop ve Apache Nutch' a dayanmaktadır [116].

Web tarayıcıları konusunda önemli bir araştırma alanı da anti-tarayıcı teknolojileridir. Anti-tarayıcılar kötü niyetli kişilerin veya sistemlerin bazı teknik araçları kullanarak toplu olarak web sitesi bilgilerini elde etmelerini önlemenin bir yoludur. Tarayıcı ne kadar başarılı olursa olsun karmaşık anti-tarayıcılar tarafından keşfedilebilmektedir. Aynı şekilde anti-tarayıcılar ne kadar iyi olursa olsun gelişmiş web tarayıcıları ile bozulabilmektedir. Anti-tarayıcılar IP kısıtlamaları, kullanıcı erişim kontrolü, oturum erişim kısıtlamaları, CAPTCHA ile doğrulama gibi kısıtlamaları kullanmaktadır. Bunlara ek olarak en başarılı sistem, insanların göremediği ve asla tıklayamayacağı, yalnızca web tarayıcılarının ziyaret edebileceği web sayfalarına kasıtlı olarak bağlantı bırakan "Honeypot" teknolojisidir [117]. Tarayıcılar ve anti-tarayıcılar zıt amaçlar için çalışır ve birbirlerinin düşmanı gibidir.

VII. SONUÇ

Web tarayıcıları, web sayfalarını tarayan ve web üzerinde bulunan verileri alarak kullanıcıların ihtiyacı olan bilgileri toplayan önemli bir bilgi toplama kaynağıdır. Web tarayıcıları, Web 'de bulunan büyük hacimli heterojen verilerin toplanmasında ve kullanıcılara sunulmasında bir ihtiyaç haline gelmiştir. Kullanıcıların ihtiyacına uygun verilerin bulunması için çeşitli tarayıcılar geliştirilmiştir. Bu

çalışmada, web tarayıcıların özellikle kapsam genişletme ve tohum URL seçim ile ilgili metotları detaylı bir şekilde incelenmiştir. Ayrıca farklı web tarayıcılarının çalışma prensipleri ile performans ölçütleri incelenmiş ve literatürde yapılmış çalışmaların performansları karşılaştırılmıştır.

Genel tarayıcıların kapsamı tüm Web 'dir ve mümkün olduğunca genişlik ve derinlik öncelikli arama yaparak Web 'i taramalıdır. Tohum URL setinin belirlenmesi hayati öneme sahip olup farklı alanlardan ve bölgelerden seçilmelidir. PageRank ve HITS gibi algoritmalar uygulanarak değeri yüksek olan sayfalardan başlanılması taramanın performansını ve kalitesini arttırmaktadır. Web 'de devasa büyüklükte verinin tamamının taranması ve indekslenmesi zor olduğundan literatürde genel web tarayıcılarının dışında çeşitli tarayıcılar ile ilgili araştırmalar yapılmıştır. Belirli konu ya da konular ile ilgili taramalar için odaklanmış tarayıcılar geliştirilmiştir. Odaklı web tarayıcıların, anahtar kelime tabanlı yaklaşım, ontoloji tabanlı yaklaşım, semantik tabanlı yaklaşım, veri madenciliği tabanlı yaklaşım ve çeşitli yaklaşımlar kullanan bir tarayıcı olarak literatürde araştırmaları yapılmıştır. Odaklanmış tarayıcılarda tohum URL seçimi ontoloji tabanlı olup alanında uzman kişilerin belirlediği sayfalardan oluşmaktadır. Genel ve odaklı tarayıcılardaki en büyük problem en güncel verileri elde edememektir. Bu nedenle araştırmacılar belirli periyotlar ile sayfaları tekrar tarayan artımlı web tarayıcılarını geliştirmişlerdir. Literatürde artımlı tarayıcılar, veri madenciliğine dayalı yaklaşım, sayfa yenileme sıklığının hesaplanması ve güncelliğe dayalı yaklaşım ve diğer çeşitli yaklaşımlar ile tasarlanmaktadır. Artımlı web tarayıcıları kapsam olarak tohum URL seçiminde genel ve odaklı tarayıcıların özelliklerini taşımaktadırlar. Gizli web tarayıcıları Web yüzeyinden çok kullanıcı sorguları ile ya da yetkili girişleri ile ulaşılabilen sayfalardaki verilere elde etmek için tasarlanmıştır. Gizli web tarayıcıları, ağaç tabanlı yaklaşım, alana özgü yaklaşım, sorgu ve form tabanlı yaklaşım ve diğer çeşitli yaklaşımlar kullanılarak tasarlanmıştır. Gizli web tarayıcılarında hedef sayfalar belli olup kapsam olarak bu sayfalardaki veriler alınmaktadırlar. Mobil web tarayıcıları temek kullanım amacı tarayıcının yükünü azaltmak ve bant genişliğini boşa harcamamak olduğu için sunucular üzerinde sayfa seçim ve filtreleme işlemleri yapmaktadırlar. Kapsam belirli bir konu ya da bir veya birkaç sunucudan oluşmaktadır. Tohum URL'ler ise konu ile ilgili sayfalar arasından genelde uzmanlar tarafından seçilmektedir. Dağıtılmış web tarayıcılarında ise temel amaç farklı bölgelerde ve aynı anda tarama yapma kabiliyeti kazanmaktır. Araştırmacılar, dağıtılmış web tarayıcılarını, veri madenciliği tabanlı, ölçeklenebilirlik tabanlı, harita azaltma tabanlı ve diğer çeşitli yaklaşımlar ile tasarlamıştır. Çalışma prensibi, tohum URL seçimi, kapsamı ve temel yapısı geleneksel web tarayıcısına benzemektedir.

