

**Ağ Anomalisi Tespiti İçin Ağ Trafiği Verilerini Scapy ile Alan ve Makine Öğrenimi Yaklaşımını Kullanan Streamlit Arayüzüne Sahip Saldırı Tespit Sisteminin Geliştirilmesi**

**Özet**

Bu çalışma, ağ trafiği analizinde ağ anomalilerini tespit etmek için makine öğrenimi algoritmalarını kullanan bir Streamlit arayüzünün geliştirilmesini ele almaktadır. Geliştirilen arayüz, kullanıcıların ağ verilerini kolayca analiz etmelerini ve potansiyel tehditleri tespit etmelerini sağlamaktadır. Bu sayede, ağ güvenliğinin arttırılması hedeflenmektedir.

**Anahtar Kelimeler: Ağ Trafiği Analizi, Saldırı Tespit Sistemi, Makine Öğrenimi**

**Hazırlayanlar**

Meryem Arpacı

Zehra Gündüz

Ayşe Sıla İnci

Elif Feyza Ünal

**Danışman**

Özal Yıldırım

**Kurum/Bölüm**

Fırat Üniversitesi Yapay Zeka ve Veri Mühendisliği

**Tarih**

**11.06.2024**

**1. Giriş**

Nesnelerin İnterneti ve 5G gibi teknolojilerin ortaya çıkmasıyla, internete bağlı kullanıcı ve cihaz sayısında büyük bir artış görülmüştür.İnsanların konforunun artırılması amacıyla gündelik hayatta ihtiyaç duyulan birçok hizmet internet üzerinden sağlanmaya başlamış ve hatta kullanılan araç-gereçler de internet üzerinden kontrol edilmeye başlanmıştır. İnternet ortamının sağlamış olduğu bu imkanlar daha çok kullanıcının daha yoğun şekilde kullanmasına sebep olmuştur. Bu durum, internet trafiğinde muazzam bir artışa sebep olmuştur.Artık sadece insanlar değil, evlerimizdeki akıllı süpürgeler, buzdolapları, kombiler ve kameralar gibi cihazlar da internete bağlanarak otonom bir şekilde veri akışı sağlamaktadır.Artan internet trafiği siber güvenlik riskleri, siber saldırılar,veri mahremiyeti endişeleri gibi problemlere yol açmaktadır. İnternet trafiği bağlamında, internette akan bu saldırılara ait veriler anomali olarak bilinmektedir.

Anomali Tespiti, normal ve beklenen veri davranışının standart ve iyi tanımlanmış özelliklerine uymayan olağandışı modelleri ve gözlemleri tanımlamak için kullanılan tekniği ifade eder. İzinsiz giriş tespiti, sahtekarlık tespiti, arıza tespiti, sağlık izleme sistemleri ve sensör ağlarında olay yönetim sistemleri dahil olmak üzere birçok uygulama alanına sahiptir. Bu beklenmedik veya olağandışı gözlemler, farklı alanlarda farklı terminolojilere sahiptir, ancak genellikle anomaliler olarak bilinir .[1]

İnternet ortamında, ağa yönelik saldırı risklerinden dolayı, internet tabanlı saldırıları engellemek amacıyla güvenlik duvarı, antivirüs ya da saldırı tespit sistemleri gibi yazılımsal veya donanımsal araçlar tasarlanmıştır. Ağ sistemi, saldırganlardan ya da hackerlerden önemli verileri ve sistemleri korumak için bu güvenlik yazılımlarından bir ya da birkaçını kullanmaktadır. Tek başına bir güvenlik duvarı sistemine güvenmek, kurumsal ağlara ya da kişisel ağlara yönelik saldırıları engellemek için yeterli değildir. Bu nedenle, güvenlik duvarının açıklarını kapatmak için aynı zamanda saldırı tespit sistemi de kullanılır.[2]

Saldırı Tespit Sistemleri (IDS) de Bilgi Güvenliğinin sağlanması amacıyla kullanılan ve günden güne geliştirilmeye de devam edilen önemli araçlardan biridir. Saldırı tespit sistemleri, tüm tedbirlere karşın bilgisayar sistemlerine yapılan saldırıları gerçekleşirken ya da gerçekleştikten sonra tespit etmek, İnternet veya yerel ağdan gelebilecek, ağdaki sistemlere zarar verebilecek, çeşitli paket ve verilerden oluşan bu saldırıları fark etmek üzere tasarlanmış sistemlerdir ve bu saldırılara yanıt vermeyi amaçlayan bir güvenlik teknolojisidir.

IDS, bir bilgisayar ağında meydana gelen tüm olayları izler ve analiz eder, izinsiz girişleri tespit eder ve güvenlik sorunlarına dair işaretler arar. IDS uygulaması, ağdaki tüm olayları izlemek için ağ tabanlı olabilir veya belirli bir bilgisayarda meydana gelen tüm olayları kaydetmek için bilgisayara dayalı olarak kullanılabilir.[3] IDS, yalnızca tehditleri tespit etmek için çalışır.Tespit edilen herhangi bir aktivite veya ihlal bir yöneticiye bildirilir veya bir güvenlik ve olay yönetimi (SIEM) sistemi kullanılarak merkezi olarak toplanır. SIEM sistemlerinde farklı kaynaklardan gelen verileri birleştirir ve filtreleme yaparak kötü niyetli alarmın yanlış alarm olup olmadığını denetler ve ayırır.

Güvenlikte oluşan açıklar yüzünden gerçekleşen saldırıların engellenebilmesi için çeşitli çalışmalar yapılmaktadır. Bu çalışmalarda izinsiz girişlerin tespit edilmesi ve anomalilerin analiz edilmesi için birçok yöntem önerilmiştir (Kasongo ve Sun, 2020).[4]

Artan veri hacmi ve karmaşıklık, IDS'lerin performansını olumsuz etkilemektedir. Yanlış alarmlar da ayrı bir sorundur ve ağ yöneticilerinin iş yükünü artırır. Bu zorlukları aşmak için birçok yöntem vardır..Bunlardan biri makine öğrenmesi (ML) yöntemidir.Son yıllarda, Saldırı Tespit Sistemlerinde makine öğrenmesi algoritmalarının kullanımı artmaktadır

Yapılan bu çalışmada Makine öğrenmesi tabanlı bir anomali tespit yaklaşımı kullanarak ağ saldırılarının tespitinde daha yüksek başarı oranları elde etme ve yeni saldırı türlerini daha kolay bir şekilde tanımlama amaçlanmıştır.

