YAŞANAN TERÖR OLAYLARINI İÇEREN BÜYÜK VERİNİN MAKİNE ÖĞRENMESİ TEKNİKLERİ İLE ANALİZİ

Barış KARABAY

Yüksek Lisans Tezi
Anabilim Dalı: Yazılım Mühendisliği
Danışman: Dr. Öğr. Üyesi Mustafa ULAŞ

AĞUSTOS-2019

T.C

FIRAT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YAŞANAN TERÖR OLAYLARINI İÇEREN BÜYÜK VERİNİN MAKİNE ÖĞRENMESİ TEKNİKLERİ İLE ANALİZİ

YÜKSEK LİSANS TEZİ

Barış KARABAY

152137102

Tezin Enstitüye Verildiği Tarih: 16.07.2019

Tezin Savunulduğu Tarih:02.08.2019

Tezin Danışmanı: Dr. Öğr. Üyesi Mustafa ULAŞ

Diğer Jüri Üyeleri: Doç. Dr. Murat KARABATAK

Doç. Dr. Muhammed Fatih TALU

ÖNSÖZ

Yüksek lisans tezim süresince yardımlarını, bilimsel katkı ve desteğini esirgemeyen değerli hocam Dr. Öğr. Üyesi Mustafa Ulaş'a, Yüksek lisans tezim süresince yanımda olan ve tez yazımımda bana yardımcı olan değerli çalışma arkadaşlarıma ve bugünlere gelmemde en büyük emeğe sahip olan aileme sonsuz teşekkürü borç bilirim.

Barış KARABAY

ELAZIĞ-2019

İÇİNDEKİLER

Sayfa No:

ÖZET

Teknoloji ve internet sayesinde veriler anlık olarak üretilmektedir. Bu verilerin çoğalması ile birlikte bilgi çıkarımı işlemleri önemli hale gelmekte ve veriden çıkarılacak bilgiler çok değerli olabilmektedir. Teknolojinin gelişmesi ve haberleşme ağlarının çoğalması ile birlikte günümüzde bilginin büyüklüğü ve değerli olması daha net bir şekilde görülmektedir. Günümüz teknolojisinde birçok cihaz anlık veri üretebilmekte, verileri kayıt altına almakta ve bu kayıtlar çoğaldıkça veriler yeterince incelenemeden veri deposuna gönderilmektedir. Araştırma ve geliştirme çalışmalarını bu yönde yürüterek karmaşık ve çeşitli verilerin altında zengin kaynaklarının yattığının farkına varan birçok araştırmacı "Büyük Veri" denilen bir yapıyı ortaya çıkartmıştır. Büyük veri işleme yöntemleri sayesinde anlamlı, faydalı ve önemli verilerin ortaya çıktığı görülmektedir. Büyük Veri, sadece büyük miktardaki veri anlamına gelmemekle birlikte geleneksel metotlar ile işlenemeyen veri anlamına da gelmektedir.

Bu çalışmada, uluslararası haber ajanslarından ve kaynaklardan elde edilen verilerin oluşturulduğu dünyada yaşanmış olan terör olaylarını içeren Global Terrorism Database (GTD) veri kümesi ele alınmıştır. Bu veri kümesi üzerinde büyük veri kapsamında makine öğrenmesi yöntemleri ile analiz ve sınıflandırma işlemleri gerçekleştirilmiştir. Yaşanan bir terör olayının hangi örgüt tarafından gerçekleştirildiğini tahmin eden bir uygulama geliştirilmiştir. Bu çerçevede, terör saldırısının saldırı tipi, kullanılan silah türü, saldırı yapılan ülke, bölge ve saldırıdaki hedef kitle bilgileri kullanılarak yaşanan terör olayının hangi örgüt tarafından gerçekleştirildiğini tahmin eden Apache Spark tabanlı açık kaynak büyük veri işleme aracı Python dili kullanılarak geliştirilmiştir. GTD veri kümesinden, en çok saldırı gerçekleştiren ilk 10 terör örgütü seçilerek altı farklı makine öğrenmesi algoritması uygulanmış olup, algoritmaların performansları arasında karşılaştırma ve değerlendirme yapılmıştır. Uygulanan algoritmalarda en yüksek doğruluk oranı K-En Yakın Komşu (KNN) algoritması ile % 98,2 olarak bulunmuştur. Lojistik Regresyon (LR) algoritmasının büyük veri kümesi için uygun olmadığı tespit edilmiştir.

Anahtar Kelimeler: Büyük veri, Apache spark, Terör olayları, Makine Öğrenme Algoritmaları

SUMMARY

Analysis of Big Data Including Terror Terms With Machine Learning Techniques

Thanks to technology and the internet, data is produced instantly. With the reproduction of this data, information extraction processes become important and the information to be extracted from the data can be very valuable. With the development of technology and the proliferation of communication networks, the size and value of information has come to the forefront today. In today's technology, many devices can produce instant data, record the data, and as these records increase, the data is considered insignificant and sent to the data store. By carrying out research and development efforts in this direction, many institutions and organizations realized that the rich resources underlie the complex and varied data have emerged as a structure called Big Data. It is seen that meaningful, useful and important data emerge thanks to large data processing methods. Big Data means not only large amounts of data, but data that cannot be processed by conventional methods.

In this study, the Global Terrorism Database (GTD) data set, which includes the terrorist incidents that took place in the world where data obtained from international news agencies and sources, was discussed. Machine learning methods and analysis and classification operations were performed on this data set within the scope of big data. An application has been developed which predicts the organization of a terrorist incident. Within this framework, the Apache Spark-based open source big data processing tool, which uses the information about the type of weapon, type of weapon, country, region and target group in the attack, was developed using Python language. Six different machine learning algorithms have been applied by selecting the first 10 terrorist organizations performing the most attacks from the GTD dataset and comparisons and evaluations have been made between the performances of the algorithms. The highest accuracy rate in the applied algorithms is 98,2 % with K-Nearest Neighbor (KNN) algorithm. The logistic regression (LR) algorithm was specified according to the situation appropriate for the big data set.

Keywords: Apache spark, Big data, Terrorist incidents, Machine Learning Algorithms

ŞEKİLLER LİSTESİ

Sayfa No:

TABLOLAR LİSTESİ

Sayfa No:

KISALTMALAR LİSTESİ

API : Application Programming Interface
BH : Boko Haram
BI : Business Intelligence
CART : Classfication and Regression Tree
DC : Decision Tree
DCT : Discrete Cosine Transform
DEAŞ : Devlet'ül Irak ve'ş Şam
DVM : Destek Vektör Makineleri
ETL : Extract-Tranform-Load
FARC : Revolutionary Armed Forces of Colombia
FMLN : Farabundo Marti National Liberation Front
GFS : Google File System
GTD : Global Terrorism Database
HDFS : Hadoop Distiributed File System
HTML : HyperText Markup Language
IRA : Irish Republican Army
JSON : Java Script Object Notation
JVM : Java Virtual Machine
KNN : K Nearest Neighborhood
LR : Logistic Regression
MLlib : Machine Learning Library
NB : Naive Bayes
NoSQL : Not Only Structured Query Language
NPA : New Peoples Army
PCA : Principal Component Analysis
PGIS : Pinkerton Global Intelligence Services
RAM : Random Access Memory
RDD : Resillient Distributem Datasets
Regex : Regular Expression
RF : Random Forest

SL	**:** Shining Path
SPSS	**:** Statistical Package for the Social Science
SQL	**:** Structured Query Language
SVM	**:** Support Vector Machine
TF-IDF	**:** Term Frequency - Inverse Document Frequency
WEKA	**:** Waikato Environment for Knowledge Analysis
XML	**:** Extensible Markup Language
YARN	**:** Yet Another Resource Negotiator
YSA	**:** Yapay Sinir Ağları

1. GİRİŞ

Yaşanan terör olayları, yüzyıllardır toplumların ve devletlerin sorunu olarak tanımlanmaktadır. Terör olayları yüzünden ülkelerin ekonomileri, insanların psikolojisi zarar görmektedir. Yaşanan terör olayları veya terör saldırıları binlerce insanın hayatının kaybetmesine ve kalıcı hasar bırakacak yaralanmalara sebebiyet vermiştir [1]. 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 incelemelerde vardır [2].Büyük veri, barındırdığı yapısal veya yapısal olmayan veriler ele alındığında, sürekli ve hızlı olarak büyüyen bir veri kümesini oluşturduğu görülmektedir. Bu ise içerisinde önemli bilgilerin de barındırıldığı büyük ve işlenmesi zor bir küme üretmektedir. Bu kadar büyük olan veri kümesinden anlamlı olan bilgiler çıkarmak önemli bir araştırma konusu haline gelmiştir. Büyük veri üzerinde bilgi keşfi ve anlamlı olan bilgiyi ortaya çıkarmak için klasik yöntemlerin kullanılması çok başarılı sonuçlar vermemektedir. Bu sorunun çözümü için büyük veri araçları, teknolojileri ve yöntemleri geliştirilmiştir. Bu çalışmada veri kümesi Global Terrorism Database (GTD) veri kümesi kullanılmıştır ve bu veri kümesi çevrimiçi olarak ulaşılabilir durumdadır [3]. GTD, 1970-2017 yılları arasındaki terör saldırıları hakkında bilgi içeren açık kaynak olarak sunulan bir veri kümesidir. Sonraki yıllarda içeriğinin güncellenmesi planlandığı ifade edilmektedir. Diğer veri tabanlarından farklı olarak hem yerel hem de uluslararası terör olayları hakkında bilgi içermektedir. Kullanılacak büyük veri kümesi, 180.000'den fazla terör saldırısının veya vakasının bilgisini içermektedir. Bir terör olayında; olayın gerçekleşme tarihi, olay yeri, hedefin niteliği ve hedef kitle, zarar gören canlı sayısı, sorumlu terörist grubu, kullanılan silah veya yöntemler gibi bilgileri içermektedir. GTD' de yer alan istatistiksel veriler, çeşitli medya kaynaklarından alınan raporlara ve bilgilere dayanmaktadır.

Bu çalışmanın amacı, bir eylem gerçekleştirildikten sonra faillerinin bulunmasında önemli bir kilometre taşı olan "eylemin hangi örgüt tarafından yapıldığına" dair olan bilginin ortaya çıkarılmasına yardımcı olmaktır. Terör eyleminin hangi terörist grubun yaptığını keşfetmek günümüzde önemli bir sorun olarak ortaya çıkmaktadır. Bu tez

kapsamında, bir saldırı ve saldırgan niteliklerine bağlı olarak gerçekleştirilen bir eylemin hangi örgüt tarafından gerçekleştirildiği konusunda karar destek sistemi ortaya koyulması hedeflemektedir. Çalışmada örnek olarak kullanılan büyük veri kümesine filtreleme işlemi uygulanarak dünyada en çok saldırı gerçekleştiren ilk 10 terör örgütü ele alınmıştır. Filtreleme işleminden geçirilen veri kümesine; K-En Yakın Komşu (KNN), Naive Bayes (NB), Destek Vektör Makineleri (SVM), Rastgele Orman (RF), Karar Ağaçları (DC) ve Lojistik Regresyon (LR) sınıflandırma algoritmaları uygulanmıştır. Bu uygulama kapsamında yaygın olarak kullanılan büyük veri işleme araçlarından olan Apache Spark ve Python programlama dili bir ara yüz geliştirilmiştir. Geliştirilen uygulamanın içeriğinde yukarıda bahsedilen algoritmalar kullanılarak sınıflandırma işlemleri gerçekleştirilmiştir. Gerçekleştirilen işlemlerde oluşturulan modellerin sonuçları karşılaştırılmıştır.

Veri kümesinde bulunan metin değerlerinin algoritma temelinde anlamlı hale gelebilmesi için sayısal formata dönüştürülmüştür. Kategorik değerlerin nümerik değerlere dönüşümü ile birlikte kullanılan algoritmalarda başarım oranının arttığı görülmüştür. Bu sınıflandırma işleminde terör olayının gerçekleştiği ülke ve bölge, terörist grubu, yapılan saldırıdaki hedef kitle ve kullanılan saldırı türü ele alınmıştır. Ayrıca bu veriler ve kıstaslar kullanılarak analiz işlemleri gerçekleştirilmiştir. Bu çalışmada uygulanan yöntemler büyük veri işleyebilen cihazlara uygun olarak tasarlanmış, veriler Apache Spark üzerinde bulunan kütüphaneler aracılığıyla işletilmiştir. Gerçekleştirilen çalışmada, 6 farklı makine öğrenmesi algoritmasının performansları ve çıktıları karşılaştırılmıştır.

1.1. Literatür Taraması

Yapılan kaynak taraması ile literatürde benzer çalışmaların ele alındığı yayınlar bulunmaktadır. Strang ve Sun, yaptıkları çalışmayla teröristin ideolojisi ve saldırı tipini ilişkisini ortaya koymak amacıyla bir uygulama geliştirmişlerdir. Google haberler servisini kullanarak elde ettikleri veriler üzerinde test etmişlerdir. Google News hizmeti üzerinde, büyük veri çatısı olan Hadoop ile geniş ölçekli, karmaşık yapıda olan verileri hızlı bir şekilde elde etmişlerdir. Ayrıca başka bir araştırmacılardan elde ettikleri verileri de ele alarak bütünleştirme işlemini gerçekleştirmişlerdir. Yaptıkları çalışmada, terörist grupların fikirleri veya ideolojileri ile saldırı tipi arasında anlamlı bir ilişki olduklarını

ortaya koymaktadırlar. Ortaya koydukları hipotezlerini SPSS yazılımı kullanarak analiz işlemlerini gerçekleştirmişlerdir. Desteklenen hipotez ile terör ideolojisi ile saldırı tipi arasındaki gizli bağlantıları ilişkili bir şekilde göstermek için benzetim modeli geliştirmişlerdir [2].

Scott, Fogg, Dugan ve LaFree, 2006 yılında yaptıkları çalışma mevut terör saldırısı verilerinin analiz edilmesindeki zorluğunun eldeki verilerin düşük kalitede olduğundan bahsetmişlerdir. Yapılan çalışmada, 1970-1997 yılları arasındaki terör verilerinin bir bütün halinde bir araya getirerek Global Terrorism Data (GTD) projesini oluşturmuşlardır. İlk olarak 67.165 veri toplanmıştır. Bu projedeki veriler PGIS (Küresel İstihbarat Servisi) tarafından doğrulanarak toplanmıştır [4].

Miller, Dugan ve LaFree, 1970 – 2015 yılları arasındaki 125.000 terör saldırısı Global Terrorism Database (GTD) veri setini ele alarak ders niteliğinde bir yayın oluşturmuşlardır [5]. Bu veri setini analiz eden araştırmacılar, çok az sayıdaki terör saldırılarının bile terörizme karşı bir tutum ve politikalar üzerinde nasıl büyük bir etkisi olduğunu göstermektedir. Yayının içeriğinde araştırmacılar, veri setinde olan terör saldırıların çoğunun tahmin edilemeyen saldırılar olduğundan "Black Swan" yani siyah kuğu olarak adlandırmaktadır ve önümüzdeki yıllarda insan ilişkileri üzerinde etki oluşturmaktadır. Yapılan çalışmada en büyük siyah kuğu olayı olarak 11 Eylül saldırılarını göstermektedir. Ayrıca 1970 yılından beri yaşanan saldırıları ve bağlantılarını tartışmaktadır [5].

Strang, Sun ve Vajihala diğer çalışmasında, terör olaylarının uygunluk analizlerini uygulamalı olarak gerçekleştirdikleri bir çalışma yapmışlardır. Bu çalışmalarında, SPSS kullanılarak terör saldırı tiplerinin ve ülkelerinin ilişkilerini ortaya çıkararak istatistiksel modellenmesini ve görselleştirilmesi konusunu ele almışlardır [6].

Khorshid ve arkadaşları, makine öğrenmesi yöntemlerinden olan Destek Vektör Makineleri (SVM) yöntemi ile diğer yöntemleri başarılı tahminsel model üretme yönünden ayrıntılı bir şekilde karşılaştırdıkları çalışma yapmışlardır. Araştırmacıların yapmış olduğu çalışmada, Orta Doğu ve Kuzey Afrika'da 2004-2008 yılları arasında meydana gelen terör saldırılarını gerçekleştiren terörist grupların tahminini ele almışlardır. Örnek veri olarak, Global Terrorism Database (GTD) veri kümesi

kullanılmıştır. Çalışmada WEKA yazılımı üzerinde belirtilen algoritmalar uygulanmıştır [7].

WEKA yazılımı sadece Java yüklü olan bilgisayarlarda çalışmaktadır. Ayrıca, masaüstü yazılımı olması ve gömülü bir sistem (Java, Python, C# vb.) olmaması sebebi ile gelecekte web ve mobil sistemler üzerinde kullanılamaması bir dezavantaj oluşturmaktadır. Ayrıca sadece 2004-2008 yıllarındaki kayıtların ele alınması ve belli bir bölgeye hedeflenmesi de eksik bir yönünü oluşturmaktadır.

Vapnik, birçok özelliği barındıran, deneylerinde umut verici performanstan dolayı Destek Vektör Makineleri (SVM) algoritmasını geliştirmiştir. Destek Vektör Makineleri(SVM), ilk olarak sınıflandırma problemlerinde yaşanan sorunların giderilmesi için geliştirildiğini, son zamanlarda regresyon problemlerinin çözümü için de genişletildiğini belirtmiştir [8].

Mahsey, büyük veri kelimesini 1999 yılında yayımlanan sunumunda ilk kez kullanmıştır. Bu sunumda veri depolamanın sürekli olarak büyüdüğünü, önlem alınmazsa iletişim altyapılarında sorunlar olacağından bahsetmiştir. Ayrıca sistem yöneticilerinin büyük sunucular üzerinde 100'lerce terabayta varan verilere ulaştığında yedekleme sorunlarıyla karşılaşacaklarını belirtmiştir. Büyük verinin, hem araştırma hem de ticari açıdan büyük fırsatlar doğuracağını söylemiştir [9].

Yapay zekâ araştırmacısı ve matematikçi Solomonoff, makine öğrenmesi konusunu ve kavramını ilk defa 1957 yılındaki yayınında bahsetmiştir. Eğitim kümelerinin öneminden bahsedip, önceki sorunların çözümünün ele alınarak ile var olan yeni veri kümesindeki sorunlara uygulanarak çözüm elde edilebileceğinden bahsetmektedir[10].

2. BÜYÜK VERİ

Büyük Veri, günümüz teknoloji ile orantılı olarak kontrol edilemeyecek şekilde sürekli artarak büyümeye devam etmektedir. Klasik ilişkisel veri tabanlarında depolanamayan ve çok çeşitli niteliklere sahip olduğu içinde yapısal bir hale getirilmesi, günümüz veri analiz yöntemleri ile işlenmesi zor olan verilerdir. Bu verilere, sosyal medya paylaşımları, ses ve görüntü aktarımları, ağ üzerinde anlık kayıtlar, sensorlardan gelen veriler, web sunucu ve istemci günlükleri, istatistikî bilgiler, haberler, log dosyaları, arşiv sistemleri ve diğer kanallardan elde edilen veriler örnek olarak verilebilir. Büyük verinin sürekli artmasındaki en büyük temel nedenler; sosyal medya kullanımının artması, kişisel bilgisayarların, akıllı telefonların, tabletlerin, hatta akıllı saatlerin artması, anlık veri depolayan sensorların artması, internet kullanımının yaygınlaşması ve bulut teknolojisinin gelişmesi olarak gösterilebilir. Bu teknolojilerin çoğalması dünyada saniyede milyarlarca verinin oluşmasına yol açmaktadır. Büyük veri işleme süreci, bu verilerin analiz edilerek fayda sağlanabilecek anlamlı hale getirilmesidir. Bu kadar büyük verileri analiz etmek için yenilikçi araçların kullanılması mecburidir. Büyük veri analizi, büyük veri kümeleri üzerinde analitik teknikler kullanarak işlemler yapılması sürecidir[11]. Büyük veriler yapılandırılmış, yapılandırılmamış ve yarı yapılandırılmış veriler olarak üçe ayrılmaktadır.

Yapılandırılmış veriler: Sabit bir şekilde tutulan verilerdir. İlişkisel veri tabanı sistemlerinde bulunan satır/sütun verileri buna örnektir.

Yapılandırılmamış veriler: Sabit bir şekilde tutulmayıp çeşitlilik gösteren verilerdir. Örnek olarak e-posta içeriği, resim, video, ses, sosyal medya etkileşimleri, makaleler ve kitaplar gibi.

Yarı-Yapılandırılmış veriler: Sabit alanlarda tutulmayıp fakat verileri kategorize ayırmak için etiket gibi unsur kullanılmaktadır [12]. Örnek olarak XML, HTML ve JSON verilebilir.

