Erzincan Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
2019, 12(1), 379-410
ISSN: 1307-9085, e-ISSN: 2149-4584
Derleme

Erzincan University
Journal of Science and Technology
2019, 12(1), 379-410
DOI: 10.18185/erzifbed.473008
Review

Akan Veri Karakterizasyonu, Üretimi ve Analitiği Üzerine Kapsamlı Bir İnceleme

Anıl UTKU*, M. Ali AKCAYOL

Gazi Üniversitesi Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Maltepe/ANKARA

Geliş / Received: 21/10/2018, Kabul / Accepted: 08/03/2019

Öz

Gelişen donanım ve yazılım teknolojileri sayesinde anlık olarak üretilen veri miktarı ve hızı giderek artmaktadır. Akan veriler, statik verilerden farklı olarak, potansiyel olarak sonsuz, hızlı, zamanla değişen ve gelişen bir yapıya sahiptir. Bu sebeple akan verilerin tamamını depolamak zaman ve depolama alanı kısıtları sebebiyle imkânsız olabilmektedir. Akan veri madenciliğinde kullanılan yöntemlerin, statik veri madenciliğinden farklı olarak akan verilerin doğasına uygun bir şekilde optimize edilmesi gerekmektedir. Bu çalışma kapsamında akan veriler, statik veriler ile karşılaştırmalı olarak analiz edilmiştir. Akan veri madenciliği, akan veri karakterizasyonu ve akan veri üretimi üzerine detaylı araştırmalar yapılmıştır. Akan verilerin türleri, kaynakları ve akan veriden öğrenme yöntemleri ile akan veri uygulamalarında kullanılan yöntemler ve algoritmalar kapsamlı bir şekilde incelenmiştir. Yapılan literatür araştırmaları ve incelemeler sonucunda, veri gizliliği, geleceğe dönük öngörülerin oluşturulması, veri ön işleme, kaynak kullanımı ve çevrimiçi güncelleme konularının geliştirilecek akan veri uygulamalarında değerlendirilmesi gereken konular olduğu ortaya konulmuştur.

Anahtar Kelimeler: Akan Veri, Akan Veri Karakterizasyonu, Akan Veri Üretimi, Akan Veri Analitiği

A Comprehensive Study on Stream Data Characterization, Generation and Analytics

Abctract

Thanks to the ever-evolving hardware and software technologies, the amount and speed of data produced instantly is increasing. The streaming data, unlike static data, has a potentially infinite, fast, changing and evolving structure. For this reason, storing all the streamingdata can be impossible due to time and storage space constraints. The methods used in streamingdata mining need to be optimized in accordance with the nature of the streamingdata unlike static data mining. In this study, streaming data was analyzed comparatively with the static data. Detailed research has been done on streamingdata mining, streamingdata characterization and streamingdata generation. The types, sources, and methods of streaming data and the methods and algorithms used in streamingdata applications are studied extensively.

Keywords: Streaming Data, Streaming Data Characterization, Streaming Data Generation, Streaming Data Analytics

1. Giriş

Günümüzde, sosyal ağlar, sağlık, pazarlama ve finans gibi heterojen kaynaklardan büyük miktarda veri üretilmektedir. Nesnelerin İnterneti, bulut bilişim ve mobil cihaz teknolojilerinin yaygınlaşması, üretilen veri miktarına katkıda bulunmaktadır. Artan veri miktarı, verinin yönetimi için kaynaklar, yeni yöntemler ve güçlü teknolojiler gerektirmektedir (Hurwitz vd., 2013).

Sensör ağları, kredi kartı işlemleri, stok yönetimi, blog gönderileri ve sosyal ağlar gibi gerçek dünya uygulamaları tarafından büyük miktarlarda veri üretilmektedir (Wrench vd., 2016). Veri madenciliği yöntemleri, büyük veri kümelerinde ve akan verilerdeki örüntüleri keşfetmek için önemlidir. Ancak büyük verilerinin boyutları, hızları ve değişkenlikleri nedeniyle birliktelik kuralları, kümeleme ve sınıflandırma gibi geleneksel veri madenciliği tekniklerini kullanarak verileri analiz etmek ve kalıcı olarak saklamak mümkün değildir. Bu nedenle, analitik tekniklerin optimize edilerek, veri kaynaklarını sınırlı kaynaklarla ve sınırlı süreler içinde işleyerek gerçek

*Sorumlu Yazar: anilutku@gazi.edu.tr

zamanlı sonuçların üretilmesi gerekmektedir (Krawczyk vd., 2017).

Artan veri miktarıyla birlikte verilerin kalıcı olarak depolandığı uygulamalardan ziyade geçici akan veriler olarak modellendiği uygulamalar kullanılmaktadır. Finansal uygulamalar, ağ izleme, güvenlik, telekomünikasyon veri yönetimi, Web uygulamaları, imalat ve sensör ağları bu tür uygulamalara örnek olarak verilebilir. Akan veri modelinde veri öğeleri, ağ ölçümleri, çağrı kayıtları, Web sayfası ziyaretleri ve sensör okumaları olabilmektedir. Bununla birlikte, çoklu, hızlı ve zamanla değişen, öngörülemeyen ve sınırsız boyutlardaki sürekli akan veriler, verilerin nasıl edileceği ile ilgili yeni araştırma problemlerini ortaya çıkarmaktadır.

Akan veriler, statik verilerden farklı olarak potansiyel olarak sonsuz, hızlı, zamanla değişen ve öngörülemeyen bir yapıya sahiptir. Bu sebeple akan veri analizlerinde kullanılacak yöntemlerin, statik veri analizlerinde kullanılan yöntemlerden farklı olarak verilerin değişimine ve gelişimine göre optimize edilmiş olması gerekmektedir. Bu çalışma kapsamında akan veri madenciliği, akan veri karakterizasyonu ve akan veri üretimi üzerine kapsamlı bir araştırma yapılmıştır. Akan veri kaynakları, akan verilerin karakteristik özellikleri, akan verilerin türleri ve akan veriden öğrenme çerçeveleri, akan veri madenciliğinde kullanılan yaklaşımlar, akan verilerde makine öğrenmesi, akan veri madenciliği uygulamaları ve kullanılan yöntemler ile akan veri karakterizasyonu ve akan veri üretimi konuları detaylı bir şekilde incelenmiştir.

2. Akan Verinin Temelleri

Akan veriler, İnternet teknolojilerindeki gelişmeler ve üretilen veri hacminin artması nedeniyle önemli bir araştırma alanı haline gelmiştir (Wrench vd., 2016). Akan veriler, büyük hacimlere sahiptir ve gerçek zamanlı uygulamalarda eylemlerin veya olayların durumunu tanımlamaktadır. Amaçları ve kaynakları ile ayırt edilen akan veriler, bir nesnenin özelliğinin veya durumunun izlendiği ölçüm akışları ve genellikle iki nesne veya kullanıcı arasındaki işlemlerin izlendiği olay akışları olarak tanımlanabilmektedir. Ölçüm akışları, imalatta kullanılan bir makineden ya da belirli bir ortamda bulunan değişimleri ölçen sensörlerden elde edilen verileri ifade etmektedir. Olay akışları ise bir Web sitesi aracılığıyla kullanıcıların takip edilerek davranış örüntülerinin belirlenmesi gibi tıklama akışlarını ifade etmektedir (Krawczyk vd., 2017). Olay akışlarına, diğer bir yaygın uygulama alanı olan borsadaki finansal veriler örnek olarak verilebilir. Finansal akan verileri analiz etmek veya sadece bir akışı diğerine karşı izlemek, alan uzmanlarının bir hisse senedinin nasıl yükseleceği veya düşeceği konusunda tahminlerde bulunmalarını sağlar (Hurwitz vd., 2013).

Akan veriler, statik verilere göre benzersiz özelliklere sahip veri kümeleridir. Bu özelliklerden en önemlisi akan verilerin potansiyel olarak sonsuz olan hacimleridir. Akan veriler statik veriler arasındaki farklılıklar Tablo 1'de görülmektedir. Gerçek zamanlı gözetim sistemleri, iletişim ağları, İnternet trafiği, finansal piyasalar ve e-ticaret alanlarındaki çevrimiçi işlemler, elektrik dağıtım şebekeleri, endüstri üretim süreçleri, bilimsel deneyler ve uzaktan algılama sensörleri gibi dinamik ortamlar tarafından potansiyel olarak sonsuz miktarda akan veri üretilmektedir. Akan veriler için hacim, hız, doğruluk ve çeşitlilik kavramları ön plana çıkmaktadır.Hız kavramı, algoritmanın akışı yalnızca bir kez işleyebileceği bir hızda veri üretilmesini ifade etmektedir. Akan veriler gerçek zamanlı veriler olduğu için kullanılan algoritma, veriler üzerinden birden fazla kez geçiş gerektiriyorsa oluşturulan model güncel olmayacaktır. Hacim kavramı, hız ile yakından ilişkili bir kavram olarak, akan verilerin toplam hacminin bilinmemesini ve tamamen işlemek için çok büyük olmasını ifade etmektedir.Büyük hacimleri nedeniyle akan verinin tamamını depolamak ya da

çoklu zaman aralıklarıyla taramak imkânsız olabilmektedir(Gaber vd., 2005).Akan verilerin tekrarlı taranamamasının sebebi yalnızca verilerin büyüklüğü değil aynı zamanda üzerinde çalışılan veri evreninin geniş olmasıdır.

.

Tablo 1: Akan veriler ve statik veriler arasındaki farklar

Parametre	Statik veri	Akan veri
Veriye erişim	Sıralı veya sırasız	Sıralı
Bellek miktarı	Esnek	Sınırlı bellek
Veri dağıtımı	Statik	Dağıtık
Hesaplama sonuçları	Kesin sonuçlar	Yaklaşık sonuçlar
Veri tarama	Esnek	Sınırlı
Algoritmalar	İşlem süresi önemli değil	İşlem süresi önemli
Örnekleme	Gerekli değil	Örneklemenin yapılacağı zamanı belirlemek karmaşıktır
Veri hızı	Önemli değil	Verinin varış hızı, işleme hızından yüksektir
Veri modelleme	Kalıcı	Akan veriye göre güncellenir
Veri şeması	Statik	Dinamik
Sorgu yapısı	Tek seferlik sorgular	Sürekli sorgular
Erişim planı	Sorguya göre belirlenen erişim işlemci ve veritabanı tasarımı	Öngörülemeyen veri girişi ve karakteristikleri
Veri işleme	Çoklu tarama mümkün	Veriler sadece bir kez işlenir
Veri işleme zamanı	Esnek	Sınırlı

Örneğin, milyonlarca insanın yaşının izlendiği bir uygulamada veri evreni nispeten küçüktür ve 0 ile 120 arasında olacaktır ve bu tür verilerin özetleri kolaylıkla korunabilmektedir. Ancak İnternet üzerindeki tüm IP adres çiftleri setine karşılık gelen evren çok büyüktür ve bu durum depolama konusunda sorun yaşanmasına sebep olmaktadır. Doğruluk kavramı, verilerin güvenilirliğinin genellikle zayıf olmasını ve incelemeye ihtiyaç duyulmasını ifade etmektedir. Çeşitlilik kavramı ise verilerin genellikle heterojen bir yapıda olmasını ve çoğu zaman farklı türlerdeki bir veya daha fazla verinin birlikte işlenmesi gerektiğini ifade etmektedir (Ellis, 2005).

2.1. Akan Veri Kaynakları

Bilgi teknolojilerindeki gelişmelerle birlikte anlık olarak giderek artan boyutlarda üretilen verileri işlemek, analiz etmek, sunmak ve bazı durumlarda meydana gelen olaylara cevap vermek için çeşitli araçlar geliştirilmiştir. Verilerin toplanması ve işlenmesi, akan verilere özgü bir altyapı ve analiz yöntemi gerektirmektedir (Psaltis, 2017). Akan veri kaynakları operasyonel izleme, Web analizi, çevrimiçi reklamcılık uygulamaları, sosyal medya ile mobil veriler ve Nesnelerin İnterneti verileri olabilmektedir.

Fiziksel sistemlerin operasyonel olarak izlenmesi, akan verilerin temel uygulamasıdır. Operasyonel denetlemenin günümüzdeki en yaygın kullanımı, İnternet'i besleyen veri merkezlerindeki ayrık bilgisayar sistemlerinin performansını izlemektir (Kleppmann, 2016). Bu sistemlerin tümü, fiziksel durumlarıyla ilgili işlemcinin sıcaklığı, fanın hızı, disk sürücülerinin durumu, işlemci yükü, ağ etkinliği ve depolama birimlerine erişim süreleri gibi birçok veriyi sürekli olarak kaydetmektedir.

Tüm bu sistemlerin izlenmesini mümkün kılmak ve sorunları tanımlamak için, bu veriler çeşitlimekanizmalar yoluyla gerçek zamanlı olarak toplanmaktadır (Xu ve Balazinska, 2011). E-ticaret ve reklamcılık

uygulamalarının gelişimiyle birlikte Web siteleri üzerindeki etkinliklerin izlenmesi yoluyla gerçekleştirilen Web analizleri giderek önemli bir hale gelmektedir. Bir Web sitesine gelen ziyaretçi sayısı, e-ticaret platformlarındaki kullanıcılar ve incelenen ürünler arasındaki ilişki gibi verileri analiz etmek için bir takım özel günlük işleme araçları geliştirilmiştir. Büyük veri ve Hadoop gibi büyük veri işleme araçlarının gelişmesiyle birlikte, Web analizleri yığın tabanlı sistemler ile birlikte yapılmaya başlanmıştır. Bu sayede büyük verilerin gerçek zamanlı olarak toplanması ve paralel olarak işlenmesi mümkün hale gelmiştir (Ellis, 2005).

Gerçek zamanlı verilerin üretildiği ve işlendiği en önemli uygulama alanlarından biri de çevrimiçi reklamcılık uygulamalarıdır. Reklamcılık uygulamaları ve gerçek zamanlı teklif verme altyapısının gelişimiyle birlikte reklam uygulamaları akan veriler ile daha önemli bir hale gelmiştir (Babcock vd., 2002). Bu uygulamalarda, farklı ortamlarda ve farklı sitelerde yapılan ticari işlemler, dakikalık olarak yönetilmektedir. Buna ek olarak, bu satın alma işlemleri, genellikle satın alma sayısı veya bir reklamdaki tıklama sayısı gibi birtakım metriklerle yönetilmektedir. Yapılan araştırmalara göre işletmeler, Google reklamları için harcadıkları her 1 ABD doları için ortalama 2 ABD doları gelir elde etmektedir.

Kullanıcılar Web sitelerine reklam uygulamaları vasıtasıyla geldiklerinde, gerçek zamanlı olarak sayfa görüntülemesi için teklif veren birçok farklı firmaya istek gönderilmektedir ve açık arttırma ile kazanan tarafın reklamı kullanıcıya sunulmaktadır (Ikonomovska vd., 2007). Bu işlemler genellikle sayfanın geri kalanı yüklenirken olur ve geçen süre yaklaşık 100 milisaniyeden daha kısadır. Bu süreçteki tüm taraflar (teklif verme aracı, reklam verenler ve yayıncı) verileri çeşitli amaçlarla gerçek zamanlı olarak toplamaktadır. Toplanan veriler gerçek zamanlı geri bildirim mekanizmaları, trafik değişimlerinin izlenmesi, erişim kısıtlamaları, kampanya yönetimi ve optimizasyonu ile risk yönetimi faaliyetleri için kullanılabilmektedir (Ellis, 2005).

Diğer büyük veri kaynaklarından biri Twitter ve Facebook gibi sosyal medya uygulamalarıdır. 2018 yılının ortalarından itibaren Twitter üzerinden saniyede yaklaşık 6.000 olmak üzere günde yaklaşık 500 milyon tweet atıldığı belirlenmiştir. 2018 yılının ortalarından itibaren ise 2.23 milyar aktif Facebook kullanıcısının olduğu raporlanmıştır. Her saniye yaklaşık 5 yeni kullanıcı profilinin oluşturulduğu Facebook üzerinden günlük yaklaşık olarak10 milyon beğeni ve paylaşım yapılmaktadır. Bu veriler, gerçek zamanlı olarak toplanmakta ve dünyadaki haber ajansları ile diğer platformlar için önemli bir bilgi kaynağı haline gelmektedir. New York'taki Twitter kullanıcıları 2011 yılında, Washington'taki bir depremin meydana gelmesinden yaklaşık 30 saniye önce sismik hareketlerden deprem hakkında bilgi almışlardır (Peary vd., 2012). Facebook ve Foursquare gibi yaygın ve gelişmekte olan iletişim platformları da düşünüldüğünde, bu veriler son derece büyük ve çeşitlidir. Web analitiği ve çevrimiçi reklamcılık gibi uygulamalardan alınan veriler çok boyutlu olmasına rağmen genellikle oldukça iyi yapılandırılmıştır. Harcanan para ya da fiziksel konum gibi boyutlar oldukça iyi anlaşılmış nicel değerlerdir. Ancak sosyal medya verileri, veri analistleri tarafından anlamlandırılan yapılandırılmamış verilerdir. Sosyal medya verileri genel olarak, doğal dil verisinin, sistemler tarafından çözümlenerek işlenip anlaşılacağı biçimdedir. Bu durum sosyal medya verilerini oldukça zenginleştirir ancak gerçek zamanlı veri kaynaklarının işlenmesi zorlaştırır (Psaltis, 2017).