Çalışmamızda Saldırı Tespit Sistemlerinde sıkça kullanılan veri kümelerinden Network Intrusion Detection kullanılmıştır. Denetime tabi tutulacak veri seti, askeri bir ağ ortamında simüle edilmiş çok çeşitli saldırılardan oluşmaktadır. Bu veri seti, tipik bir ABD Hava Kuvvetleri LAN'ını simüle ederek bir ağ için ham TCP/IP döküm verilerini elde etmek için bir ortam oluşturmuştur. LAN gerçek bir ortam gibi odaklanmış ve birden fazla saldırı ile vurulmuştur. Veri seti, her biri yaklaşık 100 bayt boyutunda olan bağlantı kayıtlarından oluşmaktadır. Her bağlantı kaydı, belirli bir süre boyunca bir kaynak IP adresinden bir hedef IP adresine veri akışını temsil eden bir dizi TCP paketinden oluşur. Her bağlantı, "normal" veya belirli bir saldırı türüyle ilişkilendirilen "saldırı" olarak etiketlenmiştir. Bu veri seti kullanılarak güncel ağ saldırılarının yüksek doğrulukla tespit edilmesi amaçlanmıştır.

**2. Literatür Taraması**

Ağ saldırı tespiti, özellikle anomali tespiti, alanında yapılan çalışmaları daha iyi anlayabilmek için bu çalışma süresince literatürde bulunan çalışmalar incelenmiştir.

İlk saldırı tespit sistemi yaklaşımının 1980'lerde bilgisayar sistemlerinde yetkiliz erişimleri tespit etmek amacıyla Doroty Denning tarafından ortaya çıkması ile birlikte çürümeye kadar IDS üzerinde birçok araştırma ve yeni ve farklı yaklaşımlar gerçekleştirilmiştir.[5] Literatürde trafik ağlarında anomali tespiti için farklı çalışmalar yapılmıştır. Chen vd. (2009), hız, yoğunluk, işgal süresi ve zaman aralığı verilerini kullanarak trafik ağlarındaki olay anomalilerinin tespiti için bir yöntem önermiştir. Bu çalışmada %80 başarı elde edilmiştir.[6] Lu vd., (2014), ortalama hız, yoğunluk ve işgal süresi verileri ile trafik ağlarındaki anormal durumları tespit etmişlerdir. Bu çalışmada %98 başarı elde edilmiştir.[7] Injadat vd.,anormallik tespiti için etkili bir çerçeve önermiştir. Rastgele Değişken (RD), DVM ve K-EYK makine öğrenmesi yöntemleri sınıflandırma için kullanılmıştır. DVM, RD ve K-EYK parametrelerini ayarlamak için Bayes Optimizasyon tekniği kullanılmıştır.[8] Hassan vd. yüksek doğruluk değerlerine sahip ağ saldırılarını tespit etmek için hibrit bir derin öğrenme modeli oluşturmuştur. Bu amaçla CNN ve ağırlığı azaltılmış Uzun Kısa Vadeli Bellek (UKVB) içeren bir derin öğrenme yöntemi kullanılmıştır.[9] Vital Ford., istenmeyen e-posta algılama, virüs algılama gibi karmaşık alanlardaki çeşitli zorlukların üstesinden gelmek için makine öğrenimi uygulamalarının öneminden bahsetmiştir. [10] Teik-Toe vd., kötü amaçlı yazılım saldırısı veri kümesini araştırmış ve benzersiz bir Bulanık K-Ortalama (FKM) kümeleme algoritması geliştirmiştir. FKM Algoritması ile anomaliyi tespit edebilen bir model sunmuşlardır. FKM algoritması ile lineer regresyon karşılaştırılması yapılmış ve %85’e %90 oranında daha fazla doğruluk oranı elde edilmiştir[11] Karaman vd., (2020) bilgisayar ağına saldırı durumunu belirlemek ve saldırı modelinin sisteme zarar verip vermeyeceğini belirleyen yapay sinir ağı temelli bir sistem tasarlamışlardır. Çalışmalarında DDOS, Botnet, DOS, BruteForce türündeki saldırıları ele alınmış olup inceledikleri veri setinde DDOS ve BruteFroce saldırıların diğer iki türe göre daha yüksek oranda geldiği tespit edilmiştir. Kurdukları sistemin başarısının ise %99, 26 gibi bir yüksek tahmin oranına sahip olduğu görülmüştür.[12] Özekes ve Karakoç (2019), zararlı ağ trafiğinin belirlenmesi için makine öğrenmesi algoritmalarından karar ağaçları ve rastgele orman (random forest) yöntemlerini kullanmışlardır. Veri kümesinde analiz edilen ve sınıflandırılan saldırı modelleri hizmet engelleme (Dos), dağıtılmış hizmet reddi (DDoS) ve Port Tarama (PortScan ) saldırılardır. Saldırıları sınıflandırmada rastgele orman yönteminin diğer yönteme göre daha başarılı olduğu tespit edilmiştir.[13]

Aytan ve Barışçı (2018), siber saldırı tespit sistemleri ile ilgili en çok kullanılan veri setli “KDD Cup’99” veri seti kullanılarak hizmet dışı bırakma saldırıları ve bilgi tarama saldırıları Weka aracında yer alan makine öğrenme algoritmaları ile tespit etmeye çalışılmıştır. En iyi sonucun Rastgele Orman Algoritması ile ortaya çıktığı tespit edilmiş olup en yakın değerler Geri Yayılma Algoritması ile elde edildiğini ortaya koymuşlardır.[14]. Karataş ve Şahingöz (2018) ise yapay sinir ağları tabanlı, çok katmanlı bir saldırı tespit sistemi oluşturarak KDD Cup 99 veri seti üzerinde “trainc, trainlm, trainbfg, trainscg, traincgp, trainoss, trainbr, trainr” eğitim fonksiyonlarının karşılaştırmasını yapmışlardır. Doğru-Pozitif (TP) metriğinin ölçek olarak kabul edildiği çalışmanın sonucuna göre en hızlı uygulama zamanı “trainscg” fonksiyonu ile, en düşük hata oranı ise “trainlm” fonksiyonu ile elde edilmiştir.[15] Ağ anomalisi tespiti ile ilgili bazı çalışmalar yukarıda belirtilmiştir ve bu alanda daha fazla çalışma mevcuttur.

