Marmara Fen Bilimleri Dergisi 2018, 4: 339-357
DOI: 10.7240/marufbd.440095

RESEARCH ARTICLE / ARAŞTIRMA MAKALESİ

Tavsiye Sistemlerinde Büyük Verinin Kullanımı Üzerine Kapsamlı Bir İnceleme

A Comprehensive Review on the Use of Big Data in Recommendation Systems

Anıl UTKU[1], **M. Ali AKCAYOL**[1]

[1]*Gazi Üniversitesi, Mühendislik fakültesi, Bilgisayar Mühendisliği Bölümü, Maltepe, ANKARA*

Öz

Web tabanlı e-ticaret platformlarındaki gelişmeler, tavsiye sistemlerinin giderek önem kazanmasına neden olmaktadır. Tavsiye sistemleri, kullanıcılar için faydalı ve kişiselleştirilmiş öneriler sunmak için geliştirilen sistemlerdir. Büyük veri çağında, artan sayıda kullanıcı ve ürün karşısında mevcut tavsiye sistemleri ölçeklenebilirlik ve verimlilik sorunları yaşamaktadır. Bu çalışma kapsamında, büyük veri ve tavsiye sistemleri üzerine kapsamlı ve karşılaştırmalı bir inceleme yapılmıştır. Literatürde büyük verinin tavsiye sistemlerinde kullanıldığı çalışmalar incelenmiş, büyük verinin tavsiye sistemlerine yüksek performans ve başarı ile uygulanabilmesi için gerekli önişlemler ve yöntemler detaylı bir şekilde incelenmiştir.

Anahtar Kelimeler: Büyük Veri, Akan Veri, Tavsiye Sistemleri

Abstract

Developments in Web-based e-commerce platforms are making recommendation systems increasingly important. Recommendation systems are developed to provide useful and personalized recommendations for users. In big data era, existing recommendation systems are experiencing scalability and productivity challenges in an increasing number of users and products. In this work, a comprehensive and comparison review of big data and recommendation systems was conducted. In the literature, the studies used in big data and recommendation systems have been examined and necessary precautions and methods have been analyzed in detail in order to be able to apply large data to recommendation systems with high performance and success.

Keywords: Big Data, Stream Data, Recommendation Systems

I. GİRİŞ

Günümüzde, donanım ve yazılım teknolojilerindeki gelişmelerle birlikte veri toplama işlemleri kolay ve sürekli yapılabilir bir hale gelmiştir. Kredi kartı işlemleri, mobil cihazlar ya da kişisel bilgisayarlar kullanılarak Web üzerinde gerçekleştirilen günlük işlemler, otomatik olarak depolanabilmektedir. Benzer şekilde bilgi teknolojilerindeki gelişmeler IP ağları arasında büyük miktarda akan verinin oluşmasını sağlamaktadır. Büyük miktardaki bu veri, çeşitli uygulamalardan farklı örüntülerin çıkarılması için kullanılabilmektedir [1].

İnternet ortamında hızın giderek daha fazla önem kazandığı günümüzde geleneksel tavsiye sistemleri, kullanıcı tercihlerindeki değişikliklere hızlı bir şekilde yanıt verememekte ve kullanıcıların ilgi alanlarını gerçek zamanlı bir şekilde yakalayamamaktadır. Büyük veri çağında, tavsiye sistemlerinin tepki sürelerinin milisaniyeler cinsinden ifade edilmesi ve yüksek hesaplama karmaşıklıkları ile baş edebilmesi gerekmektedir. Örnek olarak "Son on saniye boyunca, bir reklama Pekin' den tıklayan ve yaşı 20 ile 30 arasında olan erkek kullanıcıların oranı nedir?" sorgusu düşünüldüğünde bölge, yaş, cinsiyet ve reklam olmak üzere dört boyutun birleşiminin hesaplanması gerekmektedir. Bu gibi sorgular büyük akan verilerde yüksek hesaplama maliyetlerine neden olmaktadır [2].

Sorumlu Yazar/Corresponding Author: Anıl UTKU, **Tel:** 0312 582 31 30**, e-posta:** anilutku@gazi.edu.tr
Gönderilme/Submitted: 03.07.2018, **Düzenleme/Revised:** 01.11.2018, **Kabul/Accepted:** 06.11.2018

Günümüzde gelişmekte olan uygulamalarda veriler sonlu bir biçimde depolanan veri kümelerinden ziyade akan veri şeklinde sürekli bir şekilde elde edilmektedir. Bu gibi akan verilere borsa verileri, ağ trafiği ölçümleri, Web sunucusu kayıtları, Web üzerindeki tıklama akışları, sensör ağlarından elde edilen ölçümler ve mobil iletişim kayıtları örnek olarak verilebilir. Akan veriler üzerinde gerçekleştirilen işlemler, akan verilerin boyutlarının zamanla artmasından ve sorgulara yanıt sürelerinin kısa olması gerektiği için geleneksel veri madenciliği yaklaşımlarından farklılık göstermektedir. Bu sebeple akan verileri bütünüyle depolamak ve sorgulara yanıt verebilmek için akan verilerin tamamının taranması mümkün değildir [3].

Arka plandaki verinin hacmi çok büyük olduğu için bir takım araştırma ve hesaplama zorlukları ortaya çıkmaktadır. Verilerin hacimleri arttıkça, aynı veri üzerinden birçok kez geçiş yapılarak etkili bir şekilde işlenmesi mümkün olmamaktadır. Bir veri öğesinin en fazla bir defa işlenebilmesi, arka plandaki algoritmaların uygulanışı üzerinde kısıtlamalara yol açmaktadır. Bu nedenle akan veri madenciliğinde kullanılacak algoritmaların, veriler üzerinden tek bir geçiş ile çalışacak şekilde tasarlanması gerekmektedir [4]. Çoğu durumda veriler zaman içinde gelişebileceği için bu işleminin doğal bir zamansal bileşeni vardır. Akan verilerin bu davranışı zamansal lokalite olarak adlandırılmaktadır. Bu nedenle, tek geçişli akan veri madenciliği algoritmalarının basit bir şekilde uyarlanması görev için etkili bir çözüm olmayabilir. Akan veri madenciliği algoritmalarının, arka plandaki verilerin gelişimine odaklanılarak dikkatle bir şekilde tasarlanması gerekmektedir [5].

Tavsiye sistemleri, İnternet ortamında elektronik ticaret uygulamalarının yaygınlaşması ile birlikte giderek popüler bir hale gelmiştir. Ancak geliştirilen tavsiye sistemlerinin büyük bir bölümü gerçek zamanlı olarak tasarlanmamıştır. Mevcut sistemler, büyük ölçekli sistemlerde etkili bir şekilde çalışamadığı için çevrimiçi ortamlarda gerçek zamanlı etkileşimler kurulamamaktadır. Büyük veri algoritmalarının tavsiye sistemlerine uygulanmasındaki temel sorun bellek üzerinde yapılan işlemler için depolama alanının oldukça sınırlı olmasıdır. Ayrıca büyük veriler kullanıcılar üzerindeki bilişsel yükü artırarak, tavsiye sistemleri için ölçeklenebilirlik ve kullanıcı memnuniyeti problemlerinin yaşanmasına neden olmaktadır. Bu sebeple tavsiye sistemlerinin performansını düşürmeden daha büyük veri setlerindeki artan kullanıcı ve ürün sayıları ile hesaplama maliyetlerine karşı ölçeklenebilir bir yapıya sahip olması gereklidir.

Bu çalışma kapsamında, büyük verilerin tavsiye sistemlerinde uygulanabilirliği üzerine araştırmalar yapılmıştır. Büyük verilerin tavsiye sistemlerinde kullanıldığı çalışmalar analiz edilmiş, tavsiye sistemleri büyük veri bakış açısından değerlendirilerek büyük veri ve büyük veri analizinde kullanılan yöntemler kapsamlı bir şekilde incelenmiştir. Büyük ölçekli tavsiye sistemlerinde Hadoop ve Spark gibi büyük veri teknolojilerinin uygulanmasına yönelik literatürdeki çalışmalar değerlendirilmiştir. Tavsiye sistemlerinde kullanılan yöntemler, büyük veri kaynakları ve büyük veri analizinde kullanılan teknolojiler karşılaştırmalı olarak incelenmiştir.

Bu bölümün devamında, akan veri algoritmalarının tavsiye sistemlerine uygulanmasına yönelik literatürdeki çalışmalar incelenmiştir.

Subbian ve ark. tarafından 2016 yılında yapılan çalışmada, gerçek zamanlı öneriler sunabilmek için olasılıksal komşuluk tabanlı yeni bir yöntem geliştirilmiştir. Akan verinin kullanıldığı tavsiye sistemi uygulamalarındaki varsayım, belirli bir kullanıcının tüm değerlendirmelerinin veya belirli bir öğenin tüm değerlendirmelerinin aynı anda alınamamasıdır. Tavsiye sistemi uygulamaları, aynı kaydın tüm boyutlarının her zaman eşzamanlı olarak alındığı çok boyutlu akış uygulamalarından farklıdır. Tavsiye sistemi uygulamalarında, kullanıcı herhangi bir zamanda belirli bir öğe için bir değerlendirmede bulunabilmektedir. Ayrıca, yeni kullanıcılar veya öğeler herhangi bir zamanda sisteme dâhil olabilmektedir.

Genel olarak, değerlendirmeler hiçbir zaman silinmediğinden kullanıcıların ve öğelerin sayısı zamanla artış göstermektedir. Bu sebeple, t zamanındaki kullanıcı sayısı $m(t)$, t zamandaki öğe sayısının $n(t)$ ve t zamandaki değerlendirme matrisinin boyutunun $m(t)$ x $n(t)$ olduğu kabul edilmiştir. Kullanıcıların öğeler hakkında bulundukları değerlendirmeler (*KullanıcıId*, *ÖğeId*, *Değerlendirme Puanı*) şeklinde alınmaktadır. Kullanıcı değerlendirmeleri, +1 beğenme durumunu, – 1 ise beğenmeme durumunu ifade etmek üzere binary olarak alınmıştır. Çevrimiçi uygulama alanlarında kullanıcılara sunulacak öneri listeleri ya da belirli bir öğe ile ilgilenen kullanıcıların listesi belirlenmek istenebilir. Akan veri üzerinde tek bir kullanıcı değerlendirme güncellenmesinin olması, öğeler arasındaki mesafenin yeniden hesaplanmasını gerektirdiği için klasik komşuluk tabanlı yaklaşımlarda doğru sonuçlar üretememektedir. Ayrıca değerlendirme matrisinin tümünün hafıza tutulması beklenmemektedir. Komşuluk uzaklıkları hesaplanacak i ve j öğeleri için kullanıcılar tarafından t zamanına kadar yapılmış olumlu (+1) değerlendirmeler $P(i, t)$ ve $P(j, t)$ ile olumsuz değerlendirmeler ise $N(i, t)$ ve $N(j, t)$ ile ifade edilmektedir. t zamanda, i ve j öğeleri arasındaki benzerlik Eş. 1'de görüldüğü gibi hesaplanmaktadır.

$$S^{+}(i,j,t) = \frac{P(i,t) \cap P(j,t)}{P(i,t) \cup P(j,t)} \quad (1)$$

Olumlu değerlendirmelere dayalı olarak hesaplanan bu benzerlik Jaccard indeksidir. $\alpha = |P(i,t)| + |P(j,t)$ ve $\beta = |N(i,t)| + |N(j,t)$ olmak üzere olumlu ve olumsuz değerlendirmelerin ağırlıklandırılması ile Eş. 2 elde edilmektedir.

$$S(i,j,t) = \frac{\alpha.\, S^{+}(i,j,t) + \beta.\, S^{-}(i,j,t)}{\alpha + \beta} \quad (2)$$

Kullanıcı değerlendirmelerinin tahmin edilmesi için ilk olarak belirli bir öğenin tüm öğeler ile olan benzerliği hesaplanmaktadır. Belirli bir *i* öğesine en benzer öğelerin değerlendirmelerinin ağırlıklı ortalaması, o öğe için kullanıcı değerlendirmesi olarak öngörülmektedir. $l_i(u)$, *u* kullanıcısının değerlendirmede bulunduğu *i* öğesine benzer öğelerin kümesini ve $r_{u,j}$ ise *u* kullanıcısının *j* öğesi ile ilgili değerlendirme puanını ifade etmektedir. *u* kullanıcısı ve *i* öğesi için *t* zamanındaki komşuluk tabanlı tahmin puanı Eş. 3 kullanılarak hesaplanmaktadır.

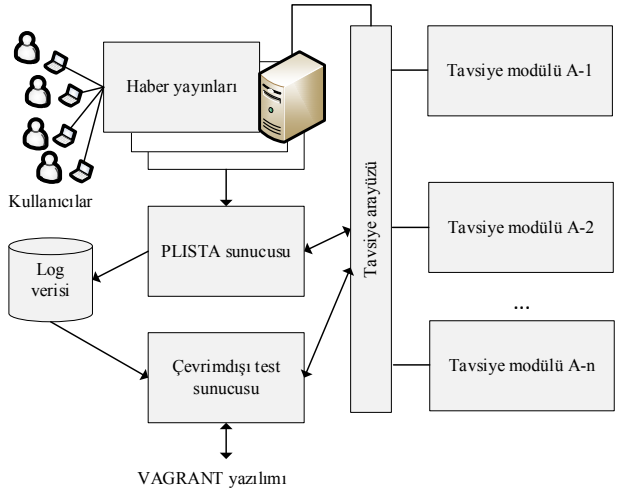
$$P_{u,i}(t) = \frac{\sum j \in I_{i(u)} (S(i,j,t).\, r_{u,j})}{\sum j \in I_{i(u)} S(i,j,t)} \quad (3)$$

Çalışma kapsamında geliştirilen olasılıksal komşuluk tabanlı yöntemler ile öğeler arasındaki benzerliklerin olasılıksal olarak hesaplandığı min-hash tekniği önerilmiştir. Benzerlikler, min-hash indeksindeki her bir kullanıcı izlenerek yaklaşık olarak hesaplanmaktadır. Temel fikir bir hash fonksiyonu kullanarak kullanıcılara sıralama düzeni uygulamaktır. Bu sıralama düzeninde *i* öğesi için pozitif değerlendirmede bulunan ilk kullanıcının, *j* öğesi için de olumlu bir değerlendirmede bulunan ilk kullanıcı olma ihtimali Jaccard indeksi ile benzerdir. $f_1(.)$.. $f_d(.)$ hash fonksiyonları, *j* öğesini olumlu olarak ve olumsuz olarak değerlendirmiş kullanıcılara uygulanmaktadır. Hash fonksiyonları uygulandıktan sonra olumlu değerlendirme yapan kullanıcılar M^{+} veri yapısında, olumsuz değerlendirme yapan kullanıcılar ise M^{-} veri yapısında tutulmaktadır. Bu veri yapılarının boyutları kullanıcı-öğe değerlendirme matrisinden daha küçük olduğu için ve kolaylıkla güncellenebildiği için bellekte muhafaza edilebilmektedir. *i* ve *j* öğeleri arasındaki benzerlik Eş. 4 kullanılarak hesaplanmaktadır.

$$S^{+}(i,j,t) \approx R^{+}(i,j,t) = \frac{\sum_{s=1}^{d} \delta(i_s = j_s)}{d} \quad (4)$$

$\delta(.)$, *i* ve *j* öğeleri benzerse 1 benzer değilse 0 değerini alan bir fonksiyondur [1].