Büyük verilerin analizini yapmak için öncelikle büyük veri olgusunun bileşenleri iyi bilinmelidir. Büyük veri, beş temel bileşenden oluşmaktadır. Bunlar: Çeşitlilik (variety), velocity (hız), volume (veri büyüklüğü), doğrulama (verification) , değer (value) ve verinin dinamikliği (variability) olarak belirtilmektedir. Bu neden literatür de 5V kuralı

olarak bilinmektedir. Ayrıca son zamanlarda altıncı kural olarak değişkenlik (variability) dâhil edilmektedir [13]. Dağıtık Sistem, Map Reduce, Google dosya sistemi (GFS), Big Table, İş zekâsı, In-Memory Database (Bellek içi veri tabanı teknolojisi), NoSQL, Cassandra, MongoDB, DynamoDB, Hadoop, YARN (Yet Another Resource Negotiator), MLlib, Apache Hive, Apache Pig, Storm, Bulut Bilişim, Elastic Search, Redis, Apache Oozie, Apache Kafka, Apache Mahout, Sqoop, Apache Flume, Hbase, Apache Spark, Python ve PySpark, DataFrame ve Esnek dağıtık veri kümeleri (Resilient Distributed Dataset) gibi birçok büyük veri analiz altyapısı veya aracı vardır.

2.1. Büyük Verinin Faydası ve Kullanım Alanları

Büyük veri olgusundan verim alabilmek için geleneksel tekniklerden ziyade yeni teknolojiler uygulayarak analiz yapılmalıdır. Büyük veri iyi analiz edildiğinde en uygun çözümlerin bulunmasında, matematiksel modelleme yapılmasında, karar destek sistemlerinin mekanizmasının kurulmasında etkili bir veri bilimidir. Büyük veri, doğru metot ve teknikler kullanıldığında kurumların ve hükümetlerin stratejik yönden doğru karar vermelerine, riskleri analizini etkin bir şekilde yapabilmesine ve teknolojide yenilik gibi çalışmalar yapmasına imkân verebilmektedir.

Büyük veri analizi için birçok yöntem, araç ve teknik geliştirilmiştir. Bu yöntemler kullanılarak işletmeler kar marjlarını arttırabilmekte, müşterilerin davranışlarına göre tahmin yapıp en fazla hangi ürünü aldığını, hangi renklerden hoşlanıldığını, hangi saat aralığında alışveriş yaptıklarını belirleyebilmektedir. Büyük veri olgusundan verim alabilmek için geleneksel tekniklerden ziyade yeni teknolojiler ile derinlemesine analiz yapılmalıdır. Eldeki veriler yeterince iyi tanımlanıp analiz edildiğinde işletmelere ve kuruluşlara ciddi oranda fayda sağlamaktadır. Güçlü tahmin yeteneği elde etme, maliyet ve zaman tasarrufu en önemli faydalarıdır.

Dünya üzerinde büyük verinin; perakende satış, e-ticaret, bankacılık, sosyal medya, işletme, reklamcılık, hava durumları, borsa, sağlık, telekomünikasyon, istihbarat faaliyetleri, eğitim, e-devlet uygulamaları, taşımacılık ve kargo işlemleri ile enerji sektörü gibi birçok uygulama alanı bulunmaktadır. Uygulama alanları şu şekilde ilişkilendirilebilir:

Telekomünikasyon: Abone veri toplayarak ve tüketici verileri ile gerçek zamanlı analiz yaparak hızlı çözümler üretilebilmektedir. Ayrıca abonelerinin mobil verilerini kayıt altına almaktadırlar. Bu veriler üzerinde analiz işlemleri yapılmakta ve müşterilerinin kullandığı verilere göre abonelerine özel indirimli paketler ile faturalı veya faturasız uygun tarifeler teklif edebilmektedirler.

Bankacılık: Bankalar, müşterilerin bilgilerini topladığı veriler ile kullanıcı davranışını modellemektedirler. Bu model ile müşterilerini özel hissettirme yeteneğine sahip olunmuştur. Menkul ve kart dolandırıcılığı için erken uyarı tespiti, kredi risk tespiti yapılabilmektedir.

Eğitim: Eğitim sistemi verilerinin toplanması ile birlikte büyük veri sayesinde öğrenciler üzerinde olumlu gelişmeler oluşturabilmektedir. Eğitim verilerinin toplanmasıyla elde edilen büyük veri analiz edilerek sistemin gerisinde olan öğrenciler bulunabilmektedir. Bu sayede öğrenciler için doğru bir yöntem seçilerek öğrencilerin derslerinde ilerlemelerinde büyük katkı sağlanabilmektedir.

Perakende: Müşteri kartları, POS cihazları, RFID etiketleri ve personel giriş kartlarından elde edilen kullanılmayan veriler vardır. Bu veriler analiz edilerek zamanında stok analizi, personel optimizasyonu, müşteri davranış analizi, ürünü doğru yerleştirme, ürün çeşitliliği ve fiyat optimizasyonu yapılabilmektedir.

Sağlık: Sağlık sisteminde elde edilen kayıtlar kullanılarak kan stoku takibi, hastalığın erken teşhisi ve durumunun takip edilmesi, DNA analizi gibi bireysel veriler kullanılabilmektedir.

Kamu: Devlet daireleri ile farklı kuruluşlar arasında entegrasyon ve birlikte çalışabilme imkanı. Hükümetler; enerji hizmet alanları, dolandırıcılık tespiti, araştırma ve geliştirme projeleri, çevre koruması gibi konular üzerinde analiz yapılarak vatandaşlar için uygun hizmeti sunabilmektedir.

Enerji: Enerji firmalar abonelerinin tüketim verilerini depolayarak büyük veri elde etmektedirler. Günümüz teknolojisinde enerji firmaları bu verileri analiz ederek müşterilerine bireysel olarak farklı fiyat tarifeleri önerebilmektedirler.

Sosyal Medya: Sayısı milyarları bulan insanın kullandığı sosyal medya platformlarında anlık olarak veriler üretilmektedir. Örnek olarak; lokasyon bildirimi, fotoğraf ve video paylaşımı, kişi durumları, kişisel bilgiler, yorum ve beğeniler gibi yapılandırılmış ve yapılandırılmamış veriler günlük olarak sosyal medyada yayımlanmaktadır. Bu verileri kullanarak sosyal medya analizi yapılmakta ve kullanıcı davranışları belirlenebilmektedir.

Taşımacılık: Taşımacılık sektöründe daha çok kargo şirketlerinin elinde bulunan gönderici ve alıcı bilgileri kullanılarak analiz ediliyor ve bu analiz verileri üzerinden avantaj sağlayacak planlamalar yapılabilmektedir.

Ulaşım: Otobüs, havayolu, tren ve gemi firmaları yolcuların bilgilerini toplayarak müşteri ile ilişkilerini geliştirebilmektedir. Havayolu şirketleri müşterisi ile olan diyaloğu geliştirmek ve onları kendilerini özel hissettirmek için büyük veriden yararlanmaktadır. Firmalar, Yolcuların sosyal durumunu analiz edip bir fırsat yaratmanın yollarını bulmuştur. Kişisel bilgiler, seyahat planları, tercihlerine, otelde konaklama ve araç kiralama alışkanlıklarına kadar birçok veri bulunmaktadır. Bu veriler kullanarak yolcuların kendilerini özel hissetmelerine ve yolculara yardımcı/avantajlı durumda olmalarını sağlamaktadır.

Büyük veri yöntem ve teknolojiler Verilerin çoğalması, yapısız hale gelmesi ve çok çeşitli olması büyük verinin işlenebilmesi nedeniyle büyük şirketler tarafından teknolojiler geliştirilmiştir. Büyük verilerin elde edilmesi, toplanması, işlenmesi ve analiz edilmesi için birçok yapı ve yöntem geliştirilmiştir.

2.2. Büyük Veri Teknolojileri

Büyük veri kümelerinden bilginin ve anlamlı verinin elde edilmesini sağlayan birçok metot ve sistem, yazılım geliştirilmektedir. Anlık olarak dünya üzerinde veri akışı bulunmakta ve sürekliliği olan bir durum oluşturmaktadır. Sürekliliği olan veri kümesinde verilerin analiz edilmesi, işlenmesi ve anlamlı olarak bilginin elde edilmesi maliyet, performans ve teknolojik alt yapı zorunlu hale gelmektedir. Bu bölümde büyük veri kapsamında yararlanılan ve kullanılan teknolojiler hakkında bilgi verilmiştir.

Dağıtık Sistem: Onlarca bilgisayar sisteminin bir ağ üzerinde birbiri ile iletişim halinde olması ve bunu birbirinden bağımsız hareket ederek yapmasıdır. Bir iş için birden fazla bilgisayarın çalışması sistemine denilmektedir.

Map Reduce: Büyük veri kümelerini paralel olarak işlemek ve analiz etmek için kullanılan bir programlama modelidir [14]. Map ve Reduce, veri analizi ve işlemesi için oluşturulmuş 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.

Google dosya sistemi (GFS): Google, geleneksel dosya sistemlerinin artık gelişen teknolojiyle birlikte artan veri hacminin işlenmesini sağlayacak bir yapının eksikliğini hissedilmesi ile birlikte GFS adında bir dosya sistemi geliştirmiştir [15]. Google, bünyesinde bulundurduğu görseller, video, haritalar, haber ve web içerikleri gibi veri hizmetlerini GFS sayesinde depolamakta ve işlemesini kolaylaştırmaktadır [15].

Big Table: Google; Map Reduce, GFS ve Big Table yöntemlerini maliyeti düşürmek için birden fazla bilgisayarın çalışması ile birlikte kümeler oluşturmaktadır [15]. Big Table, üzerinde barındırdığı çok çeşitli ve büyük verileri çözümlemek, yönetimini sağlamak için tasarlanmış dağıtık sistemdir. Google'da bulunan görüntü, web içerikleri, haber ve video gibi verileri Big Table çözümü üzerinde bulunmaktadır. Elde edilen ve anlık olarak üretilen veriler tutulmaktadır. Big Table, Google üzerinde bulunan tüm içerikleri ve büyük veri deposunu ölçeklenebilen, esnek yapıda olan ve performans çözümlü kontrol etmektedir [16].

İş Zekâsı (BI): Yarı yapılandırılmış veya yapılandırılmamış olan ham verinin iş amacı kapsamında anlamlı hale getirilmesi ve bilginin kullanımını genişleten teknolojiler bütünüdür. Kısacası büyük miktardaki veriyi elde edip anlamlı hale getiren iş araçlarıdır [17].

In-Memory Database (Bellek içi veri tabanı): Klasik veri tabanlarından farklı olarak disk tabanında veri depolamamaktadır. Veri ambarlarını RAM(Random Access Memory) üzerinde tutarak hızlı işlem yapabilmesini sağlamaktadır. Bellek maliyetinin azaldığı düşünülürse performanslı bir teknolojidir [18].

NoSQL: İlişkisel veri tabanlarına alternatif olarak ortaya çıkan gittikçe büyüyen verileri depolayabilmek ve yüksek trafiğe sahip sistemlerin ihtiyaçlarına cevap verebilmek amacıyla ortaya çıkmış yapıdır [19]. NoSQL, sadece SQL değil anlamına gelmektedir.

Cassandra: Apache tarafından NoSQL veri tabanı yapısını benimsemiş açık kaynak kodlu bir veri tabanıdır. CERN, GoDaddy, GitHub, Instagram, Netflix, Reddit gibi kurumlar kullanmaktadır [20].

MongoDB: NoSQL veri tabanı mimarini uygulamış, ilişkisel olmayan, açık kaynak kodlu, ücretsiz, doküman tabanlı ve yüksek hacimli içerikler için kullanılan veri tabanı yönetim aracıdır [21].

DynamoDB: Amazon tarafından geliştirilmiş, dağıtık veri tabanı ve tamamen NoSQL mimarisini benimsemiş, hızlı, yüksek performans sağlayan ve veri miktarının limiti olmayan veri tabanı hizmetidir [22]. Son kullanıcıya hizmet vermeden önce Amazon içinde uzun süre kullanılmış bir yapıdır.

Hadoop: Dağıtık sistem üzerinde bulunan belirli türdeki büyük veri kümelerini işlemek için geliştirilmiş yazılım çerçevesidir. Google Map Reduce ve Google dosya sisteminden esinlenerek geliştirilmiştir. Kümelenmiş bilgisayarlar arasında büyük çaplı verileri işlemek amacıyla geliştirilmiştir. Hadoop Distributed File System (HDFS) olarak adlandırılmış bir dağıtık dosya sistemi ile Hadoop Map Reduce özelliklerini bir araya getiren açık kaynaklı bir ekosistemdir. Ayrıca HDFS ve Map Reduce bileşenlerinden oluşan bir yazılım geliştirme yapısıdır [23].

YARN (Yet Another Resource Negotiator): Büyük veri, Map Reduce gibi büyük ölçekli uygulamalar ve dağıtık sistemler için kaynak yönetimini sağlamak (İşlemci, bellek gibi) için tasarlanmıştır. Apache Software Foundation tarafından açık kaynak olarak dağıtılmıştır. Arka planda kaynak yönetimini ve planlanmasını sağlar [24].

MLlib: Apache Spark'ın makine öğrenme ile ilgili algoritmalarının bir arada bulunduğu kütüphanedir [25].

Apache Hive: Hadoop üzerinde sorgulama ve veri analizi yapmak için kullanılan bir veri ambarı paketidir. Apache tarafında geliştirilmektedir. Yapılandırılmış veriyi

yönetmek, sorgulamak için HiveQL gibi SQL'e benzer bir yapı kullanır. Hive'ın en önemli özelliği Hive ile yazılan kodları arka planda Java Map Reduce kodlarına çevirir bu sebeple Java öğrenmeye ihtiyaç bulunmamaktadır [26].

Apache Pig: Hive gibi veri işlemek için geliştirilmiş bir projedir. Map Reduce ile yapılan işlemleri Pig ile daha kolay ve hızlı bir şekilde yapılmaktadır. Yahoo firması tarafından geliştirilen açık kaynak kodlu projedir [27].

Storm: Ücretsiz olarak dağıtılan gerçek zamanlı hesaplama yapabilen bir sistemdir. Storm, Hadoop ‘un toplu hesaplama işlemlerini gerçek zamanlı olarak yapmaktadır. Sınırsız veri akışlarını güvenilir bir şekilde işlemektedir [28].

Bulut Bilişim: Bulut bilişim (Cloud Computing), web servisleri aracılığıyla internet üzerinde veri depolayabilen ve aynı zamanda ortak bilgi paylaşımına izin veren bir hizmettir [29].

Elastic Search: Büyük veri ambarları içerisinde doküman endeksleme, tam metin arama ve veri analizi yapmak için güçlü ve hızlı bir araçtır [30].

Redis: Redis, açık kaynak kodlu, bellekte veri yapısı şeklinde tutulan, önbellek ve aracı olarak kullanılan veri tabanıdır. Dizeler, liste, kümeler, sıralanmış kümeler, resimler, coğrafi konumlar gibi veri yapılarını destekler. Redis, anahtar/değer tasarlanmış ve gittikçe popüler olan bir NoSQL veri tabanıdır [31].

Apache Oozie: Açık kaynak kodlu ve ücretsiz olarak geliştirilen Hadoop sisteminde bulunan tüm işleri periyodik zaman dilimlerinde yöneten bir iş akışı zamanlayıcı aracıdır [32].

Apache Kafka: Büyük veri kümelerini hızlı, hatasız ve güvenli bir şekilde çeşitli kanallardan toplayıp, diğer dağıtık sistemlere göndermek için gerekli olan mesajlaşma sistemidir. Akışı sağlanan veriler bir kuyruk yapısı şeklinde atılarak Hadoop, Pig, Hive, Spark gibi araçlara transfer edilmesini sağlar. Toplanan verilerin sistemler veya uygulamalar arasında güvenilir bir şekilde gerçek zamanlı veri akışını sağlayan araçtır [33].

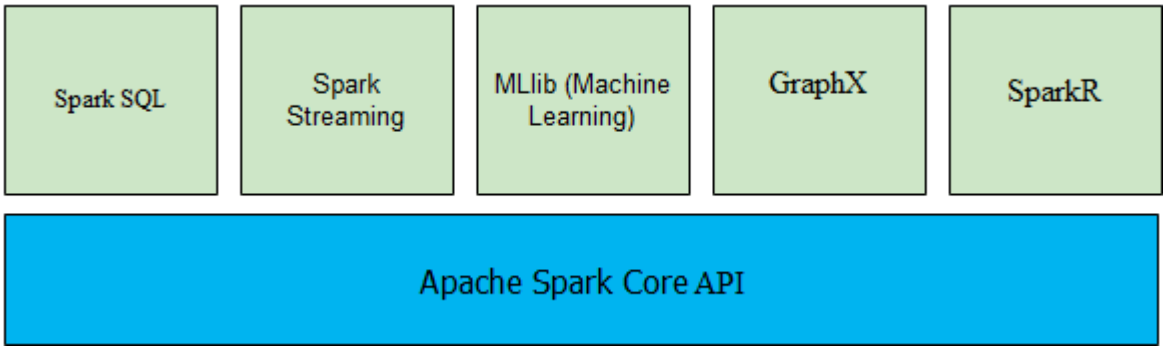
Apache Mahout: Uyumlu ve yüksek performanslı makine öğrenmesi uygulamaları oluşturmak için Apache tarafından geliştirilen açık kaynak kodlu kütüphanedir [34].

Sqoop: İlişkisel veri tabanları ile Hadoop (HDFS) arasında veri transferi yapılmasını sağlayan açık kaynak kodlu bir yapıdır [35].

Apache Flume: Flume, büyük miktardaki günlük verilerini verimli bir şekilde toplamak, bir arada tutmak ve güvenilir bir şekilde taşımak için dağıtık bir sistem olarak tasarlanmıştır [36]. Kısacası, log (günlük verisi) toplama aracı olarak tanımlanmaktadır.

Hbase: Java ile programlanmış, Hadoop dosya sisteminde çalışan, Google Big Table'dan esinlenerek geliştirilmiş, ilişkisel olmayan ve NoSQL mimarisini benimsemiş bir veri tabanıdır [37].

Apache Spark: Disk üzerinde Hadoop Map Reduce işleminden 10 kat daha hızlı veya bellekte 100 kat daha hızlı veri işlemesi yapabilen bir platformdur. Apache Spark büyük ölçekli verileri işleyebilen hızlı bir motordur. Anlık olarak akan verileri kolaylıkla işleyebilme özelliği bulunmakta ayrıca Hadoop Map Reduce işleminde yazılan kodlardan daha az kod yazarak daha hızlı ve kolay veri işlemesi yapılabilmektedir. Map Reduce yapısının bir alternatifi olarak düşünülebilir. Spark işlemlerinde bellek içi teknolojisi kullanılmaktadır. Tek başına, Hadoop veya bulut üzerinde çalışabilmektedir. HDFS, Cassandra, HBase ve birçok veri kaynaklarına erişim sağlayabilmektedir. Spark, makine öğrenimi için MLlib, akan veriler için Spark Streaming, SQL, GraphX gibi güçlü kütüphaneleri bünyesinde barındırır. Şekil 2.1'de Apache Spark sistemi ve bünyesinde barındırdığı kütüphaneler gösterilmiştir [38].



Şekil 2.1. Apache spark ekosistemi

Python ve PySpark: Python açık kaynak kodlu, ücretsiz, nesne tabanlı, modüler yapıda olan, kullanıcı ara yüzü desteğine sahip, yüksek seviyeli ayrıca bir derleyiciye gerek duymadan çalışan bir programlama dilidir [39]. Windows, Linux ve Mac OS işletim

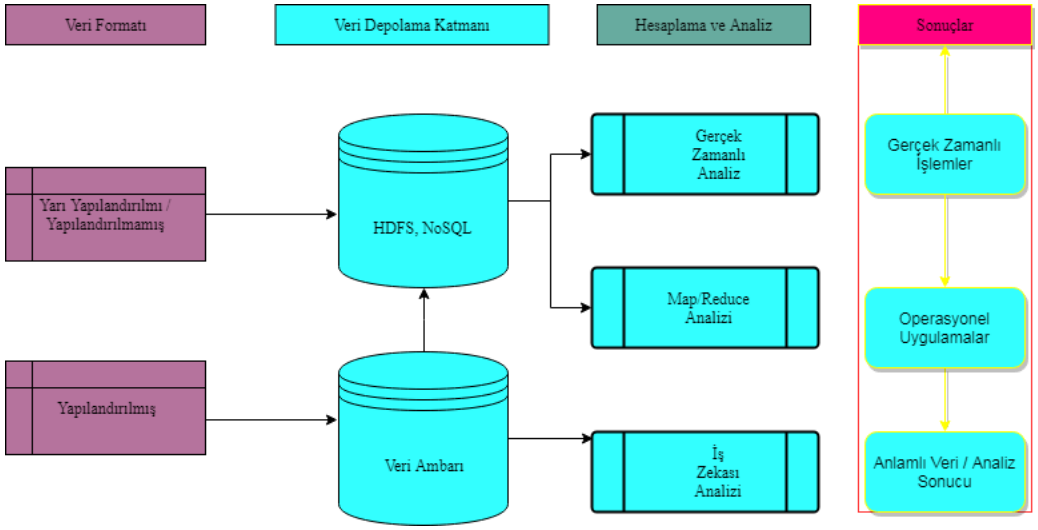
sistemlerinde çalışabilmektedir. Python, web uygulamalarında, sistem programlama, kullanıcı ara yüzü, ağ uygulamaları, modelleme ve bilimsel hesaplama alanlarında yaygın olarak kullanılmaktadır. Python üzerinde Spark uygulamaları geliştirebilmek için gerekli olan kullanıcı ara yüzü uygulamasıdır (API). PySpark API, birçok Spark fonksiyonlarından oluşmaktadır. Örneğin sorgulama, veri kümesi oluşturma ve veri kümesi kaynağından okuma işlemleri bu kütüphane aracılığıyla gerçekleşmektedir. Python üzerinde Spark işlevlerini çalıştırmak için ayrı bir kütüphane olarak kurulmaktadır.