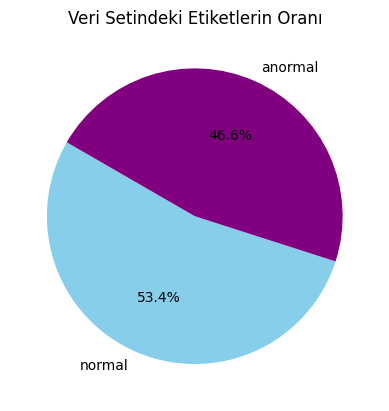
**3. Materyal ve Metodlar**

Bu çalışmamız, siber saldırıların tespiti için makine öğrenmesi algoritmalarının etkinliğini incelemeyi amaçlamaktadır. Farklı makine öğrenme algoritmaları ve hiperparametre optimizasyon teknikleri kullanılarak Network Intrusion Detection veri kümesi üzerinde deneyler gerçekleştirilmiştir. Elde edilen sonuçlar, kullanılan yöntemlerin saldırı tespit performansı üzerindeki etkisini karşılaştırmak ve en uygun modeli belirlemek için analiz edilmiştir.

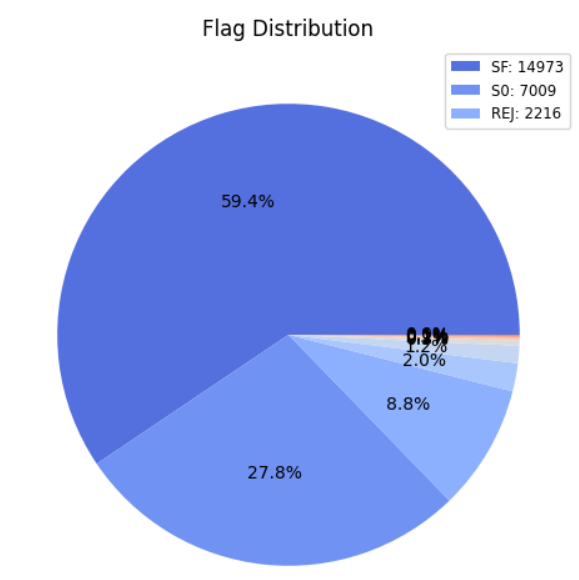
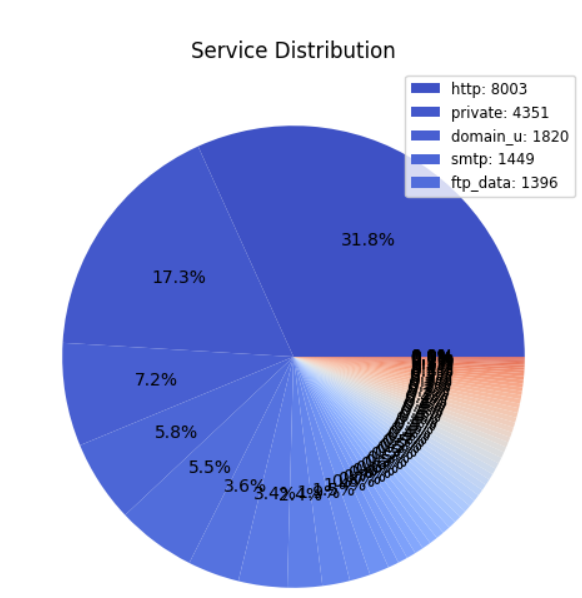
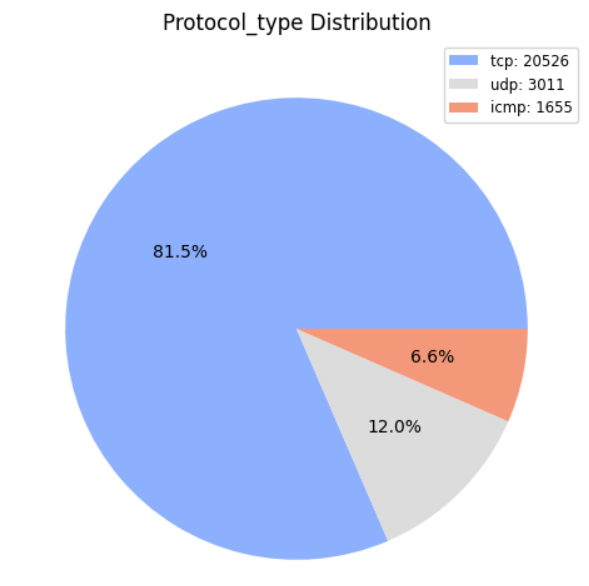
**3.1 Veri Kümesi**

Çalışmamızda Network Intrusion Detecction veri seti kullanılmıştır. Veri seti kaggle ortamından temin edilmiştir.Bu veri seti çeşitli izinsiz girişleri içeren askeri ağ ortamında simüle edilmiştir. Tipik bir ABD Hava Kuvvetleri LAN’ını temsil eden ortamda, ağ trafiği için ham TCP/IP verileri elde etmek için oluşturulmuştur. Ağ gerçek ortama benzer şekilde odaklanmış ve birden fazla saldırı ile hedeflenmiştir. Her bir bağlantı,bir kaynak IP adresinden bir hedef IP adresine belirli protokol altında veri akışını temsil eden TCP paketlerinin dizisi olarak tanımlanır. Ayrıca, her bağlantı normal veya tam olarak belirli saldırı türü olarak etiketlenmiştir. Her bir bağlantı kaydı yaklaşık 100 bayt içermektedir.