Werner ve Lommatzsch tarafından 2015 yılında yapılan çalışmada, gerçek zamanlı haber tavsiyesi sunmak amacıyla sınırlı donanım kaynakları ve zaman kısıtlamaları konularında optimize edilmiş yeni bir yöntem önerilmiştir. PLISTA adı verilen sistem ile öneri taleplerine 100 ms içerisinde yanıt verilerek, hızlı ve verimli öneriler sunmak hedeflenmiştir. Geliştirilen sistemde bir kullanıcı bir haber portalını ziyaret ettiğinde, portal PLISTA sunucusuna bir öneri talebi göndermektedir. PLISTA sunucusu, isteği rastgele seçilen bir tavsiye algoritmasına devretmektedir. İsteğe cevap verilmesi için seçilen tavsiye algoritmasının 100 ms içinde bir öneri listesi sunması gerekmektedir. VAGRANT4 adı verilen çevrimdışı test sunucusu ile geçmişte oturum açmış olan kullanıcıların etkileşim verileri kullanılarak önerilen algoritmaların performansı analiz edilmektedir. Geliştirilen sistemin mimarisi Şekil 1' de görülmektedir.



Şekil 1. Geliştirilen sistemin mimarisi

Geliştirilen sistemin sınırlamalarından biri kullanıcılar haber portallarında kullanıcı girişi yapmadıkları için benzersiz kullanıcıların belirlenmesi sorunudur. Bu durum, kullanıcıya dayalı yöntemlerin kişiselleştirilmiş önerileri hesaplamak için kapsamlı veriler elde etmesini zorlaştırmaktadır.

Geliştirilen sistemde, haber makalelerinin sınırlı kullanım ömürleri hesaba katılarak önceden belirlenmiş zaman aralığında en çok incelenen haber makaleleri kullanıcı-öğe etkileşimi istatistiğine göre belirlenerek öneri olarak sunulmaktadır. Bu yaklaşımın altında yatan fikir, en popüler makalelerin henüz bu makaleleri görmeyen kullanıcılar için de ilgi çekici olabileceğidir. $p(a)$, *a* makalesini okuyan kullanıcı sayısı ve $t(a)$, *a* makalesinin yayınlanma zamanı olmak üzere bir makalenin popülerlik ölçüsü $r(a)$, Eş. 5'te görüldüğü gibi hesaplanmaktadır.

$$r(a)_{T,P} = \log_{10}(\max(abs(p(a)-T),1.0)) + sign(p(a)-T).\frac{\text{şimdiki zaman} + T(a)}{P} \quad (5)$$

Burada *p*, makalenin yaşının popülerlik ölçüsü üzerindeki etkisini belirlemektedir. *T* ise bir makalenin öneri değeri kazanmadan önce kaç kullanıcı tarafından okunmuş olması gerektiğini belirlemektedir. *abs* fonksiyonu ifadenin mutlak değerini, *sign* fonksiyonu ise verilen ifadenin pozitif ya da negatif olacak şekilde işaretini döndürmektedir.

Klasik işbirlikçi filtreleme yaklaşımlarına benzer olarak, ortak özelliklere göre makaleler arasındaki ilişkiler hesaplanmaktadır. Örneğin, bir kullanıcı iki makaleyi okuduysa bu makaleler ilişkili olabilir denilmektedir. Algoritma, bilinen her bir makale için bir dizi ilgili makale sunmaktadır. Bu yaklaşım ile bir kullanıcıya öneri sunulacağı zaman kullanıcının daha önce okuduğu makale ile ilgili makaleler aranarak elde edilen makaleler kümesi kullanılmaktadır.

Geliştirilen sistem, kullanıcı etkileşimlerinin en yoğun olduğu zaman durumda sistemin kaynak tüketim oranı açısından, paralel istekler, maksimum güncelleme ve gösterim sayısı ile tepki süresi açısından ve bir algoritmanın aynı donanım kaynakları ile kaç tane portala cevap verebildiğine yönelik test senaryoları ile analiz edilmiştir [6].

Ludmann tarafından 2015 yılında yapılan çalışmada, akış tabanlı tavsiye sistemleri için akan veriler üzerinde sorgular yaparak kişiselleştirilmiş öneriler kümesinin hesaplandığı yeni bir yöntem önerilmiştir. Bu sorgular ilişkisel cebir işlemlerini ve veri madenciliği işlemlerini kapsamaktadır. $U=\{u_1, u_2, \dots, u_n\}$ kullanıcılar kümesi, $I=\{i, i_2, \dots, i_m\}$ öğeler kümesi ve *T* zamanı ifade etmek üzere her bir *u* kullanıcısının *i* öğesi ile *t* zamanda yaptığı etkileşimin sonucu doğrudan ya da dolaylı olarak değerlendirme puanı şeklinde elde edilmektedir. Değerlendirme puanı, $R = \{(u, i, r, t) | (u, i, r, t) \in UxIxRxT\}$ şeklinde hesaplanmaktadır.

RecSys sorguları, geliştirilen modeli eğitmek için değerlendirme girdi akışının verileri kullanılarak eğitilen model ile her bir istek talebi için öneri listeleri oluşturulmaktadır. Değerlendirme akan verileri EXTRACT TEST DATA modülü tarafından test verileri akışı ve öğrenme verileri akışı olmak üzere iki akan veriye bölünmektedir. Öğrenme verileri, bir modeli eğitmek için TRAIN RECSYS MODEL modülü tarafından kullanılmaktadır. Her yeni öğrenme grubu ile bu operatör modeli güncellemekte ve sonraki operatörler için bir kopyasını çıkarmaktadır. Bu operatör, giden modellerin geçerlilik aralıklarını şu kurallara göre ayarlamaktadır:

- Zamanın her noktasında, geçerli bir tane model vardır. Bu, yeni bir model geçerli olduğunda önceki modelin geçersiz olması gerektiği anlamına gelmektedir.
- Modeli eğitmek için kullanılan tüm öğrenme dizileri, modelin geçerlilik aralığında olmalı ve modelin geçerlik aralığında kullanılmayan bir öğrenme grubu olmamalıdır. Ayrıca, hafızada tutulan değerlendirme sayısını sınırlamaktadır.

Her bir öneri isteğinde, RECOMM CANDIDATES modülü ile bir dizi öneri adayı belirlenmektedir. Bu adaylar genellikle kullanıcı tarafından değerlendirilmemiş öğelerdir. PREDICT RATING (PREDICT RATING) modülü, her bir öneri adayının değerlendirme puanını tahmin etmek için modelleri kullanmaktadır. RECOMMEND modülü değerlendirme sonuçlarına göre önerilmesi gereken öğeleri seçmektedir. TEST PREDICTION operatörü, RMSE gibi bir değerlendirme metriği uygulayarak tahmin edilen ve gerçek değerlendirme puanlarını karşılaştırmaktadır [7].

Lommatzsch ve Albayrak tarafından 2015 yılında yapılan çalışmada, çevrimiçi haber portallarındaki kullanıcı-öğe etkileşimlerinin analiz edilerek kalite, sağlamlık, ölçeklenebilirlik ve sıkı zaman kısıtlamaları gibi gereksinimleri karşılayacak akan veri tabanlı öneriler sunmak için optimize edilmiş yeni bir algoritma geliştirilmiştir. Haber portalları ve tartışma platformlarından günlük haber verileri alınarak analiz edilmektedir. Akışların özelliklerini ve gizli kurallarını belirleyebilmek için PLISTA yarışmasındaki (ACM Recsys News Challenge 2013) akan verileri incelenmiştir. PLISTA, araştırmacılara gerçek dünya senaryoları altında algoritmalarını değerlendirme imkânı sağlayan reklamcıların ve yayıncıların bir araya geldiği bir platformdur. PLISTA' nın amacı kullanıcılara ilgi çekici makaleler önermektir. Bir kullanıcı, haber portalındaki bir web sayfasını her ziyaret edişinde makale önerileri oluşturulmakta ve öneri sayfasına yerleştirilmektedir. Bu yarışmada, bir kullanıcı yarışma programına katılan haber sitesini ziyaret ettiğinde, PLISTA sunucusu rastgele bir tavsiye ekibi seçerek talebi bu ekibe iletmektedir. Seçilen ekibin algoritması, talebe göre değişmekle birlikte altı adet öneri sunmaktadır. Öneri talepleri, iletişim süresi de dâhil olmak üzere 100 ms içerisinde cevaplanmaktadır. Ekiplerin performansı, kullanıcılar tarafından tıklanan önerilerin sayısıyla ölçülmektedir. Yaşanan sorunlar ise kullanıcılar giriş yapmadığı için benzersiz kullanıcıları tanımlamanın zor olması ve yarışmaya katılan takımların sundukları önerilerin kullanıcı davranışlarını etkilemesidir. Diğer bir problem ise sunulan önerilere kullanıcıların geribildirimde bulunma süreleridir. Bazı kullanıcılar önerileri derhal tıklarken diğerleri günlerce bekleyebilmektedir.

Çalışma kapsamında eğitim verisi olarak ACM Recsys 2013 verileri kullanılmıştır. Eğitim verisi 84 milyon kullanıcı tıklama verisi ve yaklaşık olarak 1 milyon öneri tıklama olayından oluşmaktadır. Değerlendirme ölçütü olarak kullanıcıların sunulan önerilere tıklama oranı ve çevrimiçi analizler kullanılmıştır. Çoğu kullanıcı yalnızca Web sitesindeki makalelerle ilgilendiği ve sunulan önerilere dikkat etmediği

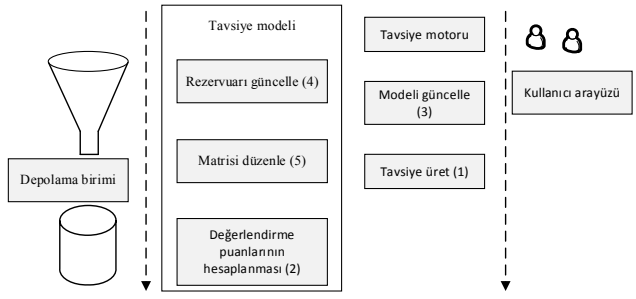
için tıklama oranı genellikle çok düşüktür. Web sayfasındaki önerilerin yerleştirilmesi, kullanıcı gruplarının alışkanlıkları ve zamanın etkisi nedeniyle, tıklama oranı büyük oranda portala bağımlıdır. Çevrimiçi değerlendirmeler tavsiye algoritmasının kullanıcı tıklamalarını ne ölçüde doğru tahmin ettiğini belirlemekte ve hassasiyet (precision) değerini hesaplamaktadır.

Gerçekleştirilen topluluk tabanlı yaklaşım ile öneri talepleri için en uygun algoritmanın seçilmesi sağlanmaktadır. Bir karar ağacı ile bağlama dayalı olarak gelen isteklerin hangi algoritma ile cevaplanacağına karar verilmektedir. Haber makalelerinin öneri olarak uygun olup olmadığının belirlenmesi için eğilim tabanlı bir yaklaşım kullanılmıştır. Bu yaklaşımda algoritmanın yakın zamanda başarılı olmasının gelecekte de başarılı olacağı varsayımı temel alınarak son 60 dakikalık veriler kullanılmıştır. Öneri listesi oluşturulurken kullanılan doldurma yaklaşımı ile sunulacak 6 elemandan oluşan öneri listesinin her elemanı için talepler birkaç farklı öneri algoritmasına iletilmektedir. Her öneri ajanı için en yüksek sırada yer alan tavsiye, sonuç kümesindeki bir öneri için kullanılmaktadır. Topluluk stratejisinin temelindeki fikir, her bir öneri algoritması hesaplama yaparken kendi özel kriterlerini kullandığından, farklı öneri algoritmalarından öneriler toplamanın çeşitliliğe yol açacağıdır.

Analiz sonuçları çevrimdışı değerlendirmelerde (sıkı zaman kısıtlamaları olmadan), topluluk stratejisinin yaklaşık olarak % 5 daha iyi öneri hassasiyetine ulaştığını göstermiştir. Topluluk stratejisinin avantajı sistemin kullanıcının davranışlarındaki değişikliklere sürekli uyum sağlamasıdır. Dezavantajı ise nihai sonuç kümesinin ancak tüm öneri algoritmaları hesaplamalarını bitirdikten sonra tamamlanmasıdır [8].

Chen ve ark. tarafından 2013 yılında yapılan çalışmada, çevrimiçi puanlama yaklaşımları genişletilerek TeRec adı verilen zamana dayalı bir tavsiye sistemi önerilmiştir. TeRec' te, kullanıcılar tweet gönderirken, gerçek zamanlı olarak ilgi alanlarına göre hashtag tavsiyeleri alabilmekte ve tavsiyeler için hızlı geribildirimler oluşturabilmektedirler. TeRec, kullanıcıların gerçek zamanlı konu önerilerine erişmesini sağlayan tarayıcı tabanlı istemci arabirimi sağlamakta ve sunucu tarafında, gerçek zamanlı akan verileri işleyip saklamaktadır. TeRec, bir akan veri ortamında çalışarak (Weibo) kullanıcılarına herhangi bir anda tercihlerine göre gerçek zamanlı tavsiyeler sağlamaktadır. TeRec kullanıcıları ve öğeleri, daha doğru sonuçlar elde edebilmek için matris çarpanlarına ayırma kullanarak modellemektedir. TeRec' in temel fikri, hashtag' leri ilginç konuların vekilleri olarak kullanmaktır. Bir kullanıcı bir tweet yayınlamak üzereyken, sistem kullanıcının mevcut ilgi alanlarını öngörmekte ve bu tweet' te kullanmak isteyebileceği çeşitli konuları (hashtag' leri) önermektedir.

