Fırat Üniversitesi Müh. Bil. Dergisi
32(1), 267-277, 2020

Terör Saldırılarını İçeren Büyük Verinin Makine Öğrenmesi Teknikleri ile Analizi

Mustafa ULAŞ[1*], Barış KARABAY[2]
[1] Yazılım Mühendisliği, Mühendislik Fakültesi, Fırat Üniversitesi, Elazığ, Türkiye
[2] Bilgi İşlem Genel Müdürlüğü, T.C. Adalet Bakanlığı, Ankara, Türkiye
[*1] mustafaulas@firat.edu.tr, [2] bariskarabay@yandex.com

(Geliş/Received: 06/10/2019; Kabul/Accepted: 08/02/2020)

Öz: Bu çalışmada 1970-2017 tarihleri arasındaki çeşitli haber kaynaklarından ve uluslararası geçerliliği kanıtlanmış haber ajanslarından elde edilen terör verilerinin bir araya gelerek oluşturulan Global Terrorism Database (GTD) isimli veri kümesi ele alınmıştır. Terör olaylarının büyük veri çerçevesinde makine öğrenmesi teknikleri ile analizi ve sınıflandırma işlemleri gerçekleştirilmiştir. GTD veri kümesine makine öğrenmesi yöntemlerinden sınıflandırma algoritmaları uygulanarak, bir terör olayının hangi terör örgütü tarafından gerçekleştirildiğini tahmin eden büyük veri işleme aracı geliştirilmiştir. Bir terör olayında saldırının tipi, saldırı yapılan ülke, bölge, saldırının hedef kitlesi ve kullanılan silah türü gibi özellikler ele alınarak tahmin edilmede kullanılmıştır. Büyük veri işleme aracının geliştirilmesinde Apache Spark (PySpark) çatısı ve Python programlama dili kullanılmıştır. GTD veri kümesi içeriğinde bulunan en çok saldırı gerçekleştiren ilk 10 terör örgütü ele alınarak, altı farklı sınıflandırma algoritması uygulanmıştır. Bu algoritmalar arasında performans değerlendirmesi yapılmış ve karşılaştırılmıştır. Uygulanan algoritmalar arasından en yüksek ağırlıklı doğruluk oranı olarak K-En Yakın Komşu (KNN) algoritması % 98,2 ile en yüksek değer bulunmuştur. Lojistik Regresyon (LR) algoritmasının büyük veri kümesi için uygun olmadığı tespit edilmiştir.

Anahtar kelimeler: Büyük veri, Apache spark, makine öğrenmesi.

Analysis of Big Data Including Terrorist Attacks by Machine Learning Techniques

Abstract: In this study, the Global Terrorism Database (GTD), which was created by gathering terrorist data obtained from various news sources and internationally proven news agencies between 1970-2017, was discussed. Analysis and classification of terrorist events with machine learning techniques within the framework of big data were carried out. By applying classification algorithms from the machine learning methods to the GTD dataset, a large data processing tool has been developed which estimates by which terrorist organization a terrorist incident occurred. In a terrorist incident, the type of the attack, the country of attack, the region, the target audience of the attack and the type of weapon used were used to predict them. Apache Spark (PySpark) framework and Python programming language were used in the development of the big data processing tool. Six different classification algorithms have been applied by considering the top 10 terrorist organizations carrying out the most attacks in the GTD dataset. Among these algorithms, performance evaluation has been made and compared. The highest weighted accuracy rate among the applied algorithms, the K-Nearest Neighbor (KNN) algorithm was the highest with 98.2%. It has been determined that the Logistic Regression (LR) algorithm is not suitable for big data set.

Key words: Big data, Apache spark, machine learning.

1. Giriş

Dünyada yaşanmış olan terör olayları sebebiyle yıllardır toplumların ve devletlerin bir sorunu olarak görülmektedir. Terör saldırıları yüzünden ülke ekonomisi ve sosyo-psikolojik yapısı zarar görmektedir. Son zamanlarda terör olaylarının doğrusal nitelikte artış görülmektedir. Terör olayları üzerinde sosyolojik olarak birçok araştırma çalışmaları yapılmıştır. Bu durum tespiti kullanılarak istatistiksel, analiz ve makine öğrenmesi gibi alanlarda da konunun işlendiği ve incelendiği görülmektedir [1]. Büyük veri, yapısal, yapısal olmayan ve yarı yapısal veriler olarak düşünüldüğünde sürekliliği olan ve hızlı bir şekilde büyüyen veri kümelerinden oluşmaktadır. Büyük veri kümelerinin içerisinden önemli bilgilerin elde edilmesi geleneksel yöntem ve teknikler ile çok başarılı olmamaktadır. Bu problemin çözümü için büyük veri analizi araçları ve teknikleri geliştirilmiştir [2]. Bu çalışmada da büyük veri ve makine öğrenmesi kapsamında terör olayları incelenmiştir ve örnek büyük veri kümesi kullanılarak Global Terrorism Database (GTD) veri kümesi kullanılmıştır [3]. Bir saldırının gerçekleştirilmesinden sonra saldırıyı hangi örgütün veya terörist grubun yaptığına dair olan bilginin ortaya çıkartılmasına yardımcı olmak çalışmanın amacıdır. Terör olaylarını içeren büyük veri kümesine makine K-En Yakın Komşu (KNN), Naive Bayes (NB), Destek Vektör Makineleri (DVM), Rastgele Orman (RO), Karar Ağaçları (KA) ve Lojistik Regresyon (LR) makine öğrenmesi tekniklerinden sınıflandırma algoritmaları

[*] Sorumlu yazar: mustafaulas@firat.edu.tr. Yazarın ORCID Numarası [1] 0000-0002-0096-9693, [2] 0000-0002-8011-4555

uygulanmıştır. Uygulamada büyük veri işleme araçlarından Apache Spark ve Python dili kullanılarak araç geliştirilmiştir.