Mobil veriler ve Nesnelerin İnterneti verileri akan verilerin elde edildiği diğer bir uygulama alanıdır. Küresel mobil veri trafiğinin 2016 ile 2021 yılları arasında yaklaşık yedi kat artacağı tahmin edilmektedir. 2018 yılında kullanıcıların mobil cihazlar ile günde ortalama 3.5 saat zaman geçirdiği belirlenmiştir. Ocak 2018

verilerine göre, küresel mobil nüfus 3,7 milyar tekil kullanıcı sayısına ulaşmıştır. Ayrıca üretilen Web trafiğinin yaklaşık %51'inin mobil cihazlar tarafından oluşturulduğu belirlenmiştir. 2021 yılında üretilen mobil trafiğin 49 exabyte boyutuna ulaşacağı tahmin edilmektedir. Mobil cihazlar ve giyilebilir teknolojiler çevrimiçi servislere geri bildirimlerde bulunmalarının yanı sıra Bluetooth gibi teknolojileri kullanarak yakınlardaki diğer nesnelerle iletişim kurma yeteneğine de sahiptir (Krishnaswamy vd., 2012). Örneğin, uyku aktivitesini ölçen bileklikler, kullanıcıların zayıf gece uykusu geçirdiğini belirleyerek ertesi gün uyarılacak bir otomatik kahve makinesini tetikleyebilir.

Akıllı toz fikri uzun zamanlar boyunca dünyanın belirli bir bölgesine dağıtılabilen ve veri toplamak için kullanılan sensörleri ifade etmektedir (Elnahrawy, 2003). Sensör cihazlarının kısıtlaması, donanım parçalarının imalatı için gereken masraflardır. Ancak bu kısıtlamalar, veri toplama donanım ve yazılımlarının (akıllı telefonlar gibi) gelişimiyle birlikte çözülmüş ve günümüzde Nesnelerin İnternet'i olarak bilinmektedir (Bifet, 2016). İnsanların nesneleri sürekli olarak izlemelerinin yanı sıra nesneler de sürekli olarak birbirlerini izleyebilmektedir. Toprak koşullarının daha iyi izlenmesi yoluyla tarımın daha verimli hale getirilmesi, şehirlerdeki trafik yönetimi gibi çeşitli potansiyel uygulama alanları bulunmaktadır. Anlık olarak elde edilen veriler analiz edilmekte ve uygulama alanlarına yönelik geliştirmeler yapılabilmektedir.

2.2. Akan Verilerin Karakteristik Özellikleri

Akan verilerin özellikleri, yerel ve dağıtık ortamlarda veri aktarımı sırasındaki sıkıştırmalardan etkilenmektedir. Bu sebeple akan verilerin özellikleri, ardışık paketler arasındaki zaman aralıklarına göre, ardışık paketler arasındaki miktarın çeşitliliğine göre ve ardışık paketler arasındaki süreklilik veya bağlantıya göre değişebilmektedir. Ardışık paketler arasındaki zaman aralıkları akan verilerin periyodiklik karakteristiğine bağlıdır (Krawczyk, 2017).

- Güçlü periyodik akan veriler: Zaman aralıkları sabit olan iki ardışık t paketi aynı uzunluğa sahipse, bu akan veriye güçlü periyodik akan veri adı verilmektedir. Geleneksel telefon anahtarlamasında kullanılan PCM kodlu konuşma örnek olarak verilebilir.

- Zayıf periyodik akan veriler: Ardışık iki paket arasındaki zaman aralıkları sabit değilse, ancak belirli bir periyodik yapıya sahipse, akan veriler zayıf periyodik olarak adlandırılmaktadır.

- Periyodik olmayan akan veriler: Süre aralıklarının güçlü ya da zayıf periyodlarda olmadığı, iletim sırasında paketler arasındaki paketler arasındaki zaman periyodlarının değişkenlik gösterdiği akan verilere periyodik olmayan akan veri adı verilmektedir.

Ardışık paketler arasındaki miktarın çeşitliliği ise akan verilerin düzenli, yarı düzenli ve düzensiz bir karaktere sahip olmasına bağlıdır.

- Düzenli akan veriler: Veri miktarının, akan verinin ömrü boyunca sabit kaldığı sıkıştırılmamış sayısal veri iletimini ifade etmektedir. Kameralardaki video akışı ve CD'lerdeki ses akışı örnek olarak verilebilir.

- Yarı düzenli akan veriler: Akan verilerin miktarının zaman içinde periyodik olarak değiştiği düzenli olmayan akan verileri ifade etmektedir. Sıkıştırılmış video akışı örnek olarak verilebilir.

- Düzensiz akan veriler: Akan verilerin miktarının sabit ya da periyodik olmamasını ifade etmektedir. Bu kategorideki akan verilerin iletimi ve işlenmesi daha karmaşıktır.

Ardışık paketler arasındaki süreklilik veya bağlantı ise akan veri kaynağının veri gönderimini sürekli ve bağlantısız göndermesini ifade etmektedir.

- Sürekli akan veriler: Ardışık paketlerin birbiri ardına ve sürekli olarak iletildiği akan verilerdir. Örneğin, Isdn B kanalı için ses verileri iletiminde 64 kbps'lik hız ile iletim gerçekleştirmektedir.

- Bağlantısız akan veriler: Bilgi birimleri arasında boşluklar olan akış verileri, bağlantısız akan veri olarak adlandırılmaktadır. Bağlantılı bir akan verinin daha yüksek kapasiteli bir kanal üzerinden iletilmesi, paketler arasında boşluklar oluşmasına neden olmaktadır. FDDI ağında 1,2 mbps ve JPEG yöntemi ile kodlanan akan veriler örnek olarak verilebilir (Steinmetz ve Nahrstedt, 2002).

Klasik algoritmalar, bellek kullanımının kısıtlamaları, işlem süresinin kısıtlanması ve gelen örneklerin tekrarlı taranamaması gibi akan verilerin gereksinimlerini karşılayamamaktadır (Ellis, 2005). Naïve Bayes gibi örnek tabanlı öğrenme algoritmaları ve sinir ağları gibi artımlı öğrenme algoritmaları, sıkı hesaplama taleplerini karşılayamamakta ve gelişen veri kaynakları karşısında yetersiz kalmaktadır.

Hafızaya ve zamana ilişkin kısıtlamalar, örnekleme ve özetleme yaklaşımları gibi farklı pencere oluşturma tekniklerinin geliştirilmesine neden olmuştur. Bununla birlikte, akan veri üreten veri kaynağındaki dağıtım zamanla değişebilmektedir. Bu nedenle, durağan olmayan akan verilerde geçmişten gelen veriler, mevcut durum için alakasız veya zararlı hale gelebilir ve sınıflandırıcıların tahminlerini olumsuz etkileyebilir. Burada veri yönetimi yaklaşımları, eski veri örneklerinin atıldığı bir unutma mekanizmasının rolünü oynayabilir.

2.3. Akan verilerden öğrenme yapıları

Denetimli bir öğrenme yapısı göz önüne alındığında, bir süre sonra gelecek veri örneğinin içinde, hedef değeri olan y^t'nin mevcut olacağı kabul edilmektedir (Stephens ve Tamayo, 2003). Bu sebeple S akan verisi, $t = 1, 2, ..., T$ için etiketli örneklerin dizisi $z^t = (x^t, y^t)$ 'dir. Genel olarak x, öznitelik değerlerinin vektörü y, sınıflandırma problemleri için ayrı bir sınıf etiketi ($y \in \{K_1, ..., K_l\}$) veya regresyon problemleri için sayısal çıktı değeridir. Burada temel amaç geçmiş verilerden (örnek bir eğitim seti) niteliklerin kümesi ile hedef çıktı arasındaki ilişkiyi öğrenmektir (Bifet, 2016). Sınıflandırma için bu ilişki, keşfedilen sınıflandırma bilgilerine karşılık gelir ve yeni gelen $x^{t'}$örneğinin sınıf etiketini belirlemede bir sınıflandırıcı kullanılır. Regresyon durumunda, öğrenilen model bir sayısal değeri tahmin etmek için kullanılır. Sınıflandırıcı ya da regresyon modellerinin, belirli bir ana kadar görülen $\{z^1, z^2, . . . , z^t\}$ veri öğelerinden öğrendiklerine dayanarak bir tahmin değeri vermesi beklenmektedir. Elde edilen tahmin sonucu $y^{t'}$ ve gerçek değer y^t, öğrenme algoritması tarafından ilave öğrenme bilgisi olarak kullanılabilir (Aggarwal, 2014).

Akan verilerde sınıflandırma problemleri için önerilen algoritmaların çoğunluğu, denetimli öğrenme çerçevesini izlemektedir. Diğer bir deyişle, işlenen örneklerin tamamı için sınıf etiketlerinin tamamına istenildiği zaman erişebilmektedir. Ancak bazı uygulamalarda, öğrenme örneklerinin eksiksiz olarak etiketlendiği varsayımı, gerçek zamanlı akan veri uygulamalarında kullanışlı olmayabilir çünkü akan verilerde yeni gelen örneklerin sınıf etiketleri hemen elde edilemez. Örneğin, mali dolandırıcılık tespitinde, dolandırıcılık işlemlerine ilişkin bilgiler uzun bir gecikmeden sonra elde edilir (bir hesap sahibinin aylık hesap özeti) ya da kredi onay problemlerinde gerçek etiketler genellikle 2-3 yıl sonra elde edilir.

Bu sebeple akan veriler için farklı çerçevelerin kullanılması gerekmektedir. Gerçek sınıf etiketlerine erişim beklenenden çok daha geç olacak ise gecikmeli etiketleme ile öğrenme yapılır ve sınıflandırıcı veriyi görmeden uyarlanabilir. Ya da gelen tüm örnekler için etiketlerin mevcut olmadığı yarı denetimli öğrenmede sistem sınıf etiketlerini elde etmek için etiketsiz örnekleri seçen aktif bir öğrenme tekniği kullanmaktadır. Denetimsiz öğrenme veya başlangıçta etiketli

örneklerden öğrenmede ise bir başlangıç sınıflandırıcısı, sınırlı sayıdaki etiketli eğitim örneğinden öğrenerek sınıf etiketlerine herhangi bir erişim olmadan etiketlenmemiş örneklerin akışını işler (Psaltis, 2017).

Akan verilerde, örnekler çevrimiçi olarak veri parçaları (bölümler, bloklar) şeklinde sağlanmaktadır. Birinci yaklaşımda, algoritmalar tek tek örnekleri zamanla ardışık anlarda işlerken, diğer yaklaşımda örnekler yalnızca veri blokları olarak adlandırılan $S = B_1 \cup B_2 \cup \ldots B_n$ kümelerini işler. Bloklar genellikle eşit boyuttadır ve sınıflandırıcıların oluşturulması, değerlendirilmesi veya güncellenmesi yeni bir bloğun tüm örnekleri mevcut olduğunda yapılır.

Akan verilerin iki temel modeli olduğu düşünülmektedir. Bunlardan ilki verilerin durağan olduğu, örneklerin sabit olduğu fakat olasılık dağılımının bilinmediği durumdur. İkincisi ise verilerin zaman içinde gelişebildiği sabit olmayan durumdur. İkinci durumda hedef kavramlar (örnek sınıfları) ve nitelik dağılımları değişebilmektedir. Burada akan veri belirli bir kararlılık süresinden sonra değişebilmektedir ve bu durum kavram kayması olarak adlandırılmaktadır (Ellis, 2005).

Kavram kaymaları, gelen veri örneklerinin özelliklerini yansıttığı için geçmiş eğitim örneklerinden öğrenilen sınıflandırıcı ve regresyon modellerinin doğruluğunu etkilemektedir. Kavram kaymasından etkilenen akan veri uygulamalarına güvenlik saldırılarının yapılabileceği bilgisayar veya telekomünikasyon sistemleri, trafik modellerinin zaman içinde değişebileceği trafik izleme uygulamaları, iklim değişiklikleri ve doğal anomalilerin tahmini etkileyebileceği hava tahmini uygulamaları, kullanıcıların ilgi alanlarının değişebileceği kişiselleştirilmiş tavsiye sistemleri, uygulanan ilaç tedavilerine ve hastaların vücut dirençlerine bağlı olarak değişebilecek tıbbi karar mekanizmaları örnek olarak verilebilir (Gaber vd., 2005). Ayrıca spam kategorizasyonu, nesne konumlandırma, endüstriyel izleme sistemleri, finansal sahtekârlık algılama sistemleri ve robotik sistemler kavram kaymasından etkilenen uygulama alanlarıdır.

3. Akan Veri Karakterizasyonu

Veri, sembollerin makineler tarafından depolanmasını ve işlenmesini tanımlamak için kullanılmaktadır. Temel olarak veri, sensörler, kullanıcı girdileri, kullanıcı etkileşimleri, çevresel gözlemler, uzay araştırmaları, uçak seyahatleri, sosyal ağlar, enerji üretim tesisleri, Web sitesi trafiği ve e-ticaret gibi platformlardan elde edilmektedir (Pyle, 1999).

Nicel veriler, fiziksel ve ölçülebilir veriler olarak tanımlanmaktadır. Bu tür verilere, bir proje sahasında bulunan sudaki bulanıklık değerlerini hesaplamak için yapılan deneylerin ölçüm sonuçları örnek olarak verilebilir. (Brief, 2012). Nitel veriler ise fiziksel olmayan veya gözlemlenebilir olan veriler olarak tanımlanmaktadır. Bu veriler izleme ve görsel analizlerin kaydedilmesi yoluyla elde edilmektedir. Bir akışın renginin veya temizliğinin gözlemlenmesi ve kaydedilmesi örnek olarak verilebilir (Johnson, 2015).

Veri öznitelikleri, veri nesnelerinin karakteristiğini temsil eden veri alanlarını ifade etmektedir. Örneğin, müşteri nesnesini açıklayan öznitelikler, müşteri ID, müşteri adı ve adres gibi bilgilerdir. Bir özniteliğin türü, özelliğin sahip olabileceği olası değerler kümesi (nominal, ikili, ordinal veya nümerik) tarafından belirlenir (Bramer, 2007).

Nominal öznitelik değerleri, semboller veya nesnelerin isimleridir. Anlamlı bir düzeni olmayan bu değerler bilgisayar bilimlerinde numaralandırma olarak bilinir (Kohler, 2002). Örnek olarak saç rengi ve medeni durum, kişi nesnesini tanımlayan niteliklerdir.

İkili öznitelikler 0 veya 1 gibi yalnızca iki kategori veya duruma sahip nominal özniteliklerdir (Brief, 2012). Burada 0 genelde özniteliğin yok olduğu, 1 ise var olduğu anlamına gelir. Örneğin, tıbbi bir test

sonucunda 0 hastanın virüs taşımadığını, 1 ise hastanın virüsü taşıdığını ifade edebilir.

Ordinal öznitelikler, anlamlı bir düzene veya aralarında bir sıralama olan değerlere sahip özniteliklerdir. Ordinal öznitelikler objektif olarak ölçülemeyen niteliklerin, subjektif değerlendirmelerinin elde edilmesi için kullanılabilir. Müşterilere, sunulan hizmetlerin değerlendirilmesi için sorulan 0 memnuniyetsiz, 1 tarafsız, 2 memnun, 3 mükemmel gibi anketler örnek olarak verilebilir.

Nümerik öznitelikler, tamsayı veya gerçek değerlerle temsil edilebilen nicel değerlerdir. Nümerik öznitelikler aralıklarla veya oran ile ölçeklendirilebilirler. Örnek olarak sıcaklık özelliği aralıklarla ölçeklendirilebilir. Oran ölçekli özniteliklere ise doküman nesnelerinde bulunan kelime sayıları örnek olarak verilebilir (Han vd.,, 2011).

Akan veriler büyük hacimleri sebebiyle statik veritabanlarında depolanamaz ve çevrimdışı sorgu işlemleri gerçekleştirilemez. Bu sebeple geleneksel çözümler akan veriler için yetersiz kalmaktadır (Gaber vd., 2005). Akan veriler, ağ izleme uygulamaları, saldırı tespit sistemleri, dolandırıcılık tespiti, finansal izleme, e-ticaret ve sensör ağları gibi uygulama alanlarında kullanılmaktadır (Ellis, 2005).