DataFrame: Spark için tasarlanmış ilişkisel veri tabanlarındaki tablo yapısına benzer tablolardır. Satır, sütunlardan ve şemalardan oluşmaktadır. Diğer veri tabanı tablolarından farkı dağıtık sistemler üzerinde çalışabilme yeteneğine sahip olmasıdır [40]. Büyük veri kümelerinin daha kolay bir şekilde işlenmesini sağlamak için geliştirilmiştir.

Esnek dağıtık veri kümeleri (Resilient Distributed Dataset):RDD, bir veri deposunu işlenmesini sağlamaktadır. İçerisinde birleştirme, işleme, sayma ve sıralama gibi metotları barındırarak dosya depolarını daha kolay işlenmesini gerçekleştirmektedir [29]. RDD, ayrı ayrı düğümlerde hesaplama işlemi yapılarak birçok parçaya bölünmektedir. Dönüşüm ve işlev fonksiyonları olarak iki durumda kullanılmaktadır. Dönüşüm işlemlerine örnek olarak veri kümesinde filtreme işlemi yapılması gösterilebilir. İşlev fonksiyonlarına bir veri kümesindeki son elemanı getiren metot örnek olarak verilebilir [41].

2.3. Büyük Veri Mimarisi

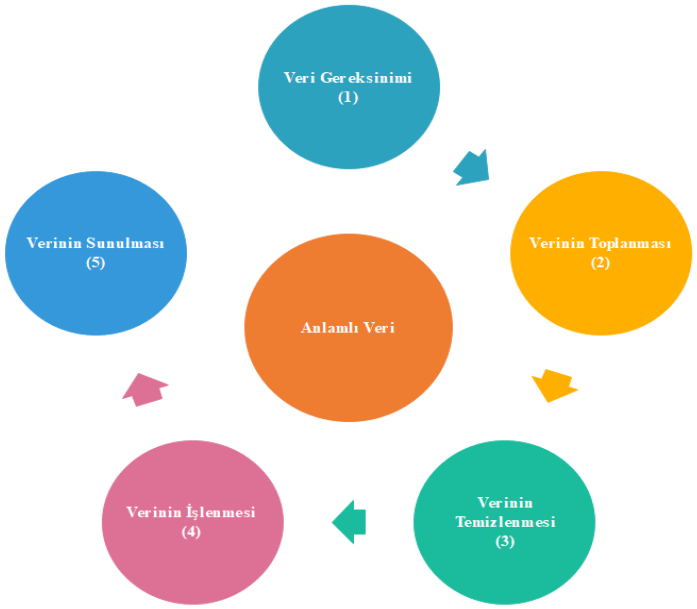
Büyük veri mimarisi içerisinde veri depolama, işleme, analiz etme ve yönetme gibi bileşenleri barındırır. Şekil 2.2'de görülen büyük veri mimarisindeki katmanlara bakılarak verilerin analizleri ve yönlendirmeleri yapılabilmektedir. Bu katmanlı mimariye göre eldeki verilerin karmaşıklığına bakılarak hangi süreçlerden geçirildiği belirlenebilmektedir. Tüm süreçler ayrı olarak da işlemden geçirilebilmektedir.



Şekil 2.2. Büyük veri katmanlı mimarisi

2.4. Büyük Veri Analiz Süreci

Büyük veri kümesinin analizini yapmak için belirli adımların yapılması gerekmektedir. Özellikle yüksek boyutlara varmış büyük veri düzeyinde, karmaşık ve yapılandırılmamış dosyalar için analizi süreçleri büyük önem arz etmektedir. Büyük veri kümesi analizi süreçleri aşağıdaki aşamalardan oluşmaktadır. Aşamalar Şekil 2.3'te görülmektedir.



Şekil 2.3. Büyük veri analizi adımları

2.4.1. Büyük veriye gereksinim duyulması

Üzerinde çalışmak istenen konunun belirlenip, veri kümesinin hangi kaynaklar üzerinden toplanmasının hedeflendiği basamaktır. Bu başlangıçta hangi veri türünün ihtiyacı var ise tespit edilmesi gerekir.

2.4.2. Büyük Verinin Elde Edilmesi ve Toplanması

İlk olarak veri kümesinin çeşitli veri kaynaklarından elde edilerek toplanmaktadır. Bu veriler uluslararası kabul görmüş veri tabanı sağlayıcıları, literatür, geçmişe yönelik haber kaynakları ve kamu veri sağlayıcılarından toplanmaktadır. Toplanan verilerdeki ve performans konusunda etkili olabileceğinden bilgiler hassas ve tutarlı olmalıdır. Toplanan veri dosyaları yanıltıcı olmamalı ve gerçek haberlerden veya kaynaklar tarafından tutulan verilerden oluşmalıdır.

2.4.3. Verinin Temizlenmesi

Veriler büyüdükçe kontrol edilmesi zor olabilmektedir. Ayrıca büyük veri kümeleri büyük ve yapılandırılmamış bir şekilde bulunmaktadır. Bu sebeple örnek veri kümesinin içinde bulunan diğer alanlarla bağlantı kurulabilmesi performans ve gerçekçi olması açısından oldukça önemlidir. Veri kümelerindeki veri alanlarının birbirleri ile bağlantı kurması ile anlamlı bilgilerin ortaya çıkması kolaylaşmaktadır. Toplanan verinin hangi amaçla kullanıldığının net olması, dosyadaki veri alanlarının altında bulunan tiplerin gerçek ve tutarlı olması (örneğin resim formatı yerine sayı formatı girilmesi), alanlar arasındaki ilişkilerin otomatik olarak sağlanması işlemleri veri analizinde performansı etkileyen işlemlerdir.

2.4.4. Büyük Verinin İşlenmesi

Büyük veri kümesinin analiz edilmesi işlemidir. Veri analizi işlemleri sınıflandırma metotları, örüntü tanıma, makine öğrenmesi, yapay sinir ağları gibi çeşitli yöntemler kullanılarak gerçekleştirilmektedir. Veriler belirli yöntem ve tekniklerden süzüldükten sonra anlamlı veriler haline getirilmektedir.

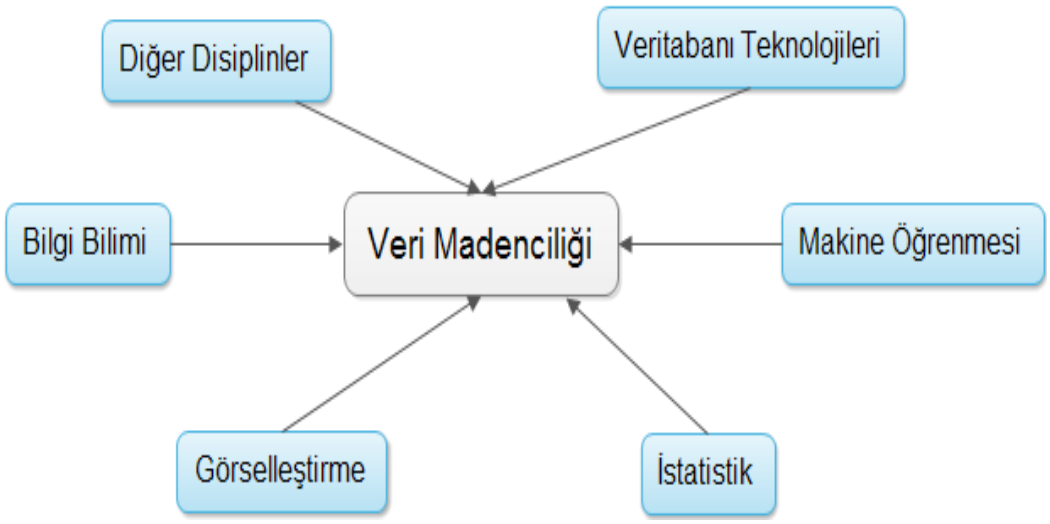
2.4.5. Verinin görselleştirilmesi

Karmaşık ve gürültülü olan veri yapısının temizlenmesiyle birlikte anlamlı verinin ortaya çıkarılması aşamasıdır. Bu süreçte verinin anlamlı hali görsel ve istatistiksel olarak sunulması işlemi gerçekleştirilir. Anlamlı veri, çeşitli yazılımlar ve kütüphaneler aracılığıyla kullanıcıya sunulur.

3. VERİ MADENCİLİĞİ VE MAKİNE ÖĞRENMESİ YÖNTEMLERİ

Veri madenciliği, büyük veri kümeleri üzerinde bilgisayar yazılımları aracılığıyla bilgi çıkarımı yapılmasını sağlayan kavramdır. Çıkarılan bilgi sayesinde gelecek ile ilgili tahminsel hesaplamalar yapmamızı kolaylaştıran bir yöntemdir. Veri madenciliği sayesinde karmaşık büyük veri kümeleri üzerinde daha kolay bilgiye ulaşılabilmektedir. Veri madenciliği uygulamaları sigortacılık, bankacılık, pazarlama, e-ticaret, telekomünikasyon, tıp ve eğitim gibi birçok alanda yaygın olarak kullanılmaktadır. Şekil 3.1'de veri madenciliği uygulama yöntemlerinin dağılımı görülmektedir. Veri madenciliği, büyük veri kümesinden bilgi çıkarımı sürecinin bir parçası olarak kabul edilmektedir. Bilgi çıkarım aşağıdaki süreçlerden geçmektedir [42]:

- Verinin temizlenmesi (Tutarsız ve yanlış verilerin çıkarılması)
- Veri bütünlüğü (Birden fazla kaynaktan verinin elde edilmesi)
- Veri kümesinden ilgili verinin seçilmesi
- ETL (Veri dönüşümü işlemlerinin gerçekleştirilmesi)
- Veri madenciliğini akıllı sistemler ile uygulamak (Makine öğrenmesi, YSA gibi)
- Elde edilen bilginin örüntüsü tanımlanması
- Sonuç olarak bilginin hedef kullanıcıya sunulması



Şekil 3.1. Veri madenciliği uygulamaları

Günümüzde teknolojinin gelişmesi veri miktarını kontrol etme güdümünü zorlaştırmaktadır. Büyük veri kümelerinin elle kontrol altına alınabilmesi ve süreçlerinin takip edilebilmesi mümkün olmamaktadır. Makine öğrenmesi, geçmişteki ve mevcut veriler kullanılarak gelecek için tahminsel hesaplamalar yapılmasına olanak sağlamaktadır. Bu sorunun çözümü için makine öğrenmesi kavramı ve yöntemleri ortaya çıkmıştır. Makine öğrenmesi yöntemleri, bilgisayar yazılımları aracılığıyla geçmişteki verileri kullanarak yeni veriler için en uygun modeli tasarlamaktadır [43].

Veri madenciliği uygulamalarında kullanılmak için birçok yöntem ve teknik geliştirilmiştir. Bu yöntemler kullanılarak sınıflandırıcı, tahminsel, davranış analizi gibi modeller oluşturulabilmektedir. Bu çalışmada, makine öğrenmesi içinde bulunan sınıflandırma yöntemlerinden yaygın olarak kullanılan ve uygulama alanı geniş olması sebebiyle Karar Ağaçları, Rastgele Orman, K-En Yakın Komşu, Destek Vektör Makineleri, Naive Bayes ve Lojistik Regresyon sınıflandırıcı algoritmaları kullanılmıştır. Bu algoritmaların her biri ile birlikte sınıflandırma işlemi için model oluşturulmuştur ve karşılaştırılmıştır. Genel olarak kullanılan veri madenciliği yöntemleri ve bu çalışmada kullanılan yöntem ve algoritma aşağıdaki gibi ele alınmaktadır.

3.1. Kümeleme

Bir veri kümesinde birbirine yakın değerler ile benzerlik gösteren verilerin birbiri içinde gruplanması işlemidir. Kümeleme basit olarak nesneleri benzerliklerine göre gruplar biçiminde ayrılmasıdır. Kümeleme algoritmaları bilgisayar yazılımı aracılığıyla kullanılarak bir veri setindeki verileri daha küçük kümelere ayrılmasını sağlamaktadır. Algoritmalar sayesinde benzer özellikteki öznitelikler aynı gruba dâhil edilmektedir.

3.2. Birliktelik Kuralları

Birliktelik kuralları, veri kaynağından elde edilen veriler arasındaki ilişkileri ifade eder. Her geçen gün artan veri miktarının çoğalması sebebiyle, kurum ve kuruluşlar bünyesinde barındırdığı veri kümelerindeki ilişkileri birliktelik kurallarını ortaya çıkarmak istemektedir. Büyük veri kümelerinden çıkarılan birliktelik kuralları ilişkisi sayesinde kurum ve kuruluşlar mali açıdan kar düzeyleri arttırabilmektedir.

Birliktelik kurallarını, en basit örneğiyle bir e-ticaret sitesinde yapılan alışverişlerde ürünler ile ürünleri alan müşteriler arasındaki ilişkiyi ortaya çıkarmak için

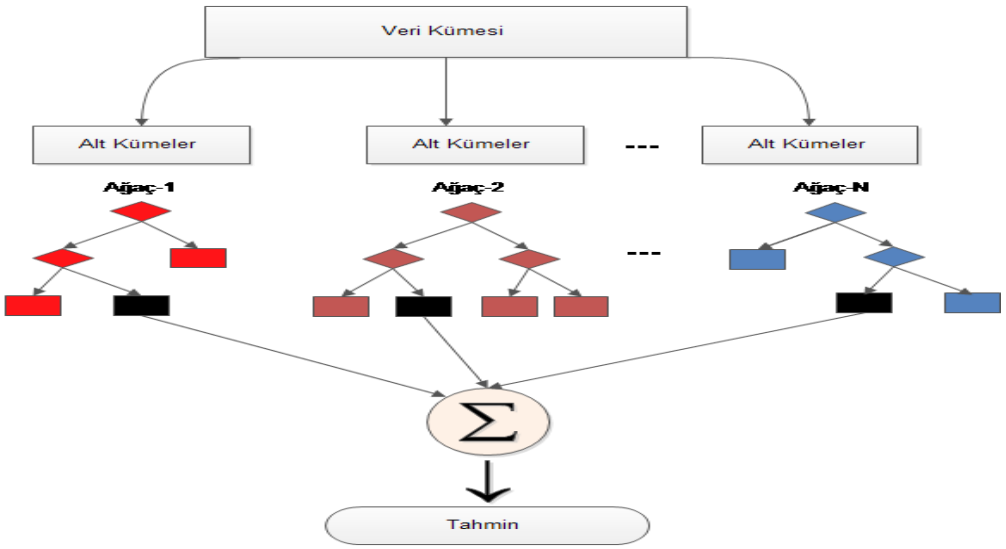
kullanılmaktadır. Bu ilişkiler sayesinde müşteri davranış analizi yapılarak satın alınan ürünler ve müşteriler arasındaki durumu ortaya çıkarabilmektedirler. Örneğin yeni bir telefon almak isteyen müşteri telefon kılıfı ile birlikte bir alım gerçekleştiriyorsa, e-ticaret sistemi otomatik olarak telefon alan müşteriye kılıfları önerebilmektedir.

3.3. Regresyon

Regresyon, sürekli bilginin bulunduğu ortamlarda tahminsel işlemlerde kullanılmaktadır. Örneğin bir alışveriş sitesinde ürün veya ürünler arayan bir müşterinin gelir modeli ve mesleği belirtilmiş ise o müşterinin gerçekleştirecek olan toplam sipariş tutarının tahmini olarak hesaplanmasında kullanılabilmektedir. Regresyon analizi, büyük veri kümesinde bulunan niteliklerin arasındaki ilişkiyi belirlemek için kullanılır.

3.4. Rastgele Orman Algoritması

Rastgele Orman algoritması, karar ağaçlarından farkı rastgele olarak birden fazla ağacından orman oluşturma felsefesine dayanmaktadır. Ağaçların sayısı arttıkça algoritmanın doğruluğu artmaktadır. Rastgele Orman, hyper-parametre (çalıştırılma sayısı gibi) olmadan bile iyi sonuçlar ve başarılı tahminler veren bir algoritmadır. Rastgele orman algoritması, sınıflandırma ve regresyon problemleri için uygun bir çözümdür. Şekil 3.2'de Rastgele orman sınıflandırıcının örneği gösterilmiştir.



Şekil 3.2. Rastgele orman çalışma yapısı

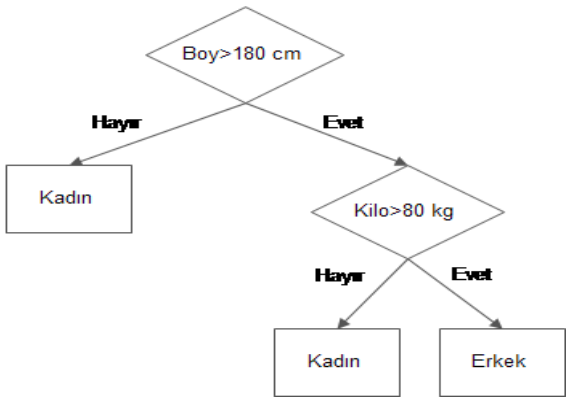
Şekil 3.2‘de yer alan yer alan ağaç yapısında veri kümesi rastgele olarak 1’den...N’e kadar olan alt kümelere ayrılarak ve bir orman yapısını oluşturmaktadır. Her bir ağaç yapısında hesaplamadan sonra ise en iyi değerler toplanarak tahmin yürütülmektedir.

3.4.1. Karar Ağaçları

Karar ağaçları, günümüzde maliyeti düşük olması, yorumlanabilir olması, kolayca diğer sistemlere uygun olması, doğrulanabilirliği ve kolay bir şekilde anlaşılması göz önüne alındığından kurum, kuruluşlar, akademik dünyada sıkça kullanılmaktadır. Karar ağaçları önemli sınıflandırıcı yöntemlerden biridir. Karar ağacı bir tahminsel yaklaşım göstermektedir. Karar ağaçları sayesinde anlaşılır kuralları yönetme, oluşturma, bilgisayar yazılımları arasındaki bağlantının ve desteğinin yoğun olması sebebiyle günümüzde yoğun olarak kullanılmaktadır. Karar ağaçlarının bazı kullanım alanları şu şekildedir:

- Elektronik posta adreslerine gelen reklam postalarının analizinde
- Kredi başvurusu yapan şahısların geçmişteki verileri analiz edip sonuçlandırmasında
- Ağ üzerinde meydana gelen etkileşimlerin ve isteklerin analiz edilerek tehlike olup olmadığının analizinde
- Ürün alış satış ilişkilerinin belirlenmesi gibi birçok kullanım alanları vardır

Karar ağaçlarının kullanımına ilişkin birçok algoritma geliştirilmiştir. ID3, C4.5, Twoing ve Gini algoritmaları örnek verilebilir. Bu çalışma kapsamında Gini yöntemi tercih edilmiştir. Şekil 3.3’de karar ağacının örnek çalışması gösterilmiştir.