Bu alanda araştırma yapacak olan araştırmacılar, çalışma alanlarının kapsamına göre hibrid bir tarayıcı geliştirebilir. Tüm Web 'in taranması için güncel, düşük ağ trafiğini kullanan ve farklı bölgelerde çalışabilen genel artımlı ve dağıtılmış bir web tarayıcısı önerilmektedir. Konu odaklı taramalar için odaklanmış, artımlı, gizli, mobil ve dağıtılmış tarayıcı özelliklerini bir arada kullanarak yüksek performanslı tarayıcı oluşturulabilir.

Bu açıklamalara dayanarak gelecekte, hedefi tüm Web olan genel ve artımlı bir web tarayıcısı tasarlanacaktır. Genel ve artımlı web tarayıcılarının avantajlarından yararlanan ve büyük verinin üç ana özelliği olan hacim, çeşitlilik ve hızı dikkate alan bir tarayıcının tohum URL seçim ve kapsam genişletme metotları üzerine çalışmalar yapılacaktır.

TEŞEKKÜR: TÜBİTAK BİDEB 2244 Sanayi Doktora Programı kapsamında 118C127 proje numarası ile desteklenmiştir.

VIII. KAYNAKLAR

[1] S. Stergiou and K. Tsioutsiouliklis, "Set Cover at Web Scale," presented at the Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 2015. [Online]. Available: https://doi.org/10.1145/2783258.2783315.

[2] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," in *SODA*, 1998, vol. 98: Citeseer, pp. 668-677.

[3] S. Zheng, P. Dmitriev, and C. L. Giles, "Graph based crawler seed selection," 2009, pp. 1089-1090.

[4] P. Dmitriev, "Host-based seed selection algorithm for web crawlers," ed: Google Patents, 2010.

[5] S. Daneshpajouh, M. M. Nasiri, and M. Ghodsi, "A Fast Community Based Algorithm for Generating Web Crawler Seeds Set," 2008, pp. 98-105.

[6] S. Sharma and A. Bhagat, "Automation of Manual Seed URLs Cull Approach for Web Crawlers," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. Volume-8, no. Issue-4, February 2019.

[7] B. Ganguly and R. Sheikh, "A review of focused web crawling strategies," *International Journal of Advanced Computer Research*, vol. 2, no. 4, p. 261, 2012.

[8] F. M. J. M. Shamrat, Z. Tasnim, A. K. M. S. Rahman, N. I. Nobel, and S. A. Hossain, "An effective implementation of web crawling technology to retrieve data from the world wide web (WWW)," *International Journal of Scientific & Technology Research*, vol. 9, no. 01, pp. 1252-1256, 2020.