Akan veriler elde edildikten sonra verilerinin niteliğinin yani akışın gerçekte sunabileceği verilerin karakterize edilmesi ön plâna çıkmaktadır. Veride bulunan değişkenlerin kullanışlılığına göre değerlendirilebilmesi için çeşitli şekillerde karakterize edilmesi gerekmektedir. Karakterizasyon, verilerde bulunan özet bilgilerin, verilerin temsil edildiği gibi görünüp görünmediğinin kontrol edilmesine yardımcı olur. Karakterizasyonun amacı, verilerin doğasını anlamaktır (Pyle, 1999). Karakterizasyon, verilerdeki ayrıntı ve detay seviyesi, tutarlılık, veri kirliliği, nesneler, ilişkiler, etki alanı, varsayılan değerler, bütünlük ve yinelemeli değişkenler açısından incelenmektedir(Pyle, 1999).

3.1. Verilerdeki ayrıntı ve detay seviyesi

Bir veri kümesinde bulunan ayrıntı düzeyi veya detay seviyesi, çıktı için mümkün olan ayrıntı seviyesini belirler. Genellikle giriş akışlarındaki ayrıntı seviyesi, çıktıdaki ayrıntı düzeyinden daha yüksektir. Verilerdeki detay seviyesinin bilinmesi, verilerin potansiyel olarak destekleyebileceği çıkarsama seviyesinin veya tahminlerin değerlendirmesini sağlar (Son, 2006).

3.2. Tutarlılık

Tutarsız veriler, farklı özelliklerin farklı sistemlerde aynı isimle temsil edilebilmesi ve aynı özelliklerin farklı sistemlerde farklı isimlerle temsil edilmesini ifade etmektedir (Han vd.,, 2011). Verilerde bulunan etiketler, değişkenlerin içeriklerini tanımlamaktadır. Veriler tek bir uygulama sisteminden elde edildiğinde veri tutarlılığıyla ilgili problemler yaşanabilmektedir. Örneğin bir sigorta şirketinin veritabanında “model” şeklinde tanımlayan bir alana girilen “Merc”, “Mercedes”, “M-Benz” ve “Mrcds” girdileri aynı üreticiyi temsil etmektedir.

3.3. Veri kirliliği

Veri kirliliği çeşitli nedenlerle ortaya çıkabilir. En yaygın olanı ise kullanıcıların bir sistemi kendi işlevselliğinin ötesine genişletmeye çalıştıklarında görülmektedir. Örneğin, bir veride "cinsiyet" alanında "B" değeri bulabilir. "B" girdisi, hem "Boy (erkek)" için hem de "Business (işletme)" için geçerlidir. Veri kirliliği farklı kaynaklar tarafından da oluşturulabilmektedir. Veri kopyalama işlemi sırasında yanlış belirlenmiş formatlar ya da alanların içeriklerinin yanlış bir alana aktarılması örnek olarak verilebilir (Pyle, 1999).

İnsan faktörü, veri kirliliğinin diğer bir kaynağıdır. Veri alanları genellikle değerli bilgiler elde edebilmek için dâhil edilmiş olsa da bu alanlar boş, eksik veya yanıltıcı bilgiler içerebilir. Bir otomobil üreticisinin araç satışı sırasında müşterilerden topladığı aile bireylerinin sayısı ve hobiler gibi elde ettiği demografik bilgiler örnek olarak verilebilir. Ancak bu bilgiler pazarlama stratejileri için

değerli olsa da müşteriler kendilerine yöneltilen bu soruları cevapla konusunda isteksizdir.

3.4. Nesneler

Veriler, nesneler ile ilgili ölçümler yapılarak elde edilmektedir. Bu nedenle, veriler karakterize edilirken ölçüm yapılan nesnelerin doğası anlaşılmalıdır (Son, 2006). Örneğin, "tüketici harcamaları" ve "tüketici satın alma örüntüsü" ifadeleri benzer görünmektedir ancak birisi tüketicilerin toplam harcamalarını belirtirken diğeri tüketicilerin satın aldığı ürün türlerini ifade etmektedir.

3.5. İlişkiler

Çoklu girdilerin bulunduğu akan verilerde, akışlar arasındaki ilişkiyi tanımlamak önemlidir. Bu ilişkiler kullanılarak, giriş akışlarındaki örnekler arasındaki ilişkiler tanımlanarak birleşmeleri sağlanır. Muhtemel tutarsızlık ve kirlilik gibi sorunlardan dolayı, verilerinin birleştirilmesi kolay olmayabilir (Pyle, 1999).

3.6. Etki alanı

Her değişken, belirli bir etki alanı veya izin verilen değerler aralığından oluşmaktadır. Özet istatistikleri ve sıklık sayıları, alanın dışındaki hatalı değerlerin ortaya çıkmasını sağlamaktadır. Bununla birlikte, bazı değişkenler yalnızca bazı koşullu alanlarda geçerli değerlere sahiptir (Tan, 2006). Örneğin, tıbbi tanı alanında yalnızca belirli bir cinsiyete sahip hastalar için geçerli olan bazı teşhisler olabileceği için koşullu alanlar olabilmektedir.

3.7. Varsayılan değerler

Veriler bazı değişkenler için varsayılan değerler içerebilmektedir. Varsayılan değerler, girilen gerçek değer girdilerine bağlı olarak koşullu olabilir. Bu gibi koşullu varsayımlar, veri eksikliği durumlarında önemli örüntüler oluşturmak için kullanılabilir. Bu örüntüler, tahmin veya çıkarım modelleri için anlamlı olabilir ancak bu örüntüler genellikle sınırlı bir değer aralığında olduğundan dikkatli bir şekilde değerlendirilmeleri gereklidir (Son, 2006).

3.8. Bütünlük

Bütünlük kontrolü, değişkenler arasında izin verilen ilişkileri değerlendirir (Pyle, 1999). Örneğin, bir çalışan birkaç tane telefon numarasına sahip olabilir ancak birden fazla çalışan numarasına sahip olamaz. Bütünlüğün kabul edilebilir değer aralıkları cinsinden düşünülmesi, uç değerlerin belirlenen sınırların dışında olarak değerlendirilmesine yol açar. Ancak uç değerlerin özellikle sigorta ve finansal veri setlerinde dikkate alınması gerekir.

3.9. Yinelemeli değişkenler

Gereksiz veriler, akan verilerin içinde mevcut olabilir ya da farklı akan verilerin birleştirilmesi sırasında ortaya çıkabilmektedir (Han vd.,, 2011). Doğum tarihi ve yaş gibi aynı bilgiyi ifade eden değişkenler bir arada bulunduğunda yinelemeli değişken kavramı ortaya çıkar. Çoğu modelleme tekniği, örnek sayısından çok değişken sayısına bağlı olduğu için gereksiz değişkenlerin kaldırılması, modelleme hızını artıracaktır.

4. Akan Veri Üretimi

Web teknolojilerindeki gelişmeler sayesinde her an giderek artan miktarda veri üretilmektedir. Kısıtlı sürelerde analiz edilecek veri boyutu petabayt boyutuna ulaşabilmektedir. Artan veri miktarı, veriler üzerinde daha çeşitli ve detaylı analizlerin yapılmasını sağlarken, özellikle hızlı bir şekilde analiz sonuçlarının elde edilmesinin gerektiği durumlarda gerçek zamanlı işleme, yığın işleme ve akan veri işleme konularında problemler yaşanabilmektedir (Klein vd., 2007).

Gerçek zamanlı sistemler, temel olarak veriler üzerinde gerçekleştirilen milisaniyeler türündeki işlem sürelerini ifade etmektedir. Gerçek zamanlı sistemlerin en temel örneklerinden biri borsa sistemleridir. Bir hisse senedinin ağ üzerinden 10 milisaniye gibi bir süre içinde gelmesi gerekiyorsa, bu

süreç gerçek zamanlı bir süreç olarak kabul edilmektedir. Burada, akan verileri işleyen bir yazılım mimarisi veya bir donanım mimarisi kullanılarak gerçekleştirilmesi ve zaman kısıtları, sistemin gerçek zamanlı olmasını sağlamaktadır. Gerçek zamanlı sistemlere banka ATM'leri, hava trafik kontrol sistemleri ve araçlarda kullanılan kilitlenmeyi önleyici fren sistemleri örnek olarak verilebilir (D'Andrea vd., 2015).

Gerçek zamanlı sistemler, program yürütme üzerinde kontrol sahibi olduğu için bir soyutlama seviyesi ön plana çıkmaktadır. Bu soyutlama seviyesi, programdaki kontrol akışı ile kaynak kod arasındaki ayrımın belli olmamasını, gerçek zamanlı sistemin programın yürütülmesi aşamasında hangi görevin yürüteceğinin seçilmesini ifade etmektedir (Huang vd., 2015). Bu durum daha yüksek üretkenliğe olanak tanıması ve karmaşık sistemlerin tasarlanmasını kolaylaştırdığı için faydalıdır ancak hata ayıklaması ve doğrulama gibi konularda kontrolün düşük kalmasına neden olmaktadır. Gerçek zamanlı işletim sistemleri ile ilgili bir diğer sorun ise görevlerin izole edilmiş olmamasıdır. Sistem, daha yüksek öncelikli görevleri zamanlamak ve göndermek için karar verir. Böylece tüm yüksek öncelikli görevler tamamlanana kadar diğer görevlerin yürütülmesi geciktirilir (Abrol vd., 2017).

Yığın işleme, büyük miktardaki verinin tek seferde işlenmesini ifade etmektedir. Bu veriler, genel olarak kesintisiz ve sıralı bir şekilde oluşturulmuş kayıtları ifade etmektedir. Yığın işlemeye, finans şirketlerinin belirli bir süre boyunca sunabileceği işlemler örnek olarak verilebilir. Yığın işleme belirli bir süre boyunca toplanan büyük miktardaki verileri işlemek için verimli bir yöntemdir (Lennert vd., 2018).

Akan veri işleme ise verileri anlık olarak analiz edebilmek için kullanılan bir süreçtir. Bu sürekli hesaplama yöntemi, sistemde çıktı üzerinde zorunlu zaman sınırlamaları olmaksızın, verilerin sistem üzerinden geçerken işlenmesi yoluyla gerçekleşir.

Akan veri işlemede, sistemlerin büyük miktarlardaki verileri depolaması gerekmez. Akan veri işlemede izlemek istenilen olaylar belirli bir periyotta sık sık gerçekleşiyorsa, olaylar anlık olarak tespit edilip hızlı bir şekilde değerlendirilebilir. Akan veri işleme, dolandırıcılık tespiti ve siber güvenlik gibi uygulama alanları için kullanışlıdır.

Akan veri işlemenin en büyük zorluklardan biri, sistemin uzun vadeli veri çıktı oranının, uzun vadeli veri girişi oranından daha hızlı olması gerektiğidir. Aksi takdirde, sistemde depolama ve bellek ile ilgili sorunlar yaşanabilecektir (Silva vd., 2013).

Büyük veri analizinde, akan veriler için hacim, hız, doğruluk ve çeşitlilik kavramları ön plana çıkmaktadır. Hız kavramı, algoritmanın akışı yalnızca bir kez işleyebileceği bir hızda veri üretilmesini ifade etmektedir. Akan veriler potansiyel olarak sonsuzdur. Bu sebeple kullanılan algoritma, veriler üzerinden birden fazla kez geçiş gerektiriyorsa oluşturulan model güncel olmayacaktır.

Statik verisetleri kullanılarak akan veri üretiminde, verilerin gelme sırası ya da zaman bilgisi kullanılabilmektedir. Zaman damgası kavramı, verisetindeki olayların gerçekleştiği tarih ve saatin bilgisini ifade etmektedir. Zaman damgası, sosyal medyadaki gönderiler ile bilgisayarda düzenlenen dosyaların saat ve tarih bilgileri örnek olarak verilebilir. Zaman damgası, bilgilerin çevrimiçi olarak oluşturulduğu, değiştirildiği veya silindiği kayıtların tutulması için önemlidir.

Statik verisetlerinde bulunan zaman damgaları kullanılarak, veriler akan veri formatına getirilerek analiz edilebilmektedir. Statik verisetindeki Unix zaman damgaları kullanılarak veriler anlık olarak okunabilmektedir. Bu sayede genel kullanıma açık akan verisetlerinin azlığı problemine çözüm üretilebilmektedir. Statik büyük verisetleri kullanılarak elde edilen akan veriler ile gerçek zamanlı analizler yapılabilmektedir.

5. Akan verilerde veri madenciliği ve makine öğrenmesi

Veritabanlarından bilgi keşfi olarak da tanımlanan veri madenciliği, veri tabanlarında veya veri ambarlarında depolanan büyük miktardaki verilerden kullanışlı ve gizli örüntüleri keşfetme görevidir. Veri madenciliği, veritabanı teknolojileri, istatistik, makine öğrenmesi, sinir ağları ve bilgiye erişim gibi çoklu disiplin tekniklerinin entegrasyonunu içermektedir. Bu süreç, temel olarak veri seçimi, veri temizleme, ön işleme ve veri dönüşümü gibi ön işleme adımlardan oluşur. Temel veri madenciliği görevleri zaman serisi analizi, birliktelik analizi, sınıflandırma, regresyon, kümeleme analizi ve özetlemedir.

Veri madenciliği, bilgisayar donanımı ve yazılım teknolojilerindeki gelişmeler sayesinde ortaya çıkan en önemli araştırma alanlarından biridir. Veriler, işletmelerin gelecek ile ilgili öngörüler oluşturması ve tahminlerde bulunabilmesi için kullanılabilecek en önemli varlıklardandır. Ancak veritabanları büyük ve dinamik olma eğilimindedir. Dolayısıyla içerikleri genellikle değişir, yeni bilgilerin eklenmesi gerekebilir, mevcut verilerin güncellenmesi veya silinmesi gerekebilir. İşletmeler sürekli olarak büyüdüğünden, ilgili veritabanları da büyüyecek ve sonuç olarak, mevcut veri madenciliği teknikleri, doğası gereği dinamik olan büyük veritabanlarıyla başa çıkmakta başarısız olacaktır (Kamburugamuve vd., 2013).

Veri madenciliği süreçleri veri madenciliği probleminin tanımlanması, verilerin toplanması, verilerin tespit edilmesi ve düzenlenmesi, modelin oluşturulması ve doğrulama aşamalarından oluşmaktadır. Veri madenciliği probleminin tanımlanması aşaması, çalışma yapılacak alana özgü bilgi ve deneyim gerektirmektedir. Değişkenlerin bağımlılıkları ve ilişkileri belirlenerek problem hakkında hipotezler oluşturulabilir. Veri toplama aşaması arklı kaynaklardan ve konumlardan veri toplanmasıyla ilgilidir. Veri toplama işlemi dâhili ve harici olarak yapılabilmektedir. Dâhili veriler genellikle mevcut veri tabanlarından ve veri ambarlarından toplanır. Kişiler tarafından kaydedilen gerçek işlemler en zengin bilgi kaynağıdır ve aynı zamanda en yararlı olanıdır. Harici veriler ise demografik veriler, psikografik veriler ve Web grafiği verileri olabilmektedir.

Verilerin tespit edilmesi ve düzenlenmesi aşaması büyük miktarda boyutlara sahip verisetlerindeki gürültü, kayıp ve tutarsız veriler üzerinde yapılan işlemleri ifade etmektedir. Bu ön işleme adımları veri temizleme, veri entegrasyonu, veri dönüşümü ve veri azaltma yoluyla gerçekleştirilebilir. Modelin oluşturulması aşaması veri madenciliği görevini seçme, yöntemi ve uygun algoritmayı seçme ve bilgi çıkarma adımları ile gerçekleştirilir. Veri madenciliği görevi seçme, geliştirilen modeli tahmine dayalı ya da tanımlayıcı olmasını ifade etmektedir. Tahmine dayalı modeller, bilinen sonuçlar kullanılarak sınıflandırma, regresyon veya zaman serileri analizi gibi veri setindeki bilinmeyen bazı değişkenleri belirlemek için kullanılmaktadır. Tanımlayıcı modeller ise verideki örüntüleri ve ilişkileri tanımlayarak incelenen verilerin özelliklerinin keşfedilmesini sağlar. Yöntemin seçilmesi ise karar ağacı veya sinir ağları gibi kullanılacak veri madenciliği yöntemlerinin belirlenmesini ifade etmektedir. Bilgi çıkarma adımı ise seçilen yöntem ve algoritmaların uygulanması sonucunda sonuçların elde edilmesini ifade etmektedir. Modelin oluşturulması ve doğrulama aşaması ise mevcut bir problem için geliştirilen modelin sonuçlarının yeterince güvenilir, kabul edilebilir ve kullanışlı olduğunun test edilmesini ifade etmektedir. Yukarıda açıklanan klasik veri madenciliği işlemleri, statik verisetlerinin durağan ve zamanla değişmeyen yapısı için uygun ve kullanışlıdır. Akan veriler, son derece yüksek hacimlere sahip ve gerçek zamanlı analizlerin yapılmasını gerektiren verilerdir. Akan veri madenciliği ile klasik veri madenciliği arasındaki farklar Tablo 2'de görülmektedir (Gagarin ve Toporivskyi, 2016).