Her TCP/IP bağlantısı için,normal ve saldırı verilerinden 41 özellik elde edilmiştir. Bu özelliklerden 3 tanesi nitel ve 38 tanesi niceldir. Network Intrusion Detecction(Ağ Sızma Tespiti), bilgisayar ağlarında bulunan kötü niyetli veya yetkisiz kullanımları tespit etmeyi amaçlayan güvenlik önlemidir. Veri seti, bu tür saldırıları simüle etmek için veya gerçek saldırıları içeren verileri içerebilir. Veri seti, ağ güvenliği algoritmalarının geliştirilmesi veya test edilmesi için kullanılır. Normal ve anormal trafiğin kapsamlı veri setini oluşturmak için çeşitli ağ saldırılarını içermektedir. Her ağ bağlantısı,kaynak ve hedef IP adreslerinin arasında değiş tokuş edilen TCP paketleri dizisi olarak temsil edilmektedir. Veri seti, her bir bağlantı için 41 özellik içermektedir. Bu özelliklerde üç ayrı başlık altında ele alınır. Bunlar içerik özellikleri,sunucu tabanlı trafik özellikleri ve zamana bağlı trafik analizleridir. İçerik özelliklerinde TCP bağlantısı ile oluşturulan öznitelikler incelenir. Sunucu tabanlı trafik özelliklerinde etki alanı(domain) bilgisi ile sağlanan sunucu tabanlı özellikler incelenir. Zamana bağlı trafik özelliklerinde ise aynı sunucu ve servis özelliklerini kullanan özniteliklerde zamana bağlı trafik özellikleri incelenir. Bu 41 özelliğin 3 tanesi nitel, 38 tanesi niceldir. Özelliklerden her biri “Normal” veya “Anormal” şeklinde etiketlenmiştir. Random Forest Classifier kullanılarak özelliklerin önem derecesi değerlendirmeye alınmış ve en önemli 13 özellik seçilmiştir. Bu özelliklerden 10 tanesi nicel, 3 tanesi niteldir. Özellikler aşağıda belirtilecektir.



*Şekil 1. Veri Setindeki Etiketlerin Dağılımı*

** **

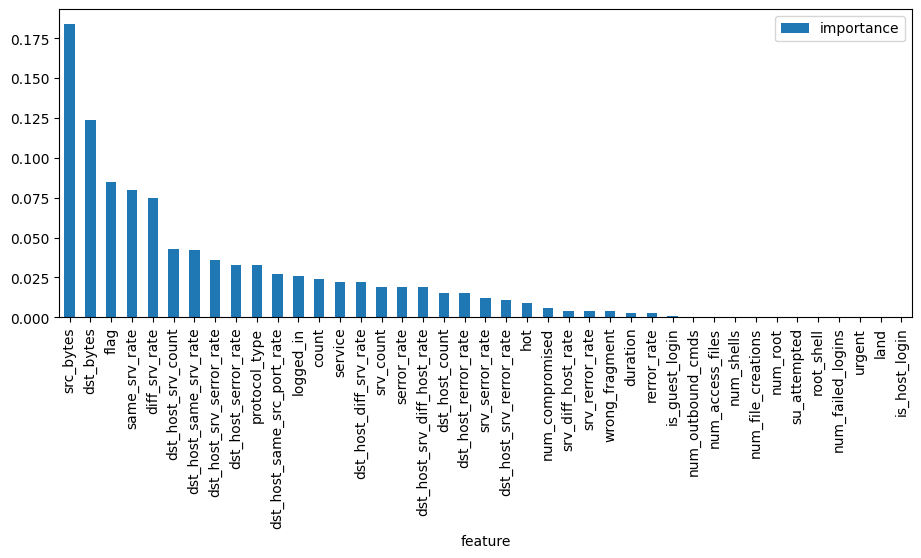
*Şekil 2. Veri Setindeki Kategorik Değişkenlerin Oranları*

**3.2 Öznitelik Seçimi**

Öznitelik seçimi ile veri setindeki mevcut öznitelik arasından uygun bir alt kümenin seçilmesi amaçlanır. Çalışmada kullanılan veri setinde 41 farklı öznitelik vardır. Oluşturulacak olan saldırı tespit sisteminde ağ trafiğinin izlenerek saldırı tespitinin başarı oranının artırılması ve çalışma zamanının da buna bağlı olarak mümkün olduğunca kısa tutulması amacıyla mevcut 41 farklı özniteliğin sayısının indirgenmesi gerekmektedir. IDS’lerin performansını geliştirmek için öznitelik seçimi ve Random Forest (RF) algoritması kullanılmıştır.Amacımız öznitelik sayısını azaltarak çalışma süresini kısaltmak ve saldırı tespit doğruluğunu korumaktır.

Veri kümesindeki özniteliklerin korelasyonları incelenerek birbirleriyle yüksek korelasyona sahip olanlar belirlenmiştir. Elediğimiz özniteliklerin etkisinin 0’a yakın olduğunu görülmüştür. Kalan öznitelikler ise RF algoritması kullanılarak önem sırasına göre sıralanmıştır. RF, her bir özniteliğin modelin tahmin doğruluğuna olan katkısını ölçerek bu sıralamayı yapmaktadır. Öznitelik seçimi sonrası 41 öznitelik sayısı 13’e indirgenmiştir. Model kurmak için bu özellikler kullanılmıştır :

* src\_bytes (Kaynak Baytlar): Bir paketin kaynağından hedefe gönderilen bayt sayısını ifade eder.
* dst\_bytes (Hedef Baytlar): Bir paketin hedeften kaynağa geri dönen bayt sayısını ifade eder.
* same\_srv\_rate (Aynı Hizmet Oranı): Belirli bir zaman diliminde, aynı hizmete (service) sahip bağlantıların oranını gösterir. Örneğin, HTTP hizmetine yapılan bağlantıların oranı.
* dst\_host\_same\_srv\_rate (Hedef Host Aynı Hizmet Oranı): Belirli bir zaman diliminde, belirli bir hedef ana bilgisayara yapılan ve aynı hizmeti kullanan bağlantıların oranı.
* diff\_srv\_rate (Farklı Hizmet Oranı): Belirli bir zaman diliminde, farklı hizmetlere sahip bağlantıların oranını gösterir. Örneğin, HTTP dışındaki hizmetlere yapılan bağlantıların oranı.
* flag (Bayrak): TCP veya UDP paketlerinin bayraklarını (flags) ifade eder. Örneğin, "SYN", "ACK", "RST", "FIN" gibi bayraklar.
* dst\_host\_diff\_srv\_rate (Hedef Host Farklı Hizmet Oranı): Belirli bir zaman diliminde, belirli bir hedef ana bilgisayara yapılan ve farklı hizmetleri kullanan bağlantıların oranı.
* logged\_in (Giriş Yapıldı mı): Kullanıcı oturum açmış mı (1: Evet, 0: Hayır). Bu, belirli bir durumda oturum açma işleminin gerçekleşip gerçekleşmediğini gösterir.
* count (Bağlantı Sayısı): Belirli bir zaman diliminde belirli bir hedef ana bilgisayara yapılan toplam bağlantı sayısını gösterir.
* dst\_host\_srv\_count (Hedef Host Hizmet Sayısı): Belirli bir zaman diliminde, belirli bir hedef ana bilgisayara yapılan ve aynı hizmeti kullanan bağlantıların sayısı.
* protocol\_type (Protokol Türü): Paketin taşıdığı protokol türünü ifade eder. Örneğin, "TCP", "UDP", "ICMP".
* dst\_host\_same\_src\_port\_rate (Hedef Host Aynı Kaynak Port Oranı): Belirli bir zaman diliminde, belirli bir hedef ana bilgisayara yapılan ve aynı kaynak portunu kullanan bağlantıların oranı.
* service (Hizmet): Bağlantının kullandığı hizmet türünü ifade eder. Örneğin, "HTTP", "FTP", "DNS".