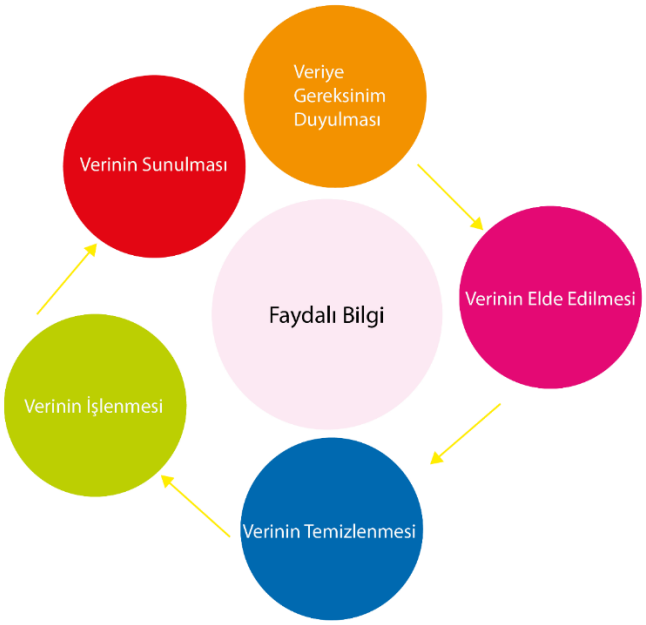
Yapılan araştırmalar neticesinde literatürde benzer çalışmaların bulunduğu yayınlar mevcuttur. Strang ve Sun yaptıkları çalışma ile Google News hizmeti üzerinden elde ettikleri veri kümesi ile terör gruplarının düşünceleri ile yaptıkları saldırılar arasında anlamlı ve bütünsel bir ilişki olduklarını ortaya koymuşlardır ve bunu model kurarak bağlantı oluşturmuşlardır [4]. Scott ve arkadaşları, 2006 yılında terör olaylarının belirgin bir nitelik düzeyinde olmadığından analiz işlemlerinde doğru sonuç alamadıklarını belirterek 1970-1997 yılları arasındaki terör olaylarını içeren verilere elde ederek Global Terrorism Database (GTD) isimli projeyi geliştirmişlerdir [5]. Miller ve Dugan, GTD veri kümesini ele alarak terör saldırılarının terör saldırılarının siyaset üzerinde nasıl bir etkisi olduğunu araştırmışlardır [6]. Khorshid ve arkadaşları, WEKA yazılımını kullanarak Orta Doğu ve Kuzey Afrika'da 2004-2008 yılları arasında meydana gelen terör olaylarının hangi örgüt tarafından gerçekleştirildiğini tahmin eden model geliştirmişlerdir [7]. Strang, Sun ve Vajihala diğer çalışmalarında, SPSS yazılımı üzerinde saldırı tipleri ve hedef ülke arasında bir ilişkisini göstermek için istatistiksel model geliştirmişlerdir [8].

2. Büyük Veri

Büyük veri, yeni teknolojiler ile birlikte sürekli ve hızlı bir şekilde artan veri kümeleridir. Çeşitliliği sebebiyle geleneksel veri tabanı teknolojilerinde depolanması zor olmaktadır. Büyük verinin içerisinden anlamlı, faydalı ve önemli bilgileri elde etmek için çeşitli analiz teknolojileri geliştirilmiştir. Büyük veri kümelerine günümüz örneklerinden sunucu ve istemci günlük kayıtları, arşiv verileri, sosyal medya etkileşimleri, hastane kayıtları, eğitim endüstrisi, müşteri hareketleri, devlet arşivleri, medya verileri ve çeşitli kaynaklardan elde edilen verilerdir. Büyük veri konusundan önemli ve faydalı bilgileri öğrenebilmek için geleneksel yöntemler yerine büyük veri olgusuna uygun teknolojiler tercih edilmelidir [9]. Büyük verinin doğru ve kararlı bir şekilde kullanılması, kurum ve devletlerin stratejik yönden, gelecekleri hakkında doğru ve etkin karar vermelerine olanak vermektedir. Dünya üzerinde büyük veri yapısı birçok alanda kurum, kuruluş ve devletler tarafından kullanılmaktadır. Eğitim sektörü, telekomünikasyon, yazılım, sağlık, enerji ve sosyal medya örnek olarak verilebilmektedir.

2.1. Büyük veri analiz

Büyük veri kümelerinden önemli bilgileri elde edebilmek için öncelikle verinin analizi yapılmalıdır. Analiz işlemlerinde belirli adımların yapılması gerekmektedir. Karmaşık ve çok boyutlu veri kümeleri için analiz işlemleri önemlidir. Şekil 1'de büyük verinin analiz aşaması gösterilmiştir.



Şekil 1. Analiz aşamaları

- Hangi alanda büyük veriye ihtiyaç tespiti ilk aşamada gerçekleşir. Burada hangi veri türüne ihtiyaç olduğu belirlenmelidir.
- Çeşitli kaynaklardan veriler elde edilerek toplanır. Bilgilerin tutarlı olmasına ve güvenilir kaynaktan elde edilmesine dikkat edilmelidir.
- Büyük veri oldukça yüksek boyutta olabilmektedir bu durum verinin kontrolünü sağlamakta zorluk çekilmesine sebep olabilmektedir. Toplanan verinin net olması, gerçek değerlerden oluşması ve tutarlı bir yapıda olması verinin temizlenmesini kolaylaştırmaktadır. Örneğin metin tipindeki bazı alanlara sayı tipleri girildiğinde verinin temizlenmesi gerekmektedir.
- Temizlenmiş olan veri kümesinin işlenmesi aşamasında büyük veri teknolojileri ile birlikte gelen makine öğrenmesi, derin öğrenme, yapay sinir ağları gibi çeşitli yöntemler kullanılmaktadır.
- Temizlenmiş ve yapısal hale getirilmiş olan verinin anlamlı ve faydalı bilgiye dönüştürüldüğü aşamadır. Burada çeşitli araçlar kullanılarak verinin sunumu gerçekleştirilir.

2.2. Kullanılan büyük veri teknolojileri

Sürekli ve doğrusal bir şekilde büyüyen veri kümelerinin kontrolü ve analizi zor olmaktadır. Büyük veri yapısından faydalı bilginin çıkarılması ve veri akışının kontrolünü sağlamak için bazı yazılımlar ve teknolojiler geliştirilmiştir. Bu çalışmada kullanılan ve uygulanan, teknoloji ve araçlar aşağıda anlatılmıştır.

Dağıtık Sistem: Birçok bilgisayarın haberleşme ağı üzerinde birbiri ile iletişim halinde olması ve bunu birbirinden bağımsız hareket ederek yapmasıdır. Bir uygulamanın işlevsel hale getirilmesi için birçok sistemin paralel olarak çalıştırılması işlemidir [10].

Map Reduce: Büyük veri kümelerini paralel olarak işlemek ve analiz etmek için kullanılan bir programlama modelidir [11]. Map ve Reduce, veri işlemesi yapabilmesine olanak sağlayan iki fonksiyondur. Elde edilen verinin işlenmesi için ilk olarak süzgeç yani filtreleme işlemi yapılması gerekmektedir. Bunu sağlayan ise Map fonksiyonudur. Reduce ise işlenmiş olan verinin analizi yapmak için kullanılır. Bu iki yöntem Map Reduce modelini ortaya çıkarmıştır.