Tablo 2: Akan veri madenciliği ile klasik veri madenciliği arasındaki farklar (Gagarin ve Toporivskyi, 2016)

Akan veri madenciliği	**Klasik veri madenciliği**
Gerçek zamanlı veri işleme	Çevrimdışı veri işleme
Veri kaynaklarından hızlı bir şekilde üretilen veriler	Durağan veri üretimi
Verilerin depolanması mümkün değil	Verilerin depolanması mümkün
Yaklaşık sonuç kabul edilebilir	Doğru sonuçlar gereklidir
Veri örneklerinin tamamının işlenmesi gerekmez	Veri örneklerinin tamamının işlenmesi gereklidir
Yalnızca toplu ve özetlenmiş veriler depolanır	Verilerin tamamı depolanır
Zamana ve konuma bağlı bilgiler önemlidir	Belirli uygulama sınıfları için mekânsal ve zamansal bağlamlar düşünülmektedir
Lineer ve lineer olmayan hesaplama teknikleri yaygın olarak kullanılmaktadır	Gerektiği takdirde yüksek alan ve zaman karmaşıklığı olan teknikler kullanılır.

Tablo 2'de görüldüğü gibi akan veri madenciliği yöntemleri gerçek zamanlı veri analizi, verilerin üretilme hızı, verilerin tamamının işlenmesi ve depolanması ve hesaplama teknikleri gibi yönlerden klasik veri madenciliği tekniklerinden farklılık göstermektedir. Akan veriler, statik veritabanlarında depolanamaz ve çevrimdışı sorgu işlemleri gerçekleştirilemez. Bu sebeple geleneksel çözümler akan veriler için yetersiz kalmaktadır (Gaber vd., 2005). Akan veriler, ağ izleme uygulamaları, saldırı tespit sistemleri, dolandırıcılık tespiti, finansal izleme, e-ticaret ve sensör ağları gibi uygulama alanlarında kullanılmaktadır (Ellis, 2005).Ancak akan verilerin süreklilik, sınırsızlık ve yüksek hız gibi dinamik özellikleri nedeniyle, hem çevrimdışı hem de çevrimiçi olarak çok büyük miktarda akan veri öğesi bulunmaktadır. Bu nedenle verilerde bir güncelleme olduğunda, tüm veritabanını tekrarlı olarak taramak ve verilerin tamamını saklamak, zaman ve depolama alanı açısından mümkün olmayabilir. Bazı durumlarda, Haber veritabanları, hisse senedi işlemleri, market veritabanları veya Web-log kayıtları gibi bazı uygulama alanlarında, yeni gelen veri öğeleri, eski veri öğelerinden daha önemli olabilmektedir. Sonuç olarak, dinamik veritabanında sıkça rastlanan bir öğe kümesi, güncellenmiş veritabanında seyrek görülse bile önemlidir.

Akan veri madenciliği tekniklerinin amacı, daha akıllı kararlar ve daha iyi ticari sonuçlar çıkarabilmek için uygulanabilir bilgiler sunmaktır (Kamburugamuve vd., 2013). Farklı analiz türleri, farklı türde bilgiler sağlamaktadır. Bu nedenle, kullanılabilecek tekniklerin ne sunduğunun belirlenmesi ve analitik işlevlerin, gayrimenkul, tesis ve varlık yönetimi işlevleri boyunca kuruluşun operasyonel yetenekleriyle eşleştirmesi önemlidir (Gagarin ve Toporivskyi, 2016).

Analitik çözümleri tanımlayıcı, tahmine dayalı ve öngörüye dayalı yaklaşımlar başlıkları ile incelenmektedir (IBM, 2013). Tanımlayıcı analitikler iş zekâsı ve veri madenciliği yöntemleri kullanılarak neler oldu sorusuna cevap vermektedir. Tahmine dayalı analitikler istatistiksel modeller kullanarak ne olabilir sorusuna dayalı tahminler üretmektedir (Huisman, 2015). Öngörüye dayalı analitikler ise optimizasyon ve simülasyon yöntemleri kullanarak ne yapmalıyız sorusuna cevap vermektedir. Tanımlayıcı analitikler diğer yaklaşımlara göre en yaygın kullanılan ve kuralcı analitiklerdir (Chandler vd., 2011). Ayrıca, bir etkinlik veya eylem için farklı yaklaşımlar geliştirme, veri ilişkilerini ortaya çıkarma, senaryo analizleri geliştirme ve iş kararlarını basitleştirerek varlık operasyonlarını geliştirmek hedeflenmektedir (Gagarin ve Toporivskyi, 2016).

5.1. Tanımlayıcı analitikler

Neler oldu sorusuna cevap vermeyi hedefleyen tanımlayıcı analitikler, kullanıcıların davranış kalıpları gibi varlık yöneticilerinin gelecekteki eylemlerinde ihtiyaç duyabilecekleri bilgileri, geçmiş veya

güncel olaylardan sağlayan yaklaşımlardır. Temel performans göstergelerinin kullanıldığı tanımlayıcı analitikler olayların sıklığı, operasyonların maliyeti ve başarısızlıkların asıl nedeni gibi ayrıntıları ortaya çıkarmak için verileri incelemektedir.

Statik veriler için en yaygın kullanılan tanımlayıcı tekniklerden biri kümeleme algoritmalarıdır. CluStream algoritması, k-means kümeleme algoritmasının mikro kümeler kullanılarak akan veriler için uyarlanmış şeklidir (Nguyen vd., 2015). Veri noktaları k kümeye ayrılmadan önce birçok mikro kümeye ayrılmaktadır. Her iki küme tipi, kümeler hakkında istatistiksel bilgiler tutan küme özellik vektörleri ile temsil edilir ve bu sayede akan verilerinin yönetilebilir bir boyuta indirgenmesi sağlanır.

5.2. Tahmine dayalı analitikler

Tahmine dayalı analitikler, geçmiş ve güncel verilere dayalı olarak gelecekteki sonuçların tahmin edilmesine yönelik çeşitli istatistiksel tekniklerden oluşmaktadır (Gama, 2010). Temel olarak örüntülerin ve veri ilişkilerinin belirlenmesi amaçlanmaktadır (Aggarwal vd., 2003). Tahmine dayalı analitikler, e-ticaret, finansal hizmetler, sigorta, telekomünikasyon, perakende, seyahat, sağlık hizmetleri ve ilaç endüstrisi gibi birçok alanda kullanılmaktadır. Karmaşık olay verilerine dayanan tahmin yöntemleri, önceden izlenen olaylara bağlı olarak sistemin özellikleriyle ilgili öngörüler yapabilir. Genel olarak bir tahmin süreci dört adımdan oluşur (IBM, 2013):

- Ham verileri toplama ve ön işleme

- Ön işleme tabi tutulan verileri, seçilen makine öğrenme yöntemi ile kolayca işlenebilecek bir hale dönüştürme

- Dönüştürülen verileri kullanarak öğrenme modeli oluşturmak

- Daha önce oluşturulan öğrenme modelini kullanarak tahminlerin oluşturulması.

5.3. Öngörüye dayalı analitikler

Öngörüye dayalı yaklaşımlar, geleceğe yönelik optimal planlama kararlarının alınmasını ve “ne yapacağız, neden bunu yapacağız" gibi soruların yanıtlanmasını hedeflemektedir. Öngörüye dayalı yaklaşımlar kullanıcıların bir dizi olası eylemi öngörmesini ve bu öngörülerden çözümler sunmasını sağlamaktadır (Leskovec vd., 2014). Temel olarak bu yaklaşımlar tavsiye sunmak ile ilişkilidir. Belirli kararlar verilmeden önce muhtemel sonuçları belirleyerek gelecekteki kararların etkileri ölçülmeye çalışılmaktadır (Huisman, 2015).

Öngörüye dayalı analitikler, bir veya daha fazla eylem önerisi sunarak tanımlayıcı ve tahminsel analitik yöntemlerin önüne geçmektedir. İş kuralları, algoritmalar, makine öğrenmesi ve hesaplama modellemesi gibi teknikler ve araçlar kullanılmaktadır. Bu analitikler, geçmiş işlem verilerine, gerçek zamanlı veri yayınlarına ve büyük verilere kadar pek çok farklı veri kümesindeki girdilere uygulanmaktadır (Bar-Yossef vd., 2002). Öngörüye dayalı analitikler, büyük şirketler tarafından doğru ürünlerin doğru zamanda tedarik edilip edilmediğini belirlemek, müşteri deneyimini optimize etmek, tedarik zincirindeki üretim, zamanlama ve envanterleri optimize etmek için kullanılmaktadır (Chandler vd., 2011).

5.4. Akan verilerde makine öğrenmesi

Makine öğrenmesi, öğrenme sistemleri ve algoritmaları teorisine, performansına ve özelliklerine odaklanan bir araştırma alanıdır. Yapay zeka, optimizasyon teorisi, bilgi teorisi, istatistik, bilişim, mühendislik ve matematik gibi disiplinler arası bir alandır. Tavsiye sistemleri, tanılama sistemleri, bilişim ve veri madenciliği gibi uygulama alanları ile bilim ve toplum üzerinde büyük etkileri olan çoğu alanda kullanılmaktadır (Najafabadi vd., 2015).

Tablo 3: Akan verilerde makine öğrenmesi (Augenstein vd., 2017)

	Özellik	**Makine öğrenmesinde yaşanan zorluklar**
Hacim	İşleme performansı	Zaman ve depolama karışıklığının artmasına yol açar.
	Modülerlik	Paralel veya iteratif veri işleme ihtiyacı vardır.
	Sınıf dengesizliği	Örnekleme teknikleri, bozuk verilere yol açabilir.
	Özellik çıkarımı	Doğru özellikleri ayıklamak, veri hacmi büyüdüğünde artan maliyetlere yol açar.
Çeşitlilik	Depolama	Tüm veriler yerel bir depolama biriminde bulundurulamaz.
	Veri Heterojenliği	Artan veri hacmi, verideki istatistikleri ve maliyeti arttırır.
	Gürültülü veriler	Veri temizleme ve eksik veri tespiti maliyeti arttırır.
Hız	Veri kullanılabilirliği	Akan verilerde eğitim ve test verisi kullanımı uygun olmayabilir. Modeller verilere göre değişebilmektedir.
	Gerçek zaman	Model eğitimi için zaman kısıtları bulunur.
	Kavram kayması	Modellerin ürettiği sonuçlar, zamana bağlı olarak değişebilir.
Doğruluk	Veri kaynağı	Makine öğrenmesi tekniklerini yapılandırmak için toplanacak veri miktarı maliyeti arttırmaktadır.
	Veri belirsizliği	Sosyal medya verilerinde ya da duygu analizi uygulamalarında elde edilen sonuçların doğruluk oranlarını azaltabilir.
	Kirli ve gürültülü veriler	Eksik veya yanıltıcı sınıf etiketleri maliyet artışına sebep olur.

Temel olarak, makine öğrenmesi denetimli öğrenme, denetimsiz öğrenme ve takviyeli öğrenme olmak üzere üç alt alana ayrılmaktadır. Denetimli öğrenme, girdileri ve istenen çıktıları olan etiketli verilerle eğitim yapılmasını gerektirir (LeCunvd., 2015). Denetimsiz öğrenme sınıf etiketlerinin bilinmediği ya da etiketlemenin maliyetli olacağı büyük veriler için etiketli eğitim verisi gerektirmez ve eğitim aşamasında hedef nitelikleri kullanmamaktadır. Takviyeli öğrenme ise harici bir çevreyle olan etkileşimler yoluyla alınan geri bildirimden öğrenmeyi mümkün kılar (Chen ve Lin., 2014). Bu üç temel öğrenme paradigmasına dayanarak, veri görevleriyle uğraşmak için birçok teori mekanizması ve uygulama hizmetleri önerilmiştir. Örneğin Google, makine öğrenmesi algoritmalarını, Google çeviri, Google sokak görünümü, görüntü arama motoru ve Android ses tanıma uygulamalarından elde ettiği dağıtık verilerin büyük bölümü ile uygulamaktadır (Qiu vd., 2016).

Bir diğer nokta, geleneksel makine öğrenmesi sistemlerinin çoğu merkezi işlemede, toplanan verilerin tamamının belleğe yükleneceği varsayımıyla tasarlanmış olmasıdır. Veriler giderek büyüdüğü zaman, mevcut makine öğrenmesi teknikleri büyük miktarda veriyi işlemede büyük güçlüklerle karşılaşmaktadır. Makine öğrenmesi yöntemlerinin amacı, bilgiyi keşfetmek ve akıllı kararlar çıkarmaktır. Büyük veri çağında, makine öğrenmesi yöntemlerinin giderek artan miktardaki verileri işleyebilecek öğrenme algoritmalarına sahip olması gerekmektedir. Makine öğrenmesi yöntemleri, akan verilerinden öğrenme, derin öğrenme, artımlı ve topluluk öğrenmesi, temsili öğrenme, dağıtık ve paralel öğrenme, aktarımlı öğrenme ve aktif öğrenmeyöntemleri başlıkları ile incelenmektedir (Oussous vd., 2017). Akan verilerde makine öğrenmesi yöntemlerinin hacim, çeşitlilik, hız, doğruluk açısından değerlendirmesi Tablo 3'te görülmektedir (Augenstein vd., 2017).

5.4.1. Akan veriden öğrenme

Sensör ağları, kredi kartı işlemleri, stok yönetimi, blog gönderileri ve ağ trafiği gibi gerçek dünya uygulamaları ile büyük miktarda veri kümeleri üretilmektedir. Veri

madenciliği yöntemleri, örüntü keşfi ve büyük veri kümelerinden ve akan verilerden çıkarımlar yapmak için önemlidir. Bununla birlikte, veri madenciliği yöntemlerinin dinamik ortamlarda büyük veri kümelerine uygulanması birliktelik kuralları, kümeleme ve sınıflandırma gibi geleneksel veri madenciliği tekniklerinin verimsizlik, ölçeklenebilirlik ve doğruluk eksikliği sorunları yaşamasına sebep olmaktadır (Aggarwal, 2007). Ayrıca akan verilerin boyutu, hızı ve değişkenliği sebebiyle tüm verilerin kalıcı olarak depolanması ve analiz edilmesi uygun değildir. Bu nedenle analitik tekniklerin optimize edilerek, veri örneklerini sınırlı kaynaklarla işlemek ve gerçek zamanlı doğru sonuçlar üretmek gerekmektedir (Gama, 2010).

Akan verilerin değişken olması, gelen akan veri örneklerinde öngörülemeyen değişikliklere neden olmaktadır. Kavram kayması adı verilen bu durum, geçmiş örneklerden elde edilen sınıflandırma modelinin doğruluğunu etkiler. Bu nedenle, kavram kaymasını tespit etmek ve uyum sağlamak için geliştirilmiş analitik yöntemlere ihtiyaç duyulmaktadır (Read vd., 2015).

5.4.2. Derin öğrenme

Derin öğrenme, makine öğrenmesi ve örüntü tanıma alanları için aktif bir araştırma alanı oluşturmaktadır. Bilgisayarla görme, konuşma tanıma ve doğal dil işleme gibi analitik uygulamalarda önemli rol oynamaktadır. Geleneksel makine öğrenmesi teknikleri, doğal verileri kendi ham biçiminde işleyebilmesi açısından sınırlıdır (LeCunvd., 2015). Derin öğrenme ise büyük veri setlerinde bulunan veri analitikleri ve öğrenme problemlerini çözmek için daha güçlüdür. Derin öğrenme, karmaşık veri örüntülerini büyük miktardaki denetimsiz ve sınıflandırılmamış ham veriden otomatik olarak çıkarmaktadır. Derin öğrenme, hiyerarşik öğrenme yapısına dayandığı için girdideki örüntüleri, algılayabileceği veya sınıflandırabileceği bir özellik vektörüne dönüştürür. Bu gibi avantajlarına rağmen, büyük veri ve derin öğrenme kavramlarının birlikte kullanılması birtakım zorlukları ön plana çıkarmaktadır (Najafabadi vd., 2015).

- Büyük verilerin hacmi: Eğitim aşamalarının büyük veriler üzerinde gerçekleştirilmesi zordur. Bu durumun sebebi öğrenme algoritmalarındaki yinelemeli hesaplamaların paralel hale getirilmesinin zorluğudur.

- Verilerin heterojenliği: Yüksek miktarda veri, derin öğrenme için büyük zorluklar getirmektedir. Büyük veri, çok sayıda girdi, çok sayıda çıktı ve yüksek boyutluluk (nitelikler) üzerinde çalışmak demektir. Bu nedenle, çalışma süresi karmaşıklığı ve model karmaşıklığının yanı sıra büyük verilerin merkezi bir işlemci ve depolama aygıtı vasıtasıyla eğitilmesi uygun değildir.