****

*Şekil 3. Random Forest Classifier ile Veri Setindeki Özniteliklerin Önem Sırası*

Şekil 3’te görüldüğü üzere bazı özelliklerin model için önemi yoktur, birçoğunun 0’a yakındır. Bu özellikler elenerek daha optimize ve hızlı çalışan bir sistem oluşturulması hedeflenmiştir.

**3.3 Topluluk Öğrenme Algoritmaları**

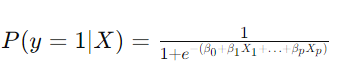
Topluluk yöntemleri olarak da adlandırılan topluluk öğrenme, birden çok ayrı modelden gelen tahminleri birleştirmekten oluşan denetimli makine öğrenimi yönteminin bir türüdür. Veri bilimi topluluğunda son zamanlarda daha fazla popülerlik kazanmıştır.Yaygın ve iyi bilinen örnekleri Torbalama (Bagging), Güçlendirme (Boosting) ve İstiflemedir (Stacking) (Opitz ve Maclin, 1999; Graczyk ve diğ., 2010; Han ve diğ., 2011;Eldardiry, Neville ve Rossi, 2020).Bir topluluk öğrenme yöntemi oluşturmadan önce alt modellerin seçimi önemlidir.Genellikle alt modeller farklı şekilde eğitilmiş aynı algoritmadan oluşur.Bu topluluk modelinin daha sonra "homojen" olduğu söylenir. İstifleme gibi yöntemler de vardır;oluşturulan topluluk "heterojen topluluklar modeli" olarak adlandırılır (Zhou, 2009).Projemizde Torbalama tabanlı topluluk öğrenme yöntemi kullanılmıştır.Torbalama, genellikle homojen zayıf modellerden oluşan bir topluluk öğrenme yöntemidir. Bu modeller veri setini birbirinden bağımsız olarak öğrenir.Çok fazla veriye ihtiyaç duyulmasından çoğunlukla boostrap örnekleme yöntemi kullanılır. Torbalamanın arkasındaki fikir çok basit; daha düşük varyansa sahip bir model elde etmek için birkaç bağımsız modele uymak ve tahminlerinin ortalamasını almaktır.

Topluluk modellerini güçlendirmek, genellikle homojen zayıf öğrenicileri dikkate alır, onları çok uyarlanabilir bir şekilde sırayla öğrenir ve bunları deterministik yolla birleştirir. Güçlendirme, sıralı olarak çok sayıda zayıf modeli çok uyarlanabilir bir şekilde uydurmaktan oluşan bir tekniktir.

**3.4 Kullanılan Algoritmalar**

**Logistic Regression**

Logistic Regression, istatistiksel bir model olup, özellikle iki sınıflı sınıflandırma problemlerinde yaygın olarak kullanılır.Bu model regresyon tekniklerini temel alır, ancak bağımlı değişkenin olasılık dağılımı üzerine odaklanır.Lojistik regresyon, özellikle büyük veri kümeleri üzerinde hızlı bir şekilde eğitilebilir.Ağ anomalisi tespiti bağlamında, normal ve anormal ağ trafiği arasındaki ayrımı yapmak için kullanılır.Lojistik regresyon, girdi değişkenleri ile çıktı değişkeni arasındaki ilişkiyi modellemek için sigmoid fonksiyonunu kullanır.Bu fonksiyon, olasılık değerlerini tahmin etmek için idealdir.Modelin katsayıları, her bir özelliğin tahmin üzerindeki etkisini anlamayı kolaylaştırır.[16]



**P(y=1|X):**Bağımlı değişkenin 1 olma olasılığı

**β0​:** Sabit terim

**β1​,…,βp​:** Modelin katsayıları

**X1​,…,Xp​:** Bağımsız değişkenler.

**Decision Tree (Karar Ağacı)**

Karar ağacı, belirli bir soruna yönelik tüm potansiyel çözümleri haritalandıran akış şeması benzeri bir diyagramdır. Bu algoritma, verileri özelliklerine göre dallara ayırarak, veri setindeki örnekleri sınıflandırır. Karar ağaçları, kolay anlaşılabilirliği ve yorumlanabilirliği ile öne çıkar[17].Ağ anomalisi tespitinde, karar ağaçları, ağ trafiği özelliklerini kullanarak anormal olayları normal olaylardan ayırt etmek için bir dizi kural oluşturur. Bu kurallar, ağ trafiğinin hangi özelliklerinin anormal davranışları en iyi şekilde tanımladığını belirlemeye yardımcı olur.Karar ağaçları, veri setini iteratif olarak bölerek çalışır. Her bölme, veri setini mümkün olan en homojen alt gruplara ayırmayı hedefler. Ağaç yapısının her düğümü, bir özelliğin belirli bir değerine dayanarak veri setini böler.