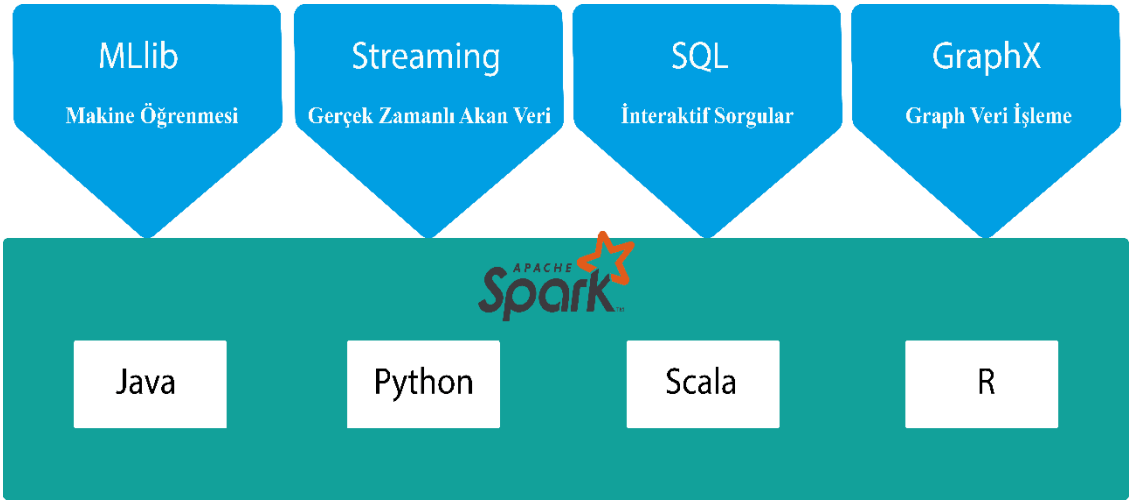
In-Memory Database (Bellek içi veri tabanı): Diğer veri tabanlarından farkı veri kümelerini disk yerine RAM(Random Access Memory) üzerinde saklar ve daha hızlı okuma, yazma ve yürütme işlemlerini yapabilmesini sağlamaktadır. Bellek maliyetinin azaldığı düşünülürse performanslı bir teknolojidir [12].

Hadoop: Dağıtık mimariler üzerinde bulunan farklı türdeki büyük verileri analizi ve işlenmesi için geliştirilmiş yazılım çerçevesidir. Google Map Reduce ve Google dosya sisteminden faydalanarak geliştirilmiştir. Hadoop Distributed File System (HDFS) olarak adlandırılmış bir dağıtık dosya sistemi ile Hadoop Map Reduce özelliklerini bir araya getiren açık kaynaklı bir ekosistemdir [13].

Apache Spark: Apache Vakfı tarafından açık kaynak kodlu olarak geliştirilen büyük veri kümelerinin analizinde ve işlenmesini kullanılan büyük veri platformudur. Apache Spark ile birlikte, büyük veri dosyaları In-Memory (bellek içi) teknolojisi sayesinde daha az maliyetli bir şekilde saklanır. Disk üzerinde Hadoop Map Reduce işleminden 10 kat daha hızlı veya bellekte 100 kat daha hızlı veri işlemesi yapabilen bir platformdur. Anlık olarak akan verileri kolaylıkla işleyebilme özelliği bulunmaktadır. Map / Reduce işleminden daha az kod yazılmasına ve zaman kazanılmasına olanak sağlamaktadır. Hadoop veya bulut ekosistemi üzerinde çalışabilmektedir. HDFS, HBase ve birçok veri tabanlarına doğrudan erişim sağlayabilmektedir. Spark, makine öğrenimi için MLlib, akan veriler için Spark Streaming, veri kaynaklarında interaktif bir şekilde sorgu yazılmasını sağlayan SQL, graf gibi veri türlerini işleyebilen GraphX gibi güçlü kütüphaneleri bünyesinde barındırır. Java, Scala, Python ve R gibi dilleri stabil bir şekilde desteklemektedir. Şekil 2 'de Apache Spark sistemi ve bünyesinde barındırdığı kütüphaneler gösterilmiştir [14].

Python: Python açık kaynak kodlu, nesne tabanlı, ara yüz desteğine sahip, yüksek seviyeli ayrıca bir derleyiciye gerek duymadan çalışan bir programlama dilidir. Sistem programlama, güvenlik, ağ uygulamaları, bilimsel hesaplama ve web uygulamalarında yaygın olarak kullanılmaktadır.

PySpark: Python üzerinde Spark uygulamaları geliştirebilmek için gerekli olan kullanıcı ara yüzü uygulamasıdır. İnteraktif sorgu oluşturma, veri kaynağı oluşturma ve okuma gibi işlemleri gerçekleştiren birçok metot barındırmaktadır.



Şekil 2. Apache spark bileşenleri [14]

Data Frame: Spark için tasarlanmış ilişkisel veri tabanlarındaki tablo yapısına benzer tablolardır. Satır, sütunlardan ve şemalardan oluşmaktadır. Diğer veri tabanı tablolarından farkı dağıtık mimariler üzerinde çalışabilme yeteneğine sahip olmasıdır [15]. Büyük veri yapısını işlevsel ve işlenebilir hale getirilmesini sağlamaktadır.

3. Makine Öğrenmesi Yöntemleri

Makine öğrenmesi, geçmişteki ve mevcut veriler kullanılarak gelecek için tahminsel hesaplamalar yapılmasına olanak sağlamaktadır. Makine öğrenmesi yöntemleri, bilgisayar yazılımları aracılığıyla geçmişteki verileri kullanarak yeni veriler için en uygun modeli tasarlayan yapılardır [16].

3.1. Sınıflandırma algoritmaları

Sınıflandırma, bir veri kümesindeki özellikleri kullanarak yeni gelen bir veri kümesinin hangi sınıfa ait olduğunu bulan yöntemdir. Sınıfları gruplamak için kullanılan eğitim veri kümesi, sınıfı bulunmak istenen veri kümesi test veri kümesi olarak adlandırılmaktadır. Sınıflandırma algoritmaları, bilgisayar yazılımları sayesinde eğitim veri kümesinden öğrendikleri bilgiyi, test veri kümesine uygulayarak yeni veri kümesinin sınıflarının bulunmasını sağlamaktadır. Sınıflandırma işlemini gerçekleştirmek için eğitim kümesi üzerinde kullanılacak olan algoritmayı uygulayarak model oluşturulur. Sonraki aşamada ise oluşturulan modelin test kümesinin üzerinde uygulanması ile sınıflandırılması yapmaktadır [17].