- Gürültülü sınıf etiketleri ve durağan olmayan dağılım: Büyük verilerin farklı ve heterojen kaynakları nedeniyle, veri eksikliği, sınıf etiketlerinde eksiklikler ve gürültülü etiketler gibi problemler ortaya çıkabilmektedir.

- Yüksek hız: Akan veriler son derece yüksek hızlarda üretilmektedir ve gerçek zamanlı olarak işlenmeleri gerekmektedir. Ayrıca veriler durağan olmayan dağılımlara sahiptirler ve zaman içinde değişen bir dağılım gösterirler.

5.4.3. Artımlı ve topluluk öğrenmesi

Artımlı öğrenme ve topluluk öğrenmesi iki dinamik öğrenme stratejisinden oluşmaktadır. Kavram kayması durumlarında akan verilerden öğrenmede temel yöntemdir. Stok eğilimi tahmini ve kullanıcı profillemesi gibi birçok uygulama alanında kullanılmaktadır. Artımlı öğrenme, yeni veriler alınırken hızlı bir şekilde sınıflandırma yapılmasına olanak tanır (Oussous vd., 2017). Çoğu geleneksel

makine öğrenmesi algoritması, artımlı öğrenmeyi desteklemektedir. Artımlı algoritmalara karar ağaçları (IDE4, ID5R), sinir ağları, Gauss RBF ağları (Learn ++, ARTMAP) ve artımlı destek vektör makinesi örnek olarak verilebilir. Topluluk algoritmaları ise daha esnektir ve kavram kayması durumlarında doğruluk değerinin sağlanması için kullanılmaktadır (Zang vd., 2014). Ayrıca, artımlı öğrenmede tüm sınıflandırma algoritmaları kullanılamaz ancak hemen hemen her sınıflandırma algoritması topluluk algoritmalarında kullanılabilir. Gerçek zamanlı veri işleme uygulamalarında, artımlı öğrenme daha uygundur. Akan verilerin karmaşık veya bilinmeyen dağılımları söz konusu olduğunda ise topluluk öğrenmesi daha kullanışlıdır (Skowron vd., 2016).

5.4.4. Temsili öğrenme

Yüksek boyutlu özelliklere sahip veri kümeleri günümüzde gittikçe daha yaygın bir hale gelmektedir. Mevcut öğrenme algoritmaları kullanılarak, bu verilerden bilgi çıkarımı yapmak ve düzenlemek zordur (Bargiela ve Pedrycz, 2016). Temsili öğrenme, belirli bir büyüklükteki öğrenilmiş veri temsili kullanılarak çok sayıdaki olası girişlerin belirlenmesini sağlayarak hesaplama verimliliği ve istatistiksel verimliliğin gelişmesini amaçlar (Bengio vd., 2013). Temsili öğrenme, özellik seçimi, öznitelik çıkarımı ve uzaklık metriği öğrenme kavramlarına dayanmaktadır. Temsili öğrenme konuşma tanıma, doğal dil işleme ve akıllı araç sistemleri gibi gerçek dünya uygulamalarında kullanılmaktadır (Huang ve Yates, 2012).

5.4.5. Dağıtık ve paralel öğrenme

Büyük verilerde, öğrenme algoritmalarının karşılaştığı sorunlardan biri de belirli bir zaman aralığında öğrenme için verilerin tamamını kullanamamalarıdır. Çözüm olarak geliştirilen dağıtık öğrenme yapıları, öğrenme sürecini çeşitli iş istasyonlarına dağıtarak öğrenme algoritmalarının ölçeklendirilmesini sağlamaktadır (Peteiro-Barral ve Guijarro-Berdiñas, 2013). Dağıtık öğrenme çerçevesinde, verilerin merkezi işleme için bir veritabanında toplanmasını gerektiren klasik öğrenme yapılarından farklı olarak, öğrenme işlemi dağıtık bir şekilde yürütülür. Dağıtık öğrenme, büyük miktarda veriyi yönetmek için dağıtık bilgi işlemenin avantajı ile merkezi işlemedeki tek bir iş istasyonuna veri toplamanın gerekliliğini ortadan kaldırarak zamandan ve enerjiden kazanç sağlar (Zheng, vd., 2011).

5.4.6. Aktarımlı öğrenme

Geleneksel makine öğrenmesi algoritmalarında temel varsayım, eğitim ve test verisinin aynı özellik aralığından seçilmesi ve aynı dağılıma sahip olmasıdır (Xiang vd., 2010). Büyük veriler göz önüne alındığında ise çeşitli kaynaklardan gelen veriler ile birlikte elde edilen veri, büyük ölçüde heterojen olmaktadır. Aktarımlı öğrenme, görevlerin ve dağılımların farklı olmasını sağlamak için bir veya daha fazla kaynak görevinden bilgi çıkarabilen öğrenme yöntemidir (Weiss vd., 2016). Aktarımlı öğrenme, büyük ölçekli belge sınıflandırması gibi gerçek dünya uygulamalarında uygulanmaktadır.

5.4.7. Aktif öğrenme

Gerçek dünya uygulamalarında büyük miktardaki verilerde bulunan veri etiketleri oldukça seyrek olabilmektedir. Bu gibi durumlarda, büyük miktarda etiketlenmemiş veriden öğrenmek oldukça zor ve zaman alıcıdır. Aktif öğrenme, etiketleme için en kritik örneklerin bir alt kümesini seçerek bu sorunu çözmeye çalışır (Settles, 2010). Aktif öğrenme mümkün olduğunca az sayıda etiketli veriyi kullanarak yüksek doğruluk elde etmeyi ve bu sayede etiketli veri elde etme maliyetini en aza indirmeyi hedeflemektedir (Crawford vd., 2013). Aktif

öğrenme yaklaşımları görüntü sınıflandırması ve biyolojik DNA tanımlaması gibi birçok veri işleme problemine uygulanmıştır (Qiu vd., 2016).

5.5. Akan veri madenciliği uygulamaları

Veri toplama teknolojilerindeki gelişmeler ile birlikte gerçek zamanlı üretilen verilere dayalı uygulamaların sayısında da artış yaşanmaktadır. Ağ trafiği izleme sistemleri ya da üretim süreçlerini ve binaları izlemek için kullanılan sensör ağları bu tür uygulamalara örnektir. Bu kaynaklardan gelen veriler, zaman aşımına uğramadan önce işlenmeli ve çıkarımlar gerçek zamanlı olarak yapılmalıdır (Wrench vd., 2016).

Geleneksel veri madenciliği algoritmaları çoğunlukla statik veri depoları üzerine odaklanmaktadır. Bununla birlikte, teknolojik gelişmeler akan verilerinin ortaya çıkmasına neden olmakta ve insanların verileri depolama, iletişim kurma ve işleme biçimlerini değiştirmektedir. Günümüzde pek çok organizasyon, her zamankinden daha yüksek hızda ve büyük miktarda veri üretmektedir. Örneğin, günlük olarak Google üzerinde 3,5 milyardan fazla arama yapılmakta, NASA uyduları yaklaşık 4TB görüntü üretmekte ve WalMart 20 milyondan fazla işlem kaydetmektedir. Akan veri üzerine yapılan araştırmalar kritik zaman kısıtları ile sonsuz sayıda, sürekli, hızlı ve zamanla gelişen akan verilerin nasıl modellenebileceği üzerinedir (Gagarin ve Toporivskyi, 2016).

Akan verilerin sürekli, sınırsız ve yüksek hızda olması nedeniyle algoritmalar tarafından işlenen veri miktarı çok büyüktür ve verileri tekrar tekrar işleyebilmek için yeterli zaman yoktur. Ayrıca sınırsız olan akan verileri depolayabilmek için yeterli alan mevcut değildir. Bu sebeple az miktarda bellek ile çalışabilen ve veriler üzerinden bir defa geçiş yapabilen algoritmalara ihtiyaç duyulmaktadır. Akan veri madenciliğinin amacı, örnek sayısına göre doğrusal olarak artan bir öğrenme süreci oluşturmaktır. Veriler sürekli olarak ulaştığı için, daha önce üretilen modele yeni bilgilerin dâhil edilmesi gerekmemekte ve zaman aşımına uğramış verilerin etkileri ortadan kalkmaktadır. Modeli yeni örneklerle yeniden eğitmek etkisiz ve yetersizdir. Bu nedenle, akan veri madenciliğinin diğer bir amacı, her örnek geldiğinde modeli kademeli olarak güncellemektir.

Akan verilerinin işlenmesi için kullanılacak algoritmaların özellikleri aşağıdaki gibi özetlenmektedir (Kamburugamuve vd., 2013):

- Veriler sürekli olarak tek bir öğe veya küçük veri yığınları halinde işlenmektedir.

- Algoritmanın veriler üzerinden bir defa geçiş yapacak şekilde tasarlanmış olması gereklidir.

- Bellek ve zaman kısıtları göz önüne alınmalıdır.

- Süreçlerin mevcut sonuçları sürekli olarak gözlemlenebilmelidir.

5.5.1. Akan verilerde yaygın örüntü madenciliği

Yaygın örüntü madenciliği, bir veri kümesi içinde sıkça bulunan örüntülerin elde edilmesine dayanmaktadır. Belirlenen minimum destek değerinden fazla sayıda olan örüntülerin sık görüldüğü temel alınmaktadır. Dinamik akan verilerdeki bu gibi örüntülerin bulunması, önemli zorluklar ortaya çıkarmaktadır. Mevcut algoritmalar, sistemin tüm veri setini birden çok kez taramasını gerektirdiği için akan verilerde kullanışlı değildir. Akan verilerde öğe kümelerinin sayısı üstel olarak arttığı için tüm verisetini taramak imkânsızdır (PhridviRaj ve GuruRao, 2014). Bu gibi zorlukların üstesinden gelmek için iki farklı yaklaşım kullanılmaktadır. Birincisi, önceden tanımlanmış, sınırlı sayıda öğe setini

taramaktır. Bu yöntem çok sınırlı kullanım ve ifade gücüne sahiptir. Çünkü sistemin, inceleme alanını yalnızca önceden tanımlanmış öğe setlerine göre önceden sınırlandırması gereklidir. İkinci yaklaşım, yaklaşık bir cevap kümesi türetmektir. Bu amaçla yaklaşık bir öğe seti sayma algoritması olan Kayıp Sayma Algoritması geliştirilmiştir. Örneğin, bir yönlendiricinin şimdiye kadar görülen tüm trafik akışının en az %1 (min. Destek) frekansında olan tüm öğelerle ilgilendiği varsayıldığında, belirlenen 0,1 destek değerine sahip öğelerin yalnızca bazılarının çıktı olarak alınacağını ifade etmektedir.

Kayıp sayma algoritmasında öncelikle minimum destek değeri olan σ ve $\pounds$ ile ifade edilen hata sınırı parametrelerinin belirlenmesi gerekmektedir. Gelen akan veriler $w = [1/\pounds]$ genişliğinde kovalara ayrılmaktadır. N şimdiye kadar görülen öğelerin sayısı yani akışın uzunluğunu ifade etmek üzere, algoritma frekansı 0'dan büyük olan tüm öğeler için bir frekans listesi kullanmaktadır. Her bir öğe için frekans sayısı olan f ve f'nin mümkün olan maksimum hatasını ifade eden Δ değerleri tutulmaktadır.

Yeni bir kova geldiği zaman kovadaki öğeler sıklık listesine eklenmektedir. Verilen bir öğe listede zaten varsa, frekans sayısı f arttırılmaktadır. Aksi takdirde verilen öğe frekans listesine değeri 1 olarak eklenmektedir. Eğer yeni öğe b. kovadan alınıyorsa, öğenin frekans sayımındaki mümkün olan maksimum hata değeri Δ, b-1 olarak belirlenmektedir. Kova boyutu w aşıldığı zaman, b kova boyutunu ifade etmek üzere $f+\Delta \leq b$ olacak şekilde bir öğe silinir. Algoritma, bu sayede frekans listesini ana hafızaya sığacak şekilde küçük tutmayı amaçlamaktadır.

Akan verilerde öğe sayısı üstel olarak artış gösterdiği için verileri kova yöntemiyle işlemek bellek tüketimi konusunda sorunlara yol açmaktadır. Bu sebeple eş zamanlı olarak birden fazla kova işlenmektedir. Yaygın örüntüleri belirleyebilmek için işlemler boyutları $w= 1/\pounds$ olan kovalara bölünmekte ve ana bellekten mümkün oldukça daha fazla kova okunmaktadır. Ana bellekten β kovanın okunduğu varsayıldığında verilen öğe kümesi frekans listesinde mevcutsa f değeri, β kovasındaki öğe setlerinin sayıları ile güncellenmektedir. Güncellenen giriş değeri, $f+\Delta \leq b$ eşitsizliğini sağlıyorsa giriş değeri silinmektedir. Ancak giriş değerinin frekansı f, β değerinden büyük eşitse ve listede bulunmuyorsa, giriş değeri $\Delta=b-\beta$ olacak şekilde listeye yerleştirilir.

Uygulama aşamasında, β değerinin büyük (Örneğin, 30 ve üzeri) seçilmesi bellekten tasarruf sağlayacaktır. Çünkü β'dan daha düşük frekanslı öğeler frekans listesine kaydedilmeyecektir. Daha küçük β değerleri, sıklık listesinin ortalama boyutunu ve yenileme hızını büyük ölçüde arttırarak zaman ve alan açısından algoritmanın verimliliğine zarar verecektir (Nguyen vd., 2015).

Tanbeer ve ark. tarafından 2009 yılında yapılan çalışmada, bir kayan pencere içerisindeki akan verilerdeki yaygın örüntülerin belirlenebilmesine yönelik yeni bir yöntem önerilmiştir. Kompakt örüntü ağacı adı verilen yöntem ile dinamik olarak akan verilerdeki güncel örüntüleri elde edip, geçmiş örüntülerin unutulması hedeflenmiştir. Deneysel çalışmalar, geliştirilen yönteminakan verilerdekigüncel örüntülerielde ederek bellek ve zaman karmaşıklığı açısından oldukça verimli olduğunu göstermektedir (Tanbeer vd., 2009).

Lee ve ark. tarafından 2014 yılında yapılan çalışmada, akan verilerdeki yaygın örüntülerin belirlenebilmesi amacıyla yeni bir yöntem sunulmuştur. Geliştirilen yöntem ile akan verideki tarama sayısı azaltılarak kayan penceredeki güncellemelerin daha sık

yapılması hedeflenmiştir. Deneysel çalışmalar, önerilen yöntemin çalışma zamanı, bellek kullanımı ve ölçeklenebilirlik açısından daha üstün performans sergilediğini göstermiştir (Lee vd., 2014).

Akan verilerde yaygın örüntü madenciliği konusunda yapılan literatürdeki çalışmalar incelendiğinde, temel olarak zaman ve bellek kısıtları ile ölçeklenebilirlik sorunları üzerinde durulduğu görülmektedir. Kayan pencereler üzerindeki güncellemelerin ve depolanan akan verilerdeki tarama sayısının azaltılması yoluyla geliştirilecek yöntemler başarılı sonuçlar verecektir.

5.5.2. Akan verilerin kümelenmesi

Kümeleme veya veri bölme, nesneleri kümeler olarak adlandırılan farklı birimlere gruplama işlemidir. Amaç, aynı kümedeki veri nesnelerinin benzer olması ve diğer kümelerdeki verilere benzememesidir. Saldırı tespiti, Web üzerindeki tıklama verilerinin analizi ve borsa analizleri kümeleme uygulamalarını içermektedir. Statik veri kümeleri üzerinde kümeleme yapmak için çok güçlü yöntemler olsa da akan verileri kümelemek, bu tür algoritmalara kısıtlamalar getirmektedir. Akan veriler için kullanılacak modellerin, akan verilerin dinamik ve zaman içerisinde gelişebilen yapılarına uygun, sınırlı bellek ve işlem süresi ile veriler üzerinden tek bir kez geçiş yapılmasını sağlayabilecek algoritmalar olması gereklidir. Akan verileri etkili bir şekilde kümelemek için kullanılacak metodolojilerin aşağıdaki özelliklere sahip olması gereklidir (Gama, 2010).

- Geçmiş verilere dair özetlerin hesaplanması ve saklanması: Sınırlı bellek alanı ve hızlı yanıt gereksinimleri nedeniyle, daha önce görülen verilerin özetlerinin hesaplanması, ilgili sonuçların saklanması ve gerektiğinde önemli istatistikleri hesaplamak için bu özetlerin kullanılmasını ifade etmektedir.

- Böl-yönet stratejisi: Akan verilerin varış sırasına göre bölümlere ayrılmasını ve bu parçaların özetlerinin hesaplanarak birleştirilmesini ifade etmektedir. Bu sayede daha küçük yapı taşlarından daha büyük modeller oluşturulabilmektedir.

- Akan verilerin artımlı kümelenmesi: Akan veriler sisteme sürekli ve artımlı olarak girdikleri için, türetilen kümelerin art arda işlenmesi gerektiğini ifade etmektedir.