TeRec, kullanıcılara tweet gönderdiklerinde kendilerine uygun hashtag' leri seçmelerine yardımcı olan tarayıcı tabanlı bir hizmet sunmaktadır. Sistemin sunucu tarafı kullanıcı tercihlerini modelleyip depolamakta, kullanıcılar ve hashtag' ler arasında değerlendirme puanı tahminlerini hesaplamakta ve kullanıcılar tweet attıklarında hashtag öneri listesinin oluşturulmasını sağlamaktadır. Geliştirilen sistem Şekil 2' de görüldüğü gibi üç katmandan oluşmaktadır. Birinci katman, öneri sonuçlarının gösterilmesi ve kullanıcı geribildirimlerinin alınması gibi kullanıcılar ve veriler arasındaki etkileşimlerin gerçekleştirildiği kullanıcı arayüzüdür.



Şekil 2. TeRec mimarisi

Depolama katmanında kullanıcı tercihleri ve öğe özelliklerinden oluşan bir matris tutulmaktadır. Üçüncü katman ise depolama katmanı ve kullanıcı arayüzü arasında kalan önerilerin oluşturulduğu ve modelin güncellendiği katmandır. TeRec' in çalışma süreci şu şekilde işlemektedir: 1 adımda kullanıcı arayüzü, kullanıcılardan gelen talepleri alarak tavsiyede bulunulmasını istemektedir. 2. adımda tavsiye modeli değerlendirme puanlarını hesaplayarak öneri listesini kullanıcı arayüzüne döndürmektedir. 3. adımda kullanıcı arayüzü, kullanıcı geribildirimini alarak öneri modelini güncellemektedir. 4. adımda tavsiye modeli, geçmiş girdilerin veri deposunu (rezervuar) güncellemektedir. 5. adımda tavsiye modeli güncellenmiş rezervuarı kullanarak matrisi güncellemektedir.

Verilerin çok büyük olabileceği ve çoğunun yararsız olduğu düşünüldüğünde, model güncellemelerini hızlandırmak için bilgilendirici girdilerin bir alt kümesinin (rezervuar) örneklenmesi gerekmektedir. Bu amaçla rezervuar örnekleme tekniği kullanılmıştır. Veri girdilerinin boyutu c sayısına ulaştığında Vitter's algoritmasında t. veri c/t olasılıkla korunacaktır. Çalışma kapsamında sunulan rezervuar örnekleme mekanizmasında ise t. veri $1-c/t$ olasılıkla korunacaktır. Yeni verilerin rezervuara konulmasına karar

verildiyse, rezervuarda bulunan s_i veri örneğinin yer değiştirme olasılığı $1 - P(s_i \in R_{t-1})$, Eş. 6 kullanılarak hesaplanmaktadır.

$$P(s_i \in R_{t-1}) \propto \exp \frac{1}{t-i} \quad (6)$$

$1 - P(s_i \in R_{t-1})$, zaman serisi analizinde yaygın olarak kullanılan üstel bozunma fonksiyonunu, *t-i* mevcut zaman sırası *t* ile sisteme gelen veri örneğinin zaman sırası *i* arasındaki farkı ifade etmektedir.

Geliştirilen sistemde kullanıcıların son tercihlerinin örneklerini saklayan rezervuar, yeni gelen her veri ile dinamik olarak güncellenmekte ve önerilerini her zaman en son güncellenen modele göre üretmektedir. Analiz sonuçları, TeRec sisteminin tweet akışları için gerçek zamanlı hashtag önerisi sunma konusunda daha başarılı sonuçlar verdiğini göstermiştir. Geliştirilen sistemde kullanıcıların son tercihlerinin örneklerini saklayan rezervuar, yeni gelen her veri ile dinamik olarak güncellenmekte ve önerilerini her zaman en son güncellenen modele göre üretmektedir. Analiz sonuçları, TeRec sistemiminin tweet akışları için gerçek zamanlı hashtag önerisi sunma konusunda daha başarılı sonuçlar verdiğini göstermiştir [9].

II. BÜYÜK VERİ

Günümüzde, insanlar ve sistemlerin oluşturduğu dijital ortamlardaki veri miktarı üstel olarak artmaktadır. Web üzerindeki veri miktarı, exabyte (1018) ve zettabyte (1021) cinsinden ölçülmektedir. Verilerin hızlı bir şekilde büyümesi dijital sensörlerin, iletişimlerin, hesaplamanın ve veri depolama hacminin artmasına bağlıdır. Büyük veri kavramı bu olguyu tanımlamak için ortaya çıkmıştır [10].

Büyük veri kavramı, Gartner tarafından daha iyi bilgi ve karar verme için maliyet etkin ve yenilikçi bilgi işleme yöntemleri talep eden yüksek hacimli, yüksek hızlı ve çok çeşitli bilgi varlıkları olarak tanımlanmıştır [11]. Benzer şekilde TechAmerica kuruluşu tarafından bilginin elde edilmesi, depolanması, dağıtımı, yönetimi ve analizi için gelişmiş teknik ve teknoloji gerektiren, yüksek hacim, yüksek hız ve yüksek karmaşıklıktaki değişken veriyi tanımlayan bir terim olarak ifade edilmiştir [12]. Büyük veri tanımlarındaki hacim kavramı terabayt ve petabaytlar ile ifade edilen verilerin büyüklüğünü belirtmektedir. Büyük verilerdeki hacim tanımları zaman ve veri tipi gibi faktörlere göre değişmektedir. Çeşitlilik, bir veri kümesindeki yapısal heterojenliği belirtmektedir [13]. Teknolojik gelişmeler firmaların yapısal, yarı yapılandırılmış ve yapılandırılmamış çeşitli veri türlerini kullanmasına izin vermektedir. Mevcut verilerin yalnızca %5'ini oluşturan yapısal veriler e-tablolarda veya ilişkisel veritabanlarında bulunan tablo verilerini ifade etmektedir [14].

Metin, görüntü, ses ve video verileri, analizler için makinelerin yapması gereken yapısal düzenlemelerden yoksun olan yapılandırılmamış verilere örnektir. Xml verileri, web logları, sosyal medya yayınları ve e-posta yarı yapılandırılmış verilere örnek olarak verilebilir [15]. Hız kavramı, verilerin üretildiği hızı ve hangi hızla analiz edileceğini ifade etmektedir. Mobil cihazlar ve sensörler gibi dijital aygıtların çoğalması büyük miktarlarda verinin oluşmasına neden olmakta ve gerçek zamanlı analizlerin yapılmasını gerektirmektedir. Mobil cihazlardan elde edilen ve mobil uygulamalar aracılığıyla akan veriler, müşterilere gerçek zamanlı ve kişiselleştirilmiş teklifler üretmek için kullanılabilecek bilgiler üretmektedir [16]. Bu bilgiler, kullanıcı profilleri oluşturmak ve gerçek zamanlı analizler yapabilmek için coğrafi konum, demografik bilgiler ve geçmiş satın alma örnekleri gibi bilgiler sağlamaktadır [17].

Büyük veri tanımında ifade edilen 3V kavramının yanı sıra doğrulama, değişkenlik ve değer kavramları ön plana çıkmaktadır. Doğrulama kavramı veri kaynaklarının güvenilirliğini temsil etmektedir. Örnek olarak sosyal medya platformlarındaki kullanıcı yorumları öznel ifadeler oldukları için belirsizdir [18]. Bu gibi kesin olmayan ve belirsiz verilerle başa çıkma ihtiyacı, belirsiz verilerin yönetimi ve incelenmesi, büyük veri analizlerinin diğer bir yönüdür. Değişkenlik kavramı, akan verilerin hızlarındaki değişime karşılık gelmektedir [19]. Değer kavramı ise orijinal haliyle alınan verilerin genellikle hacmine kıyasla daha düşük bir değere sahip olduğu ve büyük miktarlardaki verilerin analiz edilerek yüksek bir değer elde edilmesini ifade etmektedir [20].

2.1 Büyük Veri Kaynakları

Büyük veri analizi, büyük verilere gelişmiş analitik tekniklerin uygulanması olarak tanımlanmaktadır [21]. Büyük veriler, çeşitli analizler yoluyla anlam kazanmaktadır. Karar verme süreçlerinde büyük verileri hızlı ve etkin bir şekilde analiz eden süreçlere ihtiyaç vardır. Büyük veri analizleri veri yönetimi ve analiz yapma işlemlerinden oluşmaktadır. Veri yönetimi verilerin alınması, depolanması ve analizler için hazırlanması süreçlerini içermektedir [22]. Analiz yapma işlemleri ise verilerin analiz edilmesi ve çıkarımlar yapma süreçlerini içermektedir. Büyük verilerin çeşitlilik özelliklerinden dolayı, analiz işlemleri için kullanılan birçok teknik vardır [23]. Yaygın olarak tavsiye sistemlerinde kullanılan birliktelik kuralları varlıklar arasındaki ilişkileri belirlemek için kullanılmaktadır. Makine öğrenmesi yöntemleri,

bilgisayarların karmaşık örüntülerden zeki kararlar çıkarması için kullanılmaktadır. Veri madenciliği yöntemleri istatistik, makine öğrenmesi ve veritabanı yönetiminin birleşimi şeklinde kullanılmaktadır [24]. Kümeleme analizleri ise denetimsiz makine öğrenmesi yöntemlerini kullanmaktadır. Verileri, önceden bilinmeyen aynı özelliklere sahip küçük kümelere bölmeyi amaçlamaktadır [25].

Büyük veri analizleri kullanılacak veriye göre metin analizi, Web verisi analizi, ses ve video verisi analizi, sosyal medya verisi analizi, sensör verisi analizi ve mobil cihaz verisi analizi başlıkları ile incelenmektedir [26].

Metin analizleri, metinsel verilerden bilgi çıkarımı süreçlerini ifade etmektedir. E-postalar, bloglar, forumlar, anket yanıtları, kurumsal belgeler, haberler ve çağrı merkezi günlükleri metinsel verilere örnek olarak verilebilir [27]. Metin analizleri, insanlar tarafından oluşturulan metinlerin, karar vermeyi destekleyen anlamlı özetler haline dönüştürülmesini sağlamaktadır. Örneğin, finansal analizler, finansal haberlerden çıkarılan bilgileri esas alan borsa tahminleri için kullanılmaktadır. Metin analizleri, genel olarak yapılandırılmamış metinlerden kullanışlı bilgiler elde etmek için kullanılan süreçlerdir. Metin analizleri bilgiye erişim, makine öğrenmesi, istatistik, bilişimsel dilbilim ve veri madenciliği ile ilişkilidir [20].

Web verisi analizi, Web belgelerinden ve hizmetlerinden otomatik olarak aldığı verileri, ayıklayıp değerlendirerek yararlı bilgiler çıkarmayı amaçlamaktadır [28]. Web analizi veritabanı, bilgiye erişim, doğal dil işleme ve metin madenciliği gibi çeşitli araştırma alanlarıyla ilişkilidir. Web analizi Web içerik madenciliği, Web yapısı madenciliği ve Web kullanımı madenciliği başlıkları altında incelenmektedir. Web içerik madenciliği, metin, resim, ses, video ve kod gibi çeşitli veri tiplerini içeren yararlı bilgileri Web sayfalarında elde etmek için kullanılmaktadır [2]. Web yapısı madenciliği, Web sayfaları üzerinde farklı sayfalara verilen linkleri elde etmek ve model çıkarımı yapmak için kullanılmaktadır. Web kullanım madenciliği Web sunucularındaki ve proxy sunucularındaki erişim günlükleri, tarayıcı geçmişi kayıtları, kullanıcı profilleri, kayıt verileri, kullanıcı oturumları, kullanıcı sorguları ve kullanıcı tıklamaları gibi Web sayfaları üzerinde yapılan etkileşimleri incelemektedir [29].

Ses analizleri, yapılandırılmamış ses verilerinden bilgi analizi ve çıkarımı yapmayı hedeflemektedir [30]. Çağrı merkezleri ve sağlık hizmetleri, ses analizinin birincil uygulama alanlarıdır. Çağrı merkezleri, binlerce saat boyunca kaydedilen çağrıları etkili bir şekilde analiz etmek için ses analizleri kullanmaktadır. Bu teknikler, müşteri deneyiminin iyileştirilmesi, gizlilik ve güvenlik politikalarının belirlenmesi, müşteri davranışlarıyla ilgili çıkarımlar yapma gibi farklı alanlarda kullanılmaktadır [20]. Ses analiz sistemleri, canlı bir aramayı analiz etmek, müşterinin geçmiş ve günümüzdeki etkileşimlerine dayalı olarak çapraz satış önerileri belirlemek ve gerçek zamanlı geri bildirim sağlamak için geliştirilmektedir. Video analizi, video akışlarını izlemek, analiz etmek ve anlamlı bilgiler çıkarmak için çeşitli teknikleri içermektedir. Video analizleri ile sınır bölgelerindeki ihlalleri tespit etme, çalınan nesneleri belirleme, belirli bir bölgedeki dolandırıcılıkları tespit etme ve şüpheli etkinlikleri tanımlama gibi işlevler gerçekleştirilebilmektedir [31].

Sosyal medya verisi analizi, sosyal medya kanallarından elde edilen yapılandırılmış ve yapılandırılmamış verilerin analiz edilmesini ifade etmektedir [32]. Kullanıcılar tarafından oluşturulan yorumlar, resimler, videolar ve yer imleri ile kişiler, kuruluşlar ve ürünler arasındaki ilişkiler ve etkileşimler, sosyal medyadaki bilgi kaynaklarıdır [33]. Bu bilgi kaynaklarına dayalı olarak sosyal medya analizleri içerik tabanlı analiz ve yapısal analizler olarak iki başlık altında incelenebilir. İçerik tabanlı analizler, kullanıcıların sosyal medya platformlarında yayınladığı geribildirimler, ürün incelemeleri, görüntüler ve videolar gibi verilere odaklanmaktadır [34]. Sosyal medyadaki bu tür içerikler genel olarak büyük hacimli, yapılandırılmamış, gürültülü ve dinamiktir [35]. Sosyal ağ yapıları kişiler ve aralarındaki ilişkileri temsil eden düğüm ve kenarlar şeklinde modellenmiştir. Bu modellerde düğümler arasındaki kenarlar, kişiler arasındaki bir bağlantı varlığını (Örneğin, arkadaşlık) göstermektedir [36].