[9] L. Jiang and H. Zhang, "Multi-agent based individual web spider system," 2010: IEEE, pp. 177-181.

[10] S.-B. Chan and H. Yamana, "The method of improving the specific language focused crawler," 2010.

[11] J. Choudhary and D. Roy, "Priority based semantic web crawler," *International Journal of Computer Applications*, vol. 81, no. 15, pp. 10-13, 2013.

[12] P. N. Priyatam, A. Dubey, K. Perumal, S. Praneeth, D. Kakadia, and V. Varma, "Seed selection for domain-specific search," 2014, pp. 923-928.

[13] L. M. Sanagavarapu, S. Sarangi, and V. Varma, "Fine grained approach for domain specific seed URL extraction," 2018.

[14] R. Janbandhu, P. Dahiwale, and M. M. Raghuwanshi, "Analysis of web crawling algorithms," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 3, pp. 488-492, 2014.

[15] G. Gossen, E. Demidova, and T. Risse, "The iCrawl Wizard–supporting interactive focused crawl specification," 2015: Springer, pp. 797-800.

[16] A. C. Nwala, M. C. Weigle, and M. L. Nelson, "Scraping SERPs for archival seeds: it matters when you start," 2018, pp. 263-272.

[17] M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta, "The WaCky Wide Web: A collection of very large linguistically processed web-crawled corpora," *Language Resources and Evaluation*, vol. 43, pp. 209-226, 09/01 2009, doi: 10.1007/s10579-009-9081-4.

[18] H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov, "IRLbot: scaling to 6 billion pages and beyond," *ACM Transactions on the Web (TWEB)*, vol. 3, no. 3, pp. 1-34, 2009.

[19] M. Baker and M. Akcayol, "Priority queue based estimation of importance of web pages for web

crawlers," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, pp. 330-342, 2017.

[20] M. Thangaraj and P. G. Sivagaminathan, "An Improved Generic Crawler using Poisson Fit Distribution," *Communications*, vol. 6, pp. 7-13, 2016.

[21] A. Heydon and M. Najork, "Mercator: A scalable, extensible Web crawler," *World Wide Web*, vol. 2, no. 4, pp. 219-229, 1999/12/01 1999, doi: 10.1023/A:1019213109274.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, 1999.

[23] S. Chakrabarti, M. Berg, and B. Dom, "Focused crawling: A new approach to topic-specific Web resource discovery," *Computer Networks*, vol. 31, pp. 1623-1640, 04/13 2000, doi: 10.1016/S1389-1286(99)00052-3.

[24] A. Gupta and P. Anand, *Focused web crawlers and its approaches*. 2015, pp. 619-622.

[25] S. Batsakis, E. G. M. Petrakis, and E. Milios, "Improving the performance of focused web crawlers," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 1001-1013, 2009/10/01/ 2009, doi: https://doi.org/10.1016/j.datak.2009.04.002.

[26] M. S. Safran, A. Althagafi, and D. Che, "Improving Relevance Prediction for Focused Web Crawlers," in *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, 30 May-1 June 2012 2012, pp. 161-166, doi: 10.1109/ICIS.2012.61.

[27] G. H. Agre and N. V. Mahajan, "Keyword focused web crawler," in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, 26-27 Feb. 2015 2015, pp. 1089-1092, doi: 10.1109/ECS.2015.7124749.

[28] S. Age, T. Indorkar, S. Kokate, and M. Shitole, "A Self Adaptive Semantic Focused Web Crawler," *International Journal of Research In Science & Engineering*, vol. 1, no. 6, pp. 74-79, 2017.

[29] M. Kumar, A. Bindal, R. Gautam, and R. Bhatia, "Keyword query based focused Web crawler," *Procedia Computer Science*, vol. 125, pp. 584-590, 2018/01/01/ 2018, doi: https://doi.org/10.1016/j.procs.2017.12.075.

[30] S. Mali and B. B. Meshram, "Focused web crawler with revisit policy," 2011, pp. 474-479.