- Zaman çerçevesi modeli: Yeni veriler, akan veri analizinde daha eski verilerden daha farklı rol oynadıkları için özetlenen verilerin anlık görüntüsünün farklı noktalarda depolanması için bir zaman çerçevesi modelinin kullanılmasını ifade etmektedir.

- Kümeleme işleminin çevrimiçi ve çevrimdışı süreçlere bölünmesi: Akan veriler geldikçe, verilerin anlık görüntülerinin temel özetleri hesaplanmalı, saklanmalı ve aşamalı olarak güncellenmelidir. Bu nedenle, bu gibi dinamik olarak değişen kümeleri korumak için çevrimiçi bir işlem gereklidir.

Akan verilerin kümelenmesinde amaç, belirli bir ana kadar görülen dizinin, küçük bir bellek ve zaman kullanarak sürekli olarak kümelenmesini sağlamaktır. STREAM algoritması, k-Medians problemi üzerine geliştirilmiş bir algoritmadır. K-medians problemi, N veri noktasını k kümeye bölerek küme merkezleri arasındaki toplam karesel hatanın en aza indirilmesini temel almaktadır. Temel amaç diğer kümelerdeki noktalardan farklı olarak aynı kümeye benzer noktaların atanmasıdır. Yüksek kaliteli kümeleme sağlamak için STREAM algoritması, bütün kovaları (yığın) ana belleğe yerleştirerek işlemektedir. Her bir $b_i$kovası için STREAM algoritması kovadaki verileri k kümeye yerleştirir. Daha sonra, yalnızca k merkezleri ile ilgili bilgileri koruyarak kova bilgilerini özetler ve her küme merkezi, kümesine atanan puan sayısı ile ağırlıklandırılır. STREAM daha sonra noktaları atarak yalnızca merkez bilgilerini korur. Yeterli merkezler toplandıktan sonra,

ağırlıklandırılmış merkezler tekrar başka bir O (*k*) küme merkezi üretmek üzere kümelenmektedir. Bu işlem, her seviyede en fazla *m* noktanın korunacağı şekilde tekrarlanmaktadır. Bu yaklaşım akan veriler için O(*kN*) zamanda sonuçlanmaktadır.

STREAM sınırlı depolama ve zaman kısıtlamalarına göre k-medians kümelenmeleri oluştururken zamanı dikkate almadığı için akan verilerdeki eski veriler, kümelemede egemen hale gelebilir. Ancak gerçek hayatta, kümeler hesaplandığı zamana ve ölçümlerin yapıldığı zamana göre değişebilir. Örneğin, bir kullanıcı geçen hafta, geçen ay veya geçen yılda meydana gelen kümeleri incelemek isteyebilir. Bu kümeler farklı olabileceği için akan veri kümeleme algoritmasının kümeleri, etkileşimli biçimde işlemesi ve kullanıcı tanımlı zaman aralıklarında hesaplama esnekliği sağlaması gereklidir (Nguyen vd., 2015).

CluStream algoritması, çevrimiçi kümeleme sorgularına dayalı olarak gelişen akan verilerin kümelenmesini sağlamaktadır. CluStream, kümeleme işlemini çevrimiçi ve çevrimdışı süreçlere bölmektedir. Çevrimiçi bileşen, mikrokümeleri kullanarak akan veriler hakkında özet istatistikleri hesaplayıp saklar ve artımlı çevrimiçi hesaplama gerçekleştirir. Çevrimdışı bileşen ise makro kümeleme yapar ve zaman çerçevesi modelini kullanarak saklanan özet istatistikler üzerinden çeşitli kullanıcı sorgularını cevaplar. Gelişmekte olan akan verileri zaman bilgilerine dayandırarak kümelemek için, zaman çerçevesi modeli kullanılmaktadır. Buradaki amaç, yeni olaylar için daha fazla bilgiye ihtiyaç duyulmasıdır. Saklanan bilgiler, geçmişle ilgili kullanıcıya özel kümeleme sorgularının işlenmesi için kullanılabilmektedir (Gaber vd., 2005).

Mikrokümeleme işlemi, istatistiksel veri toplama ve mikrokümelerin güncellenmesi adımlarıyla gerçekleşmektedir.

Her bir yeni veri noktası, mevcut bir kümeye eklenir ya da veri noktası için yeni bir küme oluşturulur. Yeni bir kümenin gerekli olup olmadığına karar vermek için her kümeye bir maksimum sınır tanımlanır. Yeni veri noktası sınırın içine girerse kümeye eklenir, aksi halde oluşturulacak yeni kümedeki ilk veri noktası olur. Yeni bir küme oluşturulacağı zaman, yeni kümeye bellek alanı yaratmak için en nadir kullanılan mevcut bir kümenin kaldırılması veya belirli kriterlere bağlı olarak var olan iki kümenin birleşmesi gerekmektedir (Aggarwal vd., 2003).

Zhang ve ark. tarafından 2014 yılında yapılan çalışmada, akan veriler için yeni bir kümeleme yöntemi geliştirilmiştir. Akan verilerin kümelenmesindeki temel zorluklar, kümeleri temsil edecek veri öğelerinin nasıl seçileceği ve dinamik veri dağılımlarındaki gelişen örüntülerin nasıl belirleneceğidir. Geliştirilen yöntemde, Affinity Propagation algoritması kullanılarak küme merkezleri seçilmiştir. Bu algoritma istatistiksel olarak değişen veri dağılımına göre kümeleme modelinin yeniden oluşturulmasını sağlamaktadır. Geliştirilen yöntem EGEE şebekeleri üzerinde test edilmiştir (Zhang vd., 2014).

Akan verilerin kümeleme, aykırılık tespiti ya da yaygın örüntülerin belirlenmesi gibi uygulama alanlarında kullanılmaktadır. Kümeleme işlemlerinde bellek ve zaman kısıtları dışında, dinamik olarak değişen veri dağılımlarının belirlenerek kümeleme modelinin veri öğelerine göre güncellenmesi gerekmektedir.

5.5.3. Akan verilerin sınıflandırılması

Geleneksel sınıflandırma işlemlerinde, eğitim verileri statik veritabanlarında bulunmaktadır ve çoğu sınıflandırma yöntemi tarafından birden çok kez taranmaktadır. Bu nedenle,

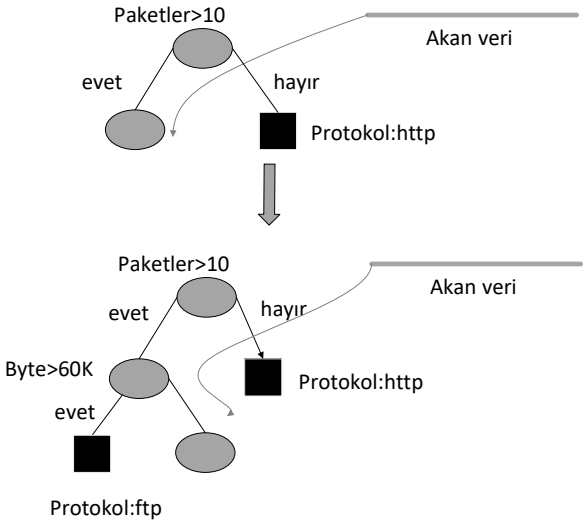
model oluşturma işlemleri çevrimdışı olarak gerçekleştirilir. Ancak akan veriler hızlı bir şekilde geldikleri için depolanmaları ve birden fazla taranmaları mümkün değildir. Akan verilerin diğer bir ayırt edici özelliği, yalnızca mevcut durumun depolandığı geleneksel veritabanı sistemlerine kıyasla zamanla değişiyor olmalarıdır. Verilerin niteliğindeki bu değişiklik, hedef sınıflandırma modelinde zaman içinde meydana gelen değişiklikleri alır ve kavram sürüklenmesi olarak adlandırılır. Akan verileri sınıflandırmak için Hoeffding Ağacı algoritması, Çok Hızlı Karar Ağacı algoritması ve Kavram Uyarlamalı Çok Hızlı Karar Ağacı algoritması geliştirilmiştir.

Hoeffding ağacı algoritması, akan verilerin sınıflandırılması için kullanılan bir karar ağacı öğrenmesi metodudur. Başlangıçta Web tıklama akışlarını izlemek ve kullanıcının hangi Web sunucularına ve Web sitelerine erişebileceğini önceden tahmin etmek için modeller oluşturmak amacıyla kullanılmıştır.

Temel olarak küçük bir örneklemin çoğunlukla optimum bir bölme özelliği seçmek için yeterli olabileceği fikrini kullanmaktadır. Bu fikir matematiksel olarak Hoeffding sınırı ile desteklenmektedir. Hoeffding Ağacı algoritması, girdi olarak nitelik değerleri A ve doğruluk parametresi δ ile tanımlanan bir dizi eğitim örneği almaktadır. Buna ek olarak bilgi kazancı, kazanç oranı, Gini indeksi veya başka bir öznitelik seçimi ölçüsü olarak değerlendirme fonksiyonu $G(A_i)$ kullanılmaktadır.

Karar ağacındaki her düğümde, her bir A_i için $G(A_i)$ değerinin maksimize edilmesi gereklidir. Hoeffding Ağacı algoritması, bölme özniteliğini seçerken bir düğümde gerekli olan örneklerin en küçük sayısı olan N değerini belirlemek için Hoeffding sınırını kullanır. Belirli bir düğüm için A_a en yüksek G'yi elde eden özellik, A_b ise ikinci en yüksek G'ye elde eden özellik olsun. £, Hoeffding sınırını ifade etmek üzere, $G(A_a)$-$G(A_b)$>£ eşitsizliği sağlandığı sürece A_a, 1- δ güven değeri ile en iyi bölme özelliği olarak seçilebilir.

Hoeffding ağaçları, aynı verinin birden çok taramasını yapmadığı ve artımlı olduğu için akan veriler için kullanımı uygundur. Şekil 1' de verilen örnekte yeni veri örneklerinin ağaç yapısına dâhil olma aşaması görülmektedir.



Şekil 1: Hoeffding Ağacına yeni veri örneklerinin eklenmesi

Bu özellik, modeli oluşturmadan önce verilerin biriktirilmesini bekleyen yöntemlerden farklıdır. Ağacın kademeli olarak oluşturulmasının diğer bir avantajı ise verilerin ağaç oluşturulurken bile sınıflandırma için kullanılabilmesidir (Kamburugamuve vd., 2013).

Çok Hızlı Karar Ağacı algoritması (ÇHKA), hız ve bellek kullanımını iyileştirmek için Hoeffding ağacı algoritmasının geliştirilmiş halidir. Nitelik seçimi sırasında yakın bağların kopması, eğitim örneklerinden sonra G fonksiyonunun hesaplanması, hafızanın tükenme durumu olduğunda en az kullanılan yaprakların devre dışı bırakılması yapılan geliştirmelere örnek olarak verilebilir. ÇHKA, akan veriler üzerinde yüksek hız ve doğruluk oranı ile çalışmaktadır. Ancak, akan verilerde yaşanan içerik kayması

durumlarının üstesinden gelememektedir (Aggarwal, 2007).

İçerik kayması durumlarını yönetebilmek için temel olarak, mevcut kavramlarla tutarlı olmayan akan veri öğelerinin zamanında tanımlanması gerekmektedir. Yaygın olarak kullanılan yöntem kayan pencere yapısının kullanılmasıdır. Buradaki amaç yeni örnekler eklemek ve eski örneklerin etkilerini ortadan kaldırmaktır. Kayan penceredeki örneklere art arda bir sınıflandırıcı algoritması uygulanabilir. Yeni örnekler geldiğinde pencerenin başına eklenir, eski örnekler pencerenin sonundan kaldırılır ve sınıflandırıcı algoritması yeniden uygulanır. Ancak bu teknik pencerenin boyutu *w* değerine duyarlıdır. *w* çok büyükse, model içerik kaymasını doğru bir şekilde belirleyemeyecektir. Öte yandan eğer w çok küçük ise, doğru bir model oluşturmak için yeterli örnek olmayacaktır ve sürekli yeni bir sınıflandırıcı modeli oluşturmak maliyetli olacaktır.

İçerik kayması durumlarının üstesinden gelebilmek için, ÇHKA algoritması üzerinden Kavram Uyarlamalı Çok Hızlı Karar Ağacı algoritması (KUÇHKA) geliştirilmiştir. KUÇHKA, kayan pencere yaklaşımını kullanmaktadır ancak her seferinde sıfırdan yeni bir model oluşturmaz. Ayrıca, yeni örneklerle ilişkili sayıları arttırarak ve eski örneklerle ilişkili sayıları azaltarak düğüm istatistiklerini güncellemektedir. Bu sayede, içerik kayması durumunda bazı düğümler artık Hoeffding sınırını geçememektedir. Bu durumda, yeni en iyi bölünme özelliği kökte olacak şekilde, alternatif bir alt ağaç oluşturulmaktadır. Akan veriden örnekler gelirken, alternatif alt ağaç sınıflandırma için kullanılmadan gelişmeye devam edecektir. Alternatif alt ağaç, mevcut ağaçtan daha doğru hale geldiğinde, eski alt ağaç değiştirilmektedir. Deneysel çalışmalar KUÇHKA'nın zamanla değişen akan verilerle ÇHKA'dan daha yüksek doğruluk oranlarına ulaştığını göstermektedir. Ayrıca, KUÇHKA'ndaki ağacın boyutu, ÇHKA algoritması ile oluşturulan ağaçtan çok daha küçüktür çünkü ÇHKA algoritması eskimiş örnekleri de biriktirmektedir (Gama, 2010).

Akan veriler için topluluk sınıflandırıcı yaklaşımında ise temel fikir, akan verilerin sıralı bölümlerinin bir sınıflandırıcı grubu (Örneğin, C4.5 ya da Naïve bayes) kullanarak eğitmektir. Sınıflandırıcılar, zamanla değişen bir ortamda, beklenen sınıflandırma doğruluğuna dayalı olarak ağırlıklandırılır. Karar ağaçları içerik kayması durumlarında etkili sonuçlar üretememektedir. Özellikle, KUÇHKA algoritmasında ağacının köküne yakın bir öznitelik Hoeffding sınırını geçmezse, ağacın büyük bir kısmının yeniden oluşturulması gerekmektedir. Ancak Naïve bayes gibi birçok sınıflandırıcı bu zayıflığı yaşamamaktadır. Naïve bayes sınıflandırıcıları, bir kararın güven değerini ifade eden sınıf etiketleri ile birlikte göreli olasılıkları da sunmaktadır. Ayrıca, KUÇHKA algoritmasında eski örneklerin elenmesi kullanışlı olmayabilir. Topluluk yaklaşımları, yalnızca en güncel örnekleri saklamak yerine, en az doğruluk oranına sahip sınıflandırıcıları elemektedir (Aggarwal, 2007).

Gomes ve ark. tarafından 2017 yılında yapılan çalışmada, gelişen akan verilerin sınıflandırılması için adaptif rastgele orman algoritması sunulmuştur. Geliştirilen algoritma, etkili bir yeniden örnekleme yöntemi ve farklı veri kümeleri için karmaşık optimizasyonlar olmadan farklı içerikkayması türleriyle başa çıkabilen adaptifyapılar içermektedir. Karşılaştırmalı deneysel çalışmalar geliştirilen algoritmanın kaynak kullanımı ve performans açısından daha verimli olduğunu göstermiştir (Gomes vd., 2017).

Akan verilerin sınıflandırılması konusunda yapılan literatür araştırmaları sonucunda,

geliştirilen algoritmaların içerik sapması durumlarını tespit edebilmesi ve gelişen veri öğelerine göre sınıf etiketlerini ve depolanan veri öğelerini güncellemesi gerektiği görülmüştür.

5.5.4. Akan verilerin filtrelenmesi

Akan verilerde filtreleme işlemi, belirli kriterlere uyan ifadelerin seçilmesi, geri kalanlarının atılmasını temel almaktadır. Örnek olarak, spam olmadığı bilinen bir milyar e-posta adresinin olduğu bir S akan verisi, e-posta adresi ve e-posta içeriği ikilisinden oluşmaktadır. Tipik e-posta adresi 20 bayt veya daha büyük bir boyutta olduğu için S akan verisinin ana bellekte depolanması kullanışlı değildir. Örnek olarak bir gigabayt hafıza olduğu düşünüldüğünde, Bloom filtrelemesi olarak bilinen teknikle ana bellek bir bit dizisi olarak kullanılmaktadır. Bu durumda bir bayt sekiz bite eşit olduğu için sekiz milyar bitlik alan kullanılabilecektir. E-posta adresleri ile sekiz milyar bit arasında bir hash fonksiyonu tasarlandığında S'nin her bir üyesi 1 bit ile gösterilir ve dizinin diğer tüm bitleri 0 kalır. S'nin bir milyar elemanı olduğundan, bitlerin yaklaşık 1/8 biti 1 olacaktır. Bir akan veri öğesi geldiğinde, onun e-posta adresi hash haline getirilir. E-posta adresinin hash biti 1 ise, üzerinden e-postaya izin verilir. Ancak 0 ise, adresin S'de olmadığı anlaşılır ve bu e-posta adresi spam olarak engellenir (Leskovec vd., 2014).