Sosyal ağların yapısından bilgi çıkarmak için topluluk belirleme, sosyal etki analizi ve link tahmini yöntemleri kullanılmaktadır. Topluluk belirleme, sosyal ağ içerisindeki dolaylı ilişkilerin belirlenmesi için kullanılmaktadır. Sosyal etki analizi, bir sosyal ağdaki kişilerin ve bağlantıların modellenmesi ve değerlendirilmesi için kullanılmaktadır [37]. Link tahmini ise ağdaki mevcut düğümler arasındaki gelecek bağlantıların öngörülmesi için kullanılmaktadır.

Sensörler ses, titreşim, akım, hava, basınç ve sıcaklık gibi fiziksel nicelikleri okunabilir dijital sinyallere dönüştürerek ölçüm ve depolama için kullanılmasını sağlamaktadır. Sensörler aracılığıyla algılanan veriler, kablolu veya kablosuz ağlar vasıtasıyla bir veri toplama noktasına aktarılmaktadır. Sensör düğümleri arasında veri iletimini sağlamak için kablosuz iletişim kullanılmaktadır. Kablosuz ağlar su kalitesi izleme, askeri gözlem ve doğal yaşam izleme gibi birçok uygulamada kullanılmaktadır [38].

Mobil cihazların işlevlerindeki gelişmelerle birlikte mobil cihazlardan elde edilen veri çeşitliliğinde de artış yaşanmaktadır. Mobil cihazlar sahip oldukları konumlandırma sistemleri aracılığıyla coğrafi konum bilgilerini, kamera ve mikrofonları aracılığıyla ses, fotoğraf ve video gibi

multimedya içeriklerini, dokunmatik ekranları ve yerçekimi sensörleri aracılığıyla da kullanıcı hareketlerini elde edebilmektedir. Kablosuz iletişim operatörleri, elde ettikleri bu tür bilgileri analiz ederek mobil internetin hizmet seviyesini geliştirmektedir [39].

2.2 Büyük Veri Teknolojileri

Farklı kaynaklardan gelen yüksek hacimli veriler, depolama ve zaman kısıtlamaları altında işlenmelidir. Tek bir fiziksel makine, karmaşık hesaplama süreçleri nedeniyle düşük gecikme ile gelen veriyi işleyemezken, bazı durumlarda belirli bir zamanda gelen büyük miktardaki veri, ağ trafiğinin sınırlarını aşabilmektedir. Bu durum, hesaplamaların birden fazla makinede gerçekleştirilmesini ve dağıtık işleme sistemlerinin geliştirilmesine neden olmuştur [40]. Dağıtık sistemlerde paylaşımı olmayan bilgisayarlar bir kümede birleştirilmektedir. Bu şekilde bir görev üzerinde birlikte çalışan makinelerin verilerinin ve hesaplamalarının sonuçlarının paylaşılması için bu makinelere, ortak bir ağ iletişiminin sağlanması gerekmektedir [41].

Apache Hadoop, donanımsal kümeler ile büyük veri kümelerini depolamak ve işlemek için kullanılan açık kaynaklı bir yapıdır. Hadoop platformu dağıtık dosya sistemi (HDFS) ve Hadoop MapReduce bilişenlerinden oluşmaktadır. HDFS, veri depolamak için kullanılan birimdir. Hadoop MapReduce ise MapReduce programlama modelinin uygulamasıdır [42]. HDFS, dağıtık Google Dosya Sisteminin (GFS) açık kaynak kodlu bir uygulamasıdır. Dağıtık makineler üzerinde büyük dosyaların güvenilir ve etkili bir şekilde depolanması için ölçeklenebilir dağıtık bir dosya sistemi sağlar [43].

MapReduce, büyük veri kümelerinin paralel olarak işlenmesini sağlamak için tasarlanmış bir programlama modelidir [44]. MapReduce, Google tarafından 2004 yılında, paralelleştirme, dağıtık depolama, yük dengeleme ve hata toleransı gibi ayrıntılardan soyutlanmış bir hesaplama aracı olarak geliştirilmiştir. MapReduce programlama modeli, kullanıcılar tarafından oluşturulan Map ve Reduce işlemlerinden oluşmaktadır. Map işlevi, girdi olarak tek bir anahtar/değer çifti alarak ara anahtar/değer çifti oluşturmaktadır. Ardından, MapReduce aynı anahtarla ilgili tüm ara değerleri bir araya getirerek bu değerleri daha küçük bir kümeye sıkıştıran Reduce işlevine iletmektedir [45]. Map aşamasında, anahtar/değer yapısına sahip her yığın için, ilgili Map fonksiyonu ile bir dizi ara anahtar/değer çifti üretilmektedir. Birleştirme aşaması, aynı ara anahtarla ilişkili tüm ara anahtar/değer çiftlerinin gruplandırılmasını amaçlamaktadır. Bölümleme aşamasında sonuçlar farklı Reduce fonksiyonlarına dağıtılmaktadır. Reduce aşamasında, aynı anahtara sahip anahtar/değer çiftleri birleştirilerek nihai bir sonuç hesaplanmaktadır [46,47].

Apache Spark, heterojen verilerin etkili bir şekilde analiz edilmesi için 2009 yılında Berkeley' de geliştirilmiştir. Hadoop' a alternatif olarak disklerdeki G/Ç sınırlamalarını aşmak ve önceki sistemlerin performansını artırmak üzere tasarlanmıştır [48]. Spark'ın temel kavramı esnek dağıtık verisetleridir (RDD). RDD, temelde bir Spark kümesine yayılmış nesnelerin değiştirilemez bir koleksiyonudur. RDD'ler üzerinde dönüşümler ve eylemler gerçekleştirilmektedir. Dönüşümler map, filter, union ve join gibi fonkiyonlar kullanılarak mevcut RDD'lerden yeni RDD'lerin oluşturulmasını, eylemler ise RDD'lerin hesaplama sonuçlarını ifade etmektedir [49]. Dönüşümler Apache Hadoop teknolojisindeki Map işlemine, eylemler ise Reduce işlemine karşılık gelmektedir.

Spark kümeleri master/slave mimarisine dayalı olarak sürücü programı, küme yöneticisi ve işçi düğüm bileşenlerinden oluşmaktadır. Sürücü programı bileşeni, Spark kümesindeki slave düğümü temsil etmektedir. Çalışan uygulamaları yöneten ve denetleyen SparkContext nesnesini tutmaktadır. Küme yöneticisi bileşeni, sürücü programı tarafından işçi düğümlere atanan uygulamaların iş akışını yönlendirmekten sorumludur [50]. Ayrıca, kümedeki kaynakları kontrol edip denetleyerek durumlarını sürücü programına döndürmektedir. İşçi düğümler ise Spark programının yürütülmesi sırasındaki bir işlemin kapsamını ifade etmektedir. Spark, Spark Core, Spark Streaming, Spark SQL, Spark MlLib ve GraphX gibi çeşitli uygulama programlama arayüzlerini (API) kullanmaktadır.

Storm, gerçek zamanlı olarak büyük yapılandırılmış ve yapılandırılmamış verileri işlemek için kullanılan açık kaynaklı bir yapıdır. Storm, gerçek zamanlı veri analizleri ve makine öğrenmesi için kullanılmaktadır [51]. Storm topolojisi, çevrimsel yönlü graflar (DAG) ile temsil edilmektedir. DAG yapısının kenarları veri aktarımını temsil etmektedir. DAG düğümleri ise spout ve bolt bileşenlerinden oluşmaktadır. Spout bileşenleri veri kaynaklarını, bolt bileşenleri ise verilere uygulanacak fonksiyonları temsil etmektedir.

Storm, Nimbus adı verilen ana düğüm ve supervisor adı verilen slave düğümlerden oluşmaktadır. Nimbus, tüm slave düğümleri arasında veri dağıtımı yapmak, slave düğümlerine görev atamak ve arızaları izlemekten sorumludur [52]. Kümede bir düğüm hatası algılanırsa, Nimbus görevi başka bir düğüme atar. Supervisor düğümler ise Nimbus tarafından atanan görevlerin yürütülmesini kontrol eder. Supervisor düğümler birden fazla işçi sürecine sahiptir ve Nimbus tarafından atanan görevleri tamamlamak için çalışan süreçleri yönetir [46].

Flink, gerçek zamanlı olarak ya da yığın halinde veri işlemek için geliştirilmiş açık kaynaklı bir çerçevedir. Flink'in programlama modeli MapReduce'a benzerdir. Ancak MapReduce'un aksine birleştirme, filtreleme ve toplama gibi üst seviye işlevler sunmaktadır. Flink, Flume ve Kafka gibi farklı araçlar tarafından toplanan akan veriler üzerinde tekrarlı ve gerçek zamanlı hesaplama yapmaya izin vermektedir [48].

MapReduce, Spark, Storm ve Flink yapılarının veri formatı, işleme modu, kullanılan veri kaynakları, programlama modeli, desteklenen programlama dilleri, küme yönetimi ve yinelemeli hesaplamaya izin vermelerine göre sınıflandırılması Tablo 1'de görülmektedir [53].

Tablo 1. Büyük veri teknolojileri

	MapReduce	**Spark**	**Storm**	**Flink**
Veri formatı	Anahtar/değer	RDD	Anahtar/değer	Anahtar/değer
İşleme modu	Yığın	Yığın ve akış	Akış	Yığın ve akış
Veri kaynakları	HDFS	HDFS, DBMS ve Kafka	HDFS, HBASE ve Kafka	Kafka, Kinesis, akış verileri
Programlama modeli	Map ve Reduce	Dönüşümler ve eylemler	Topoloji	Dönüşümler
Desteklenen programlama dilleri	Java	Java, Scala ve Python	Java	Java
Küme yönetimi	YARN	Standalone, YARN ve Mesos	YARN veya ZooKeeper	ZooKeeper
Yinelemeli hesaplama	ü	ü	ü	ü

III. BÜYÜK VERİ VE TAVSİYE SİSTEMLERİ

Tavsiye sistemleri, kişiselleştirilmiş öneriler sunmak için kullanıcıların geçmişteki satın alma durumlarını ve değerlendirme verilerini kullanmaktadır. Günümüzde mevcut olan veri hacmi tavsiye sistemlerinde öneri oluşturmak için kullanılan yöntemlerin yeniden değerlendirmesini gerektirmektedir [54]. Büyük verinin temelinde yer alan paralel ve dağıtık veri işleme, algoritma tasarımında temel teşkil etmelidir. OpenMP ve MPI gibi geleneksel paralel bilgi işleme ortamları ile MapReduce ve Spark gibi dağıtık bilgi işleme platformları bu amaçla kullanılmaktadır [55].

Tavsiye sistemlerin ele aldığı iki temel problem, değerlendirme puanı tahmini ve en iyi-N öneri listesinin oluşturulmasıdır. Değerlendirme puanı tahmininde amaç, bir kullanıcının bir öğe için vereceği puanlamayı hesaplamaktır. En iyi-N öneri listesinin oluşturulmasındaki amaç ise kullanıcıların ilgilerini çekebilecek ve muhtemelen beğenecekleri N adet öğeden oluşan bir öneri listesi sunmaktır [56]. Önerilerin hesaplanması için en yaygın kullanılan iki yaklaşım komşuluk tabanlı yaklaşımlar ve gizli faktör modeli tabanlı yaklaşımlardır. Komşuluk tabanlı yaklaşımlar, öğelerin veya kullanıcıların arasındaki benzerliklere dayalı olarak önerilerin hesaplanması için kullanılmaktadır. Gizli faktör modeli tabanlı yaklaşımlarda ise kullanıcılar ve öğeler aynı gizli alan içerisinde eşleştirilir ve bu alandaki bir kullanıcıya en yakın öğeler öneri olarak sunulur. Gizli faktör modeli tabanlı yaklaşımlar değerlendirme tahminlerinin elde edilmesinde, komşuluk temelli yaklaşımlar ise en iyi-N öneri listesinin elde edilmesinde yaygın olarak kullanılmaktadır [57].

Giderek artan veri miktarı, tavsiye sistemlerinde büyük veri analizi sorunlarının yaşanmasına neden olmaktadır. Tavsiye sistemleri, büyük ölçekli verileri işlerken veya analiz ederken genellikle ölçeklenebilirlik ve verimsizlik sorunları yaşamaktadır [58]. Bu bölümde literatürdeki tavsiye sistemlerinde büyük veri analizlerinin gerçekleştirildiği çalışmalar incelenmiştir.

Meng ve ark. tarafından 2014 yılında yapılan çalışmada, kullanıcılara kişiselleştirilmiş ve etkin öneriler sunabilmek için KASR adı verilen anahtar kelimeye dayalı tavsiye sistemi geliştirilmiştir. Anahtar kelimeler, kullanıcıların tercihlerini belirtmek için kullanılmış ve uygun öneriler oluşturmak için bir kullanıcı tabanlı işbirlikçi filtreleme yöntemi kullanılmıştır. Geliştirilen sistemde yakın kullanıcıların incelemelerinden çıkarılan anahtar kelimeler, kullanıcı tercihlerini belirlemek için kullanılmıştır. Büyük veri ortamında ölçeklenebilirliği ve verimliliği artırmak için KASR sistemi, MapReduce paralel işleme paradigması kullanılarak Hadoop platformu üzerinde geliştirilmiştir. Geliştirilen yöntemde, kullanıcıların tercihlerini elde etmek için anahtar kelime aday listesi ve uzmanlaşmış alan adı sözlüğü olmak üzere iki veri yapısı kullanılmıştır. *n*, anahtar kelime adayı listesindeki anahtar kelimelerin sayısı olmak üzere $K=\{k_1,k_2,..,k_n\}$ şeklinde ifade edilen anahtar kelime aday listesi, kullanıcıların tercihlerine ve aday hizmetlerin çoklu ölçütlerine ilişkin bir dizi anahtar kelimedir ve aday hizmetlerin kalite ölçütleriyle ilgili bir sözcük olabilmektedir. Aktif kullanıcının ve yakın kullanıcıların tercihleri anahtar kelime kümelerine eşleştirilmektedir. Aktif kullanıcıya yakın kullanıcıların hesaplanması için Jaccard benzerliği kullanılmıştır [59].