Destek Vektör Makineleri: Destek vektör makineleri, farklı sınıf etiketlerinin durumlarını birbirinden ayırmak için bir sınır oluşumu sağlamaktadır. Çok boyutlu bir alanda sınır çizgisi oluşturarak sınıflandırma işlemini gerçekleştiren makine öğrenmesi yöntemidir. Destek vektör makineleri veri kümelerinde verileri birbirinden ayırmak için kullanılmaktadır [18].

K-En Yakın Komşu Algoritması: K-En yakın komşu algoritması, eğitim kümesindeki özellikleri çok boyutlu bir özellik uzayında eşleştirilir. Uzay, Eğitim kümesinde bulunan hedef değişkenler tarafından bölgelere ayrılır. Yapılan hesaplamada bir noktaya en uygun sınıfa belirlenir. Veri kümesinin eğitim aşaması bağımsız öznitelik değerlerinden oluşan değerlerin ve eğitim örneklerinin sınıf etiketlerin bilgisini içerir. Sınıflandırma aşamasında, özellik değerleri eğitim kümesi üzerinde bulunmaktadır. Eski ile yeni değerlerin arasındaki uzaklık bir bağlantı ile hesaplanmaktadır. Daha sonra ise örnekler içerisinde k en yakın değeri seçilmektedir. Bu bağlamda, uygun sınıf bulunmaya çalışılmaktadır.

Naive Bayes: Naive Bayes sınıflandırıcı algoritması nesne tanıma ve sınıflandırma sorunlarını koşullu olasılık yöntemi ile çözmektedir. Koşullu olasılık ile sınıflandırmanın temeli Bayes teoremine dayanmaktadır. Sınıflandırma problemini çözerken bir hedef değişkenin özniteliklerini bağımlı olmadığını varsayarak hesaplama yapmaktadır. Naive Bayes sınıflandırma algoritması, makine öğrenmesi tekniklerinde çoklu sınıflandırma, metin sınıflandırma ve analizi problemlerinde sıklıkla kullanılmaktadır.

Lojistik Regresyon: Sayısal ve metinlerin hem regresyon hem de sınıflandırılma problemlerinin çözümünde uygulanan algoritmadır. Lojistik regresyon, bir veri setinde hedef (bağımlı) değişkenin iki farklı değer alması durumunda kullanılmaktadır. Örnek olarak 0 veya 1, başarılı veya başarısız gibi. Lojistik regresyon sınıflandırma algoritmasının gerçek hayatta uygulanmasında ikili (Biominal), sıralı (Ordinal) ve çoklu sınıf (Multinominal) olarak üç yöntemi bulunmaktadır. Multinominal yaklaşımda, hedef değişken üç ve daha fazla farklı değer alabilmektedir. Örneğin (0, 1, 2) veya (iyi, kötü, orta) gibi.

Karar Ağaçları: Ağaç yapısı biçiminde olan, koşullu olarak değerleri gruplayarak sınıflandırma yapabilme prensibine dayanmaktadır. Kuralları yönetme, oluşturma, bilgisayar yazılımları arasındaki bağlantının ve desteğinin yoğun olması sebebiyle günümüzde yoğun olarak kullanılmaktadır.

Rastgele Orman Algoritması: Rastgele Orman algoritmasının karar ağaçlarından farkı rastgele olarak birden fazla ağacından orman oluşturma felsefesine dayanmaktadır. Ağaçların sayısı arttıkça algoritmanın doğruluğu artmaktadır. Rastgele orman algoritması, sınıflandırma ve regresyon problemleri için uygun bir çözümdür.

3.2. Sınıflandırma başarım ölçütleri

Bir algoritmanın oluşturduğu performansı değerlendirebilmek için temel olarak kullanılan ölçütler; doğruluk oranı, f1 skoru, hata oranı, kesinlik ve duyarlılık değerleridir. Modelin başarımının iyi olması doğru ve yanlış sınıflandırılanların sayısı ile alakalıdır.

Karışıklık Matrisi: Veri kümesinin modele uygulanması ile birlikte doğruluk oranı, f1 skoru, hata oranı, kesinlik ve duyarlılık değerlerinin hesaplanabilmesi için karışıklık matrisi tablo yapısı ortaya çıkmaktadır. Karışıklık matrisi, sınıflandırmada modelin başarımını görselleştirilmesini sağlayan yapıdır. İçerisinde doğru ve yanlış tahmin edilen sınıflandırıcının değerleri görülebilmektedir. Tablo 1'de n sınıflı, çoklu sınıflandırmada kullanılan hata matrisi görülmektedir. Burada, satır kısmındaki sınıflar modelin eğitilmeden önceki durumunu belirtmektedir. Sütun kısmındaki sınıflar ise model eğitildikten sonra tahmin edilen değerleri belirtmektedir.

Tablo 1. Çoklu sınıflandırıcı için hata matrisi

Hata Matrisi		Öngörülen Sınıf		
		$X_0 \ldots X_{k-1}$	X_k	$X_{k+1} \ldots X_n$
Doğru Sınıf	$X_{k+1} \ldots X_n$	**TN**	**FP**	**TN**
	X_k	**FN**	**TP**	**FN**
	$X_0 \ldots X_{k+1}$	**TN**	**FP**	**TN**

$(0 \leq k \leq n)$Aralığında bir k sınıfı düşünüldüğünde, n sınıflı bir sınıflandırıcının hata veya karışıklık matrisinde şu 4 farklı sonuç elde edilir.