Bloom filtresi, bir öğenin bir grubun üyesi olup olmadığını test etmek için kullanılan olasılıksal bir veri yapısıdır. Bloom filtresinde, yanlış pozitif eşleşmeler mümkündür ancak yanlış negatif eşleşmeler mümkün değildir. Bu nedenle Bloom filtresi %100 duyarlılık (Recall) oranına sahiptir. Diğer bir deyişle, sorgu sonucu olarak "muhtemelen kümede" ya da "kesinlikle kümede değil" sonuçları döndürülmektedir (Aggarwal, 2007).

Bloom filtresi, başlangıçta tamamı 0 olan *n* bitlik bir diziden, *m* anahtar değerlerinin kümesi *S*'den ve hash fonksiyonundan oluşmaktadır. Bloom filtresinin amacı, anahtarları *S*'de olan tüm akan veri öğelerine izin vermek ve anahtarları *S*'de olmayan akım öğelerinin çoğunu reddetmektir. *m* farklı hash fonksiyonu, *n* dizi indisinden birine rastgele olarak yerleştirilir. Yeni bir eleman eklemek için, eklenecek eleman *m* farklı hash fonksiyonundan geçirilerek *m* farklı indis elde edilir ve bit dizisinin bu indislerdeki değeri 1 olarak ayarlanır. Yeni gelen bir anahtar değer, tüm hash fonksiyonlarında da 1 değerine eşleşiyorsa, *S* kümesinde bulunmaktadır denilir ancak tek bir tanesi bile 0 değeri ile eşleşiyorsa *S* kümesinde yoktur denilir. Yeni gelen anahtar değer eğer *S* kümesinde varsa, Bloom filtresinden kesin olarak geçmektedir. *x* toplam hedef sayısı, *yS* kümesindeki toplam eleman sayısı olmak üzere eşleşmeme olasılığı $(x\text{-}1)/x$'dir. Hiçbir *y* elemanının eşleşmeme olasılığı $(x\text{-}1/x)^y$'dir (Leskovec vd., 2014).

Bhoraskar ve ark. tarafından 2013 yılında yapılan çalışmada,akan veri öğelerinin önem düzeylerinegöre zaman ve depolama alanı açısından verimli bir indeksleme ve sorgulama düzeni geliştirmek hedeflenmiştir. Geliştirilen algoritma ile uygulamaya özel gereksinimleri dengelemek için veri semantikleri temel alınarak indeksleme ve sorgulama yapılmaktadır. Geliştirilen algoritma YouTube video akış verileri kullanılarak test edilmiştir. Analiz sonuçları, geliştirilen yöntemin geleneksel Bloom filtresine göre 4 kat daha iyi sonuç verdiğini göstermiştir (Bhoraskar vd., 2013).

Akan verilerin filtrelenmesindeki temel amaç, giderek büyüyen akan verilerin indekslenmesini sağlamaktır. Örneğin, Web üzerindeki akış verilerini işleyen sistemin önbelleği sınırlı depolama alanına sahiptir. Burada bütün veri öğelerini önbelleğe almanın maliyeti, veri öğelerinin boyutuyla orantılıdır. Akan verilerin filtrelenmesi

konusunda yapılan literatürdeki çalışmaların incelenmesi sonucunda, akan verilerin hash fonksiyonları kullanılarak indekslenmesi sayesinde sorgu ve depolama alanı optimizasyonunun sağlanabileceği görülmüştür.

5.5.5. Akan verilerde farklı eleman sayısının bulunması

Akan veri elemanlarının evrensel setlerden seçildiği bir akan veri içerisinde başlangıçtan itibaren ya da belirlenen bir arlıktan sonra kaç farklı öğenin görüldüğü belirlenmek istenebilir. Örnek olarak, birçok kullanıcıya ait istatistiklerin toplandığı bir Web sitesi düşünülürse buradaki evrensel set, o sitedeki oturum kümesidir ve kullanıcılar oturum açtığında bir akan veri öğesi oluşturulmaktadır. Bu durum, kullanıcıların benzersiz kullanıcı adlarıyla girdiği Amazon gibi bir Web sitesi için uygundur. Benzer şekilde, arama sorgusu girmek için kullanıcı girişi yapmayı gerektirmeyen ve kullanıcıları yalnızca sorguyu gönderdikleri IP adresiyle tanımlayan Google için uygundur.

Bu gibi sorunları çözmenin yolu ise ana bellekte bugüne kadar akan veride görülen tüm öğelerin bir listesini tutmaktır. Hash tablosu veya arama ağacı yapısında tutularak, yeni gelen öğenin daha önce görülüp görülmediği kontrol edilebilir. Ancak, farklı öğelerin sayısı çok yüksekse veya bir kerede işlenmesi gereken çok sayıda akış varsa (Örneğin, Yahoo! bir ayda sayfalarının her birinde görüntülenen benzersiz kullanıcı sayısını hesaplamak istemektedir) verilerin tümü hafıza tutulamaz (Bar-Yossef vd., 2002).

Flajolet-Martin algoritmasında evrensel küme elemanları uzun bir string bitine hash yapılarak farklı eleman sayısı tahmin edilebilmektedir. Bit dizisinin eleman sayısının evrensel kümenin eleman sayısından daha fazla olması gerekmektedir. Akan veri elemanları birden fazla hash fonksiyonu kullanılarak hash edilebilmektedir. Hash fonksiyonlarının akışın bir elemanı için hep aynı sonucu vermesi gerekmektedir. Flajolet-Martin algoritmasının temel fikri, akan veri içerinde ne kadar çok farklı eleman ile karşılaşılırsa görülecek farklı hash-değer ikilisi de o kadar fazla olacaktır. Daha farklı hash-değer ikilileri görüldüğünde, bu değerlerden birinin görülmemiş olma ihtimali artmaktadır. Akan veri içindeki herhangi bir a elemanı için $h(a)$ değeri sondaki maksimum sıfır sayısı R'yi ifade etmektedir. Akan veri içerisindeki farklı eleman sayısı 2^R ile hesaplanmaktadır. Bir a elemanının $h(a)$ değerinde sondaki r bitin 0 olma olasılığı 2^{-r} dir (Leskovec vd., 2014).

5.6. Akan veri madenciliğinde kullanılan yöntemler

Akan verilerini etkili bir şekilde işlemek için yeni veri yapıları, teknikler ve algoritmalara ihtiyaç duyulmaktadır. Akan verileri depolayacak sonsuz miktarda alan olmadığı için doğruluk ve depolama alanı arasında ters orantı mevcuttur. Ayrıştırma işlemleri, yapılan sorgulara yaklaşık cevapları döndürmek için kullanılabilen verilerin özetlerini sunmaktadır. Ayrıştırıcılar, esas veri kümesinden önemli ölçüde daha küçük bir veri yapısı olan özet veri yapılarını kullanmaktadır. Özetleme işlemleri, altta yatan verilerin tarafsız bir tahminini gerçekleştirmektedir. Örnekleme yöntemleri, kayıtların orijinal gösterimini kullandıkları için herhangi bir veri madenciliği uygulaması veya veritabanı işlemi ile kullanılabilmektedir (PhridviRaj ve GuruRao, 2014).

Akan verilerin örneklenmesindeki temel hedef, istatistiksel olarak akan veriyi temsil edebilecek bir altküme seçmektir. Verilerin rastgele olarak alınan küçük örneklemi ile genel olarak veri kümesinin özellikleri yakalanmaktadır. Verilerin örnekleminin hafızada tutulması ile bellek tasarrufu

sağlanarak sorgulara cevap verilebilmektedir. Akan verinin örneklenmesinde kullanılan temel yöntem rezervuar örneklemedir (Gama, 2010). Örnekleme yöntemleri, akan verilerdeki elemanların frekanslarının belirlenmesi için kullanılabilmektedir.

5.6.1. Rastgele örnekleme

Rastgele örnekleme tekniği genellikle yüksek boyutlu uygulamalar için kullanılmaktadır. Akan verinin boyutunun bilinmediği durumlarda rezervuar örneklemesi kullanılmaktadır. Akan veriden rezervuar olarak adlandırılan rastgele boyuttaki bir *s* örneği alınması temeline dayanmaktadır. Rezervuar örneklemesi, özellikle büyük boyuttaki verilerde maliyetli bir işlem olacağı için akış içerisinde belirli bir zamana kadar görülen öğelerin rastgele bir örneğini tutan bir *s* aday kümesi oluşturmaktadır. Akan veri gerçekleştikçe, her yeni elemanın rezervuardaki eski bir elemanın yerine geçme ihtimali vardır. Örnek olarak akış içerisinde *N* adet farklı eleman görülürse, yeni bir elemanın eski bir elemanın yerine geçme olasılığı *s/N* olur (Kamburugamuve vd., 2013).

5.6.2. Rezervuar örnekleme

Çevrimiçi olarak ulaşan, n öğeden oluşan bir akış göz önüne alındığında, herhangi bir anda yapılan rezervuar örneklemesi ile o zamana kadar gözlemlenen akış bölümünün *m* boyutunda rastgele bir *S* örneği elde edilmektedir. Doğal örnekleme prosedürü öncelikle akışın ilk *m* öğesinin *S*' ye eklenmesi ile başlamaktadır. İkinci adımda x_t öğesi *t* zamanında akış içerisinde görüldüğünde, x_t öğesi *S* örneklemine *m/t* olasılıkla eklenmektedir. x_t öğesi S örneklemine eklenirse, *S*' den rastgele bir öğe çıkarılmaktadır. Her seferinde $|S| = m$ eşitliğinin sağlanması amaçlanmaktadır (Gama, 2010).

Rezervuar örnekleme, *n* maddeli bir akan veriden *m* örneklerini rastgele seçmek için kullanılan yaklaşımlardır. Burada *n* değeri, genellikle ana belleğe sığmayacak kadar büyüktür. Örneğin, Google ve Facebook'daki arama sorgularının listesi *n* değerine örnek verilebilir. Burada $1 \leq m \leq n$ ve giriş dizisi akan veri olacak şekilde maksimum boyutu *m* olan bir rezervuar dizisi oluşturmak en basit çözümdür. [0..*n*-1] akışından rastgele olarak tek tek öğelerin seçilerek rezervuar dizisine eklenmesi, algoritmanın zaman karmaşıklığını $O(m^2)$ olmasına neden olmaktadır. Ancak *m* değeri büyüdükçe ve girişler akan veri formunda oldukça bu yöntem verimli olmamaktadır. O(*n*) zamanda bu işlemi gerçekleştirmek için [0..*m*-1] boyutunda rezervuar dizisi oluşturularak akan veri dizisinin ilk *m* öğesi rezervuar dizisine eklenmektedir. Daha sonra *m*+1. öğeden *n*. öğeye kadar olan öğeler düşünüldüğünde *i* mevcut öğenin akan veri dizisindeki indeksini ifade etmek üzere 0 *ile i* arasında rastgele bir sayı üretilmektedir. Üretilen sayının *j* olduğu varsayılırsa ve eğer *j*, 0 ile *k*-1 aralığında ise rezervuar dizisinin *j*. elemanı ile akış dizisinin *i*. elemanı değiştirilmektedir (Gama, 2010).

5.6.3. Kayan pencereler

Akan verileri rastgele örneklemek yerine akan verileri analiz etmek için kayan pencere modeli kullanılmaktadır. Temel fikir belirli bir ana kadar görülen veriler üzerinde hesaplamalar yapmadan, yalnızca son verilere dayanan kararlar verebilmedir. Her *t* zamanında yeni bir veri öğesi geldiği ve *w* pencere uzunluğunu ifade etmek üzere bu elemanın, *t* + *w* zamanında sona erdiği temel alınmaktadır. Kayan pencereler modeli son olayların önemli olabileceği hisse senetleri veya sensör ağları için kullanışlıdır. Kayan pencereler modeli, yalnızca küçük bir veri penceresi depolandığı için bellek gereksinimlerini azaltmaktadır (Aggarwal, 2007).

5.6.4. Histogramlar

Histogram tabanlı yöntemler statik veri kümeleri için yaygın olarak kullanılmaktadır. Histogram, bir akan verideki öğe değerlerinin sıklık dağılımını yaklaşık olarak hesaplayabilmek için kullanılabilen özet bir veri yapısıdır. Histogramlar, verileri bitişik kümeler halinde bölümlere ayırmaktadır. Kullanılan bölümlendirme kuralına bağlı olarak, genişlik (kova değeri aralığı) ve derinlik (kova başına öğe sayısı) değişebilmektedir. Kümeleme işlemine benzer mantıkla çalışan V-Optimal histogramlar, verilerin dağılımına göre her kova içindeki frekans değişimini en aza indirgeyen kova boyutları belirlemektedir. Oluşturulan histogramlar ile sorgu ifadelerine cevap döndürülmektedir (Gagarin ve Toporivskyi, 2016).

5.6.5. Eskizler

Özetleme yöntemleri, depolama hassasiyetini nasıl değiştirdiklerine göre farklılık göstermektedir. Örnekleme teknikleri ve kayan pencere modelleri, verilerin küçük bir bölümüne odaklanırken, diğer özetler genel olarak çok detaylı ayrıntıları özetlemeye çalışmaktadır. Histogramlar ve dalgacıklar gibi yöntemler veriler üzerinde birden çok geçiş gerektirirken, eskizler gibi yöntemler ise tek bir geçişte çalışabilmektedir. Örnek olarak U= (1,2, .., v) öğelerin evrenini ve A= (a_1, a_2, .., a_N) ise akışı ifade ettiği bir histogramda evren büyüdükçe A sırasındaki i frekansının belirlenmesi maliyetli olabilmektedir. Bu durumda an tahmini ön plana çıkmaktadır. Eskizler akan verideki farklı eleman sayısı (F_1) ve ikinci anın tahmini (F_2) uygulamalarında kullanılabilmektedir.

$$F_k = \sum_{i=1}^{v} m_i^k \quad (1)$$

Eş. 1' de görüldüğü gibi, v evren veya etki alanı boyutu, m_i, i' nin frekansı ve $k \geq 0$ olmak üzere F_0, dizideki farklı öğelerin sayısını ifade etmektedir. F_1, dizinin uzunluğunu (N), F_2 ise tekrar oranını ifade etmektedir. Bir veri kümesinin sıklık anları, sorgu cevaplama gibi veritabanı uygulamalarında veriler hakkında bilgiler sağlamaktadır. Ayrıca, uygun bölümleme algoritmasını belirlemek için paralel veritabanı uygulamalarında kullanışlı olan verilerdeki çarpıklık veya asimetri derecesini göstermektedir. U evreni, v değer sayısını ve N eleman sayısını ifade etmek üzere F_0, F_1 ve F_2 eskizleri O(logv+logN) uzayına yakınsamaktadır. Temel fikir her elemanı rastgele olarak z_i € {-1,+1} aralığına hashlemek ve $X = \sum_i m_i z_i$ rastgeleliği korumaktır. Daha iyi tahmin değerleri elde etmek için, birden fazla rastgele X_i değişkeni tutulabilmektedir. Bu değişkenlerin karesinin medyan değeri seçilerek, tahmin değerinin F_2'ye yakın olduğu doğrulanabilir. Veritabanı açısından eskiz bölme, akan veri sorgulama optimizasyonlarında eskiz performansını artırmak için kullanılmaktadır (Aggarwal, 2007).

6. Akan Verilerde Araştırma Konuları

Bilgi ve iletişim teknolojilerindeki gelişmeler, veri toplama ve işleme yöntemlerinde önemli ölçüde değişime yol açmıştır. Örnek olarak sensör teknolojisindeki gelişmeler ile çevre koşulları ile ilgili ayrıntılı ve zamansal veriler toplanabilmektedir. Bu sensörler yüksek hızda akanveri üretmektedir.

Veri madenciliği açısından bakıldığında, bu problem dinamik ve durağan olmayan bir ortamda sürekli bir olarak veri üreten çok sayıda değişkenle (sensörler) karakterize edilmektedir. Elde edilen akan veriler analiz edilerek çeşitli problemler için kararlar alınması aşamasında kullanılmaktadır.

Burada sensörler sürekli olarak bir akan veri ürettiği için bellek ve hesaplama gücü gibi kaynaklar sınırlıdır. Sınırlı kaynaklar ve hızlı veri üretimi göz önüne alındığında, bilgilerin

gerçek zamanlı olarak işlenmesi ve çok boyutlu akış analizi senaryoları oluşturulması gerekmektedir.