Şekil 3.3. Karar ağacı örneği

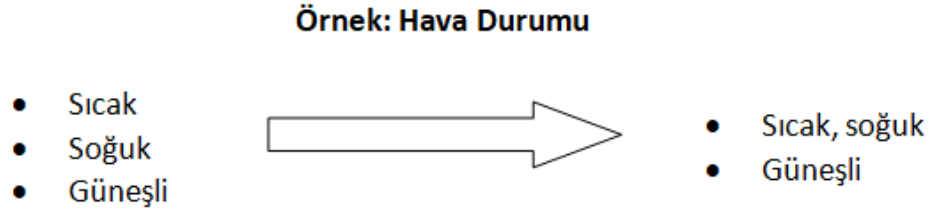
3.4.2. Gini Algoritması

Gini algoritması karar ağacı yöntemlerinde kullanılan bir algoritmadır. Sınıflandırma ve regresyon ağacı (CART) olarak bilinmektedir [44]. CART, ikili bölünmeler şeklinde gerçekleşen sınıflandırma ve regresyon ağacı yöntemidir. Her bir karar düğümden itibaren iki dala ayrılmaktadır [45]. Bölünme değeri ve ilk bölünme aşaması Gini değerine bakılarak hesaplanmaktadır. Gini katsayısı, büyük veri kümesindeki özniteliklerin değerlerin oranı olarak hesaplanmaktadır. Gini değeri eşit olan niteliklerde dağılım ve ağaç içerisindeki yolu aynı demektir denilebilir. Gini algoritmasının amacı, en iyi durumu öğrenmek için her zaman veri kümesindeki en büyük değeri bulmaktır. Gereksiz verileri dışarıda bırakır, sınıflandırma bölgesini ele alınır ve birbirine yakın olan seçenekler sınıflandırılır. Şekil 3.4'de örneği verilmiştir. Gini algoritması işlemleri şu şekildedir:

- Tahmin edilen en iyi bölünme katsayısını bul,
- Düğümün en iyi bölünmesini bul: 1. Adımda bulunan en iyi bölünme değerleri arasında en yüksek değeri seç
- Karar ağacı sonlanmadığı durumda 2. Adımda bulunan en büyük katsayı değerini seçerek ilgili düğümü bölün

Gini algoritmasının uygulaması şu şekilde gerçekleşmektedir [45]:

- Nitelik değerlerinin her biri ikili bölünmeler olacak şekilde sınıflanır. Elde edilen sol ve sağ bölünmelere karşılık gelen sınıf değerleri gruplandırılır.
- Sağ düğüm ve sol düğüm için bölünmeler ayrı ayrı Gini hesaplamaları yapılmaktadır.



Şekil 3.4. Bölünme örneği

Büyük veri kümesindeki niteliklerin sağ veya sol düğümde bulunup bulunmadığını anlamak için Gini değerleri hesaplanmaktadır. Sağ düğüm için $Gini_{Sağ}$, sol düğüm için

Gini_{Sol} değeri hesaplanmaktadır. $\text{Gini}_{\text{Sağ}}$ ve Gini_{Sol} değerinin hesaplanması için aşağıda (3.1) ve (3.2) denklemleri verilmiştir.

$$\text{Gini}_{\text{Sağ}} = 1 - \sum_{i=1}^{k} \left[\frac{L_i}{|T_{Sağ}|}\right]^2 \quad \textbf{(3.1)}$$

$$\text{Gini}_{\text{Sol}} = 1 - \sum_{i=1}^{k} \left[\frac{R_i}{|T_{Sol}|}\right]^2 \quad \textbf{(3.2)}$$

(3.1) ve (3.2) denklemlerindeki simgelerin anlamları şu şekilde söylenebilir:

$\boldsymbol{L_i}$ = Ağacın sol kısmında bulunan i grubunun örneklerinin sayısı

$\boldsymbol{R_i}$= Ağacın sağ kısmında bulunan i grubunun örneklerinin sayısı

k = Sınıf sayısı

T = Düğümdeki örneklerin sayısı

$|\boldsymbol{T_{Sol}}|$ = Sol kısımdaki nitelik sayısı

$|\boldsymbol{T_{Sağ}}|$ = Sağ kısımdaki nitelik sayısı

Büyük veri kümesindeki n tane kayıtın her bir j nitelik değeri için aşağıdaki denklem kullanılır.

$$Gini_j = \frac{1}{n}\left(|\boldsymbol{T_{Sol}}|\text{Gini}_{\text{Sol}} + |\boldsymbol{T_{Sağ}}|\text{Gini}_{\text{Sağ}}\right) \quad \textbf{(3.3)}$$

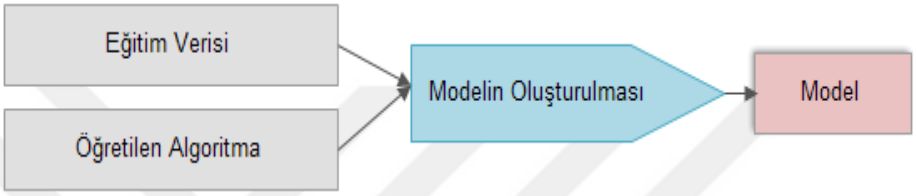
Her j niteliği için hesaplanmış olan $Gini_j$ değerinden en küçük olan seçilir ve bölünme aşaması bu nitelik değeri üzerinden yapılır. Büyük veri kümesi üzerinde kalan nitelik değeri için yineleme işlemleri tekrar edilerek işlem tamamlanır. Gini algoritmasının basit, kullanışlı ve daha az kod karmaşıklığı olması sebebiyle bu çalışmada kullanılması uygun bulunmuştur.

3.5. Sınıflandırma

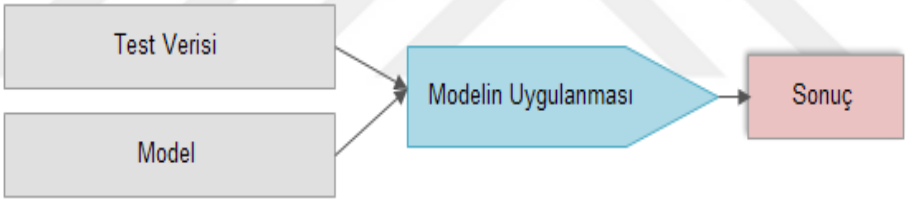
Veri madenciliği alanında sıklıkla kullanılan bir yöntemdir. Bir veri kümesinde bulunan öznitelik özelliklerini kullanarak başka bir veri kümesinin hangi sınıfa ait olduğunu bulan yöntemdir. Sınıfları gruplamak için kullanılan öğrenme yani eğitim veri kümesi (Training Data Set), sınıfı bulunmak istenen veri kümesi test veri seti (Test Data Set) olarak adlandırılmaktadır. Kısacası yeni gelen verinin hangi sınıfta olduğunu

oluşturur. Sınıflandırma algoritmaları bilgisayar yazılımları kullanılarak sayesinde eğitim veri kümesinden öğrendikleri bilgiyi, test veri kümesi kullanılarak onların sınıflarını bulunmasına yardımcı olur.

Sınıflandırma işlemini gerçekleştirmek için elde edilen eğitim verisi sınıflandırma algoritmaları kullanılarak model oluşturulur. Daha sonra test verileri ele alınarak oluşturulan modelin uygulanması ile birlikte test verilerinin sınıflandırmasını yapmaktadır. Bu durum Şekil 3.5 ve Şekil 3.6'da gösterilmiştir.



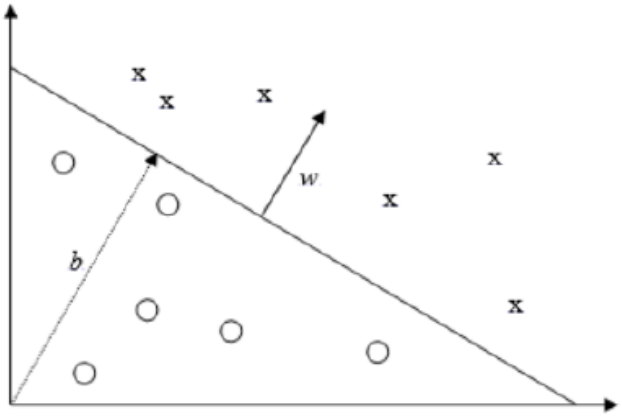
Şekil 3.5. Eğitim verisi ile modelin oluşturulması



Şekil 3.6. Test verisine modelin uygulanması

3.5.1. 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. DVM, regresyon ve sınıflandırma işlemleri için kullanılmaktadır. Destek vektör makineleri veri kümelerinde verileri birbirinden ayırmak için kullanılmaktadır. Veri kümesinin eğitilmesi için konveks olarak optimize edilen uygun bir yöntem seçilebilir. Doğrusal olarak ayrılamayan veri setini, daha büyük boyuta taşıyarak düzlemde ayırmayı sağlamaktadır.



Şekil 3.7. İki farklı veri tipinin doğrusal ayrılması

Şekil 3.7'de iki farklı veri tipi birbirinden doğrusal olarak ayrılmaktadır. Bu ayrım fonksiyonu aşağıdaki şekilde ifade edilir:

$$f(x) = \sum_{i=1}^{n} W_i X_i + b \quad \textbf{(3.4)}$$

Denklem (3.4)'te X ayrım yapılan girdi verisi, b ise eşik değeri olarak tanımlanmaktadır. W ağırlıklandırılmış vektördür. Bu vektör veri kümesinin eğitimi sırasında belirlenmektedir.

3.5.2. K-En Yakın Komşu Algoritması

K-En yakın komşu algoritması eğitim kümesindeki nesneleri sınıflamak için geliştirilmiş bir algoritmadır. Bu algoritmada, eğitim kümesindeki öznitelik değerleri çok boyutlu bir özellik uzayında eşleştirilir. Uzay, Eğitim kümesinde bulunan hedef değişkenler tarafından bölgelere ayrılır. Yapılan hesaplamada bir noktaya en uygun sınıfa belirlenir. Bu atama işleminde genellikle Öklid, Manhattan ve Minkowski uzaklık yöntemleri kullanılmaktadır [46].

Veri kümesinin eğitim aşaması bağımsız öznitelik değerlerinden oluşan değerlerin ve eğitim örneklerinin sınıf etiketlerin depolanmasını içerir. Sınıflandırma aşamasında, öznitelik değerleri eğitim kümesi üzerinden bulunmaktadır. Eski ile yeni olan değerlerin uzaklığı hesaplanmaktadır. Daha sonra ise örnekler içerisinde k en yakın değeri seçilmektedir. Bu bağlamda, uygun sınıf bulunmaya çalışılmaktadır.

Algoritmanın k değeri veri kümesindeki öznitelik değerlerine bağlı olarak belirlenmektedir. Öznitelik değerleri, k değerinin büyük seçilmesi sınıflamadaki gürültünün temizlenmesi açısından önemlidir [46].

KNN algoritması beş adımdan oluşmaktadır [46]:

1. Öncelikle k değeri belirlenir.

2. Hedef noktaya olan uzaklıkları hesaplanır.

3. Uzaklıklar sıralanır ve en minimum uzaklığa bağlı olarak en yakın komsular bulunur.

4. En yakın komsu kategorileri toplanır.

5. En uygun komsu kategorisi seçilir.

Öklid Uzaklığı: $$d(X,Y)=\sqrt{\sum_{i=1}^{k}(X_i - Y_i)^2} \quad \textbf{(3.5)}$$

Manhattan Uzaklığı: $$d(X,Y)=\sqrt{\sum_{i=1}^{k}|X_i - Y_i} \quad \textbf{(3.6)}$$

Minkowski Uzaklığı: $$d(X,Y) =\left[\sum_{i=1}^{k}(|X_i - Y_i)^q\right]^{1/q} \quad \textbf{(3.7)}$$

Denklem (3.5), (3.6) ve (3.7)'de uzaklık bağlantıları verilmiştir. Bu tez çalışmasında iki nokta arasındaki uzaklık hesaplaması için Minkowski bağlantısı tercih edilmiştir.

3.5.3. 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. Kullanım alanları mail spam filtreleme, tahmin hesaplama ve tavsiye motorlarıdır. P(A\B) olayının olması olasılığı Denklem (3.8)'de belirtilmiştir.

$$P(A\backslash B) = \frac{P(B\backslash A)*P(A)}{P(B)} \quad \textbf{(3.8)}$$

- P(A | B) : B'nin olasılığı verildiğinde A olayının olma olasılığı
- P(B | A) : A'nın olasılığı verildiğinde B olayının olma olasılığı
- P(A) : A olayının olma olasılığı
- P(B) : B olayının olma olasılığını vermektedir.

3.5.4. Multinominal Lojistik Regresyon

Lojistik Regresyon, makine öğrenmesi yöntemlerinde sıklıkla kullanılan, 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. Aşağıda Lojistik Regresyon ve Multinominal Lojistik Regresyon denklemi belirtilmiştir. Denklem (3.9) ve (3.10)'da Lojistik ve Multinominal Lojistik Regresyon formülü temsil edilmiştir.

Lojistik Regresyon: $$P(x) = \frac{e^{\beta_0+\beta_1 X_1}}{1+e^{\beta_0+\beta_1 X_1}} \quad \textbf{(3.9)}$$

Denklem (3.9)'da Lojistik Regresyon için;

$\boldsymbol{e}$, 2.7182

$\boldsymbol{\beta_0}$**:** Bağımsız değişkenin ilk değeri 1 olduğu için ortaya çıkan katsayıdır.

$\boldsymbol{X_1}$**:** Bağımsız değişken

P(x) = y: Hedef yani bağımlı değişkenin değeridir.

Multinominal Lojistik Regresyon: $$P(i,j) = \frac{e^{\sum_{j=1}^{K} \alpha+\beta_{kj} X_{kji}}}{\sum_{j=1}^{J} e^{\sum_{j=1}^{K} \alpha+\beta_{kj} X_{kji}}} \quad \textbf{(3.10)}$$

Denklem(3.10)' da Multinominal Lojistik Regresyon için [47];

P(i, j) olayında, j hedef değişkenin i'inci durumunun olma veya seçilme olasılığıdır.

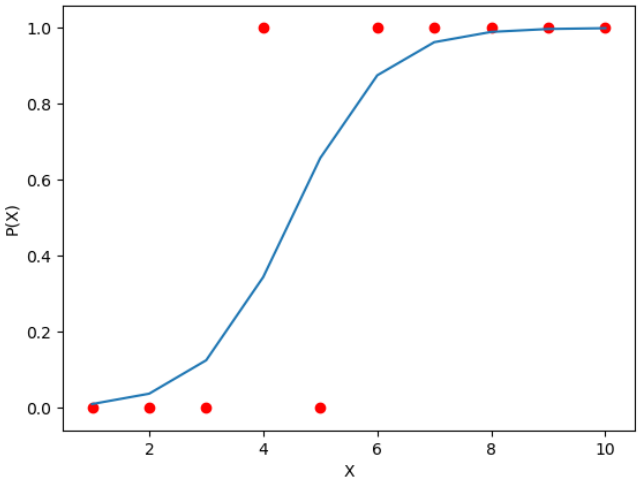
k = bağımsız değişkenler, **K**= Özelikler kümesi, **J**=Hedef değişkenlerin kümesi

,j= bağımlı, hedef değişken ve i = olay olmak üzere;

α: sabit katsayı

X_{kji}:k özelliğinin j üzerinde i'inci durumunun olma katsayısı.

β_{kj}:k özelliğinin j üzerindeki katsayısı



Şekil 3.8. Lojistik regresyon

Şekil 3.8'de gösterildiği gibi lojistik regresyon sınıflandırıcının hedef değişken dağılımı gösterilmektedir.

3.6. Sınıflandırmada Yardımcı Teknikler

Makine öğrenmesi teknikleri kullanılarak yapılan sınıflandırma işlemlerinde veri kümesinin boyutundan ziyade içeriğindeki veri yapısı çok önemlidir. Sınıflandırmaya katılacak olan değişken ve niteliklerin biçimi, yapısı, türü ve dönüşümü belirli teknik ve yöntemlerden geçirilerek özellik seçimi gerçekleştirilir.

3.6.1. 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 [48]. 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.

3.6.2. Özellik Çıkarımı

Bir veri kümesinde yer alan özniteliklerin bellek boyutunu düşürme yöntemidir. Örneğin bir resim dosyasında kırmızı elma meyvesinin tespitinin yapılması isteniyor ise dosyanın tamamının ele alınması yerine sadece renk tonlarının belirlendiği kısmın ele alınması ile bellek boyutu düşürülebilmektedir. Bu işlemde zor problem minimuma indirgenerek çözülmesi hedeflenmektedir. Özellik çıkarımının doğru yapılması oluşturulan modelin doğruluğunu ve performansını etkilemektedir. TF-IDF, Word2Vec, Count Vectorizer, Feature Hasher yöntemleri özellik çıkarımlarında sıklıkla kullanılmaktadır [49]. Genel olarak bir cümle veya doküman yapısından özellik çıkarım işlemleri gerçekleştirilmektedir. Bu tez kapsamında cümle veya doküman içeriğinden bilgi çıkarımı hedefi kapsamında olmadığından özellik çıkarım yöntemleri kullanılmamıştır.

3.6.3. 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 [49]. Tez kapsamında String Indexer, Index To String, One Hot Encoder ve Vector Assembler yöntemleri kullanılmıştır.

String Indexer; Veri kümesi içerisinde bulunan kategorik özniteliklerin farklı değerlerine indeks biçiminde nümerik değere dönüştürülmesi işlemidir. Veri kümesinde bulunan, her bir değerin sıralı olarak değerlerin karşılığı verilmiştir. Tablo 3.1'de kategorik değer olan terör örgütü isminin nümerik karşılığına dönüştürülmesi gösterilmiştir.

Tablo 3.1. String indexer yöntemi

Kategorik Değişken	Nümerik Karşılığı
Kara Milliyetçiler	2
Beyaz aşırılık yanlıları	5
Grevciler	10
Öğrenci Radikalleri	14

One Hot Encoder; Kategorik değişkenlerin "0" veya "1" olarak temsil edilmesini sağlayan yöntemdir. Bu yöntem öznitelik bir vektörde temsil edilmektedir. Tablo 3.2'de verilen kategorik değerlerin vektör biçimde temsil edilmesi gösterilmiştir.

Tablo 3.2. One hot encoder yöntemi

Kara Milliyetçiler	Beyaz aşırılık yanlıları	Savaşçılar	Öğrenci Radikalleri	Vektörel Gösterimi
1	0	0	0	(14,[2],[1])
0	1	0	0	(14,[5],[1])
0	0	1	0	(14,[10],[1])
0	0	0	1	(14,[14],[1])

Vektörel gösterim olarak açıklamak gerekirse **"**Kara Milliyetçiler**"** isimli terör örgütünün vektörel biçimindeki karşılığı **"(14, [2], [1])"** görülmektedir.**(14)** değeri veri

kümesinde 14 farklı örgüt olduğunu temsil etmektedir. **[2]** değeri ise örgütün veri kümesinde 2 numaralı indiste bulunduğunu göstermektedir. **[1]** değeri ise vektörel düzlemde o satırda bu değişkenin varlığını göstermektedir.

Vector Assembler; Bir veri kümesinde bulunan özniteliklerin bir sütun halinde birleştirilip vektörel biçimde temsil edilmesini sağlayan bir yöntemdir. Vektörel biçimde temsil edilmesi için kategorik değerlerin numerik biçime dönüştürülmesi daha sonra ise tek bir sütun haline getirilip özellik bütünü oluşturulması hedeflenir. Burada kullanılan yöntem sınıflandırma modellerinin ayrı ayrı özellik olarak değil de tek bir özellik formatını kullanarak başarım oranını arttırmak ve hesaplama süresini azaltmaktır. Tablo 3.3'te özniteliklerin vektörel biçimde temsili gösterilmiştir.

Tablo 3.3. Özelliklerin vektörel biçimi

Ülke	Bölge	Atak Tipi	Hedef Kitle	Silah Türü	Özellik (Feature)
58	2	1	5	13	**(18, [1, 58, 2, 1, 5, 13])**

Ülke, bölge, atak tipi, hedef kitle ve silah türü niteliklerinin numerik biçimi ele alınarak tek bir sütun haline getirilip vektörel biçime dönüştürülmüştür. Burada özellikler **"(18, [1, 58, 2, 1, 5, 13])"** vektörel olarak temsil edilmiştir. **(18)** değeri veri kümesinde bulunan özellik sütununun en üst sınır indisini ifade etmektedir. Diğer değerler ise özniteliklerin karşılığını göstermektedir. Burada dizinin en başındaki **"1"** değeri vektörün değerini ifade etmektedir.

Index To String; Veri kümesi içerisinde bulunan kategorik özniteliklerin farklı değerlerine indeks biçiminde nümerik değere dönüştürülmesi işleminin tam tersidir. Örneğin bir kategorik değişkenin numerik değere dönüşümünden sonra ise bu değerin kategorik karşılığına ihtiyaç duyuldu ise bu yöntem kullanılmaktadır. Tablo 3.4'te numerik değer olan terör örgütü isminin kategorik karşılığına dönüştürülmesi gösterilmiştir.

Tablo 3.4. Numerik değerlerin karşılığı

Nümerik Değer	Veri Kümesindeki Karşılığı
2	Siyah Milliyetçiler
5	Aşırı Beyaz Yanlıları
10	Grevciler

4. KULLANILAN VERİ KÜMESİ VE ANALİZ İŞLEMLERİ

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 olaylarını detayları ile beraber bilgiler içermektedir. Veri kümesi yaklaşık 170 MB, içeriğinde 181.692 satır bulunmaktadır. Örnek veri kümesinde 135 adet sütun bulunmaktadır. Analiz ve istatistik işlemlerinde veri kümesinin tamamı ele alınmıştır. Sınıflandırma işlemlerinde ise, 6 adet sütun, terör saldırısının en fazla gerçekleştiren ilk 10 örgüt ele alınmıştır. Tablo 4.1'de bu çalışmadan kullanılan örnek büyük veri kümesinin alanlarının içeriği belirtilmiştir.