**Gini Katsayısı:** Bir düğümdeki örneklerin saflığını ölçer. Gini katsayısı minimum olan bölme seçilir.



**Bilgi Kazancı:** Entropi farkına dayanır ve düğümün belirsizliğini azaltan bölmeleri seçer.

**GaussianNB (Gaussian Naive Bayes)**

Gaussian Naive Bayes, Bayes teoremine dayanan özellikle sürekli veri özelliklerine sahip veri setlerinde kullanılan bir sınıflandırma algoritmasıdır. Ağ anomalisi tespitinde, GaussianNB, her bir özelliğin normal dağılım gösterdiğini varsayarak, bir ağ olayının anormal olma olasılığını hesaplar.[18]

Gaussian Naive Bayes, Bayes teoremini kullanarak sınıfların olasılıklarını hesaplar ve her bir sınıf için olasılığı en yüksek olanı seçer. Bayes teoremi şu şekilde ifade edilir:



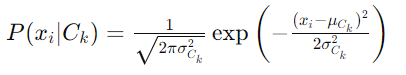
**P(C∣X):** X veri noktasının sınıfı C olma olasılığı.

**P(X∣C):** C sınıfında X veri noktasının ortaya çıkma olasılığı.

**P(C):** C sınıfının a priori olasılığı.

**P(X):** X veri noktasının a priori olasılığı.

Gaussian Naive Bayes, özelliklerin normal dağılıma sahip olduğunu varsayarak, P(X∣C)P(X|C)P(X∣C) olasılığını Gaussian dağılım fonksiyonu ile hesaplar:

​

**CatBoost**

CatBoost, özellikle kategorik özellikler içeren veri kümeleri üzerinde yüksek performans gösteren bir gradient boosting algoritmasıdır.Özellikle büyük veri setlerinde ve kategorik özelliklerin bulunduğu durumlarda etkili sonuçlar verir. Önişleme süresi yerine eğitim süresi boyunca kategorik öznitelikleri ele almaktadır. CatBoost, düşük hiperparametre ayarlaması gereksinimi ve otomatik özellik önem sıralaması gibi kullanışlı özelliklere sahiptir.Ağ trafiğindeki IP adresleri, port numaraları gibi kategorik özellikleri işleyerek, normal ve anormal davranışları tespit eder.[19]

**Sıralı Kodlama:** Kategorik özellikleri sıraya göre kodlayarak, veri sızıntısını (data leakage) önler.

**Destek Vektörleri:** Kategorik özelliklerin işlenmesi sırasında daha dengeli ve doğru tahminler yapılmasını sağlar.

**XGBoost**

XGBoost algoritması Gradient Boosting algoritmasının optimize edilmiş bir türüdür. Büyük ölçekli ve yüksek boyutlu veri setlerinde etkili sonuçlar verir.XGBoost, genellikle diğer birçok makine öğrenmesi algoritmasından daha yüksek doğruluk oranları elde eder.Ağ anomalisi tespitinde, XGBoost, ardışık olarak ağaçlar oluşturarak ve her bir ağacın bir önceki ağacın hatalarını düzeltmesini sağlayarak güçlü bir sınıflandırma modeli oluşturur.Her iterasyonda hatayı azaltarak modelin performansını artırır. Bu algoritma, her bir yeni ağacı eklerken mevcut modelin hatalarını düzeltmeye çalışır[20]. XGBoost, hedef değişkenin ve özelliklerin bir doğrusal kombinasyonunu optimize eder:



**L(f):** Toplam kayıp fonksiyonu.

**ℓ(yi​,y^​i​):** Tahmin edilen ve gerçek değerler arasındaki kayıp fonksiyonu.

**Ω(fk​):** Model karmaşıklığı.

**Random Forest**

Random forest algoritması,denetimli sınıflandırma algoritmalarından biridir. Hem regresyon hem de sınıflandırma problemlerinde kullanılmaktadır. Algoritma ,birden fazla karar ağacı üreterek sınıflandırma işlemi esnasında sınıflandırma değerini yükseltmeyi hedefler. Random forest algoritması birbirinden bağımsız olarak çalışan birçok karar ağacının bir araya gelerek aralarından en yüksek puan alan değerin seçilmesi işlemidir.Her bir ağaç, veri setinin rastgele bir alt kümesinde eğitilir ve bu sayede modelin çeşitliliği artırılır.[21]

Bootstrap Örnekleme: Her ağaç, eğitim veri setinin rastgele bir örnekleminde eğitilir.

Rastgele Özellik Seçimi: Her ağaç için rastgele bir alt özellik kümesi seçilir.

Ağaçların çıktıları şu şekilde birleştirilir:

Sınıflandırma: Her ağacın tahmini oylanır ve en çok oyu alan sınıf seçilir.

Regresyon: Her ağacın tahminleri ortalanarak nihai tahmin elde edilir.

Random Forest, ağ anomalisi tespitinde, ağ trafiğindeki karmaşık ilişkileri modellemek ve anormal davranışları belirlemek için kullanılır. Her bir ağaç, veri setinin farklı bir yönünü öğrenir ve bu sayede anomali tespitinde yüksek doğruluk sağlar.

Projede, veri setimizin yapısını etkili bir şekilde modelleme amacıyla, yüksek doğruluk oranları ve düşük overfitting riski sunması nedeniyle modelleme sürecinde Random Forest algoritması tercih edilmiştir.

**Performans Metrikleri**

Farklı makine öğrenmesi algoritmaları ile farklı modeller oluşturulmuştur.bu modellerden hangisinin daha iyi sonuç vereceğini ölçmek için performans metriklerine ihtiyaç duyulmaktadır.Makine öğrenmesi modellerinin doğruluğu ,4 farklı metrik ile ölçülmüştür.



*Tablo 1. Confusion matrix*

**Doğruluk (Accuracy)**

Doğru tahmin edilen değerlerin, tüm değerlere bölünmesi ile elde edilmektedir. Doğruluk değeri, 0 ile 1 arasındadır. Doğruluk değeri 1’e yaklaştıkça başarı artmaktadır.

Doğruluk: (GP + GN) / (GP + GN + FP + FN)

**Duyarlılık (Recall)**

Doğru olarak tahmin etmemiz gereken değerlerin, ne kadarını doğru tahmin ettiğimizi belirtmektedir. Duyarlılık değeri, gerçekte doğru olan ve doğru olarak tahmin edilen değerlerin, tüm doğru değerlere bölünmesi ile elde edilmektedir.