Yu ve ark. tarafından 2015 yılında yapılan çalışmada, yapılandırılmamış büyük sağlık hizmeti verilerini güvenli bir ortamda yöneterek bu verilerden yararlı bilgiler üretmek ve bilgileri faydalı bir pratik modele çevirmek amaçlanmıştır. Çalışma kapsamında hastalıkların erken teşhisi için bir uygulama sistemi oluşturmak hedeflenmiştir. Uygulama

sistemi, kullanıcıların sağlık koşulları ile ilgili önerilerde bulunmak, tedavi optimizasyonu sağlamak ve olumsuz olayları önlemek için Apache Mahout'un üzerinde çalışan Naïve Bayes (NB) sınıflandırma algoritması kullanılarak oluşturulmuştur [60].

Gu ve ark. tarafından 2015 yılında yapılan çalışmada, Spark üzerine kurulmuş dağıtık matris hesaplama kütüphanesi olan Marlin önerilmiştir. Sosyal ağ madenciliği, tavsiye sistemleri ve doğal dil işleme gibi veri analitiği uygulamalarının temeli olan matris hesaplaması, büyük veri çağında matris ölçekleri büyüdüğü için geleneksel tek düğümlü matris hesaplama sistemleri, bu gibi veri boyutları ve hesaplamaları konusunda yetersiz kalmaktadır. Geliştirilen Marlin kütüphanesi ise içerdiği dağıtık matris işlem algoritmaları ile yüksek seviyeli matris hesaplaması sağlamaktadır. Deneysel sonuçlar, Marlin'in R ve MapReduce'u temel alan dağıtık matris işlem algoritmalarından daha hızlı olduğunu göstermiştir [61].

Verma ve ark. tarafından 2015 yılında yapılan çalışmada, Hadoop çerçevesi kullanılarak Web üzerinde ürün, etkinlik, birey ve hizmetler gibi herhangi bir öğe hakkında değerlendirme, inceleme, görüş, şikayet, açıklama, geri bildirim ve yorum gibi büyük miktarda veri sağlayan bir tavsiye sistemi geliştirilmiştir. Geliştirilen sistemde inceleme, görüş, açıklama, yorum ve şikâyet gibi farklı türdeki verileri filtrelemek için bir hibrid filtreleme tekniği kullanılmıştır. Kullanıcılara sunulan öneriler, kullanıcı değerlendirmelerine, içeriğe, inceleme yapan kullanıcının davranışına ve farklı kullanıcılar tarafından üretilen incelemelerin zamanlamasına dayanmaktadır. Geliştirilen sistem Hadoop platformu üzerinde MovieLense verisetinin farklı boyuttaki dosyaları ile test edilmiştir. Sonuç grafiği, dosya boyutundaki artışa paralel olarak değerlendirme, inceleme ve geri bildirim biçiminde olan veri boyutunun da arttığını ancak veri işleme süresinde aynı oranda artış yaşanmadığını göstermektedir [62].

Dai ve ark. tarafından 2015 yılında yapılan çalışmada, büyük rota verileri kullanılarak kullanıcılara kişiselleştirilmiş rota önerilerinin sunulmasına yönelik bir tavsiye sistemi geliştirilmiştir. Geliştirilen sistem, farklı sürücülerin sürüş tercihlerini kullanarak modelleme ve güncelleme yapmaktadır. Kişiselleştirilmiş rota tavsiyesi oluşturmak için, sürücünün belirlediği kaynak ve varış noktası ile kalkış saatine göre belirlenecek rota için en etkin alt küme rotaların çıkarılması hedeflenmiştir. Geliştirilen sistem, Pekin'deki 52.211 taksi şoförü ile analiz edilmiştir. Test sonuçları geliştirilen sistemin verimli ve etkili bir şekilde tavsiyeler sunduğunu göstermiştir [63].

Huang ve ark. tarafından 2015 yılında yapılan çalışmada, büyük veriler üzerinde gerçek zamanlı ve doğru öneriler sunabilmek için Storm platformu kullanılarak TencentRec adı verilen sistem geliştirilmiştir. Akan veriler, geliştirilen bir veri erişim bileşeni ve bir veri depolama bileşeni ile birlikte Storm kullanılarak analiz edilmiştir. Farklı türdeki uygulamalar için öğe tabanlı işbirlikçi filtreleme, içerik tabanlı filtreleme ve demografik özelliklere dayalı filtreleme yöntemleri uygulanmıştır. Gerçek zamanlı veri toplama ve işleme süreci kullanılarak öneri değişiklikleri gerçek zamanlı olarak sunulmaktadır. Geliştirilen sistem ön işleme katmanı, algoritma katmanı ve depolama katmanı bileşenlerinden oluşmaktadır. Ön işleme katmanı, alınan verileri ayrıştırır ve niteliksiz verileri süzerek algoritma katmanına gönderir. Algoritma katmanı ana algoritma hesaplamalarından sorumludur. İşbirlikçi filtreleme, içerik tabanlı filtreleme ve demografik özelliklere dayalı filtreleme yöntemleri algoritma katmanında uygulanmaktadır. Depolama katmanı, algoritma katmanı tarafından üretilen sonuçlara, farklı uygulamaların kurallarına göre filtreleme uygulamakta ve hesaplama sonuçlarını güncellemektedir [64].

Riyaz ve ark. tarafından 2016 yılında yapılan çalışmada, geleneksel tavsiye sistemlerinin artan kullanıcı ve ürün verilerinin analizinde yaşadığı ölçeklenebilirlik ve verimlilik sorunları sebebiyle MapReduce paradigması kullanılarak Apache Hadoop üzerinde işbirlikçi filtreleme yöntemleri ile yeni bir tavsiye sistemi geliştirilmiştir. Geliştirilen sistem Hadoop düğümleri, dağıtık tavsiye motoru ve HBase depolama alanından oluşmaktadır. Geliştirilen sistemde Amazon ürün veriseti kullanılmıştır. İşbirlikçi filtreleme yönteminde Pearsons Correlation Coefficient yöntemi kullanılarak benzerlikler hesaplanmıştır. Üretilen kullanıcı tavsiyeleri HBase dağıtık veritabanında saklanmaktadır. HBase, disk üzerinde fazladan arama yapmayı azaltan Bloom Filtresini kullanmaktadır [65].

Shang ve ark. tarafından 2016 yılında yapılan çalışmada, bir mikro-video öneri sistemi geliştirilmiştir. Geliştirilen sistem videonun yapımcısına kaç kullanıcının videoyu beğendiği gibi bilgileri sağlamaktadır. Ayrıca kullanıcıların favori videolarını ve izleme geçmişini analiz ederek kullanıcılara video önerisi sunmaktadır. Geliştirilen sistemde, Web crawler yazılımı kullanılarak video siteleri ve forumları gibi alanlardan veri toplanmıştır. Elde edilen veriler kullanılarak mikro-video modeli ve kullanıcı modeli oluşturulmuştur. Web crawler yazılımı ile elde edilen veriler Hadoop platformunda depolanmış ve verileri işlemek için Mahout kullanılmıştır. Mahout üzerinde işbirlikçi filtreleme yöntemlerinin gerçekleştirilmesini sağlayan Slope algoritması kullanılmıştır [66].

Chang ve ark. tarafından 2017 yılında yapılan çalışmada, sonsuz ve değişken boyuttaki akan veriler için gerçek zamanlı güncelleme yapabilen sRec sistemi geliştirilmiştir.

Geliştirilen sistemde girdi akışı, kullanıcı geri bildirim etkinlikleri, yeni kullanıcılar ve yeni öğeler olarak modellenmiştir. sRec değişen kullanıcı ve ürün sayısı ile kullanıcı tercihlerinde yaşanabilecek içerik kayması durumlarını belirleyerek gerçek zamanlı öneriler sunmaktadır. sRec temel olarak çevrimiçi öneri modülü ve çevrimdışı parametre öğrenme modülünden oluşmaktadır. Sistem, kullanıcı dinamiklerini yakalamak ve gerçek zamanlı öneriler sunabilmek için modelini sürekli olarak güncellemektedir. sRec sisteminde, kullanıcı öğe değerlendirmeleri zamana dayalı bir fonksiyon ile modellenmektedir. Çevrimdışı parametre güncellemeleri, yalnızca olaylar meydana geldiğinde ve önceki olaylara bağlı olarak gerçekleştirilmektedir. Çevrimiçi öneri modülü ise oluşturulan kullanıcı-öğe-zaman modellerini kullanarak kullanıcıların yapmış oldukları tercihlerin son olasılık dağılımına (posterior) göre öneri sunmaktadır [67].

Prando ve ark.tarafından 2017 yılında yapılan çalışmada, e-ticaret platformlarındaki kullanıcıların tercihlerini belirleyebilmek için kullanıcıların sosyal ağlardaki etkileşimlerine dayalı yeni bir tavsiye sistemi geliştirilmiştir. Geliştirilen tavsiye sistemi, içerik tabanlı filtreleme teknikleri ile yeni kullanıcıların sosyal ağ verilerini analiz ederek kullanıcıları belirli kategorilere atamaktadır. Kullanıcı tercihleri, kullanıcıların doğrudan yaptığı paylaşımlar, beğendikleri paylaşımlar ve beğendikleri sayfalar kullanılarak elde edilmiştir. Geliştirilen sistem, soğuk başlangıç sorununu gidermek amacıyla bir e-ticaret platformu üzerinde test edilmiştir. Analiz sonuçları, geliştirilen sistemin ilk kez e-ticaret platformuna erişen yeni kullanıcılar için başarılı sonuçlar verdiğini göstermiştir [68].

Ajantha ve ark. tarafından 2017 yılında yapılan çalışmada, kullanıcı konum vektörü adı verilen bir vektör kullanılarak, kullanıcılar ile ilgi alanları arasındaki ilişkileri belirlemek için yeni bir yöntem geliştirilmiştir. Geliştirilen sistem kullanıcı bilgisi toplama modülü, kullanıcı kümeleme modülü, konum bilgisi toplama modülü, kullanıcı-konum vektörü hesaplama modülü ve konum profili modülünden oluşmaktadır. Kullanıcı profil bilgileri ve yaş, cinsiyet gibi bilgileri Facebook'tan kullanıcı bilgisi toplama modülü kullanılarak toplanmaktadır. Elde edilen kullanıcı bilgileri k-means algoritması kullanılarak yakın kullanıcıların belirlenebilmesi için kümelenmektedir. Konum bilgileri ise trip advisor ve seyahat bloglarından toplanmaktadır. Kullanıcı-konum vektörleri ise konum bilgileri ve kullanıcı profil bilgilerine göre hesaplanmaktadır. Konum profil modülü ise kullanıcı profillerinden çıkarılan kullanıcıların ilgi alanları ile yakın kullanıcıların ilgi alanlarına göre ziyaret ettikleri konumları eşleştirmektedir [69].

Zhou ve ark. tarafından 2017 yılında yapılan çalışmada, öğe tabanlı işbirlikçi filtreleme yöntemi ve Hadoop programlama modeli kullanılarak bir film tavsiye sistemi geliştirilmiştir. Geliştirilen sistemde dağıtık dosya sistemi HDFS ve MapReduce kullanarak artan veri hacminin depolanması ve verilerin paralel olarak işlenerek algoritmanın performansının ve sistemin yanıt hızının artırılması amaçlanmıştır. Geliştirilen sistem, MoviLense veritabanı üzerinde test edilmiştir. Deneysel sonuçlar, sistemin büyük veri kümelerinde klasik yöntemlere göre yüksek verimlilik ve güvenilirlik sağladığını göstermektedir [70].

Seo ve ark. tarafından 2017 yılında yapılan çalışmada, sosyal ağlardaki kullanıcılar arasındaki yakınlığı hesaplamak için yeni bir yöntem sunulmuştur. Twitter üzerindeki büyük sosyal veriler analiz edilerek kullanıcılara konu veya ilgi alanı öneren arkadaşlık temeline dayalı, kişiselleştirilmiş bir tavsiye sistemi geliştirilmiştir. Geliştirilen sistem, bir aylık Twitter verileri kullanılarak precision, recall, f ölçütü ve ortalama mutlak hata metriklerine göre karşılaştırmalı deneyler gerçekleştirilmiştir. Deney sonuçları, geliştirilen sistemin kullanıcılar arasındaki yakınlık derecesini belirlemede ve kişiselleştirilmiş öneriler hesaplamada daha başarılı olduğunu ortaya koymuştur [71].

Wei ve ark. tarafından 2017 yılında yapılan çalışmada, işbirlikçi filtreleme yaklaşımlarının yaşadığı soğuk başlangıç problemlerine çözüm olarak derin öğrenme tabanlı yeni bir model geliştirilmiştir. Geliştirilen modelde ürünler ile ilgili kullanıcı değerlendirmelerin eksik olması veya hiç olmaması durumlarında yaşanan problemler aşılmaya çalışılmıştır. Öğelerin içerik özelliklerini çıkarmak için sinir ağı tabanlı bir derin öğrenme mimarisi kullanılmıştır. Kullanıcı tercihlerinin ve öğe özelliklerinin zamansal dinamiklerini modelleyen işbirlikçi filtreleme modeli, tekil değer ayrışımı (Singular Value Decomposition-SVD) kullanılarak soğuk başlangıç sorunu yaşanan öğelerin içerik özelliklerini öngörmek üzere değiştirilmiştir. Geliştirilen sistem, Netflix değerlendirmelerinin bulunduğu büyük veri kümesi üzerinde test edilmiştir. Test sonuçları geliştirilen modelin, soğuk başlangıç sorunu yaşanan öğelerinin değerlendirme puanlarının tahmininde klasik modellerden daha iyi performans sergilediğini göstermiştir [72].

3.1 Tavsiye Sistemlerinde Kullanılan Veri Toplama Yaklaşımları

Kullanıcılara öneri olarak sunulacak öğelerin tahmini aşamasında, kullanıcı profilleri ve modelleri oluşturmak için kullanıcıların nitelikleri, davranışları veya kullanıcıların eriştiği kaynakların içerikleri gibi bilgiler kullanılmaktadır. Örnek olarak e-öğrenme platformlarında bilişsel beceriler,

zihinsel yetenekler, öğrenme stilleri, ilgi, tercihler ve sistemle olan etkileşimler kullanıcı profillerinin oluşturulması aşamasında kullanılmaktadır [7].