[31] M. S. Safran, A. Althagafi, and D. Che, "Improving relevance prediction for focused Web crawlers," 2012: IEEE, pp. 161-166.

[32] D. Taylan, M. Poyraz, S. Akyokuş, and M. C. Ganiz, "Intelligent focused crawler: learning which links to crawl," 2011: IEEE, pp. 504-508.

[33] M. S. Safran, A. Althagafi, and D. Che, "Improving relevance prediction for focused Web crawlers," 2012 2012: IEEE, pp. 161-166.

[34] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199-220, 1993.

[35] D. Mukhopadhyay, A. Biswas, and S. Sinha, "A New Approach to Design Domain Specific Ontology Based Web Crawler," in *10th International Conference on Information Technology (ICIT 2007)*, 17-20 Dec. 2007 2007, pp. 289-291, doi: 10.1109/ICIT.2007.20.

[36] M. Ehrig and A. Maedche, "Ontology-focused crawling of web documents," 2003, pp. 1174-1178.

[37] G. Agre and S. Dongre, "A keyword focused web crawler using domain engineering and ontology," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no.

3, pp. 463-465, 2015.

[38] Y. Du, Y. Hai, C. Xie, and X. Wang, "An approach for selecting seed URLs of focused crawler based on user-interest ontology," *Applied Soft Computing*, vol. 14, pp. 663-676, 2014/01/01/ 2014, doi: https://doi.org/10.1016/j.asoc.2013.09.007.

[39] Y. B. Yu, S. L. Huang, N. Tashi, H. Zhang, F. Lei, and L. Y. Wu, "A survey about algorithms utilized by focused web crawler," *Journal of Electronic Science and Technology*, vol. 16, pp. 129-138, 06/01 2018, doi: 10.11989/JEST.1674-862X.70116018.

[40] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Vips: a vision-based page segmentation algorithm," 2003.

[41] J. Wu and K. Aberer, *Using SiteRank for Decentralized Computation of Web Document Ranking*. 2004.

[42] C. Kohlschütter and W. Nejdl, "A densitometric approach to web page segmentation," presented at the Proceedings of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA, 2008. [Online]. Available: https://doi.org/10.1145/1458082.1458237.

[43] K. S. S. Prabha, C. Mahesh, and S. P. Raja, "An Enhanced Semantic Focused Web Crawler Based on Hybrid String Matching Algorithm," *Cybernetics and Information Technologies*, vol. 21, no. 2, pp. 105-120, 2021.

[44] W. Wang, X. Chen, Y. Zou, H. Wang, and Z. Dai, "A Focused Crawler Based on Naive Bayes Classifier," in *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, 2-4 April 2010 2010, pp. 517-521, doi: 10.1109/IITSI.2010.30.

[45] L. Ying, X. Zhou, J. Yuan, and Y. Huang, *A Novel Focused Crawler Based on Breadcrumb Navigation*. 2012, pp. 264-271.

[46] N. Luo, W. L. Zuo, F. Y. Yuan, and C. L. Zhang, "A new method for focused crawler cross tunnel," in *Rough Sets and Knowledge Technology, Proceedings*, vol. 4062, 2006, ch. 1st International Conference on Rough Sets and Knowledge Technology, pp. 632-637.

[47] P. Bedi, A. Thukral, H. Banati, A. Behl, and V. Mendiratta, "A Multi-Threaded Semantic Focused Crawler," *Journal Of Computer Science And Technology,* vol. 27, no. 6, pp. 1233-1242, NOV 2012, doi: 10.1007/s11390-012-1299-8.

[48] N. Le Huy Hien, T. Tien, and N. V.H, "Web Crawler: Design And Implementation For Extracting Article-Like Contents," *Cybernetics and Physics*, vol. 9, pp. 144-151, 11/20 2020, doi: 10.35470/2226-4116-2020-9-3-144-151.

[49] D. k. Sharma and M. A. Khan, "SAFSB: A self-adaptive focused crawler," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, 4-5 Sept. 2015 2015, pp. 719-724, doi: 10.1109/NGCT.2015.7375215.

[50] H. Dong and F. K. Hussain, "Self-Adaptive Semantic Focused Crawler for Mining Services Information Discovery," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1616-1626, 2014, doi: 10.1109/TII.2012.2234472.