TP (True Positive) : Doğru pozitif
FP (False Positive) : Yanlış pozitif
TN (True Negative) : Doğru negatif
FN (False Negative) : Yanlış negatif

i tahmin edilen sınıf, **j** ise öngörülen sınıf olmak üzere; TP, TN, FP ve FN için Denklem 1-4 elde edilir.

$$TP = X_{kk} \quad \textbf{(1)}$$

$$TN = \sum_{i \in N} X_{ij} \quad \textbf{(2)}$$

$$FP = \sum_{i \in N} X_{ik} \quad \textbf{(3)}$$

$$FN = \sum_{i \in N} X_{ki} \quad \textbf{(4)}$$

Doğruluk: Algoritmanın veri kümesine uygulanması ile ortaya çıkan doğru tahmin edilenlerin uygulanan veri kümesine oranıdır.

$$\frac{TP+TN}{FP+FN+TP+TN} \quad \textbf{(5)}$$

Hata Oranı: Yanlış tahmin edilenlerin toplam tahmin edilenlere oranıdır. Doğruluk oranını 1'den çıkarılması ile de sonuç bulunabilir.

$$\frac{FP+FN}{FP+FN+TP+TN} \quad \textbf{(6)}$$

Hassasiyet: Pozitif durumda olan tahminlerin ne kadar başarılı olduğunu bulan ölçüttür.

$$\frac{TP}{TP+FN} \quad \textbf{(7)}$$

Kesinlik: Doğru ve pozitif olan tahminlerin başarısını gösteren ölçüttür.

$$\frac{TP}{TP+FP} \quad \textbf{(8)}$$

F1 Skoru: Hassasiyet ve kesinlik ölçütlerinin harmonik ortalamasından elde edilen ölçüttür.

$$2*\frac{(Hassasiyet*Kesinlik)}{(Hassasiyet+Kesinlik)} \quad \textbf{(9)}$$

3.3. Özellik seçimi

Özellik seçimi, temel anlamda veri kümelerinde bulunan sütunlardan, sınıflandırma işlemlerinde sonucu etkileyecek özniteliklerin seçilmesini ve elimine edilmesini sağlayan yöntemlerdir. Sınıflandırma yapılacak alana bağlı olarak ilgisiz özellikler veri kümesinden atılarak doğruluk oranını arttırmak, hesaplama ve çalışma sürelerini azaltmaktır. Özellik seçimi, veri yönetimini kolaylaştırmak, özellikler arasındaki ilişkileri daha iyi tanımlamak ve algoritmanın oluşturduğu modeli daha net olarak görebilmek için yapılmaktadır. Özellik seçimi yöntemleri kullanılarak gereksiz öznitelik değerleri temizlenir, veri boyutunun minimize edilmesini sağlar. Özellik belirlemede çeşitli yöntemler geliştirilmiştir.

Bu çalışma da, Özyinelemeli özellik eleme yöntemi kullanılmıştır. Öncelikle veri kümesine uygulanacak olan model belirlenmektedir. Daha sonra ise model oluşturularak veri kümesi içerisinde yer alan tüm öznitelikler modele eklenmektedir. Bu durum özyinelemeli olarak uygulanan modelin doğruluk oranının en yüksek seviyeye ulaşana kadar devam etmektedir. Doğruluk oranı yüksek olan aşamada eklenen öznitelikler seçilmektedir [19]. Örnek büyük veri kümesine, özyinelemeli özellik eleme yöntemi uygulandığında 135 adet öznitelik değerinden 67 tanesinin uygun olduğu tespit edilmiştir. Seçilen bu özelliklerden ilk 10 değeri ele alındığında ülke, bölge saldırı tipi, hedef ana kitle ve silah tipi özellikleri seçilmiştir. Bu bilgiler dışındaki özelliklerin boş değer fazla içermesi sebebiyle elimine edilmiştir.

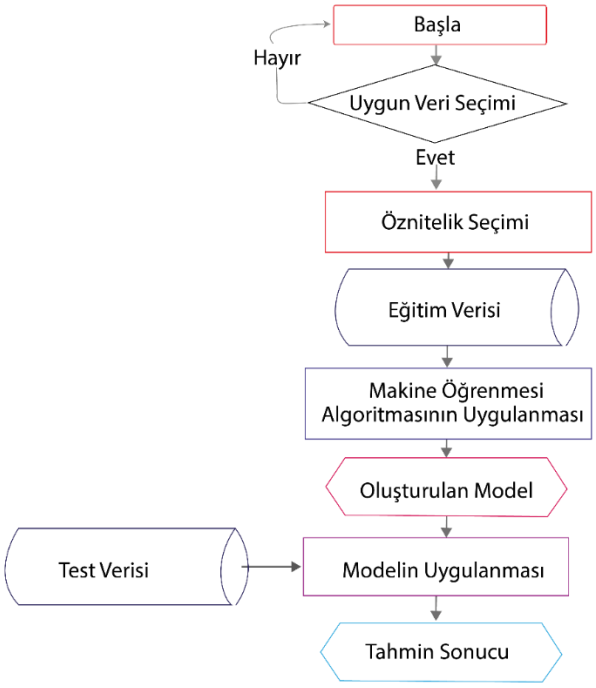
3.4. Özellik dönüşümü

Veri kümesinde yer alan özniteliklerin ağırlıklandırma, format dönüştürme ve değişim işlemlerinin gerçekleştirerek sınıflandırma işleminde sıkça kullanılan yöntemlerdir. Örneğin kategorik olan bir değişkenin nümerik bir değere dönüştürülmesi işleminin yapılmasıdır. Kullanılan öznitelik değerlerinin bir çıktı olarak vektörel bir biçime dönüştürülmesi işlemi de bu konuya örnek verilebilir. Tokenizer, Stop Words Remover, n-gram, Binarizer, PCA, Polynomial Expansion, Discrete Cosine Transform (DCT), String Indexer, One Hot Encoder, Vector Assembler, Index To String, Vector Indexer ve Min-Max Scaler yöntemleri özellik dönüşümünde kullanılmaktadır [20]. Çalışma kapsamında String Indexer, Index To String, One Hot Encoder ve Vector Assembler yöntemleri kullanılmıştır.

4. Uygulama

Geliştirilen uygulama ile birlikte büyük veri kümesindeki özellikler ve nitelik kullanılarak makine öğrenmesi algoritmaları ile model geliştirilmiştir. Sınıflandırma işlemlerini örnek büyük veri kümesi üzerinde uygulanması için büyük veri teknolojileri incelenmiştir. Literatür ve bilişim alanında yapılan araştırmalar sonucunda Apache Hadoop büyük veri teknolojisinin disk tabanlı kayıt işlemleri gerçekleştirmesi nedeniyle yavaşlık problemi

bulunmaktadır. Bu sebeple bu çalışmada Apache Spark büyük veri teknolojisi tercih edilmiştir. Kullanılan algoritmaların sınıflandırma problemlerinde yaygın olarak kullanılması, kaynak taramasının geniş olması ayrıca kütüphane desteğinin olması sebebiyle bu çalışmada tercih edilmiştir. Bu çalışmada uygulanan algoritmalar ile model oluşturma aşaması Şekil 3'te ki gibi gerçekleştirilmiştir.