Tavsiye sistemlerinde sunulan önerilerin kalitesi ve başarısı oluşturulacak kullanıcı profilleri ile doğrudan ilişkilidir. Kullanıcıların beklentilerine cevap verebilecek öneriler sunabilmek için kullanıcılar hakkında mümkün oldukça fazla bilgiye sahip olunması gerekmektedir. Kullanıcıların davranışlarının gözlemlenmesi yoluyla elde edilen dolaylı geribildirimlerin kullanıcı tercihlerini daha doğru bir şekilde yansıtacağı temel alınmaktadır [73].

3.1.1 Dolaylı geribildirimler

Dolaylı geribildirimler, kullanıcıların satın alma geçmişi, gezinme geçmişi, Web sayfalarında kalınan süreler, kullanıcı tarafından tıklanan bağlantılar ve kullanıcı arayüzü üzerindeki tıklamalar gibi farklı kullanıcı eylemlerinin gözlemlenmesi yoluyla elde edilmektedir. Dolaylı geribildirimler kullanıcı tercihlerini, kullanıcıların sistemle olan etkileşimlerinden çıkararak kullanıcı yükünü azaltmaktadır. Dolaylı geribildirimler, herhangi bir kullanıcı çabası gerektirmemesi ve doğrudan kullanıcı davranışlarının analiz edilmesi yoluyla elde edildiği için daha objektif bir yaklaşım olarak ön plana çıkmaktadır [74].

3.1.2 Doğrudan geribildirimler

Doğrudan geribildirim kullanılan sistemlerde, bir kullanıcı arayüzü aracılığıyla kullanıcılardan sistemde bulunan öğeler hakkında değerlendirmelerde bulunmaları beklenmektedir. Bu sistemlerde sunulan önerilerin doğruluğu, kullanıcı tarafından yapılan değerlendirmelerin sayısına bağlıdır. Bu yöntemin eksikliği ise kullanıcı çabası gerektirmesi ve kullanıcıların daima yeterli bilgi vermeye hazır olmamasıdır. Doğrudan geribildirimlerin daha fazla kullanıcı çabası gerektirmesi gerçeğine rağmen, kullanıcı davranışlarından çıkarımlar yapılması gerekli olmadığından elde edilen veriler daha güvenilir olarak görülmekte ve öneri sürecinde şeffaflık sağlanmaktadır [75].

3.1.3 Hibrit geribildirimler

Doğrudan ve dolaylı geribildirimlerin zayıf yönlerini en aza indirgemek ve en iyi performansı elde edebilmek için, bu yöntemler birlikte kullanılabilmektedir. Hibrit geribildirimler, doğrudan geribildirimlerin üzerine bir kontrol mekanizması şeklinde eklenen dolaylı geribildirimler ile ya da dolaylı geribildirimler alınırken, kullanıcıların öğeler hakkındaki değerlendirmeleri alınarak gerçekleştirilebilmektedir [76].

3.2 Tavsiye Sistemlerinde Kullanılan Yöntemler

Kullanıcılara faydalı ve kaliteli öneriler sunabilmek için etkin ve doğru öneri tekniklerinin kullanılması önemlidir. Farklı öneri sunma yöntemlerinin özelliklerinin ve potansiyellerinin doğru bir şekilde belirlenmesi ve kullanılacak sisteme göre değerlendirilmesi gerekmektedir.

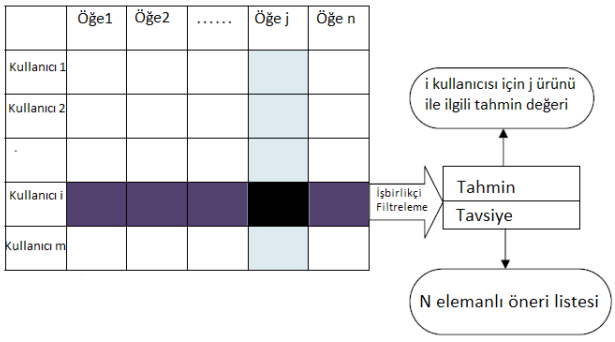
3.2.1 İçerik tabanlı filtreleme

İçerik tabanlı filtreleme yöntemleri, kullanılan alana bağımlı algoritmalardır ve tahminlerin oluşturulması için öğe özelliklerinin analizi yüksek önem taşımaktadır. Web sayfası, yayın ve haber önerisi gibi uygulama alanlarında içerik tabanlı filtreleme teknikleri daha yüksek başarı göstermektedir. İçerik tabanlı filtreleme yöntemlerinde, kullanıcıların geçmişte değerlendirdikleri öğelerin özelliklerine göre kullanıcı profilleri oluşturulmaktadır. Çoğunlukla kullanıcılar tarafından olumlu olarak değerlendirilmiş öğelere yakın özellikteki öğeler öneri olarak sunulmaktadır [77]. İçerik tabanlı filtreleme yöntemleri anlamlı öneriler üretebilmek için öğeler arasındaki benzerlikleri hesaplamada farklı modeller kullanmaktadır. Farklı öğeler arasındaki ilişkileri modellemek için TF / IDF gibi Vektör Uzay Modeli veya Naive Bayes Sınıflandırıcısı, Karar Ağaçları ve Yapay Sinir Ağları gibi olasılıksal modeller kullanabilmektedir. Bu teknikler, istatistiksel analizler yoluyla ya da makine öğrenmesi yöntemleri ile temel alınan modeli öğrenerek tavsiyelerde bulunmaktadır. İçerik tabanlı filtreleme yöntemleri, kullanıcı profillerindeki olası değişimlere karşı çok kısa bir süre içinde sunacağı önerileri düzenleme potansiyeline sahiptir. Bu yöntemin en büyük dezavantajı, öğelerin özelliklerine ilişkin derinlemesine bilgi ve açıklamaya ihtiyaç duymasıdır [78].

3.2.2 İşbirlikçi filtreleme

İşbirlikçi filtreleme yöntemleri, film ve müzik gibi kolaylıkla ve yeterince tanımlanamayan içeriklerin bulunduğu uygulama alanlarında kullanılan bir tahmin yöntemidir [79]. İşbirlikçi filtreleme, kullanıcıların öğeler ile ilgili değerlendirmelerinin bulunduğu bir veritabanı (kullanıcı-öğe matrisi) oluşturarak çalışmaktadır. Öneri sunma aşamasında kullanıcı profilleri arasındaki benzerlikleri hesaplayarak kullanıcılarla alakalı ve ilgili tercihleri eşleştirmektedir. Aktif kullanıcıya daha önce görmediği ancak kendi komşuluğundaki kullanıcılar tarafından olumlu olarak değerlendirilen öğeler öneri olarak sunulmaktadır [80]. İşbirlikçi filtreleme tarafından tahminler ve tavsiyeler üretilebilmektedir. Tahmin, *j* ürünü için *i* kullanıcısının yapacağı değerlendirmeyi gösteren R_{ij} ile ifade edilmektedir. Tavsiye ise

Şekil 3'te görüldüğü gibi kullanıcıların en çok beğeneceği *N* elemanlı bir öğe listesidir [81].



Şekil 3. Kullanıcı-öğe değerlendirme matrisi

Kullanıcı tabanlı yöntemler

Kullanıcı tabanlı yöntemlerde, hedef kullanıcıya benzer bir dizi kullanıcı, kullanıcının komşusu olarak belirlenir. Kullanıcılar arasındaki benzerlikler, genellikle olarak iki kullanıcının değerlendirme puanlarını belirten vektörler arasındaki kosinüs benzerliği veya Pearson correlation coefficient kullanılarak hesaplanmaktadır. *k* komşu kullanıcıları ifade etmek üzere *u* kullanıcısının *i* öğesi hakkındaki değerlendirme puanını tahmin etmek için Eş.7'de görülen eşitlik kullanılmaktadır [82].

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{u \in N_u^k(i)} sim(u,v)\,(r_{vi} - \mu_v)}{\sum_{u \in N_u^k(i)} sim(u,v)} \qquad (7)$$

$sim(u,v)$ *u* ve *v* kullanıcıları arasındaki benzerliği, μ_u ve μ_v ise *u* ve *v* kullanıcılarının değerlendirme puanlarının ortalamasını ifade etmektedir. $N_u^i(k)$ ise *i* öğesi hakkında değerlendirmede bulunan, *k* yakın kullanıcılar kümesini ifade etmektedir [83].

Öğe tabanlı yöntemler

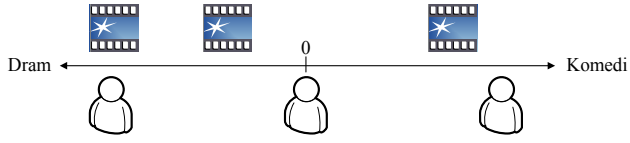
Öğe tabanlı yöntemler, belirli bir kullanıcı tarafından puanlanan öğelerin, hedef öğeye benzerlikleri kullanılarak *u* kullanıcısının hedef öğe üzerindeki değerlendirme puanını hesaplamaktadır [84]. Kullanıcı tabanlı yöntemlere benzer şekilde, öğeler arasındaki benzerliği hesaplamak için kosinüs benzerliği veya Pearson correlation coeffcient kullanılabilir. *u* kullanıcısının *i* öğesi hakkındaki değerlendirme puanı Eş.8'de görülen eşitlik kullanılarak hesaplanabilmektedir.

$$\hat{r}_{ui} = \mu_i + \frac{\sum_{j \in N_i^k(u)} sim(i,j)\,(r_{uj} - \mu_j)}{\sum_{j \in N_i^k(u)} sim(i,j)} \qquad (8)$$

$sim(i,j)$ *i* ve *j* öğeleri arasındaki benzerliği, μ_i ve μ_j ise *i* ve *j* öğeleri için yapılmış değerlendirme puanlarının ortalamasını ifade etmektedir. $N_u^i(k)$ ise *u* kullanıcısının değerlendirmede bulunduğu, *i* öğesine benzer *k* yakın öğeler kümesini ifade etmektedir [85].

Gizli Faktör Modeli Tabanlı İşbirlikçi Filtreleme

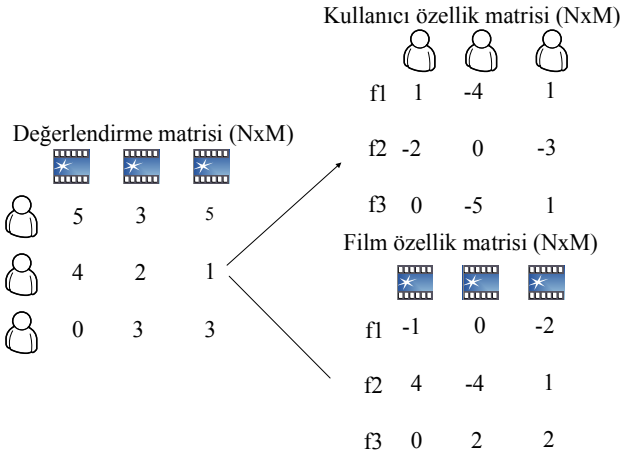
Gizli faktör tabanlı yöntemler, nesnelerin niteliklerini tanımlayan özelliklerin bulunmasını temel almaktadır. Öğe özellikleri ve kullanıcı tercihleri Şekil 4'te görüldüğü gibi sayısal faktör değerleri ile ifade edilmektedir. Kullanıcı değerlendirmelerine dair tahminler ise daha az sayıda parametrenin bir araya getirilmesi ile elde edilen modellerden çıkarılmaktadır [86].



Şekil 4. Gizli faktör modelleri

Gizli faktör modeli tabanlı yaklaşımlarda matris çarpanlara ayırma yöntemleri kullanılmaktadır. Matris çarpanlara ayırma yöntemleri, değerlendirme puanı tahmini için kullanılmaktadır. Kullanıcı değerlendirme matrisini ifade eden *R* kullanılarak *P* ve *Q* kullanıcı ve öğe matrisinin hesaplanabileceği temel alınmaktadır. *u* kullanıcısının *i* öğesi üzerindeki değerlendirme puanı, Şekil 5'te görüldüğü gibi öğeler ve kullanıcılar faktör vektörüyle ilişkilendirilerek Eş. 9 kullanılarak hesaplanmaktadır.

$$\hat{r}_{ui} = P_u^T q_i \qquad (9)$$



Şekil 5. Öğe ve kullanıcı özellik matrisleri

Eş. 9 kullanılarak tamamlanan $\hat{R} = PQ^t$ matrisi, kullanıcıların değerlendirmede bulunmadığı öğeler hakkındaki değerlendirme puanlarının belirlenmesinde kullanılmaktadır [87].

3.2.3 Hibrit yöntemler

Hibrit yöntemler, tavsiye sistemlerinin bazı sınırlamalarını ve sorunlarını ortadan kaldırmak ve daha iyi sistem optimizasyonu elde etmek için farklı öneri tekniklerinin birleştirilmesi ile oluşturulmaktadır [88]. Hibrit tekniklerin ardındaki düşünce, bir algoritmanın dezavantajlarının başka bir algoritma ile ortadan kaldırılacağı ve birden fazla algoritma kombinasyonunun tek bir algoritmadan daha doğru ve etkili tavsiyeler sunabileceğidir. Algoritmaların ayrı ayrı uygulanması ve sonucun birleştirilmesi, işbirlikçi filtreleme yöntemlerinde içerik tabanlı filtreleme yöntemlerini kullanma ya da her iki yaklaşımı bir araya getiren birleşik bir öneri sistemi oluşturma yoluyla hibrit sistemler oluşturulabilmektedir [89].

3.3 Bağlamsal Öneriler

Kullanıcıların değerlendirme puanlarının dışında, öneri kalitesini artırmak için kullanılabilecek çok miktarda bağlamsal veri mevcuttur [90]. Bağlamsal veriler zaman, konum ya da kullanıcılar, öğeler veya değerlendirmelerle ilişkili ek bilgiler olabilmektedir. Bağlamsal öneriler, her bağlamı farklı bir boyut olarak ele alarak ilişkilerin modellenmesini sağlamaktadır. Bu sayede kullanıcılar ve öğelerden oluşan iki boyutlu bir değerlendirme matrisi yerine çok boyutlu bir ilişki modeli (kullanıcı, öğe, zaman, konum gibi) oluşturmaktadır [91].