[51] Q. Zhu, "An Algorithm OFC for the Focused Web Crawler," in *2007 International Conference on Machine Learning and Cybernetics*, 19-22 Aug. 2007 2007, vol. 7, pp. 4059-4063, doi: 10.1109/ICMLC.2007.4370856.

[52] G. A. F. Alfarisy and F. A. Bachtiar, "Focused web crawler for Indonesian recipes," in *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, 24-25 Nov. 2017 2017, pp. 196-202, doi: 10.1109/SIET.2017.8304134.

[53] T. Suebchua, A. Rungsawang, and H. Yamana, "Adaptive Focused Website Segment Crawler," in *2016 19th International Conference on Network-Based Information Systems (NBiS)*, 7-9 Sept. 2016 2016, pp.

181-187, doi: 10.1109/NBiS.2016.5.

[54] J. Hernandez, H. M. Marin-Castro, and M. Morales-Sandoval, "A Semantic Focused Web Crawler Based on a Knowledge Representation Schema," *Applied Sciences*, vol. 10, no. 11, 2020, doi: 10.3390/app10113837.

[55] J. Cho and H. Garcia-Molina, "Estimating frequency of change," *ACM Transactions on Internet Technology (TOIT)*, vol. 3, no. 3, pp. 256-290, 2003.

[56] S. Sharma and P. Gupta, "The anatomy of web crawlers," in *International Conference on Computing, Communication & Automation*, 15-16 May 2015 2015, pp. 849-853, doi: 10.1109/CCAA.2015.7148493.

[57] M. Singh and B. Varnica, "Web crawler: Extracting the web data," *International Journal of Computer Trends and Technology*, vol. 13, no. 3, pp. 132-137, 2014.

[58] A. Gupta and A. Dixit, "A novel user trend-based priority assigner and URL scheduler for dynamic incremental crawling," *Concurrency and Computation: Practice and Experience*, https://doi.org/10.1002/cpe.6555 vol. n/a, no. n/a, p. e6555, 2021/08/08 2021, doi: https://doi.org/10.1002/cpe.6555.

[59] G. Pavai and T. V. Geetha, "Improving the freshness of the search engines by a probabilistic approach based incremental crawler," *Information Systems Frontiers*, vol. 19, no. 5, pp. 1013-1028, 2017/10/01 2017, doi: 10.1007/s10796-016-9701-7.

[60] A. S. R. Santos, C. R. de Carvalho, J. M. Almeida, E. S. de Moura, A. S. da Silva, and N. Ziviani, "A genetic programming framework to schedule webpage updates," *Information Retrieval Journal*, vol. 18, no. 1, pp. 73-94, 2015.

[61] Q. Tan and P. Mitra, "Clustering-based incremental web crawling," *ACM Transactions on Information Systems (TOIS)*, vol. 28, no. 4, pp. 1-27, 2010.

[62] Z. Shi, M. Shi, and W. Lin, "The Implementation of Crawling News Page Based on Incremental Web Crawler," in *2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD)*, 12-14 Dec. 2016 2016, pp. 348-351, doi: 10.1109/ACIT-CSII-BCD.2016.073.

[63] Y. Nagar and N. Singhal, "A users search history based approach to manage revisit frequency of an Incremental Crawler," *International Journal of Computer Applications*, vol. 63, no. 3, 2013.

[64] M. Pavkovic and J. Protic, "SInFo – Structure-Driven Incremental Forum Crawler That Optimizes User-Generated Content Retrieval," *IEEE Access*, vol. 7, pp. 126941-126961, 2019, doi: 10.1109/ACCESS.2019.2939872.

[65] R. Madaan, A. Dixit, A. K. Sharma, and K. K. Bhatia, "A framework for incremental hidden web crawler," *International Journal on Computer Science and Engineering*, vol. 2, no. 3, pp. 753-758, 2010.

[66] C. Bouras, V. Poulopoulos, and A. Thanou, "Creating a polite adaptive and selective incremental crawler," in *IADIS International Conference 2005*, 2005 2005, vol. 1: Citeseer, pp. 307-314.