Şekil 3. Uygulamanın çalışma mantığı

4.1. Kullanılan veri kümesi

Bu çalışmada veri kümesi olarak dünyada yaşanan terör olaylarını içeren Küresel Terörizm Veri Tabanı (GTD) kullanılmıştır. Veri kümesinde, 1970 yılından itibaren 2017 yılına kadar olan terör olaylarının detayları ile beraber bilgisini içermektedir. Veri kümesi yaklaşık 170 MB, içeriğinde 181.692 satır bulunmaktadır. Örnek veri kümesinde 135 adet sütun bulunmaktadır. Burada veri kümesine filtreleme işlemi uygulanarak en fazla eylem gerçekleştiren ilk 10 örgüt ele alınmıştır.

Tablo 2. Veri kümesi içeriği

Alan	**Açıklaması**
country	Ülke
region	Bölge
attacktype	Saldırı tipi
targtype_txt	Hedef ana kitle
Gname	Terör örgütünün adı
Weaptype	Silah tipi

Eğitim kümesi için **29.540**, test kümesi için ise **7.403** adet veri ayrılmıştır. Sınıflandırma işlemlerinde 6 adet sütun ele alınmıştır. Tablo 2'de bu çalışmada kullanılan örnek veri kümesinin alanlarının içeriği belirtilmiştir.

4.2. Uygulama sonuçları

Uygulama, Python programlama dili, makine öğrenmesi ve büyük veri teknolojileri olarak PySpark ve SkLearn kütüphaneleri, grafik ara yüzünün tasarlanması için Tkinter kullanılmıştır. Yapılan çalışmanın sonucunda çıktıların hesaplanmasında örnek büyük veri kümesi olan GTD'ye makine öğrenmesinde sıkça kullanılan 6 temel algoritmanın uygulanması ile ilgili saldırıları düzenleyen terör örgütü tahmin etmede oluşan çıktıların performans ölçütleri ele alınmıştır. Uygulamanın geliştirme aşaması, doğruluk oranlarının hesaplanmasında ve test işlemlerinde Intel i7-4570 3.14 GHZ CPU ve 128 GB SSD ve 1 TB SATA disk kullanılmıştır. Sistem üzerinde bulunan RAM 'in 32 GB'lik kısmı bellek hatalarını önlemek adına JVM üzerinden Apache Spark için ayrılmıştır. Veri kümesinin yüklenmesi aşamasında, modelin başarımının artması ve iyileştirilmesi için yaşanan terör olaylarında ilk 10 örgüt ele alınmıştır. Gerçekleştirilen uygulamanın analizi için karışıklık matrisleri oluşturulmuştur. Tablo 3 Karar Ağaçları algoritması ile elde edilen sonuçlar görülmektedir. Aynı şekilde Tablo 4.'e Lojistik Regresyon, Tablo 5'te Naive Bayes, Tablo 6. 'da Rastgele Orman, Tablo 7.'de KNN ve Tablo 8.'de Destek Vektör Makinelerine ait karışıklık matrisleri de verilmiştir. Bu sonuçlara bakıldığında her bir sınıf için karışıklık değerleri belirtilmiştir.

Tablo 3. KA için karışıklık matrisi

	Taliban	DEAŞ	SL	FMLN	EŞ	NPA	IRA	FARC	BH	PKK
Taliban	1444	0	0	0	0	0	12	0	0	0
DEAŞ	0	971	0	0	152	5	0	0	0	4
SL	0	0	942	0	0	0	0	1	0	0
FMLN	0	0	0	683	0	0	0	2	0	0
EŞ	0	0	0	0	637	0		0	0	0
NPA	0	0	0	0	0	562	0	0	0	0
IRA	0	0	0	0	0	0	545	0	0	2
FARC	0	0	0	0	0	0	0	489	0	0
BH	0	0	0	0	424	0	0	0	59	0
PKK	0	3	0	0	435	0	1	0	0	30

Tablo 4. LR için karışıklık matrisi

	Taliban	DEAŞ	SL	FMLN	EŞ	NPA	IRA	FARC	BH	PKK
Taliban	1449	3	1	3	0	0	0	0	0	0
DEAŞ	184	947	1	0	0	0	0	0	0	0
SL	330	0	492	121	0	0	0	0	0	0
FMLN	409	0	83	193	0	0	0	0	0	0
EŞ	135	499	0	0	3	0	0	0	0	0
NPA	476	4	76	6	0	0	0	0	0	0
IRA	7	3	0	0	0	0	537	0	0	0
FARC	407	0	42	40	0	0	0	0	0	0
BH	103	380	0	0	0	0	0	0	0	0
PKK	144	324	0	0	1	0	0	0	0	0

Tablo 5. NB için karışıklık matrisi

	Taliban	DEAŞ	SL	FMLN	EŞ	NPA	IRA	FARC	BH	PKK
Taliban	1444	0	0	3	2	2	0	0	0	5
DEAŞ	1	832	1	1	32	3	0	160	6	96
SL	1	1	819	36	0	85	0	1	0	0
FMLN	0	81	9	561	1	2	0	31	0	0
EŞ	0	159	0	0	393	0	0	2	25	58
NPA	0	0	207	65	0	289	0	0	0	1
IRA	0	6	2	0	1	0	535	2	1	0
FARC	0	47	0	128	0	0	0	314	0	0
BH	0	94	0	0	116	0	0	21	235	17
PKK	4	32	0	0	35	20	0	19	4	355

Tablo 6. RO için karışıklık matrisi

	Taliban	DEAŞ	SL	FMLN	EŞ	NPA	IRA	FARC	BH	PKK
Taliban	1448	0	8	0	0	0	0	0	0	0
DEAŞ	4	1122	0	0	0	5	0	0	0	1
SL	0	0	940	0	0	0	0	3	0	0
FMLN	11	0	2	671	0	0	0	1	0	0
EŞ	0	42	1	0	583	0	0	0	11	0
NPA	0	0	6	0	0	556	0	0	0	0
IRA	1	8	1	0	0	0	537	0	0	0
FARC	110	0	0	97	0	0	0	282	0	0
BH	45	203	0	0	45	0	0	0	190	0
PKK	1	249	0	0	0	0	0	0	0	219

Tablo 7. KNN için karışıklık matrisi

	Taliban	DEAŞ	SL	FMLN	EŞ	NPA	IRA	FARC	BH	PKK
Taliban	528	0	0	0	0	0	1	0	0	0
DEAŞ	0	514	0	0	0	1	0	0	0	4
SL	0	0	482	4	1	0	0	0	0	0
FMLN	0	0	1	913	0	0	0	0	0	0
EŞ	0	0	9	0	674	0	0	0	0	0
NPA	0	4	0	0	0	476	0	3	0	7
IRA	5	0	0	0	0	0	1524	0	0	0
FARC	0	0	0	0	0	2	0	653	19	5
BH	0	0	0	0	0	2	0	8	453	1
PKK	5	2	0	0	0	37	1	7	1	1042