Tablo 4.1. Büyük veri kümesi içeriği

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

Tablo 4.1'de gösterildiği gibi veri kümesi içeriği belirli alanlardan oluştuğu görülmektedir. Fakat bu alanlar veri kümesi içeriğinde klasik veri tabanı tabloları gibi belirtilmemiştir. Dosya yapısı yarı yapılandırılmış veri biçimi şeklinde olduğu görülmektedir. Bu alanlar boşluk ve özel karakterler bulunduğundan veri temizleme işlemi yapılmıştır. Regex (Regular Expression) yani düzenli ifadeler fonksiyonu kullanılarak veri temizleme ve düzenleme işlemi yapılmıştır. Düzenli ifadeler, metinler üzerinde seçme, ayrıştırma ve ayırma işlemleri yapılmasına olanak sağlayan yapıdır. Bu yapı kullanılarak istediğimiz veriyi bir alana bölerek üzerinde analiz işlemleri yapmamızı daha sağlıklı kılmaktadır.

4.1. Dünyada Yaşanan Terör Olayları

Dünya üzerinde çeşitli nedenlerle terör saldırıları yaşanmaktadır. Bu saldırıların bazılarında terör olayının kimin yaptığı yıllarca sorun haline gelmiş kamuoyu nezdinde sürekli tartışılmaktadır. Bazı saldırılar faili meçhul olarak kayıtlara geçmiş bazıları da teröristlerin veya terörist grubun kendileri tarafından itiraf niteliğinde açıklamalar yapılmıştır. Yaşanan terör olaylarında insanlar hayatlarını kaybetmiş, yaralanmalara ve psikolojik travmalara neden olmuştur. Terör örgütünün kimliğinin belirlenmesi o örgütle mücadele etmede önemli bir ayrıntı olmaktadır. Bu çalışmada terör örgütü kimliğinin belirlenmesinden ziyade yapılan saldırı davranışlara ve olgulara bakılarak örgüt hakkında fikir sahibi olması amaçlanmıştır.

4.2. Terör Örgütleri ve Saldırı Analizleri

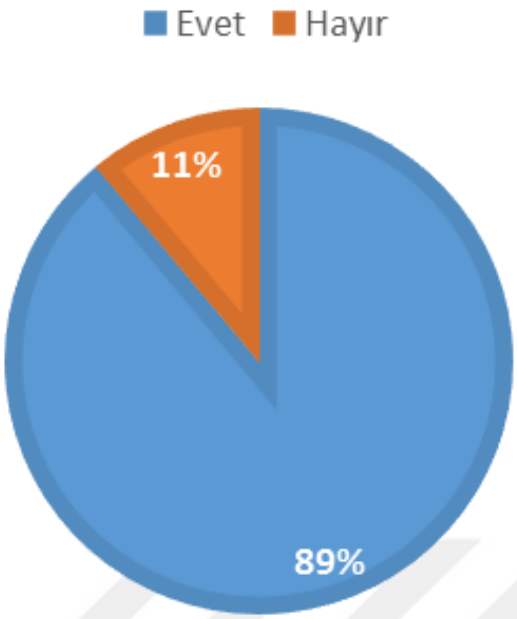
Terör örgütleri dünya üzerinde birçok saldırı gerçekleştirmiştir. Bu durum birtakım istatistik bilgilerin oluşmasına sebebiyet vermiştir. Bu bölümde terör örgütlerinin gerçekleştirdiği saldırıların istatistiki verileri tablolar ve şekiller üzerinde gösterilmiştir.

4.2.1. Bölge, ülke ve şehirlere göre analiz

Bu bölümde dünya üzerinde yaşanan saldırıların analizi genel, ülke ve bölgesel olarak yapılmıştır. Veri kümesi ele alınarak incelendiğinde 181.692 adet terör olayı gerçekleşmiştir. Tablo 4.2'ye bakıldığında bu saldırılardan 20.060 terör olayının başarısız, 161.632 terör olayının ise başarılı saldırı olarak kayıt altına geçmiştir.

Tablo 4.2. Saldırı başarı durumu

Başarı Durumu	Sayısı
Evet	161.632
Hayır	20.0560



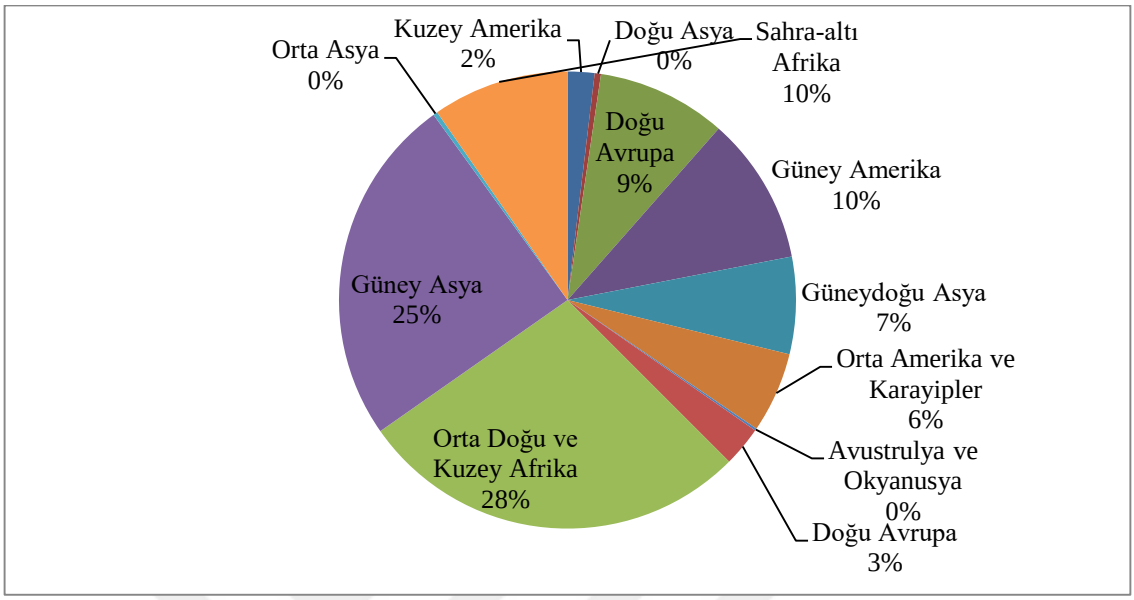
Şekil 4.1. Başarı oranı

Şekil 4.1'de görüldüğü gibi oransal olarak başarılı saldırıların sayısı epey yüksek görünmektedir. GTD'ye göre 1970-2017 yılları arasındaki terör saldırıları % 89 oranında başarılı olmuş, başarısız oran ise % 11 olarak kalmıştır.

Tablo 4.3. Bölgelere göre saldırı sayısı

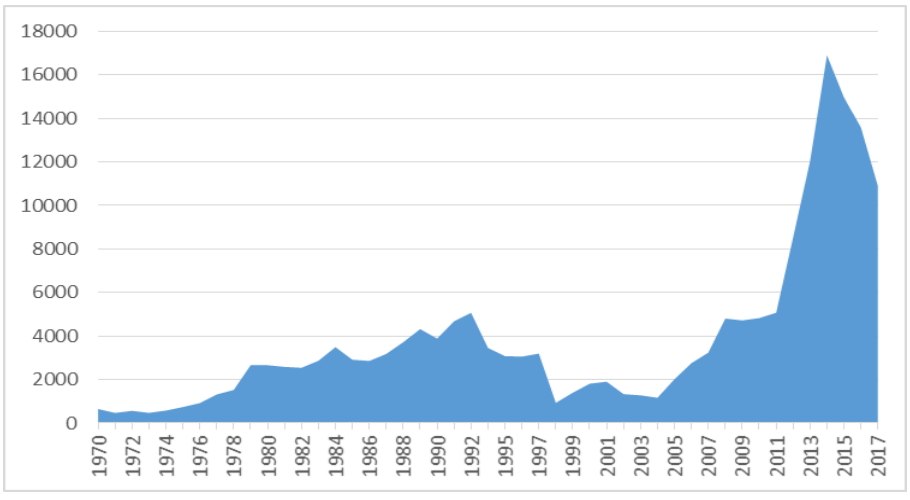
Bölge	**Saldırı Sayısı**
Kuzey Amerika	3.456
Doğu Asya	802
Doğu Avrupa	16.639
Güney Amerika	18.978
Güneydoğu Asya	12.485
Orta Amerika ve Karayipler	10.344
Avustralya ve Okyanusya	282
Doğu Avrupa	5.144
Orta Doğu ve Kuzey Afrika	50.474
Güney Asya	44.974
Orta Asya	563
Sahra-altı Afrika	17.551
Toplam	**181.692**

Tablo 4.3‘de terör saldırılarının kıtalara göre sayısı verilmiştir. Bu tabloya bakıldığında 50.474 terör saldırı ise Orta Doğu ve Kuzey Afrika bölgesi ilk sırada oluşmaktadır.



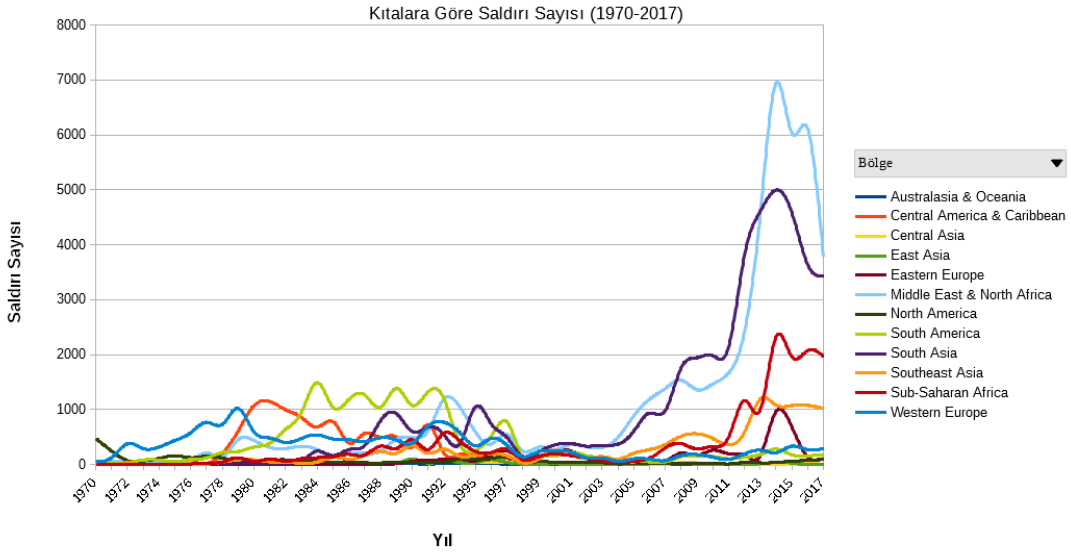
Şekil 4.2. Bölgelere göre saldırı sayısı

Oransal olarak bakıldığında Şekil 4.2’de belirtildiği gibi % 28 olarak görülmüştür. Bunun anlamı dünyadaki tüm terör olaylarının % 28’lik kısmı bu bölgede yaşanmaktadır. Ondan sonra gelen ise 44.974 terör saldırısı ve Şekil 4.2’de belirtildiği gibi % 25’lik oran ile Güney Asya görülmektedir. En az terör saldırısının yaşandığı bölgeler olarak Avustralya ve Okyanusya kıtası görülmektedir. Burada yaşanan terör olayı sadece 282’dir.



Şekil 4.3. Yıllara göre saldırı sayıları

Şekil 4.3 incelendiğinde 1990-1992 arasında terör olaylarında biraz artış gözlemlenmiştir. 2003 yılından sonra ise olaylar gittikçe artmıştır.



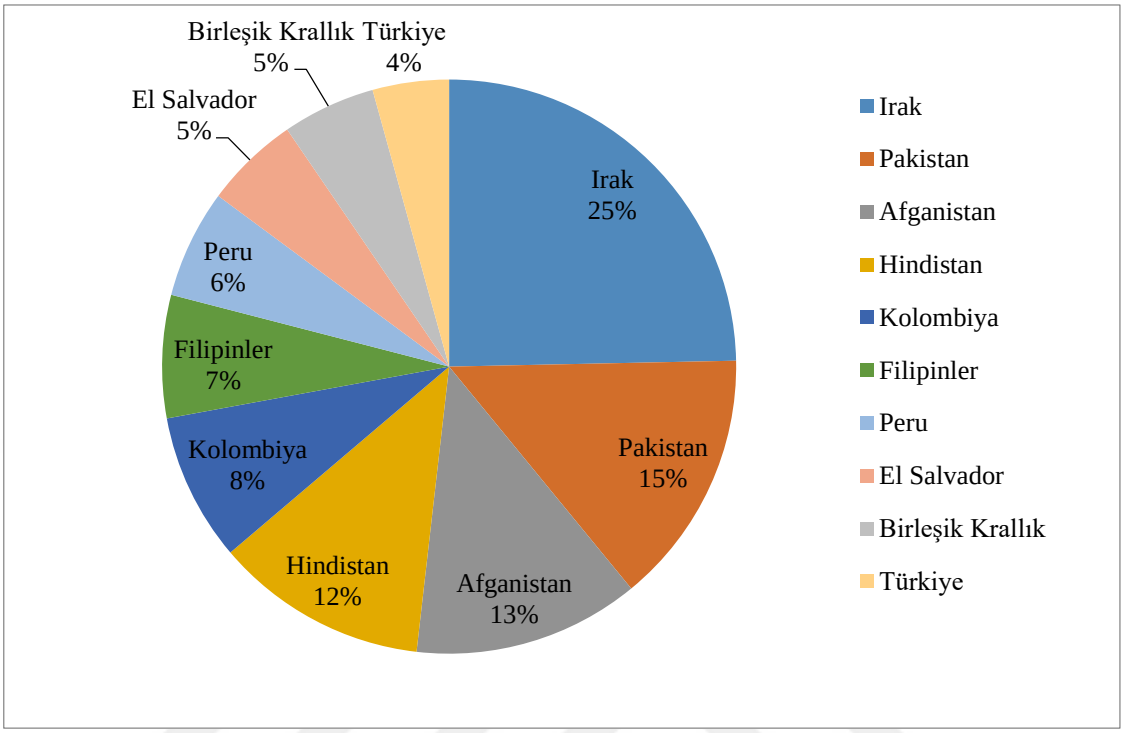
Şekil 4.4. Bölgelerde yıllara göre terör saldırısı

Şekil 4.4'e bakıldığında terör olaylarının 2003-2015 yılları arasında en çok Orta Doğu ve Kuzey Afrika bölgesinde ciddi oranda artış olduğu gözlenmektedir.

Tablo 4.4. Ülke bazında terör saldırıları

Ülke	Saldırı Sayısı
Irak	24.636
Pakistan	14.368
Afganistan	12.731
Hindistan	11.960
Kolombiya	8.306
Filipinler	6.908
Peru	6.096
El Salvador	5.320
Birleşik Krallık	5.235
Türkiye	4.292

Tablo 4.4'te terör saldırılarının ülke bazında saldırı sayısı verilmiştir. Görüldüğü gibi 24.636 terör saldırı sayısı ile Irak ilk sırada oluşmaktadır.



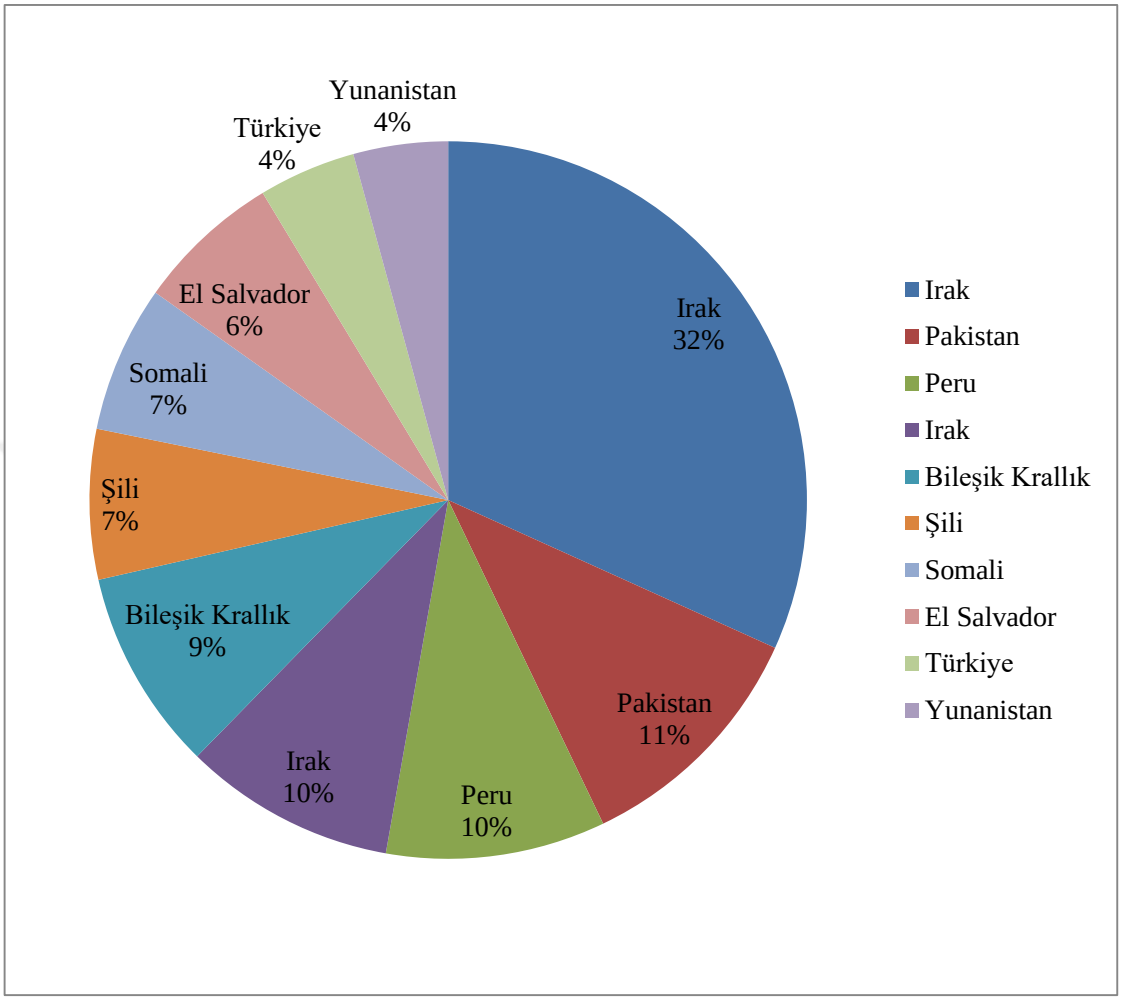
Şekil 4.5. Ülkelere göre saldırı sayısı

Oransal olarak bakıldığında şekil 4.5'de % 25 olarak görülmüştür. Bunun anlamı terör olaylarının yaşandığı ilk 10 ülkedeki % 25'lik kısmı bu ülkede yaşanmıştır. Türkiye'de ise GTD'ye göre 4.292 terör saldırı gerçekleşmiştir. Tablo 4.4'te ve Şekil 4.5'de diğer ülkelerin durumu da gösterilmiştir.

Tablo 4.5. Şehirlere göre terör saldırısı

Şehir	**Ülke**	**Saldırı Sayısı**
Bağdat	Irak	7.586
Karaçi	Pakistan	2.659
Lima	Peru	2.363
Musul	Irak	2.282
Belfast	Bileşik Krallık	2.172
Santiago	Şili	1.614
Mogadişu	Somali	1.582
San Salvador	El Salvador	1.562
İstanbul	Türkiye	1.048
Atina	Yunanistan	1.018

Tablo 4.5'te görüldüğü gibi dünyada en çok saldırıya uğrayan şehir olarak Bağdat şehri ön plana çıkmıştır. İstanbul'da ise 1.048 saldırı gerçekleşmiştir.



Şekil 4.6. Şehirlere göre oranlar

Şekil 4.6'ya bakıldığında terör olaylarının Irak ve Pakistan ülkelerinde yoğun olduğu görülmektedir. Bölge ile orantılı olarak terör olayının arttığı saptanmıştır.