[67] M. Kumar, R. Bhatia, and D. Rattan, "A survey of Web crawlers for information retrieval," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 6, p. e1218, 2017.

[68] P. Zerfos, J. Cho, and A. Ntoulas, "Downloading textual hidden web content through keyword queries," in *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '05)*, 7-11 June 2005 2005, pp. 100-109, doi: 10.1145/1065385.1065407.

[69] S. Kaur and G. Geetha, "SIMHAR - Smart Distributed Web Crawler for the Hidden Web Using

SIM+Hash and Redis Server," *IEEE Access*, vol. 8, pp. 117582-117592, 2020, doi: 10.1109/ACCESS.2020.3004756.

[70] S. Gupta and K. K. Bhatia, "HiCrawl: A Hidden Web Crawler for Medical Domain," in *2013 International Symposium on Computational and Business Intelligence*, 24-26 Aug. 2013 2013, pp. 152-157, doi: 10.1109/ISCBI.2013.39.

[71] K. K. Bhatia, A. K. Sharma, and R. Madaan, "AKSHR: A novel framework for a Domain-specific Hidden Web Crawler," in *2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010)*, 28-30 Oct. 2010 2010, pp. 307-312, doi: 10.1109/PDGC.2010.5679916.

[72] S. Raghavan and H. Garcia-Molina, "Crawling the hidden web," Stanford, 2000.

[73] P. Liakos, A. Ntoulas, A. Labrinidis, and A. Delis, "Focused crawling for the hidden web," *World Wide Web*, vol. 19, no. 4, pp. 605-631, 2016/07/01 2016, doi: 10.1007/s11280-015-0349-x.

[74] M. Kumar and R. Bhatia, "Design of a mobile Web crawler for hidden Web," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 3-5 March 2016 2016, pp. 186-190, doi: 10.1109/RAIT.2016.7507899.

[75] Y. Li, Y. Wang, and J. Du, "E-FFC: an enhanced form-focused crawler for domain-specific deep web databases," *Journal of Intelligent Information Systems*, vol. 40, no. 1, pp. 159-184, 2013.

[76] A. I. El-desouky, H. A. Ali, and S. M. El-ghamrawy, "An Automatic Label Extraction Technique for Domain-Specific Hidden Web Crawling (LEHW)," in *2006 International Conference on Computer Engineering and Systems*, 5-7 Nov. 2006 2006, pp. 454-459, doi: 10.1109/ICCES.2006.320490.

[77] L. Jiang, Z. Wu, Q. Zheng, and J. Liu, *Learning Deep Web Crawling with Diverse Features*. 2009, pp. 572-575.

[78] T. A. Patil and S. Chobe, "Web Crawler for Searching Deep Web Sites," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 17-18 Aug. 2017 2017, pp. 1-5, doi: 10.1109/ICCUBEA.2017.8463648.

[79] Q. Zheng, Z. Wu, X. Cheng, L. Jiang, and J. Liu, "Learning to crawl deep web," *Information Systems*, vol. 38, no. 6, pp. 801-819, 2013/09/01/ 2013, doi: https://doi.org/10.1016/j.is.2013.02.001.

[80] S. Anbukodi and K. M. Manickam, "Reducing web crawler overhead using mobile crawler," in *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, 23-24 March 2011 2011, pp. 926-932, doi: 10.1109/ICETECT.2011.5760252.

[81] R. Nath and S. Bal, "A novel mobile crawler system based on filtering off non-modified pages for reducing load on the network," *Int. Arab J. Inf. Technol.*, vol. 8, no. 3, pp. 272-279, 2011.

[82] H. Takeno, M. Muto, N. Fujimoto, and K. Hagihara, "Developing a Web Crawler for Massive Mobile Search Services," in *7th International Conference on Mobile Data Management (MDM'06)*, 10-12 May 2006 2006, pp. 44-44, doi: 10.1109/MDM.2006.69.

[83] Y. Li, Y. Wang, and E. Tian, "A New Architecture of an Intelligent Agent-Based Crawler for Domain-Specific Deep Web Databases," in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 4-7 Dec. 2012 2012, vol. 1, pp. 656-663, doi: 10.1109/WI-IAT.2012.103.