Duyarlılık: GP / (GP + FN)

**Kesinlik (Precision)**

Doğru olarak tahmin ettiğimiz değerlerin, ne kadarının gerçekte doğru olduğunu göstermektedir. Kesinlik değeri, gerçekte doğru olan ve doğru olarak tahmin edilen değerlerin, doğru olarak tahmin edilen tüm değerlere bölünmesi ile elde edilmektedir.

Kesinlik: GP / (GP + FP)

**F1 Skoru (F1 Score)**

F1 skoru, duyarlılık ve kesinlik değerlerinin harmonik ortalamasının hesaplanması ile elde edilmektedir. Her iki değerinde hesaplamaya katılarak dengeli bir değer elde edilmesi amaçlanmaktadır. Eşit dağılıma sahip olmayan veri setlerinde başarılı sonuçlar elde etmek için kullanılmaktadır.

F1 Skoru: 2 \* Kesinlik \* Duyarlılık / (Kesinlik + Duyarlılık)

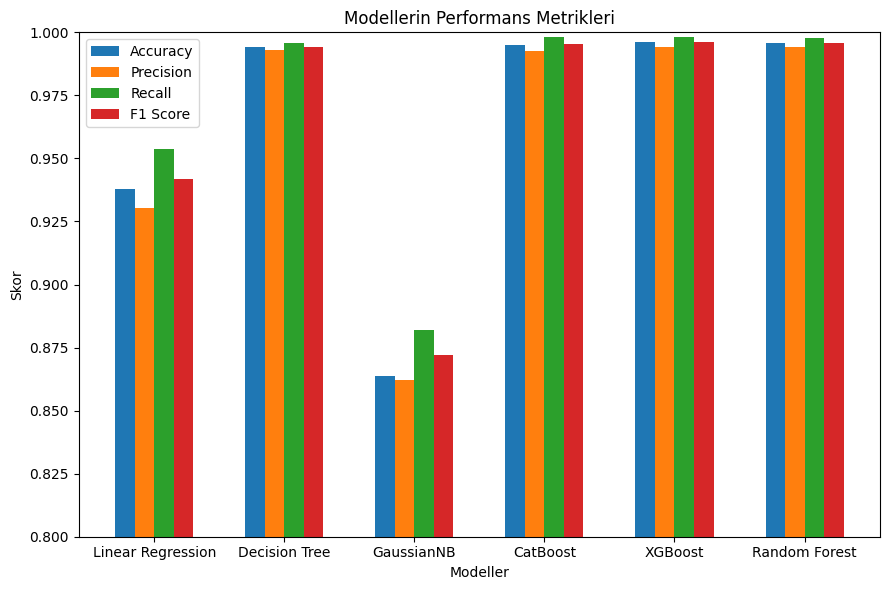
**4. Deneysel Sonuçlar ve Tartışma**

Çalışmada Kaggle üzerinden alınmış Network Intrusion Detection veri kümesi ile ağ trafiklerinin yapay zeka kullanılarak ‘normal’ veya ‘anormal’ olma durumları tahmin edilmeye çalışılmıştır.

|  | **Precision** | **Recall** | **F1-Score** | **Accuracy** |
| --- | --- | --- | --- | --- |
| **GaussianNB** | **0.86** | **0.86** | **0.86** | **0.86** |
| **Logistic Regression** | **0.94** | **0.94** | **0.94** | **0.93** |
| **Random Forest Classifier** | **1.00** | **1.00** | **1.00** | **0.99** |
| **CatBoost** | **1.00** | **1.00** | **1.00** | **0.99** |
| **XGBoost** | **1.00** | **1.00** | **1.00** | **0.99** |
| **Decision Tree Classifier** | **0.99** | **0.99** | **0.99** | **0.99** |

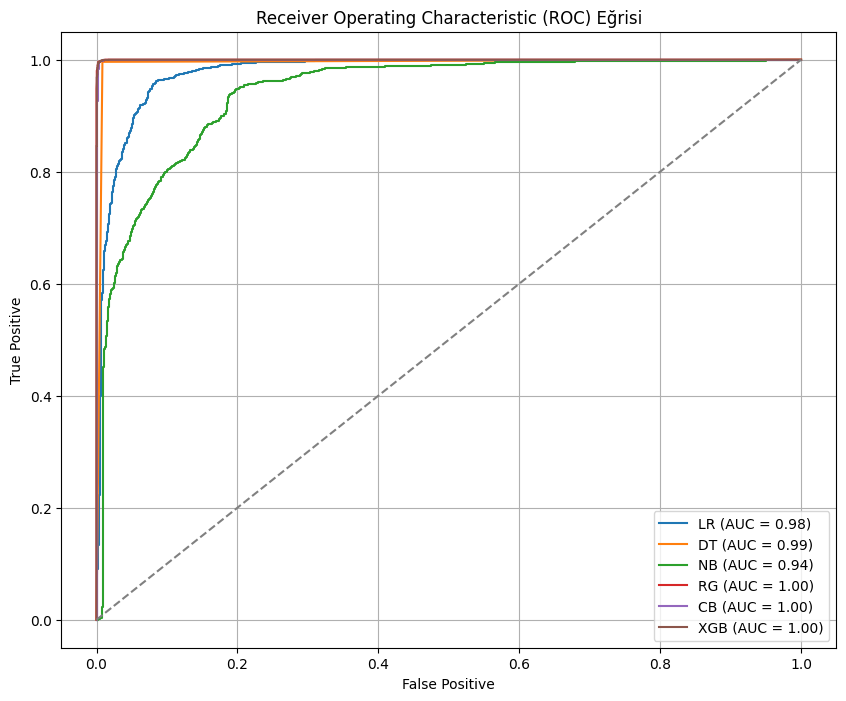
*Tablo 2.**Modellerin Performans Metrikleri*

Tablo 2’de verilen performans metrikleri grafiği incelendiğinde algoritmaların tamamının iyi sonuç verdiği gözlemlenmiştir. Logistic Regression ve Gaussian NB algoritmaları dışındaki algoritmalar ortalama %99 gibi bir doğruluk değeri performansı göstermiştir.

****

*Şekil 4. Bütün Modellerin Doğruluk Metriklerinin Grafiği*

Şekilde 4’te modellerin performans metrikleri daha iyi göz önünde bulundurulmuş ve Random Forest algoritması seçilmiştir.