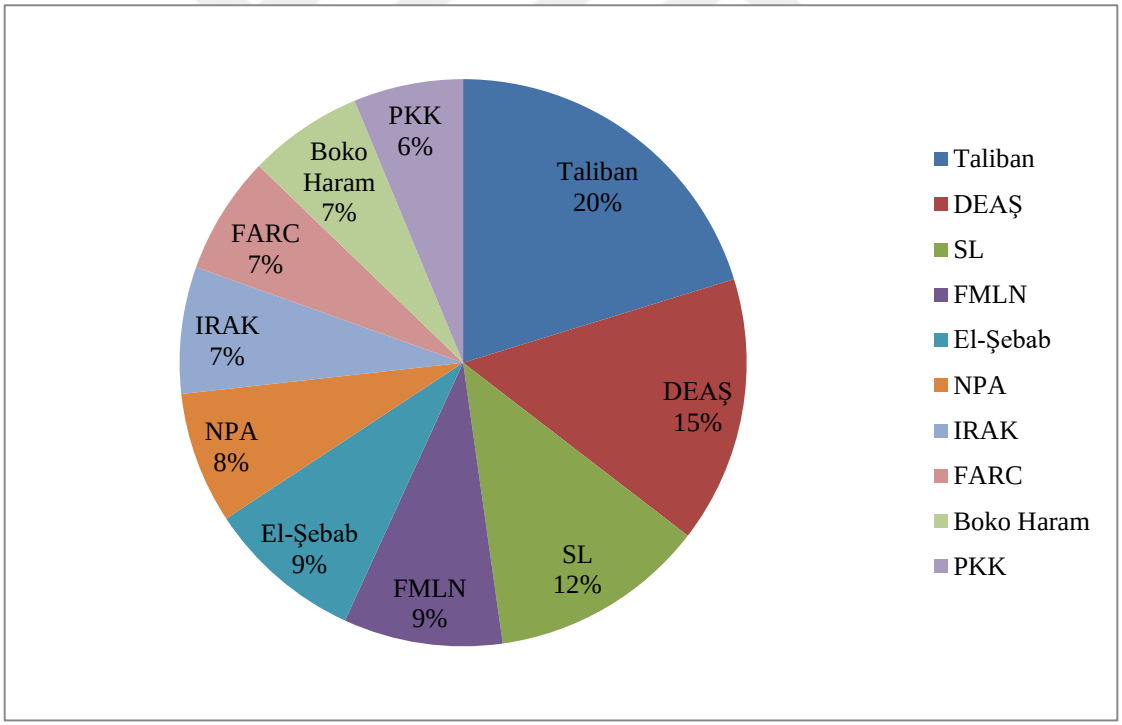
4.2.2. Terör Örgütleri ve Sayısal Değerler

GTD'ye göre dünyada 1970-2017 yılları arasında gerçekleştiren saldırılarda terör gruplarından bazıları tespit edilebilmiş büyük bir kısmı ise tespit edilememiştir. Tablo 4.6'da değerlere bakıldığında dünyadaki saldırıların 82.782 adedini hangi terör grubun gerçekleştirdiği hala bilinmemektedir. Bu saldırılar tüm saldırıların % 46'sını oluşturmaktadır.

Tablo 4.6. Terör örgütleri

Terör Örgütü	Saldırı Sayısı
Bilinmeyen	82.782
Taliban	7.478
DEAŞ	5.613
SL	4.555
FMLN	3.351
El-Şebab (EŞ)	3.288
NPA	2.772
IRA	2.671
FARC	2.487
Boko Haram(BH)	2.418
PKK	2.311
…	**…**
Toplam	**181.692**

En çok saldırı gerçekleştiren terör örgütü Taliban olarak görülmektedir. Bu örgüt 7.478 sayı ve Tablo 4.6'da belirtilen değerler üzerinden bulunamayan saldırıların sayısı çıkarıldığı zaman % 20'lik kısmı oluşturmaktadır.



Şekil 4.7. Örgütlerin oransal dağılımı

4.2.3. Saldırı Tipi ve Sayıları

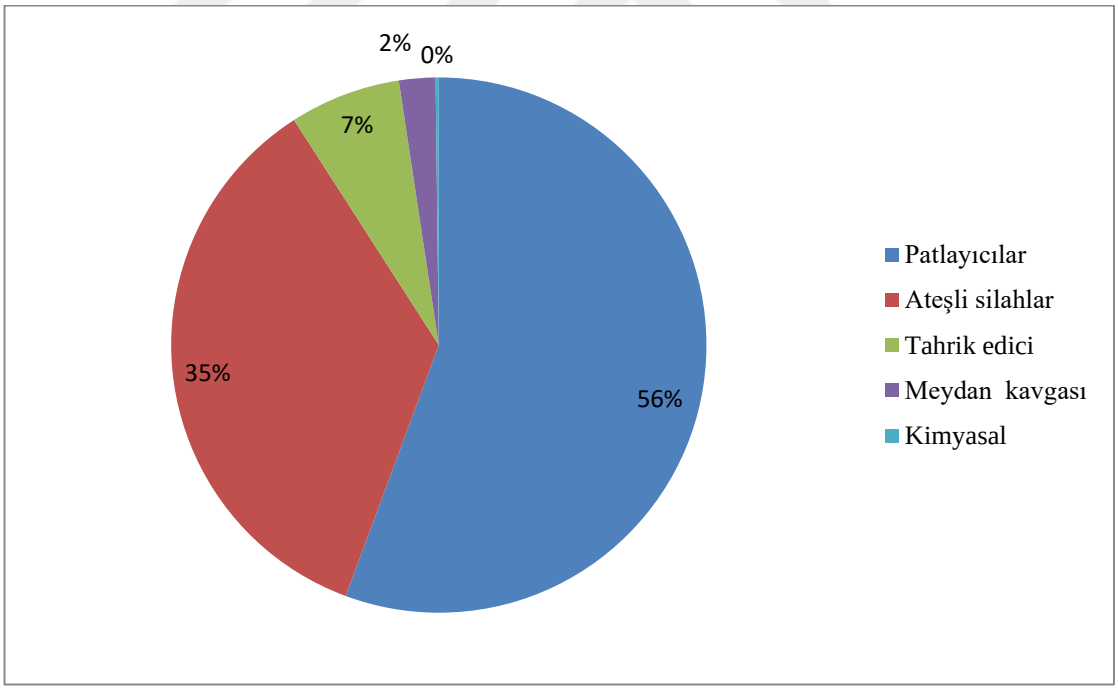
Dünyada yaşanan terörizm olaylarının yaşattığı travma ve etkisi olayın yapılma veya saldırma şekli ile doğrudan ilgilidir. Tablo 4.7'ye bakıldığında en çok yaşanan ilk üç

saldırı tipi olarak sırasıyla patlayıcılar, ateşli silahlar ve kundaklama olarak görülmektedir.

Tablo 4.7. Saldırı türleri

Saldırı Tipi	Sayı
Patlayıcılar	92.426
Ateşli silahlar	58.524
Tahrik edici	11.135
Meydan kavgası	3.655
Kimyasal	321
Sabotaj Ekipmanı	141
Araç (Otomobil veya kamyon bombaları)	136
Diğer	114
Biyolojik	35
Sahte Silahlar	33
Radyoloji	14

Şekil 4.8'de görüldüğü gibi % 56 oranında patlayıcı madde ile saldırı düzenlenmesi ilk sırada almaktadır. Ateşli silahlar ise % 35 oranında en büyük ikinci saldırı türüdür.



Şekil 4.8. Saldırı türleri oransal dağılımı

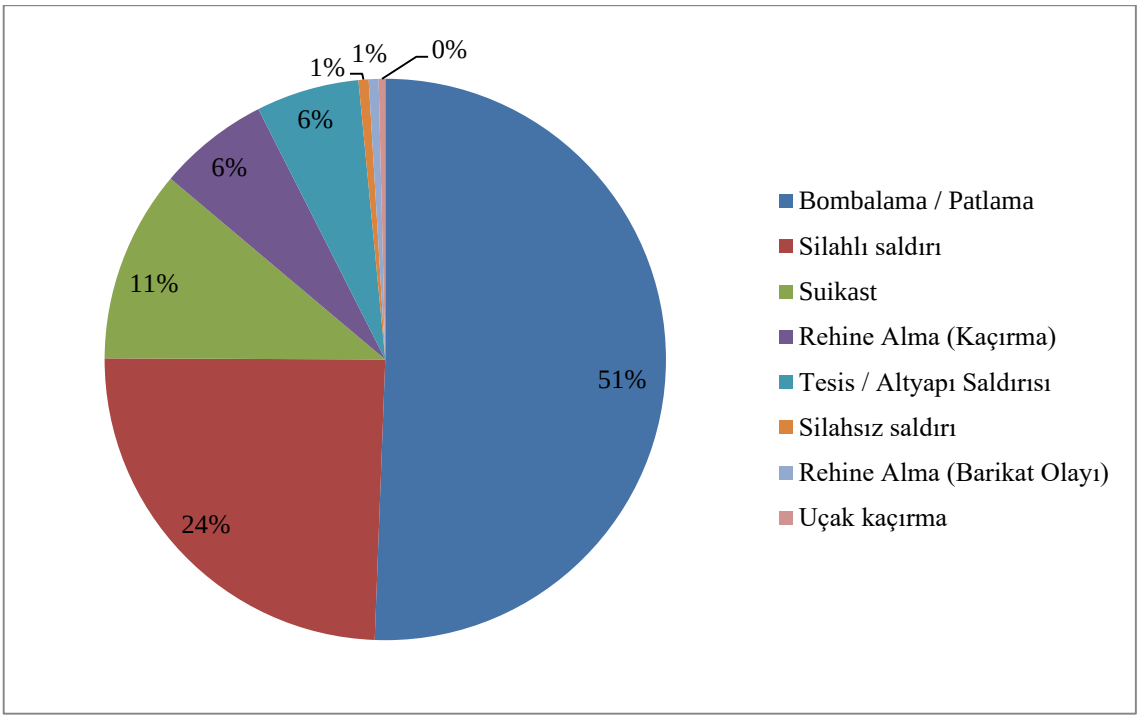
4.2.4. Kullanılan Atak Türleri

Bir terör olayından kullanılan saldırı türü kadar bu saldırıyı nasıl bir yöntemle veya atakla yapmış olması da önemlidir. Atak tipine göre (Tablo 4.8) olay verilerine bakıldığında saldırı türünün bir türevi olan bombalama/patlama ilk sırayı almaktadır. Silahlı saldırı ve suikast yöntemi ise takip etmektedir.

Tablo 4.8. Atak tipleri

Atak tipi	Sayı
Bombalama / Patlama	88.255
Silahlı saldırı	42.669
Suikast	19.312
Rehine Alma (Kaçırma)	11.158
Tesis / Altyapı Saldırısı	10.356
Silahsız saldırı	1.015
Rehine Alma (Barikat Olayı)	991
Uçak kaçırma	659

Aşağıda Şekil 4.9'da görüldüğü gibi % 51 oranında bombalama/patlama ile atak tipi ilk sırada almaktadır. Silahlı saldırı ise % 24 oranında en büyük ikinci atak tipidir.



Şekil 4.9. Atak tipi

4.2.5. Saldırı Hedef Kitlesi

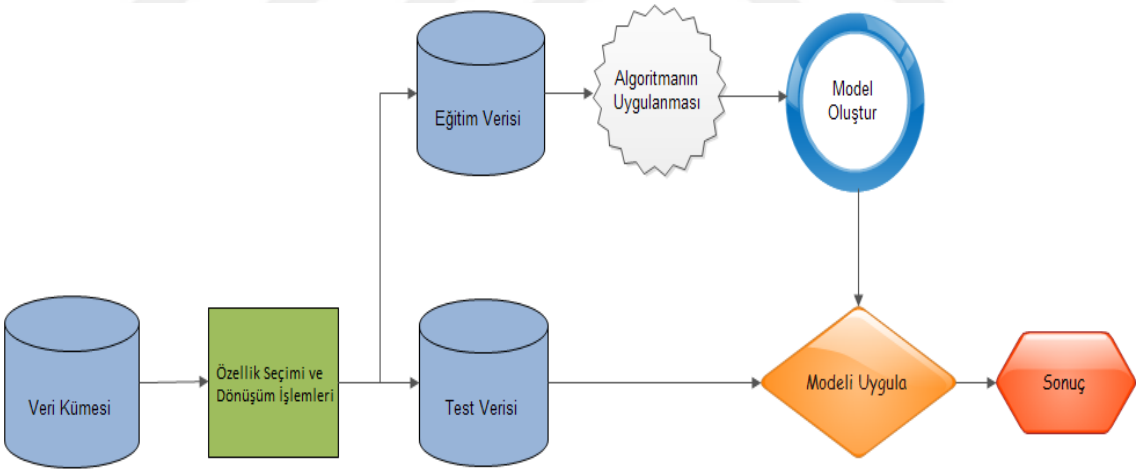
Saldırı tipinden ve silah türünden sonra önemli bir ölçüt olan saldırının hedefinde olan kişi, kurum ve kitlelerdir. Terörist oluşumun saldırısının hedefinde kendi ideolojisine ters durumda kitle bulunabilir. Dünyadaki saldırılardan hedef olarak bulunan kitlelere bakıldığında Tablo 4.9'da görüldüğü gibi ilk sırada siviller bulunmaktadır. Daha sonra ise asker, polis, hükümet görevlileri ve iş insanları hedef olmuştur.

Tablo 4.9. Hedef kitle ve sayısı

Hedef Kitle	**Sayı**
Özel Vatandaşlar ve Mülkiyet	43.511
Askeri	27.984
Polis	24.506
Hükümet (Genel)	21.283
İş	20.669
Taşımacılık	6.799
Araçlar	6.023
Dini Figürler /Kurumlar	4.440
Eğitim Kurumu	4.322
Hükümet (Diplomatik)	3.573

5. GELİŞTİRİLEN UYGULAMA VE SINIFLANDIRMA İŞLEMLERİ

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. Geliştirilen model, algoritmanın hızı, veri kümesini eğitim zamanı bir sonraki başlıklarda belirtilmiştir. Bu tez çalışmasında uygulanan algoritmalar ile model oluşturma aşaması Şekil 5.1'deki gibi gerçekleştirilmiştir.



Şekil 5.1. Uygulamanın çalışma mantığı

5.1. Büyük Veri Kümesinin Sınıflandırma İşlemleri

GTD veri kümesi içerisinde 135 adet nitelik ve nicelik bulunmaktadır. Bu tez kapsamında yaşanan saldırılardan hangi terör örgütünün yaptığını tespit edilmesi sebebiyle hedef değişken veya sınıf olarak Terör Örgütü Adı seçilmiştir. Hedef değişken altında 3500 farklı terör örgütünün ismi yani sınıf bulunmaktadır. Bu sayının yüksek

olması model eğitiminin başarımını azaltmaktadır. Bu sebeple en fazla terör olayını gerçekleştiren örgütler temel alınmıştır. Baz alınırken istatistiki bilgiler incelenmiş olup en fazla saldırı gerçekleştiren ilk 10 terör örgütünün temel alınması uygun bulunmuştur. Veri kümesi indirgenmeden önce 3500 adet sınıf içermektedir. İndirgenen veri setinde 10 adet sınıf bilgisi bulunmaktadır. Sadece 1 tane terörist eylem gerçekleştiren örgütün de var olduğu bu veri tabanında tüm örgütler ile bir model kurulması mümkün değildir. 1 tane terör eylemi gerçekleştiren örgütü de tahmin edecek modelin eğitilmesi için ne eğitim veri kümesi ne de test veri kümesi oluşturulamayacaktır. Ayrıca veri kümesi içerisinde terör saldırılarında bilinmeyenlerin sayısı oldukça fazla görünmektedir. Bu durumlar veri kümesinden temizlendiğinde **36.943** adet terör saldırısı modelin oluşturulması için kullanılmıştır. Bu sebeple, en fazla terör olaylarını gerçekleştiren ilk 10 örgüt ismi modelin oluşturulması aşamalarına dâhil edilmiştir. Özellikler arasından terör saldırısının yaşandığı Ülke ve Bölge, saldırıda kullanılan Silah Türü, saldırının hedefinde bulunan kitle, topluluk veya unsurlar, saldırının türü seçilmiştir. Belirtilen bu nitelik ve niceliklere K-En yakın komşu, Naive Bayes, Rastgele Orman, Karar Ağacı, Destek Vektör Makineleri ve Lojistik Regresyon sınıflandırma algoritmaları uygulanmıştır. Uygulanan algoritmalar sayesinde test veri kümesindeki örgüt isminin tespit edilmesindeki doğruluk oranı bulunmaya çalışılmış ve model oluşturulmuştur.

5.2. Model 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.

5.2.1. 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 5.1'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 5.1. 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 [50].

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 (**5.**1), (**5.**2), (**5.**3) ve (**5.**4) elde edilir.

$$TP = X_{kk} \quad \textbf{(5.1)}$$

$$TN = \sum_{i \in N} X_{ij} \quad \textbf{(5.2)}$$

$$FP = \sum_{i \in N} X_{ik} \quad \textbf{(5.3)}$$

$$FN = \sum_{i \in N} X_{ki} \quad \textbf{(5.4)}$$

5.2.2. Doğruluk

Algoritmanın test veri kümesine uygulanması ile ortaya çıkan doğru tahmin edilenlerin uygulanan test veri kümesine oranıdır. Tersi durum olan yanlış tahmin edilenlerin test verisine oranı da hata oranıdır. Doğruluk değerinin 1'den çıkarılarak da bulunabilmektedir. TP, TN, FP ve FN değerlerine göre ise doğruluk oranı şu şekilde hesaplanmaktadır;

$$\frac{TP+TN}{FP+FN+TP+TN} \quad \textbf{(5.5)}$$

5.2.3. Hata Oranı

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

$$\frac{FP+FN}{FP+FN+TP+TN} \quad \textbf{(5.6)}$$

5.2.4. Hassasiyet

Pozitif durumda olan tahminlerin ne kadar başarılı olduğunu bulan ölçüttür. Denklem (5.7)'de görüldüğü gibi hesaplanmaktadır.

$$\frac{TP}{TP+FN} \quad \textbf{(5.7)}$$

5.2.5. Kesinlik

Doğru ve pozitif olan tahminlerin başarısını gösteren ölçüttür. Denklem (5.8)'de görüldüğü gibi hesaplanmaktadır.

$$\frac{TP}{TP+FP} \quad \textbf{(5.8)}$$

5.2.6. F1 Skoru

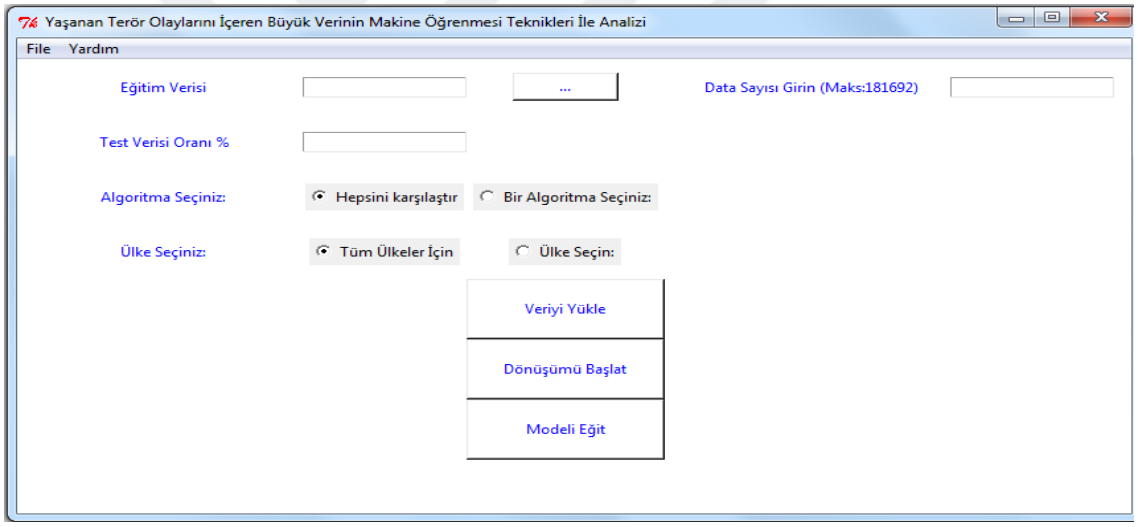
Hassasiyet ve kesinlik ölçütlerinin harmonik ortalamasından elde edilen ölçüttür. Denklem (5.9)'da görüldüğü gibi hesaplanmaktadır.

$$2*\frac{(Precision*Recall)}{(Precision+Recall)} \quad \textbf{(5.9)}$$

5.3. Geliştirilen Uygulama ve Çalışma Adımları

Geliştirilen uygulama ile terör örgütü verilerini içeren CSV uzantılı dosyadan, daha önceden belirlenen öznitelik değerleri kullanılmıştır. Bu bağlamda uygulamanın amacı olarak terör olayını gerçekleştiren örgüt grubunun tahmin edilmesi modellenmiştir. Uygulama, Python programlama dili, makine öğrenmesi ve büyük veri teknolojileri olarak PySpark ve SkLearn kütüphaneleri, grafik ara yüzünün tasarlanması için Tkinter kullanılmıştır. Uygulama yerel bilgisayar üzerinde çalışmaktadır, çalıştırmak için Apache Spark ve Hadoop çatısının kütüphanesi yüklü olması gerekmektedir. Uygulamayı başlatmak için Windows veya Linux sistemlerde konsol açılarak Apache Spark'ın bulunduğu dizine gidilip aşağıdaki parametre girilmesi gerekmektedir. Uygulamayı çalıştırdığında Şekil 5.2 gibi görüntü çıkmaktadır.