Tablo 8. DVM için karışıklık matrisi

	Taliban	DEAŞ	SL	FMLN	EŞ	NPA	IRA	FARC	BH	PKK
Taliban	528	0	0	0	0	0	1	0	0	0
DEAŞ	0	516	0	0	0	3	0	0	0	0
SL	0	0	483	4	0	0	0	0	0	0
FMLN	0	0	1	913	0	0	0	0	0	0
EŞ	0	0	0	0	683	0	0	0	0	0
NPA	0	4	0	0	0	483	0	0	0	0
IRA	0	0	0	0	0	0	1529	0	0	0
FARC	0	0	0	0	0	0	0	591	88	0
BH	0	0	0	0	0	0	0	0	464	0
PKK	5	2	0	0	0	152	0	0	1	935

Tahmin edilen ve öngörülen sınıfların her biri ele alınarak ağırlıklı ortalama değerleri bulunmuştur. Tablo 9'da ağırlıklı ortalamalar hesaplanarak gösterilmiştir. Sonuçlara bakıldığında; gerçekleştirilen uygulama ile K-En Yakın Komşu algoritmasının veri seti üzerinde terör örgütünü tahmin etmedeki ağırlıklı doğruluk oranı % 98,2 olarak hesaplanmıştır. Bu değerlendirme sırasıyla Destek Vektör Makineleri, Rastgele Orman, Karar Ağacı, Naive Bayes ve Lojistik Regresyon algoritmaları için de gerçekleştirilmiştir. ***Çalışma süresi*** bir algoritmanın hız bakımından bir ölçüsünü ifade etmektedir. K-En Yakın Komşu algoritması veri kümesine uygulandığında çalışma süresi saniye cinsinden 0,125 olarak en iyi bulunmuştur. K-En Yakın Komşu algoritmasını çalışma süresi bakımından sırasıyla Destek Vektör Makineleri, Naive Bayes, Lojistik Regresyon, Karar Ağacı ve Rastgele Orman algoritmaları takip etmiştir. ***Kesinlik*** durumu tüm sınıflardan kaç adet doğru tahmin edildiğini belirten bir ölçüttür. Burada kaç adet örgüt isminin doğru tahmin edildiğini ifade etmektedir. Bu ölçütün yüksek olması başarımın daha iyi olduğunun göstergesidir. Kesinlik durumuna göre tablolara bakıldığında K-En Yakın Komşu algoritmasının veri kümesindeki çıktısına göre ağırlıklı ortalaması 0,983 olarak en iyi sonuç bulunduğu görülmüştür. K-En Yakın Komşu algoritmasını kesinlik ölçütü bakımından sırasıyla Destek Vektör Makineleri, Karar Ağacı, Rastgele Orman, Naive Bayes ve Lojistik Regresyon algoritmaları takip etmiştir. ***Duyarlılık*** ölçütü algoritmanın doğru değerlerinden yani istenilen değerlerin ne kadarı doğru tahmin edildiğini göstermektedir. Veri kümesindeki terör örgütlerinin ne kadarının doğru tahmin ettiğini belirtmektedir. Bu ölçütün çıktılarına göre K-En Yakın Komşu algoritmasının veri kümesindeki çıktısına göre ağırlık ortalaması 0,982 olarak en iyi sonuç olarak bulunmuştur. Bu değerlendirme sonucu diğer algoritmalara bakıldığında sırasıyla Destek Vektör Makineleri, Rastgele Orman, Karar Ağacı, Naive Bayes ve Lojistik Regresyon algoritmaları izlemiştir. ***F1 skoru***, duyarlılık ve kesinlik başarım

ölçütlerinin harmonik ortalamasından elde edilen bir ölçüttür. Burada F1 skoru, test edilmiş olan veri kümesinin duyarlılık ve kesinlik değerlerini ele alarak doğruluğunu ölçmektedir. F1 skor ölçütüne bakıldığı zaman K-En Yakın Komşu algoritması veri kümesindeki çıktısına göre ağırlık ortalaması 0,982 olarak bulunmuştur. Bu skor sonucuna göre diğer algoritmalara bakıldığında sırasıyla Destek Vektör Makineleri, Rastgele Orman, Karar Ağacı, Naive Bayes ve Lojistik Regresyon algoritmaları izlemiştir.

Tablo 9. Sınıflandırma algoritmaları başarım ölçütleri

Sınıflandırma Algoritması	Doğruluk Oranı(%)	Hata Oranı(%)	Çalışma Süresi(sn.)	F1 Skoru	Kesinlik	Duyarlılık
KA	85,9	14,1	5,801	0,840	0,933	0,859
KNN	98,2	1,8	0,125	0,982	0,983	0,982
RO	88,2	11,8	11,103	0,871	0,903	0,882
LR	48,8	51,2	4,151	0,384	0,423	0,488
DVM	96,4	3,6	0,889	0,965	0,971	0,964
NB	78,0	22,0	2,262	0,777	0,786	0,780

5. Sonuç ve Değerlendirme

Teknolojinin gelişmesi ve haberleşme ağlarının çoğalması ile birlikte günümüzde bilginin büyüklüğü ve değerli olması daha net bir şekilde görülmektedir. Günümüz teknolojileri ile birçok cihaz anlık veri üretebilmekte, verileri kayıt altına almakta ve bu kayıtlar çoğaldıkça veriler yeterince incelenemeden veri deposuna gönderilmektedir. Son yıllarda ise bu terör olaylarının belirgin bir şekilde artış gösterdiği görülmektedir. Bu alanda sosyal olarak gerçekleştirilen araştırmalar var olduğu gibi istatistiksel analiz, büyük veri analizi ve makine öğrenmesi gibi birçok farklı yöntemler ile araştırmalar vardır. Bu çalışmada dünyada yaşanmış olan terör olaylarının içerdiği Global Terrorism Database (GTD) veri kümesi ele alınmıştır. Bu veri kümesi üzerinde büyük veri kapsamında makine öğrenmesi yöntemleri ile analiz ve sınıflandırma işlemleri gerçekleştirilmiştir.