Akan veri madenciliği, makine öğrenmesi ve veri madenciliği konularında çalışma yapan araştırmacılar tarafından ele alınması gereken önemli bir çalışma alanıdır. 2. bölümde belirtildiği gibi akan verilerin özellikleri, akan veriler üzerinde makine öğrenmesi ve veri madenciliği tekniklerinin geliştirilmesinde, geleneksel statik verisetlerine göre dikkate alınması gereken daha fazla konu olduğunu göstermektedir. Akan verinin sürekliliğini yönetmek, enerji tüketimi, bellek gereksinimleri, elde edilen sonuçların doğruluğu, veri madenciliği sonuçlarının sınırlı bant genişliğine sahip bir kablosuz ağ üzerinden aktarılması, modelleme işlemlerindeki değişikliklerin zaman içinde sonuçlanması, sonuçların değişimine göre algoritmaları geliştirmek, kullanıcı gereksinimlerine göre etkileşim, farklı sistemler ile uyum, gerçek hayat uygulamalarındaki ihtiyaçlar ve veri ön işleme gibi konular ön plana çıkmaktadır (Gaber vd., 2005). Akan veri madenciliği uygulamalarında, verilerin sabit olmayan bir dağılımgöstermesi sebebiyle, öğrenme sisteminineski verileri unutmaya yönelik bir biçimde tasarlanması gerekmektedir. Akan verilerden öğrenme modellerinin, kavram sapmasını hesaba katan artımlı öğrenme algoritmalarını kullanması gerekmektedir.

Akan veriler doğası gereği düzensiz ve öngörülemez olabileceğinden, geliştirilen algoritmanın kaynakları dengeli bir şekilde kullanarak oluşan trafiği yönetebilmesi gereklidir. Kullanılan algoritmaların, büyük miktarlardaki verileri işlemek için indeksleme ve depolama gibi konularda kaynak kullanımı konusunda verimli olması gereklidir (Ikonomovska vd., 2007). Elde edilen sonuçların zaman ve kaynak kısıtlamaları göz önüne alınarak kabul edilebilir hata sınırları içinde olması gereklidir. Sonuçların kablosuz ortam üzerinden iletildiği durumlarda, süreci sınırlı bant genişliği ile tamamlamak için bazı ek önlemler alınmalıdır. Verilerin dinamik yapısı sebebiyle, geliştirilen algoritma ve modellerin yeni veri örneklerine göre güncellenmesi ve gerçek hayat uygulamalarındaki kullanıcı beklentilerinin karşılanması gereklidir (Debnath ve Chobe, 2014).

7. Sonuçlar

Akan veriler, anlık ve kalıcı sorgu gereksinimleri bakımından statik veri analizinden farklılık göstermektedir. Geleneksel veritabanı uygulamaları, verilerin kalıcı olarak depolandığı ve karmaşık sorgulamalar gerektiren uygulamalarda kullanılmaktadır. Veritabanı sorguları ihtiyaç anında gerçekleştirilmekte ve sorgu sonuçları veritabanındaki statik durumunu yansıtmaktadır. Ancak İnternet trafiği, finansal işlemler, e-ticaret ve sensör ağları gibi uygulama alanlarında, veriler gerçek zamanlı olarak akış halinde üretilmekte ve bu verilerin gerçek zamanlı olarak işlenmesi gerekmektedir. Akan verilerde yapılan sorgular ise veri kümesinin belirli bir zaman aralığındaki anlık görüntüsü üzerinden yapılan değerlendirmelere cevap döndürür. Akan veri üzerinden yapılan sorgular, veri öğeleri gelmeye devam ettikçe sürekli olarak değerlendirilir, saklanabilir ve güncellenebilir.

Dinamik verilerin sürekli, sınırsız ve yüksek hız özelliklerinden dolayı, hem çevrimdışı hem de çevrimiçi akan verilerde büyük miktarda veri bulunmaktadır. Bu sebeple veritabanlarını tekrarlı olarak taramak, akan verilerin ulaşma sırasını kontrol etmek ve çevrimiçi işlemlere ait tüm akan verileri saklamak mümkün değildir. Bu durum, arka plandaki algoritmaların işleyişi üzerinde kısıtlamalara yol açmaktadır. Verilerin dinamik özelliği, akan veri algoritmalarının yalnızca sınırlı bir bellek miktarına erişmelerine ve sınırlı sürelerde işlem

yapmalarına neden olmaktadır. Bu kısıtlamalara, bir algoritmanın bellekteki akış verisinin bir özetine veya taslağına dayalı olarak yaklaşık bir cevap üretmesi çözüm olarak sunulabilir. Akan veriler, statik verisetlerinden farklı olarak işlendikçe atılan veri öğelerine sahiptir. Anlık olarak ulaşan veri öğelerinin tamamının işlenmesi, depolanması ve tekrarlı olarak analiz edilmesi mümkün değildir. Bu sebeple akan veri analitiğinde veri ön işleme, veri analizi ve kaynak kullanımı konuları ön plana çıkmaktadır.

Bu çalışma kapsamında akan veri kaynakları, akan verilerin karakteristik özellikleri, akan verilerin türleri ve akan veriden öğrenme çerçeveleri, akan veri madenciliğinde kullanılan yaklaşımlar, akan verilerde makine öğrenmesi, akan veri madenciliği uygulamaları ve kullanılan yöntemler ile akan veri karakterizasyonu ve akan veri üretimi konuları kapsamlı ve detaylı bir şekilde incelenmiştir. Akan veriler, statik veriler ile karşılaştırmalı olarak analiz edilmiştir. Akan veriler, veri işleme özellikleri bakımından statik veriler ile birlikte analiz edilerek, statik veri madenciliği yöntemlerinin akan verilerin dinamik yapısına uyarlanması konusunda detaylı araştırmalar yapılmıştır. Akan veri madenciliğinde kullanılan yöntemler ve algoritmalar karşılaştırmalı olarak incelenmiştir.

Yapılan literatür araştırmaları ve incelemeler sonucunda, akan veri analitiği uygulamalarında göz önüne alınması gereken konular aşağıda listelenmiştir:

- Verilerin gelişmekte olan doğası göz önünde bulundurularak, veriler anlık olarak geldikçe gizliliğin sağlanması için yöntemler geliştirmek,

- Veriler arasındaki ilişkileri kullanarak geleceğe dönük öngörülerde bulunmak,

- Oluşturulacak geribildirim mekanizmaları ile modellerin veri kullanımını değerlendirmek,

- Veri ön işleme adımlarının sistematik bir metodoloji ile gerçekleştirilmesi,

- Sistem kaynaklarının kullanımını optimize eden modeller oluşturulması,

- Çevrimiçi güncellemeler yaparak verilerin güvenliğinin ve kaynak kullanımının dengelenmesini sağlamak.

8. Kaynaklar

Abrol, S., Rajasekar, G., Khan, L., Khadilkar, V., Nagarajan, S., Mcdaniel, N., Thuraisingham, B. 2017. "Real-time, stream data information integration and analytics system", *U.S. Patent Application No. 14/746,576*.

Aggarwal, C.C. 2014."Data classification: Algorithms and applications",*CRC Press*.

Aggarwal, C.C. 2007. "Data streams: models and algorithms", *Springer Science & Business Media*.

Aggarwal, C.C., Han, J., Wang, J., Yu, P. S. 2003."A framework for clustering evolving data streams",*In Proceedings of the 29th international conference on Very large data bases*, 81-92.

Augenstein, C., Spangenberg, N.,Franczyk, B. 2017."Applying machine learning to big data streams: An overview of challenges",*In Soft Computing & Machine Intelligence (ISCMI), 2017 IEEE 4th International Conference on*, 25-29.

Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J. 2002."Models and issues in data stream systems",*In Proceedings of the twenty-first ACM SIGMOD-*

SIGACT-SIGART symposium on Principles of database systems, 1-16.

Bargiela, A., Pedrycz, W. 2016."Granular computing",*In Handbook On Computational Intelligence: Volume 1: Fuzzy Logic, Systems, Artificial Neural Networks, and Learning Systems*, 43-66.

Bar-Yossef, Z., Jayram, T. S., Kumar, R., Sivakumar, D., Trevisan, L. 2002."Counting distinct elements in a data stream",*In International Workshop on Randomization and Approximation Techniques in Computer Science*, 1-10.

Bengio, Y., Courville, A., Vincent, P. 2013."Representation learning: A review and new perspectives", *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798-1828.

Bhoraskar, R., Gabale, V., Kulkarni, P., Kulkarni, D. 2013. "Importance-aware bloom filter for managing set membership queries on streaming data",*In 2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*, 1-10.

Bifet, A. 2016."Mining Internet of Things (IoT) Big Data Streams",*In SIMBig*, 15-16.

Bramer, M. 2007. "Principles of data mining",*Springer*.

Brief, A. I. 2012."Qualitative and Quantitative Research Techniques for Humanitarian Needs Assessment",*Acaps*.

Chandler, N., Hostmann, B., Rayner, N.,Herschel, G. 2011."Gartner's business analytics framework. Processes and platforms that need to be integrated and aligned to take a more strategic approach to business intelligence", *Gartner*.

Chen, X. W., Lin, X. 2014."Big data deep learning: challenges and perspectives", *IEEE access*, 2, 514-525.

Crawford, M. M., Tuia, D., Yang, H. L. 2013."Active learning: Any value for classification of remotely sensed data?", *Proceedings of the IEEE*, 101(3), 593-608.

Debnath, P., Chobe, S. 2014. "A Quick Survey on Data Stream Mining",*Int. J. Comput. Sci. Inf. Technol.*, 5(3).

D'Andrea, E., Ducange, P., Lazzerini, B., Marcelloni, F. 2015."Real-time detection of traffic from twitter stream analysis", *IEEE transactions on intelligent transportation systems*, 16(4), 2269-2283.

Ellis, B. 2014."Real-time analytics: Techniques to analyze and visualize streaming data",*John Wiley & Sons*.

Elnahrawy, E. 2003."Research directions in sensor data streams: solutions and challenges", *Rutgers University, Tech. Rep*.

Gaber, M. M., Zaslavsky, A., Krishnaswamy, S. 2005."Mining data streams: a review", *ACM Sigmod Record*, 34(2), 18-26.

Gagarin, O. O., Toporivskyi, B.P. 2016."Research issues of mining big data streams", *Infocom 2016*.

Gama, J. 2010. Knowledge discovery from data streams,*CRC Press*.

Gomes, H.M., Barddal, J. P., Enembreck, F., Bifet, A. 2017. "A survey on ensemble learning for data stream classification",*ACM Computing Surveys (CSUR)*, 50(2), 23.

Han, J., Pei, J., Kamber, M. 2011. "Data mining: Concepts and techniques",*Elsevier*.

Huang, F., Yates, A. 2012."Biased representation learning for domain adaptation",*In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1313-1323.

Huang, Y., Cui, B., Zhang, W., Jiang, J., 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.

Huisman, D. O. 2015. "To What Extent do Predictive, Descriptive and Prescriptive Supply Chain Analytics Affect Organizational Performance", *Yüksek lisans tezi, Twente Üniversitesi, Twente*.

Hurwitz, J., Nugent, A., Halper, F., Kaufman, M. 2013."Big data for dummies", *John Wiley & Sons*.

IBM. 2013."Descriptive, predictive, prescriptive: Transforming asset and facilities management with analytics", *IBM Software Thought Leadership White Paper*.

Ikonomovska, E., Loskovska, S., Gjorgjevik, D. 2007."A survey of stream data mining",*In Proceedings of 8th National Conference with International participation*, 19-21.

Johnson, A. 2015."Quantitative vs. Qualitative Data.Natural Resources", *Online Courses*.

Kamburugamuve, S., Fox, G., Leake, D., Qiu, J. 2013."Survey of Streaming Data Algorithms", *Tech. Rep., Indiana University*.

Klein, A., Do, H. H., Hackenbroich, G., Karnstedt, M., Lehner, W. 2007."Representing data quality for streaming and static data",*In Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, 3-10.

Kleppmann, M. 2016."Making Sense of Stream Processing", *O'Reilly Media Inc*.

Kohler, H. 2002. "Statistics for Business and Economics: Minitab Enhanced",*South-Western, Thomason Learning*.

Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J.,Woźniak, M. 2017."Ensemble learning for data stream analysis: A survey",*Information Fusion*, 37, 132-156.

Krishnaswamy, S., Gama, J., Gaber, M.M. 2012."Mobile data stream mining: from algorithms to applications",*In Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, 360-363.

Lee, G., Yun, U., Ryu, K. H. 2014. "Sliding window based weighted maximal frequent pattern mining over data streams",*Expert Systems with Applications*, 41(2), 694-708.

LeCun, Y., Bengio, Y., Hinton, G. 2015."Deep learning", *Nature*, 521(7553), 436-444.

Lennert, D., Maynard, W., Mehta, V., Huss, T. 2018. "Systems and methods for cross-platform batch data processing", *2018. U.S. Patent Application No. 15/397,583*.

Leskovec, J., Rajaraman, A., Ullman, J.D. 2014. "Mining of massive datasets",*Cambridge University Press*.

Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald,

R., Muharemagic, E. 2015."Deep learning applications and challenges in big data analytics", *Journal of Big Data*, 2(1).

Nguyen, H. L., Woon, Y. K., Ng, W.K. 2015."A survey on data stream clustering and classification", *Knowledge and information systems*, 45(3), 535-569.

Oussous, A., Benjelloun, F.Z., Lahcen, A.A.,Belfkih, S. 2017."Big Data Technologies: A Survey", *Journal of King Saud University-Computer and Information Sciences*.

Peary, B. D., Shaw, R., Takeuchi, Y. 2012."Utilization of social media in the east Japan earthquake and tsunami and its effectiveness", *Journal of Natural Disaster Science*, 34(1), 3-18.

Peteiro-Barral, D., Guijarro-Berdiñas, B. 2013. "A survey of methods for distributed machine learning", *Progress in Artificial Intelligence*, 2(1), 1-11.

PhridviRaj, M.S.B., GuruRao, C.V. 2014."Data mining–past, present and future–a typical survey on data streams", *Procedia Technology*, 12, 255-263.

Psaltis, A.G. 2017."Streaming Data: Understanding the Real-Time Pipeline", *Manning*.

Pyle, D. 1999. "Data preparation for data mining", *Morgan Kaufmann*.

Qiu, J., Wu, Q., Ding, G., Xu, Y., Feng, S. 2016."A survey of machine learningfor big data processing", *EURASIP Journal on Advances in Signal Processing*, 1(67).

Read, J., Perez-Cruz, F., Bifet, A. 2015."Deep learning in partially-labeled data streams",*In Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 954-959.

Settles, B. 2010."Active learning literature survey", *University of Wisconsin, Madison*, 52, 55-66.

Silva, J.A., Faria, E.R., Barros, R.C., Hruschka, E.R., De Carvalho, A.C., Gama, J. 2013."Data stream clustering: A survey",*ACM Computing Surveys (CSUR)*, 46(1), 13.

Skowron, A., Jankowski, A., Dutta, S. 2016."Interactive granular computing", GranularComputing, 1(2), 95-113.

Son, N.H. 2006."Data cleaning and Data preprocessing", *Mimuw University*.

Steinmetz, R., Nahrstedt, K. 2002. "Multimedia fundamentals, Volume 1: Media coding and content processing",*Pearson Education*.

Stephens, S., Tamayo, P. 2003."Supervised and unsupervised data mining techniques for the life sciences", *Curr. Drug Discov*, 34(36).

Tan, P.N. 2006. "Introduction to data mining",*Pearson Education India*.

Tanbeer, S.K., Ahmed, C.F., Jeong, B.S., Lee, Y.K. 2009. "Sliding window-based frequent pattern mining over data streams",*Information sciences*, 179(22), 3843-3865.

Weiss, K., Khoshgoftaar, T.M. ve Wang, D. 2016."A survey of transfer learning", *Journal of Big Data*, 3(1).

Wrench, C., Stahl, F., Di Fatta, G., Karthikeyan, V., Nauck, D.D. 2016."Data stream mining of event and complex event streams: A survey of existing and future technologies and applications in big data",*In Enterprise*

*Big Data Engineering, Analytics, and Management***,** 24-47.

Xiang, E. W., Cao, B., Hu, D. H., Yang, Q. 2010."Bridging domains using world wide knowledge for transfer learning", *IEEE Transactions on Knowledge and Data Engineering*, 22(6), 770-783.

Xu, S., Balazinska, M. 2011."Sensor Data Stream Exploration for Monitoring Applications",*DMSN*.

Zang, W., Zhang, P., Zhou, C., Guo, L. 2014."Comparative study between incremental and ensemble learning on data streams: Case study", *Journal of Big Data*, 1(1).

Zhang, X., Furtlehner, C., Germain-Renaud, C., Sebag, M. 2014. "Data stream clustering with affinity propagation",*IEEE Transactions on Knowledge and Data Engineering*, 26(7), 1644-1656.

Zheng, H., Kulkarni, S.R. ve Poor, H.V. 2011."Attribute-distributed learning: models, limits, and algorithms", *IEEE Transactions on Signal processing*, 59(1), 386-398.