spark-submit TEZ_APP.py --master local --driver-memory 32g --executor-memory 32g



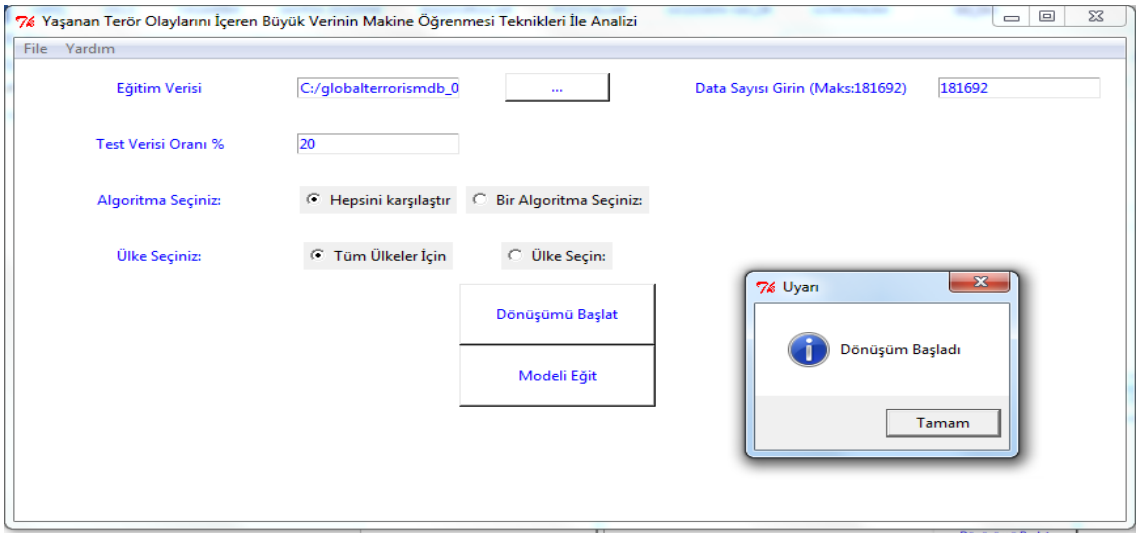
Şekil 5.2. Uygulama ekran görüntüsü

Öncelikle eğitim verisi kısmından dosya seçilmesi gerekmektedir. Dosyanın uzantısı olarak CSV olarak belirtilmiştir. Daha sonra ise veri sayısı kısmından kullanılacak veri miktarı girilmektedir. Burada dikkat edilmesi gereken husus örnek olarak kullanılan veri kümesinin maksimum **181.692** satırdan oluşmaktadır. Burada veri kümesine filtreleme işlemi uygulanarak en fazla eylem gerçekleştiren ilk 10 örgüt, geliştirilen uygulamanın arka planında süzgeçten geçerek ele alınmaktadır. Eğitim kümesi için **29.540**, test kümesi için ise **7.403** adet veri ayrılmıştır. Gerekli girişler yapıldıktan sonra kullanılmak istenen test verisi oranı girilmektedir. Tüm veri giriş işlemleri gerçekleştikten sonra Veri Yükle

düğmesine tıklanarak verinin sisteme girilmesi sağlanmaktadır. Veri girildiğinde sistem otomatik olarak uyarı vermektedir. Şekil 5.3'te veri kümesinin seçimi gösterilmiştir.

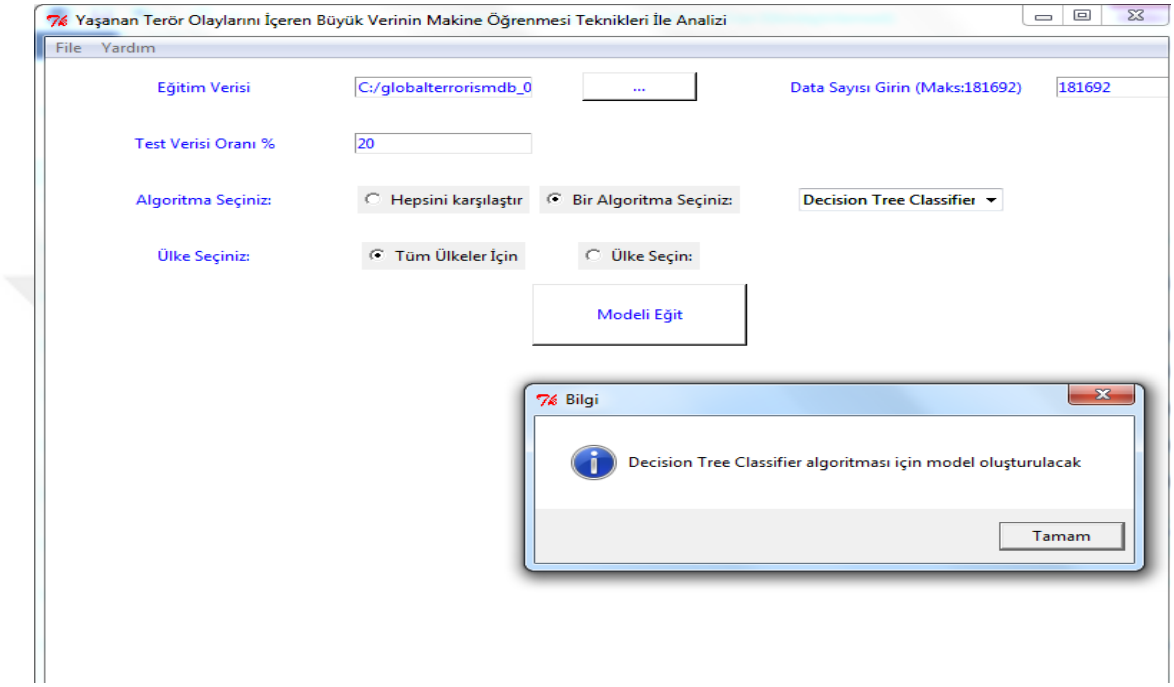
Şekil 5.3. Veri kümesinin seçimi

Dosya yapısının seçiminden sonra ise veri kümesinde bulunan öznitelik değerlerinin ve hedef değişkenin seçiminin arka planda yapılması için Dönüşümü Başlat düğmesine tıklanmalıdır. Burada yapılan işlemler nitelik değerlerinden olan kategorik değişkenlerin (Target Type, Gname) arka planda dönüşümü gerçekleştirilmektedir. Dönüşümlerin tamamlanması için String Indexer, Index To String, One Hot Encoder, Vector Indexer, Standard Scaler, Min-Max Scaler, Vector Assembler algoritmaları kullanılmıştır. Şekil 5.4'te dönüşüm işlemi gösterilmiştir.



Şekil 5.4. Dönüşümün başlatması

Veri dönüşüm işlemleri tamamlandıktan sonra modelin oluşması için algoritma seçimi yapılması gerekmektedir. Algoritma Seçiniz kısmında 6 farklı algoritma veya tüm algoritmaların seçimi seçenekleri sunulmuştur. Örnek olarak Şekil 5.5'te Karar Ağaçları algoritması seçilmiştir. Algoritma seçimi sonrasında Modeli Eğit düğmesinden seçilen algoritmaya göre model oluşturulma işlemi gerçekleştirilmektedir.



Şekil 5.5. Modelin eğitilmesi

Model eğitimi bitiminden sonra sonuçların ekrana yansıtılması için Sonucu Göster düğmesi aktif hale gelmektedir. Düğmeye tıklandığında, ekrana gelen sonuç görüntüleme tablosunda (Şekil 5.6) doğruluk ve hata oranı, F1 skoru, hesaplama zamanı, kesinlik ve duyarlılık değerleri gösterilmektedir. Şekil 5.6'da bu değerlere ilişkin örnek sonuçlar verilmiştir.

Sonuç Görüntüleme

Algoritma	Data Sayisi	Dogruluk Orani	Hata Orani	Hesaplama Süresi (sn.)	F1 Skoru	Kesinlik	Duyarlilik
Karar Agaci (Egitim)	29540	0.860	0.140	0.093	0.839	0.929	0.859
Karar Agaci (Test)	7403	0.859	0.141	0.161	0.840	0.933	0.859

Şekil 5.6. Sonuçların gösterilmesi

5.4. Uygulamanın Çıktıları

Geliştirilen uygulamanın sonucunda çıktıların hesaplanmasında örnek büyük veri kümesi olan GTD'nin 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. Sonraki aşamalarda uygulanan algoritmaların çıktıları tablolar şeklinde gösterilmiştir. Veri kümesinin % 80'lik kısmı yani 29.540 satır eğitim kümesi, % 20'lik kısmı ise test kümesi olarak ayrılmıştır. Tablo 5.2'de belirtilmiştir. Veri kümesinin yüklenmesi aşamasında, modelin başarımının artması ve iyileştirilmesi için yaşanan terör olaylarında ilk 10 örgüt ele alınmıştır.

Tablo 5.2. Eğitim ve test veri kümeleri

Veri Kümesi	Veri Sayısı
Eğitim Veri Kümesi	29.540
Test Veri Kümesi	7.403

Tablo 5.3. Karar ağacı algoritması başarım ölçütleri

Veri Kümesi	Doğruluk Oranı(%)	Hata Oranı(%)	Hesaplama Süresi(sn.)	F1 Skoru	Kesinlik	Duyarlılık
Eğitim	86,0	14,0	0,093	0,839	0,929	0,860
Test	85,9	14,1	0,161	0,840	0,933	0,859

Tablo 5.4. Lojistik regresyon algoritması başarım ölçütleri

Veri Kümesi	Doğruluk Oranı(%)	Hata Oranı(%)	Hesaplama Süresi(sn.)	F1 Skoru	Kesinlik	Duyarlılık
Eğitim	48,9	51,1	0,077	0,383	0,435	0,489
Test	48,8	51,2	0,093	0,384	0,423	0,488

Tablo 5.5. Naive Bayes algoritması başarım ölçütleri

Veri Kümesi	Doğruluk Oranı(%)	Hata Oranı(%)	Hesaplama Süresi(sn.)	F1 Skoru	Kesinlik	Duyarlılık
Eğitim	78,1	21,9	0,092	0,779	0,788	0,781
Test	78,0	22,0	0,141	0,777	0,786	0,780

Tablo 5.6. Rastgele orman algoritması başarım ölçütleri

Veri Kümesi	Doğruluk Oranı(%)	Hata Oranı(%)	Hesaplama Süresi(sn.)	F1 Skoru	Kesinlik	Duyarlılık
Eğitim	88,4	11,6	0,117	0,873	0,906	0,884
Test	88,2	11,8	0,115	0,871	0,903	0,882

Tablo 5.7. K-En yakın komşu algoritması başarım ölçütleri

Veri Kümesi	Doğruluk Oranı(%)	Hata Oranı(%)	Hesaplama Süresi(sn.)	F1 Skoru	Kesinlik	Duyarlılık
Eğitim	98,5	1,5	0,749	0,985	0,986	0,985
Test	98,2	1,8	0,187	0,982	0,983	0,982

Tablo 5.8. Destek vektör makineleri algoritması başarım ölçütleri

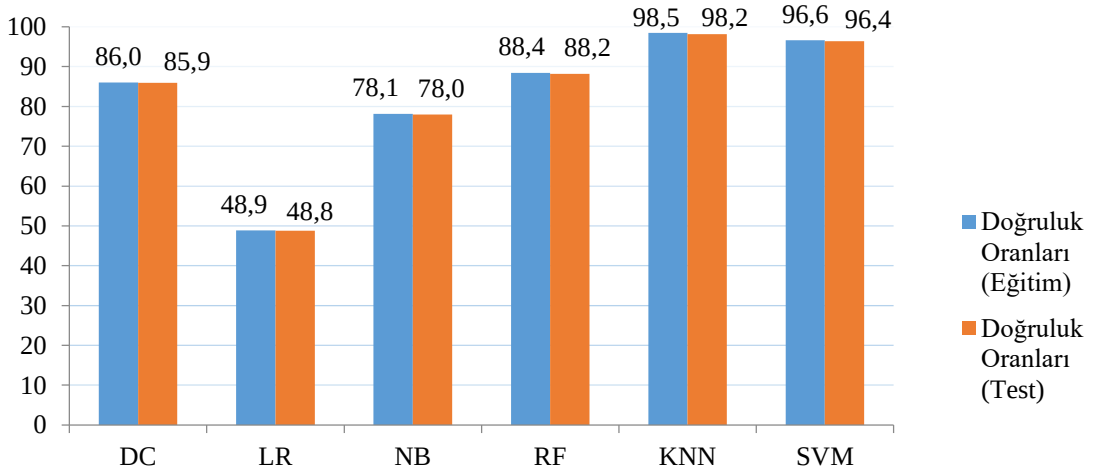
Veri Kümesi	Doğruluk Oranı(%)	Hata Oranı(%)	Hesaplama Süresi(sn.)	F1 Skoru	Kesinlik	Duyarlılık
Eğitim	96,6	3,4	1,888	0,967	0,972	0,966
Test	96,4	3,6	0,467	0,965	0,971	0,964

Eğitim ve test kümesi ait doğruluk oranları yukarıdaki tablolarda verilmiştir. Bu tablolara bakıldığında genel olarak sırasıyla Eğitim ve Test veri kümesi için; Tablo 5.3'te Karar Ağacı algoritması % 86 ve % 85,9; Tablo 5.4'te Lojistik Regresyon algoritması % 48,9 ve % 48,8; Tablo 5.5'te Naive Bayes algoritması % 78,1 ve % 78,0; Tablo 5.6'da Rastgele Orman sınıflandırıcı % 88,4 ve % 88,2; Tablo 5.7'de K-En yakın komşu algoritması % 98,5 ve % 98,2; Tablo 5.8'de Destek vektör makineleri % 96,6 ve % 96,4 olduğu görülmüştür. Tablo 5.7'de belirtilen K-En yakın komşu algoritması ile oluşturulan modelin tahmin sonuçlarının %98,2 ile en yüksek değere sahip olduğu görülmüştür. Tablo 5.9'da algoritma çalışma süreleri gösterilmiştir.

Tablo 5.9. Algoritma çalışma süreleri

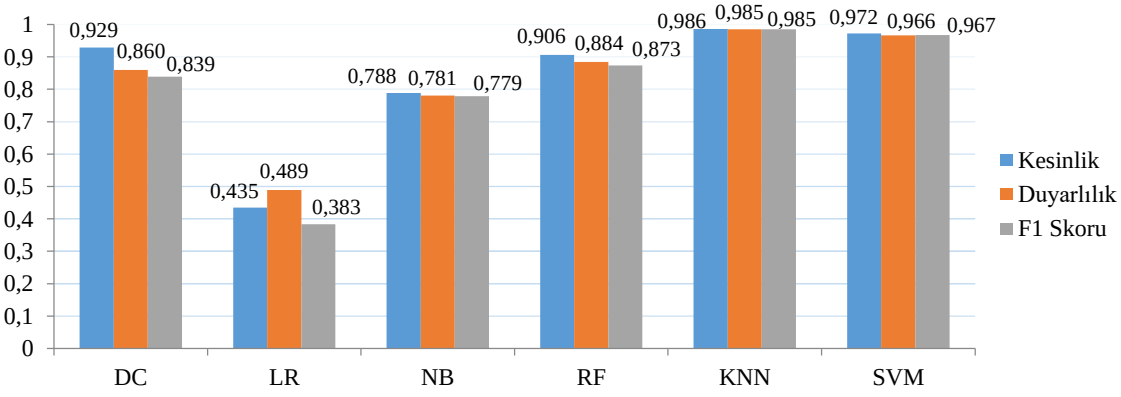
Uygulanan Algoritma	Çalışma Süresi (sn.)
Karar Ağaçları	5,801
Lojistik Regresyon	4,151
Naive Bayes	2,262
Rastgele Orman	11,103
K-En Yakın Komşu	0,125
Destek Vektör Makineleri	0,889

Doğruluk oranı, F1 skoru, kesinlik ve duyarlılık değerleri grafiksel biçimde gösterilerek sonuçlar daha net bir şekilde görülmektedir. Yukarıdaki tablolara bağlı olarak Şekil 5.7'de eğitim ve test kümesi için algoritmaların doğruluk ölçütleri gösterilmiştir. Şekil 5.8'de eğitim kümesi ve Şekil 5.9'da test kümesi için F1 skoru, kesinlik ve duyarlılık ölçütleri grafiksel biçimde gösterilmiştir. Eğitim ve test kümeleri için hesaplama süreleri bir diğer anlamıyla tahmin etme (Predict time) süreleri Şekil 5.10 ve Şekil 5.11'de grafiksel biçimde gösterilmiştir. Algoritmaların çalışma (Model'i üretme / oluşturma) süreleri Şekil 5.12'de gösterilmiştir.

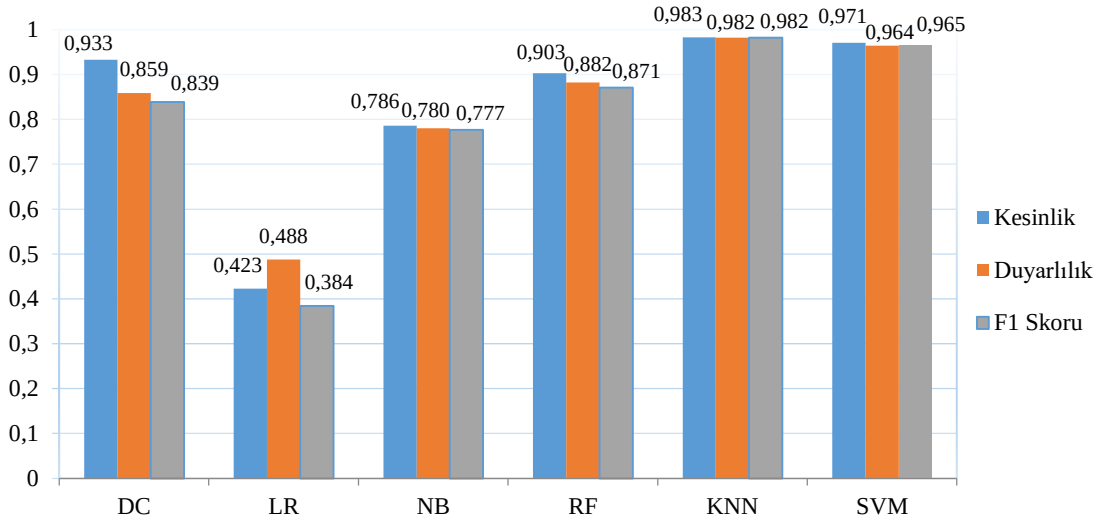


Şekil 5.7. Eğitim ve test kümesi için doğruluk oranları

Şekil 5.7'ye bakıldığında K-En Yakın Komşu algoritması doğruluk oranı açısından ilk sırada yer alarak örnek veri kümesi için başarılı olduğu görülmektedir. Lojistik regresyon sınıflandırıcı algoritması ise örnek veri kümesinin eğitim ve test kısmında, doğruluk açısından en düşük orana sahiptir.