Bu çalışma kapsamında geliştirilen uygulamaya dâhil edilen algoritmalar pek çok makine öğrenmesi ve veri madenciliği uygulamalarında gerek bilimsel çalışmalar niteliğinde gerekse yapay zekâ alanında uygulama çözümlerinde kullanılmıştır. Bu çalışma kapsamında gerçekleştirilen modeller, farklı nitelik ve nicelik değerleri elde edilerek ortaya koyulmuştur. Kullanılan algoritmaların hızı, başarımı ve uygulanabilirliği veri kümesinin içeriğine bağlı olmakla beraber verinin bütünlüğü ve hedef nitelik çeşitliliğine bağlıdır. Bu çalışmada kullanılan ve indirgenen veri seti içerisinde dünyada en çok terör saldırısı gerçekleştiren ilk 10 terör örgütü ele alınarak sınıflandırma işlemleri gerçekleştirilmiştir. Kullanılan bazı algoritmaların başarımının düşük olmasının sebebi hedef niteliğinin fazla çeşit içermesi ve çoklu sınıflandırma kullanılmasıdır. Bu algoritmalar farklı veri kümelerinde ve hedef değişkenin daraltılması ile birlikte daha kapsamlı sonuçlar elde edildiği ve başarım ölçütlerinin arttığı görülmektedir. Fakat bu çalışma kapsamında örnek veri kümesinin büyük veri teknolojileri kapsamında ele alınarak makine öğrenmesi tekniklerinin uygulanması ve sonucunda bir uygulama geliştirilmesi diğer çalışmalardan farkını ortaya koymaktadır.

Bu çalışmada ve geliştirilen uygulamanın verdiği çıktılara göre doğruluk, duyarlılık, kesinlik ve F1 skoru başarım ölçütleri ele alınarak K-En Yakın Komşu algoritması daha başarılı bulunmuştur. Ayrıca çalışma süresi, duyarlılık ve kesinlik ölçütlerine göre de K-En Yakın Komşu algoritması yine öne çıkmıştır. Gerçek sistem üzerinde modelin veri kümesine uygulandığında (% 98,2-% 96,4 ortalama doğruluk Aralığında) % 1,8 hata tolerans edilerek Destek Vektör Makineleri tercih edilebilir. Büyük veri kapsamında değerlendirildiğinde makine öğrenmesi çözümlerinde Lojistik Regresyon algoritmasının doğruluk parametresi açısından uygun olmadığı ve Rastgele Orman algoritmasının çalışma süresi bakımından uygun olmadığı görülmektedir.

Bu çalışmada Apache Spark ile birlikte örnek bir veri kümesi üzerinde analiz ve örnek model oluşturma uygulamaları gerçekleştirilmiştir ve karşılaştırmaları doğruluk, F1 skoru, duyarlılık ve kesinlik gibi başarım ölçütlerine göre yapılmıştır. Ortaya koyulan karar destek sistemi ile % 98,2 ağırlıklı doğruluk oranı ile bir terör olayının hangi terör örgütü tarafından gerçekleştirildiğine dair tahmin üretilebilmektedir. Bu ise faillerinin bulunmasında çok önemli bir adım olan "eylemin hangi örgüt tarafından yapıldığına" dair bilgiyi tahmin edebilmeyi sağlamaktadır. Daha sonraki çalışmalarda büyük verilerden elde edilen anlamlı ve faydalı bilgilerin derin öğrenme ve yapay zekâ yöntemleri ile analiz edilmesi hedeflenmektedir.

Kaynaklar

[1] Scime A, Murray GR, Hunter LY. Testing terrorism theory with data mining. International Journal of Data Analysis Techniques and Strategies; 2010; 2(2), 122-139.
[2] Karabay B, Ulaş M. Comparison of Commonly Used Tools in Big Data Processing. 8th International Advanced Technologies Symposium; Eylül 2017; Elazığ; 3880-3897.
[3] Overview of the GTD, Erişim Tarihi: 10.10.2019, http://www.start.umd.edu/gtd/about/.
[4] Vajihala NR, Strang KD, Sun Z. Statistical modeling and visualizing open big data using a terrorism case study. 3rd International Conference on Future Internet of Things and Cloud; Ağustos 2015; 489-496
[5] LaFree G. Building a global terrorism database. DIANE Publishing; 2011
[6] LaFree G, Dugan L, Miller E. Putting terrorism in context: Lessons from the Global Terrorism Database. Routledge; 2014
[7] Khorshid M, Abou-El-Enien T, Soliman G. A comparison among support vector machine and other machine learning classification algorithms. IPASJ International Journal of Computer Science; 2015; 3(5); 26-35.
[8] Strang KD, Sun Z. Analyzing relationships in terrorism big data using Hadoop and statistics. Journal of Computer Information Systems; 2017; 57(1); 67- 75.
[9] Karabay B, Ulaş M. Example of Log Analysis with Apache Spark. 8th International Advanced Technologies Symposium; Eylül 2017; Elazığ; 4004-4011.
[10] Harter A, Hopper A. A distributed location system for the active office. IEEE network; 1994; 8(1), 62-70.
[11] Chu CT ve diğerleri. Map-reduce for machine learning on multicore. Advances in neural information processing systems; 2007; 281-288.
[12] Berkowitz BT, Simhadri S, Christofferson PA, Mein G.US. Patent No. 6,457,021. Washington, DC: U.S. Patent and Trademark Office; 2002.
[13] Shvachko K, Kuang H, Radia S, Chansler R. The hadoop distributed file system. IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST); Mayıs 2010; 1-10.
[14] Alsheikh MA, Niyato D, Lin S, Tan HP, Han Z. Mobile big data analytics using deep learning and apache spark. IEEE network; 2016; 30(3); 22-29.
[15] Zaharia M ve diğerleri. Apache spark: a unified engine for big data processing. Communications of the ACM; Kasım 2016; 59(11); 56-65.
[16] Altay O, Gurgenc T, Ulas M, Özel C. Prediction of wear loss quantities of ferro-alloy coating using different machine learning algorithms. Friction; 2019, 8(1); 107-114.
[17] Aggarwal CC. Data classification: algorithms and applications. CRC press; Taylor & Francis G.; 2014.
[18] Keerthi SS ve diğerleri. A fast iterative nearest point algorithm for support vector machine classifier design. IEEE transactions on neural networks; 2000; 11(1); 124-136.
[19] Spark, Apache. Extracting, transforming and selecting features, Spark 2.2.0 Documentation. Erişim Tarihi: 10.10.2019,https://spark.apache.org/docs/2.2.0/ml-features.html,
[20] Krüger F. Activity, Context, and Plan Recognition with Computational Causal Behaviour Models, Doktora, University of Rostock, Almanya, 2016.