[84] Y. J. Du, Y. Xu, and M. Wang, "A Novel Cooperatıon And Competıtıon Strategy Among Multı-Agent Crawlers " *Computıng And Informatıcs,* vol. 35, no. 5, pp. 1050-1078, 2016.

[85] S. Deshmukh and K. Vishwakarma, "A Survey on Crawlers used in developing Search Engine," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 6-8 May 2021 2021, pp. 1446-1452, doi: 10.1109/ICICCS51141.2021.9432368.

[86] V. Shkapenyuk and T. Suel, "Design and implementation of a high-performance distributed web crawler," 2002: IEEE, pp. 357-368.

[87] J. F. Cai and H. Zhang, "Dis-Dyn Crawler: A Distributed Crawler for Dynamic Web Page," presented at the Proceedıngs Of The 4th Internatıonal Conference On Mechatronıcs, Materıals, Chemıstry And Computer Engıneerıng 2015 (ICMMCCE 2015), 2015.

[88] J. K. Yu, M. R. Li, and D. Y. Zhang, "A Distributed Web Crawler Model based on Cloud Computing," presented at the Proceedıngs Of The 2nd Informatıon Technology And Mechatronıcs Engıneerıng Conference (ITOEC 2016), 2016.

[89] D. L. Quoc, C. Fetzer, P. Felber, R. É, V. Schiavoni, and P. Sutra, "UniCrawl: A Practical Geographically Distributed Web Crawler," in *2015 IEEE 8th International Conference on Cloud Computing*, 27 June-2 July 2015 2015, pp. 389-396, doi: 10.1109/CLOUD.2015.59.

[90] Q. Pu, "The Design and Implementation of a High-Efficiency Distributed Web Crawler," in *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 8-12 Aug. 2016 2016, pp. 100-104, doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.34.

[91] X. X. Liu and Z. P. Jin, "ChainMR Crawler: A Distributed Vertical Crawler Based on MapReduce," presented at the Securıty, Prıvacy And Anonymıty In Computatıon, Communıcatıon And Storage (SPACCS 2016), 2016.

[92] P. Boldi, B. Codenotti, M. Santini, and S. Vigna, "Ubicrawler: A scalable fully distributed web crawler," *Software: Practice and Experience*, vol. 34, no. 8, pp. 711-726, 2004.

[93] S. K. Bal and G. Geetha, "Smart distributed web crawler," in *2016 International Conference on Information Communication and Embedded Systems (ICICES)*, 25-26 Feb. 2016 2016, pp. 1-5, doi: 10.1109/ICICES.2016.7518893.

[94] M. E. ElAraby, H. M. Moftah, S. M. Abuelenin, and M. Z. Rashad, "Elastic Web crawler service-oriented architecture over cloud computing," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 8111-8126, 2018.

[95] D. Gunawan, A. Amalia, and A. Najwan, "Improving data collection on article clustering by using distributed focused crawler," *Data Science: Journal of Computing and Applied Informatics*, vol. 1, no. 1, pp. 1-12, 2017.

[96] H. T. Yani Achsan and W. C. Wibowo, "A Fast Distributed Focused-Web Crawling," *Annals of DAAAM & Proceedings*, vol. 24, no. 1, 2013.

[97] C. Tsai, T. Ku, P. Yang, and M. Chen, "A distributed multi-tasking job scheduling mechanism for web crawlers," in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, 11-14 Aug. 2014 2014, pp. 243-248, doi: 10.1109/SOCPAR.2014.7008013.

[98] Y. Shi and T. Zhang, "Design and implementation of a scalable distributed web crawler based on Hadoop," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, 10-12 March 2017 2017, pp. 537-541, doi: 10.1109/ICBDA.2017.8078691.

[99] K. P. Zhu, Z. M. Xu, X. L. Wang, and Y. M. Zhao, "A full distributed Web crawler based on structured network," presented at the Informatıon Retrıeval Technology, 2008.