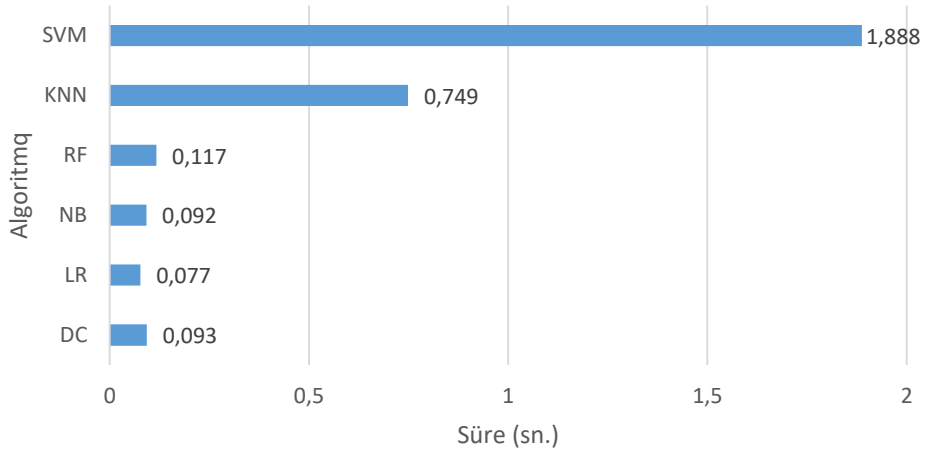


Şekil 5.8. Eğitim kümesi için çıktıların grafiksel gösterimi



Şekil 5.9. Test kümesi için çıktıların grafiksel gösterimi

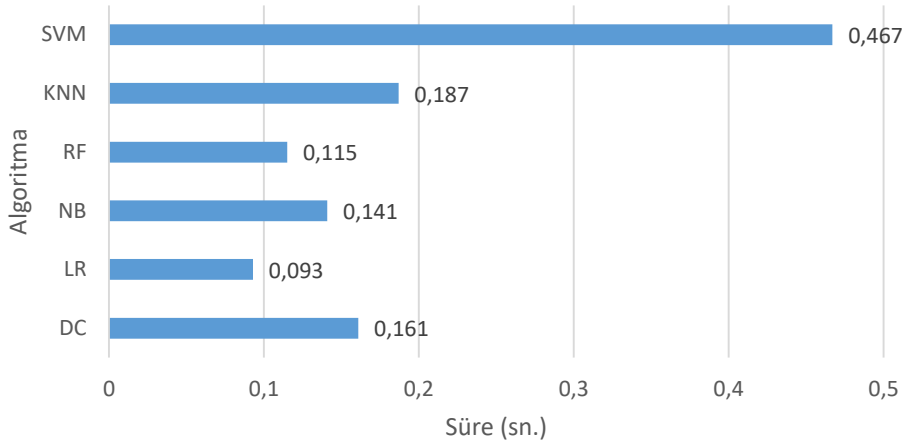
Şekil 5.8 ve Şekil 5.9'a bakıldığında K-En Yakın Komşu algoritması kesinlik, duyarlılık ve F1 skoru açısından ilk sırada yer alarak örnek veri kümesi için başarılı olduğu görülmektedir. Lojistik regresyon sınıflandırıcı algoritması ise örnek veri kümesinin eğitim ve test kısmında, kesinlik, duyarlılık ve F1 skoru açısından en düşük orana sahiptir.



Şekil 5.10. Eğitim kümesi hesaplama süreleri grafiksel gösterimi

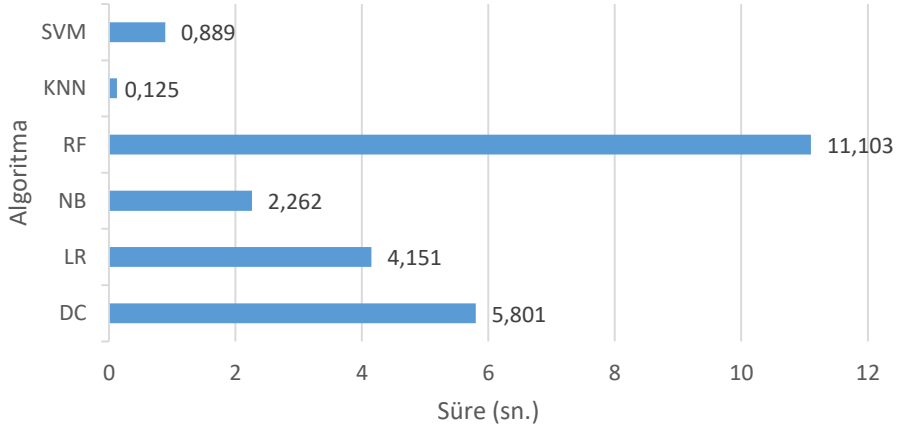
Şekil 5.10'a bakıldığında ise Destek Vektör Makineleri hesaplama süresi açısından 1,888 sn. ile ilk sırada yer alarak eğitim veri kümesi için uzun sürede tahmin ettiği görülmektedir. Lojistik Regresyon sınıflandırıcı algoritması ise eğitim veri kümesinde

hesaplama süresi açısından en düşük orana sahiptir. Lojistik Regresyon, bu veri kümesi için hızlı tahmin etme yeteneğine sahip olduğu görülmektedir.



Şekil 5.11. Test kümesi hesaplama süreleri grafiksel gösterimi

Şekil 5.11'e bakıldığında eğitim veri kümesinde olduğu gibi Destek Vektör Makineleri hesaplama süresi açısından 0,467 sn. ile ilk sırada yer alarak test kümesi için uzun sürede tahmin ettiği görülmektedir. Lojistik Regresyon ise test veri kümesinde hesaplama süresi açısından en düşük orana sahiptir ve test veri kümesi için hızlı tahmin etme yeteneğine sahip olduğu görülmektedir.



Şekil 5.12. Algoritma çalışma süreleri grafiksel gösterimi

Çalışmada kullanılan büyük veri kümesine uygulanan sınıflandırma algoritmalarının uygulanması ile çalışma süreleri (Runtime) ortaya çıkmıştır. Şekil 5.12'ye bakıldığında

bu veri kümesi için Rastgele Orman algoritması 11,103 sn. ile en uzun çalışma süresine ulaşan algoritma olmuştur. K-En Yakın Komşu algoritması ise bu veri kümesi için 0,125 sn. ile en hızlı çalışan algoritma olmuştur. Bir algoritmanın çalışma süresini tespit edebilmek için üzerinde çalışılan veri kümesi, kullanılan sistemin işlemci ve bellek kapasitesi kaynakları önemli olmaktadır. Bu tabloda bulunan süre değerleri her sistemde aynı sonucu vermemektedir. Kullanılan sistemin yetkinliğine ve performansına bağlı olarak çalışma sürelerinin azalması veya artması söz konusu olabilmektedir.

5.5. Sonuçlar ve Değerlendirme

Çalışma süreleri bir algoritmanın hız bakımından bir ölçüsünü ifade etmektedir. Tablo 5.9'daki sonuçlara bakıldığında 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.

Hesaplama veya *Tahmin Zamanı (Predict Time)*, bir algoritmanın model üzerinde üretilen değerleri ne kadar sürede tahmin ettiğini gösteren bir ölçüyü ifade etmektedir. Tablodaki çıktılara bakıldığında Lojistik Regresyon algoritmasının eğitim veri kümesine uygulandığında hesaplama süreleri saniye cinsinden 0,077 ve test veri kümesine uygulandığında ise 0,093 olarak en iyi sonucu verdiği görülmüştür. Lojistik Regresyon algoritmasını eğitim veri kümesinde hesaplama süresi bakımından sırasıyla Naive Bayes, Karar Ağacı, Rastgele Orman, K-En Yakın Komşu ve Destek Vektör Makineleri algoritmaları takip etmiştir. Lojistik Regresyon algoritmasını test veri kümesinde hesaplama süresi bakımından sırasıyla Rastgele Orman, Naive Bayes, Karar Ağacı, K-En Yakın Komşu ve Destek Vektör Makineleri 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 eğitim ve test veri kümesindeki çıktısına göre 0,986 ve 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 eğitim ve test veri kümesindeki çıktısına göre 0,985 ve 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 eğitim ve test veri kümesindeki çıktısına göre 0,985 ve 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.

Aşağıda algoritmaların karışıklık matrisi verilmiştir. Tablo 5.10'a bakıldığında Karar Ağacı algoritması için EŞ (El-Şebab) terör örgütü % 39,0 ile en az tahmin yürütüldüğü tespit edilmiştir. En yüksek doğruluk oranı ise % 99,9 ile Taliban örgütü bulunmuştur. Tablo 5.11'de Lojistik Regresyon için PKK, Boko Haram, NPA ve FARC terör örgütlerinin tahmin edilemediği tespit edilmiştir. En yüksek doğruluk oranı ise % 99,8 ile IRA örgütü bulunmuştur. Tablo 5.12'de Naive Bayes için FARC terör örgütü % 57,0 ile en az tahmin yürütülmüştür. En yüksek doğruluk oranı ise % 99,2 ile Taliban örgütü bulunmuştur. Tablo 5.13'de Rastgele Orman için DEAŞ terör örgütü % 69,0 ile en az tahmin yürütülmüştür. En yüksek doğruluk oranı ise % 99,9 ile PKK terör örgütü bulunmuştur. Tablo 5.14'de K-En Yakın Komşu algoritması için NPA terör örgütü % 92,0 ile en az tahmin yürütülmüştür. En yüksek doğruluk oranı ise % 99,9 ile IRA örgütü bulunmuştur. Tablo 5.15'de Destek Vektör Makineleri için NPA terör örgütü % 76,0 ile en az tahmin yürütülmüştür. En yüksek doğruluk oranı ise % 99,9 ile PKK terör örgütü bulunmuştur.

Tablo 5.10. DC 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 5.11. 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 5.12. 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 5.13. RF 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 5.14. 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 5.15. SVM 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

6. SONUÇ

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.

Uygulamanın çıktıları ve sonuçları tablolar şeklinde önceki bölümde belirtilmiştir. Gerçekleştirilen uygulama ile K-En Yakın Komşu algoritmasının eğitim setin üzerinde terör örgütünü tahmin etmedeki doğruluk oranı % 98,5 test verisi üzerindeki doğruluk oranı ise % 98,2 olarak hesaplanmıştır. Bu değerlendirme sırasıyla Destek Vektör Makineleri, Rastgele Orman, Karar Ağacı, Naive Bayes ve Lojistik Regresyon algoritmaları için de gerçekleştirilmiştir. Sonuçlar karşılaştırıldığında en uygun algoritmanın K-En Yakın Komşu olduğu görülmüştür.

Bu tez 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. Hesaplama süresine bakılacak olursa Lojistik Regresyon ön plana çıkmaktadır. Gerçek sistem üzerinde modelin test verisine uygulandığında (% 98,2-% 96,4 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 Şekil 5.7'de ve Rastgele Orman algoritmasının çalışma süresi bakımından uygun olmadığı Şekil 5.12'de görülmektedir.

Büyük veri kümesini geleneksel yöntemler ile analiz etme işleminin gerçekleştirilmesi ciddi zaman kaybına neden olabilmektedir. Bu zaman kaybının önüne geçilmesi için büyük verilere uygun geliştirilmiş veri analizi ve işlenmesinde kullanılan araçlardan olan açık kaynak kodlu ve ücretsiz olan Apache Spark tercih edilmiştir. Bu çalışmada Apache Spark ile birlikte örnek bir veri kümesi üzerinde analiz ve örnek model oluşturma uygulamaları gerçekleştirilmiştir ve karşılaştırmaları doğruluk, hesaplama süreleri, F1 skoru, duyarlılık ve kesinlik gibi başarım ölçütlerine göre yapılmıştır. Çalışmada yazılım geliştirme aracı olarak Python (IDLE, PyCharm), Apache Spark, scikit-learn, MLlib ve sınıflandırma fonksiyonları gibi araçlar tercih edilmiştir. Bu araçlar geliştirilen uygulamalara uyum sağlayabilmesi açısından faydalı araçlardır. Geliştirilen uygulamalarda bu teknolojiler veri kümesi ve niteliklerine uygun olarak modifiye edilmiştir.

Geliştirilen bu uygulama ile birlikte, gerçekleşen bir terör saldırısının hangi terör örgütün yaptığını ve saldırı tipi veya kimliği belirlenen saldırganın hangi terör örgütüne mensup olduğuna dair tahminler üreten bir karar destek sistemi ve Apache Spark üzerinde büyük veri işleme aracı ortaya koyulmuştur. Ortaya koyulan karar destek sistemi ile test ortamında % 98,2 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.

KAYNAKÇA

[1] **Demirli, A.** (2011). Terörizm, psiko sosyal etkileri ve müdahale modelleri. Türk Psikolojik Danışma ve Rehberlik Dergisi, 4(35), 66-76.

[2] **Vajihala, N. R., Strang, K. D., & Sun, Z.** (2015, August). Statistical modeling and visualizing open big data using a terrorism case study. In *2015 3rd International Conference on Future Internet of Things and Cloud* (pp. 489-496). IEEE.

[3] **Overview of the GTD** (2019), (online), Available: http://www.start.umd.edu/gtd/about/

[4] **LaFree, G.** (2011). *Building a global terrorism database*. DIANE Publishing.

[5] **LaFree, G., Dugan, L., & Miller, E.** (2014). *Putting terrorism in context: Lessons from the Global Terrorism Database*. Routledge.

[6] **Strang, K. D., & Sun, Z.** (2017). Analyzing relationships in terrorism big data using Hadoop and statistics. *Journal of Computer Information Systems*, *57*(1), 67-75.

[7] **Khorshid, M., Abou-El-Enien, T., & Soliman, G.** (2015). A comparison among support vector machine and other machine learning classification algorithms. *IPASJ International Journal of Computer Science*, *3*(5), 26-35.

[8] **Vapnik, V.** (1963). Pattern recognition using generalized portrait method. Automation and remote control, 24, 774-780.

[9] **Mashey, J. R.** (1999, June). Big Data and the Next Wave of InfraStress: Problems, Solutions, Opportunities. In 1999 USENIX Annual Technical Conference.

[10] **Dowe, D. L.** (2013). Introduction to Ray Solomonoff 85th memorial conference. In *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence* (pp. 1-36). Springer, Berlin, Heidelberg.

[11] **Russom, P.** (2011). Big data analytics. *TDWI best practices report, fourth quarter*, *19*(4), 1-34.

[12] **Manyika, J.** (2011). Big data: The next frontier for innovation, competition, and productivity. *http://www. mckinsey. com/Insights/MGI/Research/Technology_and_Innovation/Big_data_The_next_frontier_for_innovation*.

[13] **Demchenko, Y.** (2013). Defining the big data architecture framework (BDAF). *Outcome of the Brainstorming Session at the University of Amsterdam. SNE Group. University of Amsterdam, Amsterdam*.

[14] **Lämmel, R.** (2008). Google's MapReduce programming model—Revisited. Science of computer programming, 70(1), 1-30.

[15] **Sanjay, G.** (2003). The Google file system. In SOSP'03: Proceedings of the nineteenth ACM symposium on Operating systems principles (pp. 29-43). ACM Press.

[16] **Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., ... & Gruber, R. E.** (2008). Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, *26*(2), 4.

[17] **What are business intelligence tools?** (2019), (online), Available: https://azure.microsoft.com/tr-tr/overview/what-are-business-intelligence-tools/

[18] **Berkowitz, B. T., Simhadri, S., Christofferson, P. A., & Mein, G.** (2002). U.S. Patent No. 6,457,021. Washington, DC: U.S. Patent and Trademark Office.

[19] **Han, J., Haihong, E., Le, G., & Du, J.** (2011, October). Survey on NoSQL database. In *2011 6th international conference on pervasive computing and applications* (pp. 363-366). IEEE.

[20] **Waage, T., Jhajj, R. S., & Wiese, L.** (2015, October). Searchable encryption in apache cassandra. In *International Symposium on Foundations and Practice of Security* (pp. 286-293). Springer, Cham.

[21] **Banker, K.** (2011). *MongoDB in action*. Manning Publications Co.

[22] **Vogels, W.** (2012). Amazon dynamodb–a fast and scalable nosql database service designed for internet scale applications. *Retrieved July*, *30*, 2012.

[23] **Shvachko, K., Kuang, H., Radia, S., & Chansler, R.** (2010, May). The hadoop distributed file system. In *MSST* (Vol. 10, pp. 1-10).

[24] **Murthy, A. C., Vavilapalli, V. K., & Eadline, D.** (2014). *Apache Hadoop YARN: moving beyond MapReduce and batch processing with Apache Hadoop 2*. Pearson Education.

[25] **Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., ... & Xin, D.** (2016). Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research*, *17*(1), 1235-1241.

[26] **Huai, Y., Chauhan, A., Gates, A., Hagleitner, G., Hanson, E. N., O'Malley, O., ... & Zhang, X.** (2014, June). Major technical advancements in apache hive. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* (pp. 1235-1246). ACM.

[27] **Fuad, A., Erwin, A., & Ipung, H. P.** (2014, September). Processing performance on apache pig, apache hive and MySQL cluster. In *Proceedings of International Conference on Information, Communication Technology and System (ICTS) 2014* (pp. 297-302). IEEE.

[28] **Storm, A.** (2013). Storm, distributed and fault-tolerant realtime computation. *Google Scholar*.

[29] **Küçüksille, E., Özger, F., & Genç, S.** (2013). Mobil bulut bilişim ve geleceği. *Akademik Bilişim Konferansı Bildirileri*, 23-25.

[30] **Gormley, C., & Tong, Z.** (2015). *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.".

[31] **Smola, O., & Žďánský, J.**(2013). Škálovatelný distribuovaný systém pro detekci podobných dokumentů.

[32] **Islam, M. K., & Srinivasan, A.** (2015). *Apache Oozie: The Workflow Scheduler for Hadoop*. " O'Reilly Media, Inc.".

[33] **Dobbelaere, P., & Esmaili, K. S.** (2017, June). Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations: Industry Paper. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems* (pp. 227-238). ACM.

[34] **Varghese, L., MH, S., & Jacob, K. P.** (2017). Spam: A Big Data Challenge. *International Journal of Advanced Research in Computer Science*, *8*(1).

[35] **Ting, K., & Cecho, J. J.** (2013). *Apache Sqoop Cookbook: Unlocking Hadoop for Your Relational Database*. " O'Reilly Media, Inc.".

[36] **Hoffman, S.** (2013). *Apache Flume: distributed log collection for Hadoop*. Packt Publishing Ltd.

[37] **Vora, M. N.** (2011, December). Hadoop-HBase for large-scale data. In *Proceedings of 2011 International Conference on Computer Science and Network Technology* (Vol. 1, pp. 601-605). IEEE.

[38] **Alsheikh, M. A., Niyato, D., Lin, S., Tan, H. P., & Han, Z.** (2016). Mobile big data analytics using deep learning and apache spark. *IEEE network*, *30*(3), 22-29.

[39] **Lutz, M.** (2010). *Programming Python: powerful object-oriented programming*. " O'Reilly Media, Inc."

[40] **Damji, J.** (2016). A tale of three apache spark apis: Rdds, dataframes, and datasets. *URL https://databricks. com/blog/2016/07/14/a-tale-of-three-apache-spark-apis-rdds-dataframes-and-datasets. html*.

[41] **Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., ... & Stoica, I.** (2012, April). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation* (pp. 2-2). USENIX Association.

[42] **Kudyba, S.**(2014). "Managing Data Mining", CyberTech Publishing, 146-163.

[43] **Han, J., Pei, J., & Kamber, M.** (2011). *Data mining: concepts and techniques*. Elsevier.

[44] **Özekes, S.** (2003). Veri madenciliği modelleri ve uygulama alanları.

[45] **Diri, B.** (2012). Makine Öğrenmesine Giriş (Machine Learning–ML).

[46] **Zortuk, M., Koç, E., & Bayrak, S**. (2014). HANEHALKLARI SATIN ALMA KRİTERLERİNİN ANALİZİ: MULTİNOMİNAL LOJİSTİK REGRESYON YAKLAŞIMI. *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi*, 163-176.

[47] **Guyon, I., Weston, J., Barnhill, S.**, &**Vapnik, V.** (2002). Gene selection for cancer classification using support vector machines. Machine learning, 46(1-3), 389-422.

[48] **Spark, Apache**. Extracting, transforming and selecting features, Spark 2.2.0 Documentation. URL: https://spark.apache.org/docs/2.2.0/ml-features.html,

[49] **Krüger, F.**(2016). Activity, Context, and Plan Recognition with Computational Causal Behaviour Models *(Doctoral dissertation, University).*

ÖZGEÇMİŞ

1991 yılında Manisa'nın Soma ilçesinde doğdu. İlk ve orta eğitimini Soma Canlar İlköğretim okulunda tamamladı. Soma Teknik Lisesi Bilişim Teknolojileri bölümü ağ işletmenliği alanından 2009 yılında mezun olduktan sonra Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliğinden 2015 yılında mezun oldu. Lisans eğitim döneminde yaz stajlarını sırasıyla 2013 yılında Kentkart (İzmir), 2014 yılında HAVELSAN(Ankara)firmalarında ve 2015 yılında ise iş yeri stajını Sanrobo Bilişim (Elazığ) firmasında tamamladı.

2015 Temmuzunda İzmir'de bulunan Kuasar Teknoloji firmasında yazılım geliştirici olarak çalıştı. 2016 yılının Şubat ayında Fırat Üniversitesi Fen Bilimleri Enstitüsü Yazılım Mühendisliği Anabilim dalında yüksek lisans eğitimine başladı.2017 yılından itibaren Adalet Bakanlığı Bilgi İşlem Genel Müdürlüğünde yazılım mühendisi olarak çalışmaktadır. Temel araştırmalarını makine öğrenmesi, derin öğrenme ve büyük veri alanlarında yapmaktadır.

Tezin çalışmaları kapsamında "Example of Log Analysis with Apache Spark" ve "Comparison of Commonly Used Tools in Big Data Processing" adlı çalışmalar bildiri olarak 8TH International Advanced Technologies Symposium, 2017 'de sunulmuştur.