3.4 Büyük Veriler İçin Komşuluk Tabanlı Yaklaşımların Ölçeklenmesi

Filtreleme ve yaklaşık en yakın komşu tabanlı yaklaşımlar, hesaplanmış benzerliklerin sayısını azaltmak için veri yapılarını kullanmaktadır [92]. Diğer yaklaşımlar, yüksek kalitede sonuç garantisi olmaksızın komşuları belirlemek için bir kullanıcı veya öğe alt kümesi seçmektedir. Performans artışı, çok kanallı (multithread) ve dağıtık sistemler aracılığıyla gerçekleştirilmektedir. Komşulukların belirlenmesi için k-en yakın komşu ve e-en yakın komşu yaklaşımları kullanılmaktadır. K-en yakın komşu (kNN) yaklaşımı, bir sorgu nesnesine en yakın komşulukta olan k nesneyi bulmaya amaçlamaktadır. e-en yakın komşu (eNN) veya benzerlik arama yaklaşımı, yapılan sorguya en az e benzerliğine sahip tüm nesnelerin bulunmasını amaçlamaktadır [93].

Günümüzde, veri miktarındaki artış ile birlikte komşu olamayan nesne çiftlerinin göz ardı edildiği veya filtrelendiği seyrek vektörler için en yakın komşu yöntemleri önerilmiştir [94]. Bir kullanıcıyı veya öğe değerlendirmesini temsil eden vektör, kullanıcıların genellikle öğelerin çoğunu değerlendirmediği için seyrektir. Arama yöntemleri, ters indeks yapısı kullanarak hiçbir özelliği olmayan nesnelerin karşılaştırılmasını önler [95]. İndeks yapısı, tüm nesneler arasındaki her özellik için bir tane olmak üzere bir dizi oluşturur. Bu sayede *j*. özellik için sıfırdan farklı bir $r_{i,j}$ değere sahip *i* öğelerinden ve (i, $r_{i,j}$) ikililerinden oluşan *j* listesi elde edilmektedir [96].

Komşulukların belirlenmesi için kullanılan yöntemler bellek tabanlı çalışmaktadır. Bellek paylaşımlı paralel veri işleme yöntemleri iş parçacıklarının çalışma süresinin ve kaynak rekabetinin minimize edilmesini amaçlamaktadır. En yakın komşuların belirlenmesi için mevcut olan dağıtık çözümler genellikle MapReduce çerçevesini kullanmaktadır. MapReduce çerçevesiyle birlikte nesneler daha küçük alt gruplara bölünerek blok çiftleri arasında en yakın komşu arama yöntemleri uygulanabilmektedir. Bazı blok karşılaştırmaları, blok düzeyinde filtreleme tekniklerine dayanarak ortadan kaldırılabilmektedir [97].

3.5 Değerlendirme Puanı Tahmini

Tavsiye sistemleri temel olarak kullanıcıların belirli öğeler hakkında yaptıkları değerlendirme puanlarını kullanarak öğelerin geri kalanı için kullanıcı değerlendirme puanlarını tahmin etmeyi hedeflemektedir [98]. *n* kullanıcı ve *m* öğeden oluşan bir sistem, *u* kullanıcılarının *i* öğeleri için yapmış olduğu r_{ui} değerlendirmelerini içeren $n \times m$ boyutunda bir *R* matrisi ile ifade edilmektedir. *R* matrisi, kullanıcıların henüz değerlendirmede bulunmadıkları öğeler sebebiyle seyrektir [55].

Matris ayrıştırma teknikleri, kullanıcı-öğe değerlendirme matrisini düşük seviyeli matrislere bölmek için kullanılmaktadır. Bu sayede kullanıcı-öğe değerlendirme matrislerindeki eksik değerlerin tamamlanması hedeflenmektedir. Tavsiye sistemlerinde matris çarpanlarına ayırma yöntemlerinin kullanılabilmesi için least squares ve gradient descent algoritmaları kullanılmaktadır [99]. Least squares, büyük ölçekli verilere uygulanan matris tamamlama algoritmalarından biridir. *m* kullanıcı sayısını, *n* ise öğe sayısını ifade etmek üzere oluşturulacak faktör vektörleri ile öğeler ve kullanıcılar temsil edilebilmektedir. Least squares metodunun amacı, bir fonksiyona uyan parametrelerin bir tahmini bulmaktır [100]. Gradient descent metodu ise makine öğrenmesi alanında yaygın olarak kullanılan ve düşük hesaplama karmaşıklığı için çok sayıda yineleme yapan bir optimizasyon algoritmasıdır. Temel olarak, optimizasyon değişkenleri için bir maliyet fonksiyonunun ve başlangıç değerlerinin varlığını varsayan, basit ve yinelemeli bir optimizasyon işlemi gerçekleştirir. Gradient descent algoritmasında amaç bir fonksiyonun minimum noktasını bulmaktır [101].

3.6 Tavsiye Sistemlerinin Değerlendirilmesinde Kullanılan Metrikler

Tavsiye algoritmalarının kalitesini ölçmek için kullanılacak metrik, algoritmaya göre değişebilmektedir [102]. Tavsiye sistemlerinin doğruluğunu ölçen metrikler istatistiksel ölçümler ve karar destek ölçümleri olmak üzere ikiye ayrılmaktadır. Her metriğin uygunluğu, veri kümesinin özelliklerine ve sistemin yapacağı görev türlerine bağlıdır [103].

İstatistiksel doğruluk ölçümleri, tahmin edilen değerlendirme puanlarını gerçek kullanıcıların değerlendirme puanlarıyla doğrudan karşılaştırarak kullanılan tavsiye algoritmasının doğruluğunu değerlendirmektedir. Ortalama mutlak hata (Mean Absolute Error-MAE), ortalama karesel hatanın karekökü (Root Mean Squared Error-RMSE) ve korelasyon genellikle istatistiksel doğruluk metrikleri olarak kullanılmaktadır. Ortalama mutlak hata, önerinin kullanıcının özel değerinden sapmasının ölçüsünü ifade etmektedir ve Eş. 10'da görüldüğü gibi hesaplanmaktadır.

$$MAE = \frac{1}{N}\sum_{u,j} | p_{u,j} - r_{u,j} | \qquad (10)$$

$p_{u,i}$, *u* kullanıcısı ve *i* öğesi için tahmin puanını, $r_{u,i}$ *u* kullanıcısının *i* öğesi için gerçekte vermiş olduğu değerlendirme puanını ve *N* değeri verisetindeki toplam değerlendirme puanı sayısını ifade etmektedir. MAE değeri düştükçe, tavsiye sistemi kullanıcı değerlendirmelerini daha doğru bir şekilde tahmin etmektedir. Ortalama karesel hatanın karekökü Eş. 11'de görüldüğü gibi, ortalama mutlak hata metriğine benzerdir ancak daha büyük sapmalara daha yüksek ağırlık vermektedir [104].

$$RMSE = \sqrt{\frac{1}{n}\sum_{u,i}(p_{u,i} - r_{u,i})^2} \qquad (11)$$

Karar destek ölçümleri ise hassasiyet (Precision), duyarlılık (Recall) ve F ölçütü' dür (F – Measure). Bu ölçütler, kullanıcıların sistemde mevcut öğeler arasından yüksek kalitede olan öğeleri seçmelerine yardımcı olmaktadır. Hassasiyet, Eş. 12'de görüldüğü gibi öneri olarak sunulan öğelerin kullanıcı ile gerçekten ilgili olup olmadığını belirlemektedir [105].

$$Hassasiyet = \frac{\text{Öneri listesinden seçilen öğe sayısı}}{\text{Öneri listesinin boyutu}} \qquad (12)$$

Duyarlılık, Eş. 13'te görüldüğü gibi kullanıcıların seçtikleri öğelerin kaç tanesinin kendisine öneri olarak sunulduğunu belirlemektedir.

$$Duyarlılık = \frac{\text{Öneri listesinden seçilen öğe sayısı}}{\text{Seçilen toplam öğe sayısı}} \qquad (13)$$

F-ölçütü ise Eş. 14'te görüldüğü gibi hassasiyet ve duyarlılık metriklerinin tek bir metrik içerisinde hesaplanmasını sağlamaktadır [106].

$$F\ ölçütü = \frac{2*Hassasiyet*Duyarlılık}{Hassasiyet+Duyarlılık} \qquad (14)$$

IV. SONUÇLAR VE ÖNERİLER

Tavsiye sistemleri, e-ticaret başta olmak üzere Web üzerinde hizmet veren çoğu platformda uzun zamandır kullanılmaktadır. Tavsiye sistemleri kullanıcıların ilgilerini çekebilecek kişiselleştirilmiş içerikleri öneri olarak sunarak kullanıcılar üzerindeki aşırı bilgi yüklemesi sorununun hafifletilmesini hedeflemektedir. Büyük veri çağında, tavsiye sistemlerinde artan kullanıcı tıklamaları ve ürün miktarı ile arka plandaki veri hacminin artması sebebiyle aynı verilerin tekrarlı bir şekilde işlenmesi mümkün olmamaktadır. Bir veri öğesinin en fazla bir defa işlenebilmesi ise arka plandaki klasik tavsiye sistemi algoritmaları üzerinde kısıtlamalara neden olmaktadır. Bu sebeple tavsiye sistemlerinde kullanılan geleneksel veri madenciliği yaklaşımları etkisiz kalmaktadır.

Bu çalışma kapsamında büyük verilerin tavsiye sistemlerinde kullanıldığı çalışmalar analiz edilmiş, tavsiye sistemleri büyük veri bakış açısından değerlendirilerek büyük veri ve büyük veri analizinde kullanılan yöntemler kapsamlı bir şekilde incelenmiştir. Tavsiye sistemlerinde kullanılan komşuluk tabanlı yaklaşımlar ve gizli faktör modeli tabanlı

yaklaşımlar, geleneksel tavsiye sistemleri ve büyük veri tavsiye sistemleri açısından incelenmiştir.

Yapılan literatür araştırmaları sonucunda, yapılan çalışmaların paralel işleme platformları olan MapReduce ve Spark kullanılarak geliştirildiği görülmüştür. Klasik sosyal ağ madenciliği, tavsiye sistemleri ve doğal dil işleme gibi veri analitiği uygulamalarında kullanılan matris tabanlı yöntemlerin, büyük verilerin dağıtık yapıları sebebiyle yetersiz kaldığı, bu sebeple paralel ve dağıtık veri işleme ortamlarının büyük veri çağında ön plana çıktığı görülmüştür.

Büyük veri ve tavsiye sistemleri konusunda çalışma yapacak araştırmacılara, statik büyük veri kümeleri için paralel işleme teknikleri önerilmektedir. Mevcut kullanıcı ve ürün değerlendirmelerin tamamı kullanılarak daha etkin, kullanışlı ve hızlı öneriler elde edilebilir. Akan veriler ile gerçekleştirilecek tavsiye sistemlerinde ise verilerinin tamamının tekrarlı bir şekilde işlenmesi mümkün olmayacağı için gelen veri örneklerine göre güncellenebilecek bir karar modelinin kullanılması önerilmektedir.

KAYNAKLAR

[1] Muthukrishnan, S. (2005). Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2), 117-236.

[2] Isinkaye, F., Folajimi, Y. ve Ojokoh B. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 261-273.

[3] Amatriain, X. (2013). Mining large streams of user data for personalized recommendations. *ACM SIGKDD Explorations Newsletter*, 14(2), 37-48.

[4] Subbian, K., Aggarwal, C. ve Hegde, K. (2016). Recommendations for streaming data. *In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2185-2190.

[5] Anceaume, E., Busnel, Y. ve Rivetti, N. (2015). Estimating the Frequency of Data Items in Massive Distributed Streams. *In Network Cloud Computing and Applications (NCCA), 2015 IEEE Fourth Symposium*, 59-66.

[6] Werner, S. ve Lommatzsch, A. (2014). Optimizing and Evaluating Stream-based News Recommendation Algorithms. *In CLEF (Working Notes)*, 813-824.

[7] Ludmann, C.A. (2015). Online Recommender Systems based on Data Stream Management Systems. *In Proceedings of the 9th ACM Conference on Recommender Systems*, 391-394.

[8] Lommatzsch, A. ve Albayrak, S. (2015). Real-time recommendations for user-item streams. *In Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 1039-1046.

[9] Chen, C., Yin, H., Yao, J. ve Cui, B. (2013). Terec: A temporal recommender system over tweet stream. *Proceedings of the VLDB Endowment*, 6(12), 1254-1257.

[10] Sri, P.A. ve Anusha, M. (2016). Big Data-Survey. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 4(1), 74-80.

[11] Beyer, M. (2011). Gartner Says Solving'Big Data'Challenge Involves More Than Just Managing Volumes of Data. *Gartner. Archived from the original on*.

[12] TechAmerica Foundation's Federal Big Data Commission. (2012). Demystifying bigdata. *A practical guide to transforming the business of Government*".

[13] Chen, C.P. ve Zhang, C.Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, 275, 314-347.

[14] Cukier, K. (2010). Data, data everywhere: A special report on managing information. *Economist Newspaper*.

[15] Khan, N., Yaqoob, I., Hashem, I.A.T, Inayat, Z., Mahmoud, A.W.K., Alam, M. ve Gani, A. (2014). Big data: survey, technologies, opportunities, and challenges. *The Scientific World Journal*.

[16] Ward, J.S. ve Barker, A. (2013). Undefined by data: a survey of big data definitions. *arXiv preprint*.

[17] Chen, M., Mao, S. ve Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2), 171-209.

[18] LaValle, S., Lesser, E., Shockley, R., Hopkins, M.S. ve Kruschwitz, N. (2011). Big data, analytics and the path from insights to value. *MIT sloan management review*, 52(2), 21.

[19] Sagiroglu, S. ve Sinanc, D. (2013). Big data: A review. *In Collaboration Technologies and Systems (CTS)*, 42-47.

[20] Gandomi, A. ve Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2), 137-144.

[21] Tsai, C.W, Lai, C.F., Chao, H.C. ve Vasilakos, A.V. (2015). Big data analytics: A survey. *Journal of Big Data*, 2(1), 21.

[22] Zikopoulos, P. ve Eaton, C. (2011). Understanding big data: Analytics for enterprise class hadoop and streaming data. *McGraw-Hill Osborne Media*.