[100] L. Fei, F. Y. Ma, Y. M. Ye, M. L. Li, and J. D. Yu, "Distributed high-performance web crawler based on peer-to-peer network," in *Parallel And Dıstrıbuted Computıng: Applıcatıons And Technologıes, Proceedıngs*, vol. 3320, 2004, pp. 50-53.

[101] F. Ye, Z. Jing, Q. Huang, C. Hu, and Y. Chen, "The Research and Implementation of a Distributed

Crawler System Based on Apache Flink," in *Algorithms and Architectures for Parallel Processing*, Cham, T. Hu, F. Wang, H. Li, and Q. Wang, Eds., 2018// 2018: Springer International Publishing, pp. 90-98.

[102] L. Su and F. Wang, "Web crawler model of fetching data speedily based on Hadoop distributed system," 2016: IEEE, pp. 927-931.

[103] B. Marr, "Why Only One of the 5 Vs of Big Data Really Matters IBM Big Data & Analytics Hub: IBM. 2015," ed.

[104] U. R. Pol, "Big data analysis using Hadoop MapReduce," *Am. J. Eng. Res. AJER*, vol. 5, pp. 146-151, 2016.

[105] A. C. Eberendu, "Unstructured Data: an overview of the data of Big Data," *International Journal of Computer Trends and Technology*, vol. 38, no. 1, pp. 46-50, 2016.

[106] Y. Zhao and J. Chen, "A survey on differential privacy for unstructured data content," *ACM Computing Surveys (CSUR)*, vol. 54, no. 10s, pp. 1-28, 2022.

[107] L. Zhang, N. Li, and Z. Li, "An Overview on Supervised Semi-structured Data Classification," in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, 6-9 Oct. 2021 2021, pp. 1-10, doi: 10.1109/DSAA53316.2021.9564205.

[108] M. Bahrami, M. Singhal, and Z. Zhuang, "A cloud-based web crawler architecture," in *2015 18th International Conference on Intelligence in Next Generation Networks*, 17-19 Feb. 2015 2015, pp. 216-223, doi: 10.1109/ICIN.2015.7073834.

[109] M. Hurst and A. Maykov, "Social Streams Blog Crawler," in *2009 IEEE 25th International Conference on Data Engineering*, 29 March-2 April 2009 2009, pp. 1615-1618, doi: 10.1109/ICDE.2009.146.

[110] M. Boanjak, E. Oliveira, J. Martins, E. Mendes Rodrigues, and L. Sarmento, "TwitterEcho - A distributed focused crawler to support open research with twitter data," 04/16 2012, doi: 10.1145/2187980.2188266.

[111] F. Psallidas, A. Ntoulas, and A. Delis, "Soc web: Efficient monitoring of social network activities," in *International Conference on Web Information Systems Engineering*, 2013: Springer, pp. 118-136.

[112] G. Gossen, E. Demidova, and T. Risse, "iCrawl: Improving the Freshness of Web Collections by Integrating Social Web and Focused Web Crawling," presented at the Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries, Knoxville, Tennessee, USA, 2015. [Online]. Available: https://doi.org/10.1145/2756406.2756925.

[113] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[114] J. Wang, W. Zhu, J. Lai, and Z. Wang, "FDataCollector: A Blockchain Based Friendly Web Data Collection System," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, 13-15 Dec. 2021 2021, pp. 732-739, doi: 10.1109/MSN53354.2021.00115.

[115] Y. Kalmukov and I. Valova, "Design and development of an automated web crawler used for building image databases," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 20-24 May 2019 2019, pp. 1553-1558, doi: 10.23919/MIPRO.2019.8756790.

[116] A. Ali, R. Ali, A. M. Khatak, and M. S. Aslam, "Large Scale Image Dataset Construction Using Distributed Crawling with Hadoop YARN," in *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*, 5-8 Dec. 2018 2018, pp. 394-399, doi: 10.1109/SCIS-ISIS.2018.00075.

[117] F. Zhou and Y. Wang, "Exploring The Role of Web Crawler and Anti-Crawler Technology in Big Data Era," in *2022 11th International Conference of Information and Communication Technology (ICTech))*, 4-6 Feb. 2022 2022, pp. 316-319, doi: 10.1109/ICTech55460.2022.00070.