[23] Snijders, C., Matzat, U. ve Reips, U.D. (2012). Big Data: Big gaps of knowledge in the field of internet science. *International Journal of Internet Science*, 7(1), 1-5.

[24] Liu, X., Iftikhar, N. ve Xie, X. (2014). Survey of real-time processing systems for big data. *In Proceedings of the 18th International Database Engineering & Applications Symposium*, 356-361.

[25] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y. ve Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, 2(3), 267-279.

[26] Verma, J.P., Agrawal, S., Patel, B. ve Patel, A. (2016). Big data analytics. *Challenges and applicatıons for text, audio, video, and social media data*.

[27] Evelson, B.(2015). Vendor Landscape: Big Data Text Analytics. *For Application Development & Delivery Professionals*.

[28] Lohr, S. (2012). The age of big data. *New York Times*, 11(2012).

[29] Russom, P. (2013). Managing big data. *TDWI Best Practices Report, TDWI Research*, 1-40.

[30] Snoek, C.G., Worring, M. ve Smeulders, A.W. (2005). Early versus late fusion in semantic video analysis. *In Proceedings of the 13th annual ACM international conference on Multimedia*, 399-402.

[31] Davenport, T.H., Barth, P. ve Bean, R. (2012). How big data is different. *MIT Sloan Management Review*, 54(1), 43.

[32] Cambria, E., Rajagopal, D., Olsher, D. ve Das, D. (2013). Big social data analysis. *Big data computing*, 2013, 401-414.

[33] Tufekci, Z. (2014). Big questions for social media big data: Representativeness, validity and other methodological pitfalls. *arXiv preprint*.

[34] Manovich, L. (2011). Trending: The promises and the challenges of big social data. *Debates in the digital humanities*, 2, 460-475.

[35] Bravo-Marquez, F., Mendoza, M. ve Poblete, B. (2014). Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems*, 69, 86-99.

[36] Cambria, E., Wang, H. ve White, B. (2014). Guest editorial: Big social data analysis. *Knowledge-Based Systems*, (69), 1-2.

[37] Lazer, D., Kennedy, R., King, G. ve Vespignani, A. (2014). The parable of Google Flu: Traps in big data analysis. *Science*, 343(6176), 1203-1205.

[38] Chen, H., Chiang, R.H. ve Storey, V.C. (2012). Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4), 1165-1188.

[39] Laurila, J.K., Gatica-Perez, D., Aad, I., Bornet, O., Do, T.M.T., Dousse, O. ve Miettinen, M. (2012). The mobile data challenge: Big data for mobile computing research. *In Pervasive Computing*.

[40] Hashem, I.A.T, Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A. ve Khan, S.U. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47, 98-115.

[41] Wang, G., Ng, T.S. ve Shaikh, A. (2012). Programming your network at run-time for big data applications. *In Proceedings of the first workshop on Hot topics in software defined networks*, 103-108.

[42] Fan, W. ve Bifet, A. (2013). Mining big data: current status, and forecast to the future. *ACM sIGKDD Explorations Newsletter*, 14(2), 1-5.

[43] Dittrich, J., Quiané-Ruiz, J.A. (2012). Efficient big data processing in Hadoop MapReduce. *Proceedings of the VLDB Endowment*, 5(12), 2014-2015.

[44] Katal, A., Wazid, M. ve Goudar, R.H. (2013). Big data: issues, challenges, tools and good practices. *In Contemporary Computing (IC3), 2013 Sixth International Conference on*, 404-409.

[45] Wang, Y. (2016). Stream Processing Systems Benchmark. *StreamBench*.

[46] Inoublia, W., Aridhib, S., Meznic, H. ve Jungd, A. (2016). An Experimental Survey on Big Data Frameworks. *arXiv preprint*.

[47] Grolinger, K., Hayes, M., Higashino, W.A., L'Heureux, A., Allison, D.S. ve Capretz, M.A. (2014). Challenges for mapreduce in big data. *In Services (SERVICES), 2014 IEEE World Congress on*, 182-189.

[48] Karau, H., Konwinski, A., Wendell, P. ve Zaharia, M. (2015). Learning spark: lightning-fast big data analysis. *O'Reilly Media, Inc*.

[49] Reyes-Ortiz, J.L., Oneto, L. ve Anguita, D. (2015). Big data analytics in the cloud: Spark on hadoop vs mpi/openmp on beowulf. *Procedia Computer Science*, 53, 121-130.

[50] Madden, S. (2016). From databases to big data. *IEEE Internet Computing*, 16(3), 4-6.

[51] Landset, S., Khoshgoftaar, T.M., Richter, A.N. ve Hasanin, T. (2015). A survey of open source tools for machine learning with big data in the Hadoop ecosystem. *Journal of Big Data*, 2(1), 24.

[52] Markl, V. (2014). Breaking the chains: On declarative data analysis and data independence in the big data era. *Proceedings of the VLDB Endowment*, 7(13), 1730-1733.

[53] Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S. ve Tzoumas, K. (2015). Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 36(4).

[54] Zhang, Y. (2016). GroRec: a group-centric intelligent recommender system integrating social, mobile and big data technologies. *IEEE Transactions on Services Computing*, 9(5), 786-795.

[55] Anastasiu, D.C., Christakopoulou, E., Smith, S., Sharma, M. ve Karypis, G. (2016). Big Data and Recommender Systems.

[56] Bobadilla, J., Ortega, F., Hernando, A. ve Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-based systems*, 46, 109-132.

[57] Ricci, F., Rokach, L. ve Shapira, B. (2011). Introduction to recommender systems handbook. *Springer US*.

[58] Chen, K., Chen, T., Zheng, G., Jin, O., Yao, E., & Yu, Y. (2012). Collaborative personalized tweet recommendation. *In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 661-670.

[59] Meng, S., Dou, W., Zhang, X. ve Chen J. (2014). KASR: A Keyword-Aware Service Recommendation method on MapReduce for big data applications". *IEEE Transactions on Parallel and Distributed Systems*, 25(12), 3221-3231.

[60] Weider, D.Y., Pratiksha, C., Swati, S., Akhil, S. ve Sarath, M. (2015). A Modeling Approach to Big Data Based Recommendation Engine in Modern Health Care Environment. *In Computer Software and Applications Conference (COMPSAC)*, 75-86.

[61] Gu, R., Tang, Y., Wang, Z., Wang, S., Yin, X., Yuan, C. ve Huang, Y. (2015). Efficient large scale distributed matrix computation with spark. *In Big Data (Big Data), 2015 IEEE International Conference on*, 2327-2336.

[62] Verma, J.P., Patel, B. ve Patel, A. (2015). Big data analysis: recommendation system with Hadoop framework. *In Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on*, 92-97.

[63] Dai, J., Yang, B., Guo, C. ve Ding, Z. (2015). Personalized route recommendation using big trajectory data. *In Data Engineering (ICDE), 2015 IEEE 31st International Conference*, 543-554.

[64] Huang, Y., Cui, B., Zhang, W., Jiang, J. ve Xu, Y. (2015). Tencentrec: Real-time stream recommendation in practice. *In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 227-238.

[65] Riyaz, P.A. ve Varghese, S.M. (2016). A Scalable Product Recommendations Using Collaborative Filtering in Hadoop for Bigdata. *Procedia Technology*, 24, 1393-1399.

[66] Shang, S., Shi, M., Shang, W. ve Hong, Z. (2016). A micro-video recommendation system based on big data. *In Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on*, 1-5.

[67] Chang, S., Zhang, Y., Tang, J., Yin, D., Chang, Y., Hasegawa-Johnson, M.A. ve Huang, T.S. (2016). Streaming Recommender Systems. *arXiv preprint*.

[68] Prando, A.V., Contratres, F., Souza, S., ve De Souza, L. (2017). Content-based recommender system using social networks for cold-start users. *In Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 181-189.

[69] Ajantha, D., Vijay, J. ve Sridhar, R. (2017). A user-location vector based approach for personalised tourism and travel recommendation. *In Big Data Analytics and Computational Intelligence (ICBDAC), 2017 International Conference on,* 440-446.

[70] Zhou, T., Chen, L. ve Shen, J. (2017). Movie Recommendation System Employing the User-Based CF in Cloud Computing. *In Computational Science and Engineering (CSE) and Embedded and Ubiquitous Computing (EUC), 2017 IEEE International Conference on,* 46-50.

[71] Seo, Y. D., Kim, Y. G., Lee, E. ve Baik, D. K. (2017). Personalized recommender system based on friendship strength in social network services. *Expert Systems with Applications*, 69, 135-148.

[72] Wei, J., He, J., Chen, K., Zhou, Y. ve Tang, Z. (2017). Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications*, 69, 29-39.

[73] Lee, T.Q., Park, Y. ve Park, Y.T. (2008). A time-based approach to effective recommender systems using implicit feedback. *Expert systems with applications*, 34(4), 3055-3062.

[74] Hu, Y., Koren, Y. ve Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. *In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, 263-272.

[75] Núñez-Valdéz, E.R., Lovelle, J.M.C., Martínez, O.S., García-Díaz, V., de Pablos, P.O. ve Marín, C.E.M. (2012). Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, 28(4), 1186-1193.

[76] Jawaheer, G., Weller, P. ve Kostkova, P. (2014). Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(2).

[77] Karatzoglou, A., Baltrunas, L. ve Shi, Y. (2013). Learning to rank for recommender systems. *In Proceedings of the 7th ACM conference on Recommender systems*, 493-494.

[78] Lops, P., De Gemmis, M. ve Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. *In Recommender systems handbook*, 73-105.

[79] Tatiya, R.V. ve Vaidya, A.S. (2011). A Survey of Recommendation Algorithms. *IOSR Journals (IOSR Journal of Computer Engineering*, 1(16), 16-19.

[80] Ekstrand, M.D., Riedl, J.T. ve Konstan, J.A. (2011). Collaborative filtering recommender systems. *Foundations and Trends in Human–Computer Interaction*, 4(2), 81-173.

[81] Tso-Sutter, K.H., Marinho, L.B. ve Schmidt-Thieme, L. (2008). Tag-aware recommender systems by fusion of collaborative filtering algorithms. *In Proceedings of the 2008 ACM symposium on Applied computing*, 1995-1999, 2008.

[82] Hameed, M.A, Al Jadaan, O. ve Ramachandram, S. (2012). Collaborative filtering based recommendation system: A survey. *International Journal on Computer Science and Engineering*, 4(5), 859.

[83] Bobadilla, J., Ortega, F., Hernando, A. ve Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-based systems*, 24(8), 1310-1316.

[84] Pham, M.C., Cao, Y., Klamma, R. ve Jarke, M. (2011). A clustering approach for collaborative filtering recommendation using social network analysis. *J. UCS*, 17(4), 583-604.

[85] Gao, M., Wu, Z. ve Jiang, F. (2011). Userrank for item-based collaborative filtering recommendation. *Information Processing Letters*, 111(9), 440-446.

[86] Schafer, J. B., Frankowski, D., Herlocker, J. ve Sen, S. (2007). Collaborative filtering recommender systems. In The adaptive web, Springer, Berlin, Heidelberg 291-324.

[87] Koren, Y. ve Bell, R. (2015). Advances in collaborative filtering. *In Recommender systems handbook,* 77-118, 2015.

[88] Wang, C. ve Blei, D.M. (2011). Collaborative topic modeling for recommending scientific articles. *In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 448-456.

[89] Klašnja-Milićević, A., Vesin, B., Ivanović, M. ve Budimac, Z. (2011). E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 56(3), 885-899.

[90] Porcel, C., Tejeda-Lorente, A., Martínez, M.A. ve Herrera-Viedma, E. (2012). A hybrid recommender system for the selective dissemination of research resources in a technology transfer Office. *Information Sciences*, 184(1), 1-19.

[91] Elahi, M., Ricci, F. ve Rubens, N. (2016). A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*.

[92] Adomavicius, G. ve Tuzhilin, A. (2015). Context-aware recommender systems. *In Recommender systems handbook*, 191-226.

[93] Desrosiers, C. ve Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, 107-144.

[94] Zhou, X., Xu, Y., Li, Y., Josang, A. ve Cox, C. (2012). The state-of-the-art in personalized recommender systems for social networking. *Artificial Intelligence Review*, 37(2), 119-132.

[95] Cacheda, F., Carneiro, V., Fernández, D. ve Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)*, 5(1).

[96] Yang, X., Guo, Y., Liu, Y. ve Steck H. (2014). A survey of collaborative filtering based social recommender systems. *Computer Communications*, 41, 1-10.

[97] Noulas, A., Scellato, S., Lathia, N. ve Mascolo, C. (2012). A random walk around the city: New venue recommendation in location-based social networks. *In Privacy, security, risk and trust (PASSAT), 2012 international conference on and 2012 international confernece on social computing*, 144-153.

[98] Schelter, S., Boden, C. ve Markl, V. (2012). Scalable similarity-based neighborhood methods with MapReduce. *In Proceedings of the sixth ACM conference on Recommender systems*, 163-170.

[99] Gantner, Z., Rendle, S., Freudenthaler, C. ve Schmidt-Thieme, L. (2011). MyMediaLite: A free recommender system library. *In Proceedings of the fifth ACM conference on Recommender systems*, 305-308.

[100] Takács, G. ve Tikk, D. (2012). Alternating least squares for personalized ranking. *In Proceedings of the sixth ACM conference on Recommender systems*, 83-90.

[101] Yu, H.F., Hsieh, C.J., Si, S. ve Dhillon, I. (2012). Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. *In Data Mining (ICDM), 2012 IEEE 12th International Conference on*, 765-774.

[102] Gemulla, R., Nijkamp, E., Haas, P.J. ve Sismanis, Y. (2011). Large-scale matrix factorization with distributed stochastic gradient descent. *In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 69-77.

[103] Lü, L., Medo, M., Yeung, C.H., Zhang, Y.C, Zhang, Z.K. ve Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1), 1-49.

[104] Shani, G. ve Gunawardana, A. (2011). Evaluating recommendation systems. *Recommender systems handbook*, 257-297.

[105] Bellogin, A., Castells, P. ve Cantador, I. (2011). Precision-oriented evaluation of recommender systems: An algorithmic comparison. *In Proceedings of the fifth ACM conference on Recommender systems*, 333-336.

[106] Zaier, Z., Godin, R. ve Faucher, L. (2008). Evaluating recommender systems. *In Automated solutions for Cross Media Content and Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on*, 211-217.