*Şekil 5. ROC Eğrisi Grafiği*

Bu modeller hiperparametre optimizasyon işlemine tabi tutulmuştur. Optimizasyon sonrası model performanslarında artış olmuştur.

|  | **Hiper-Parametre** | **Değer** |
| --- | --- | --- |
| **Logistic Regression** | max\_iter | 1000 |
| c | 0.1 |
| penalty | l2 |
| solver | newton-cg |
| **Random Forest Classifier** | bootstrap | False |
| max\_depth | None |
| max\_features | sqrt |
| min\_samples\_leaf | 1 |
| min\_samples\_split | 2 |
| n\_estimators | 100 |
| **XGBoost** | colsample\_bytree | 0.7 |
| learning\_rate | 0.1 |
| max\_depth | 7 |
| min\_child\_weight | 1 |
| n\_estimators | 200 |
| subsample | 1.0 |

*Tablo 3. Model Hiperparametreleri*

**5. Sonuç**

Bilgisayar ağlarına yönelik saldırıların önlenebilmesi, ilgili ağlarda ortaya çıkabilecek güvenlik sorunlarına tedbir alınması ve daha çok kullanıcıların oluşturduğu güvenlik açıklarının kapatılması ile mümkün olabilmektedir. Yapılan bu çalışmayla saldırı tespit sürecinde insan müdahalesinin olabildiğince az, makine öğrenmesi esaslarına dayanan ve ağ trafiğindeki olağan dışı hareketleri önceden tanımlanması gerek kalmadan yorumlayabilen bir sistem hedeflenmiştir.

Ağ üzerindeki saldırıların analizi için elde edilen veriler en önemli bileşenler olarak kabul edilmektedir. Çalışma sonucunda elde edilen verilere dayanarak saldırı tespitinde kullanılacak olan özniteliklerin en iyi şekilde seçilmesi gerektiği görülmüştür. Mevcut olan çalışmalara benzer doğrultuda algoritmaların analiz edilmesinde doğruluk, hassasiyet, f-ölçütü ve ROC eğrisi dikkate alınmıştır. Yapılan deneyler sonucunda en başarılı sınıflandırma modeli Random Forest Classifier algoritması olduğu görülmüştür.

Gelecekteki çalışmalarda ise ağ üzerinden daha fazla veri alarak öznitelik sayımızı arttırmayı ve yeni özellikler ile model kurup anomali tespiti analiz edilecektir.Sistemi geliştirip ağa yönelik saldırı tespiti için makine öğrenmesi ve derin öğrenme kullanan kapsamlı bir saldırı tespit sistemi (IDS) geliştirmesi planlanmıştır.. Derin öğrenme modellerinin dahil edilmesiyle önerilecek bir Saldırı Tespit Sistemi’nin başarımı literatürdeki diğer sistemler ile kıyaslanacaktır.

Gelecek çalışmalarda bahsedilen bu tekniklerin kullanılması ve farklı veri kümelerinin topluluk öğrenme yöntemleri ile bir araya getirilmesi ile daha iyi sonuçlar elde edilmesi planlanmaktadır.

**6.Kaynakçalar**

[1] Thottan, M., Ji, C., Anomaly detection in IP networks. IEEE T SIGNAL PROCES, 51(8): 2191-2204, 2003.

[2] Kaya, Ç., & Yıldız, O. (2015). Intrusion Detection with Machine Learning Techniques: Comparative Analysis. Marmara Fen Bilimleri Dergisi, 26(3), 89-104. <https://doi.org/10.7240/mufbed.24684>

[3] Akhan Baykan, N. ve Khorram, T. (2021). Optimize Edilmiş Makine Öğrenimi Algoritmalarını Kullanarak Ağa İzinsiz Giriş Tespiti. Avrupa Bilim Ve Teknoloji Dergisi ( 25 ), 463-474. https://doi.org/10.31590/ejosat.849723

[4] Kuş, İ., Bozkurt Keser, S., & Yolaçan, E. (2021). Saldırı Tespit Sistemlerinde Topluluk Öğrenme Yöntemlerinin Kıyaslanması. Avrupa Bilim Ve Teknoloji Dergisi(31), 725-734. <https://doi.org/10.31590/ejosat.971875>

[5] E. C. Bayazit, O. K. Sahingoz and B. Dogan, "Neural Network Based Android Malware Detection with Different IP Coding Methods," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2021, pp. 1-6, doi: 10.1109/HORA52670.2021.9461302. keywords: {Training;Operating systems;Cellular phones;Transforms;Tablet computers;Malware;Encoding;Machine Learning;ANN;Android System;Malware Detection},

[6,7] Örnek, Ö., Vatan, S., Sarıoğlu, S., Yazıcı, A. (2018). Trafik Ağlarında Anomali Tespiti. Eskişehir Osmangazi Üniversitesi Mühendislik Ve Mimarlık Fakültesi Dergisi, 26(3), 132-138. <https://doi.org/10.31796/ogummf.440285>

[8,9] Demir, F. (2021). Siber saldırı tespiti için makine öğrenmesi yöntemlerinin performanslarının incelenmesi. Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 23(2), 782-791. https://doi.org/10.25092/baunfbed.876338

[10,11] Gökdemr, A., & Çalhan, A. (2022). Nesnelerin interneti ortamlarında derin öğrenme ve makine öğrenmesi tabanlı anomali tespiti. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 37(4), 1945-1956. <https://doi.org/10.17341/gazimmfd.962375>

[12] Karaman, MS, Turan, M. ve Aydın, MA (2021). Yapay Sinir Ağı Kullanılarak Anomali Tabanlı Saldırı Tespit Modeli Uygulaması. Avrupa Bilim Ve Teknoloji Dergisi ( Ejosat Ek Özel Sayı (HORA) , 10-17. https://doi.org/10.31590/ejosat.1115825

[13] Hatipoğlu, C. ve Tunacan, T. (2021). Türkiye'de Siber Saldırı ve Tespit Yöntemleri: Bir Literatür Taraması. Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi, 8(1), 430-445. https://doi.org/10.35193/bseufbd.838732

[14]Aytan, B., & Barisci, N. (2018, December). Siber savunma alanında yapay zekâ tabanlı saldırı tespiti ve analizi. In *Proceeding of the 2nd International Symposium on Innovative Approaches in Scientific Studies, Samsun,(2018, December)*.

[15]Karataş, G. and Şahingöz, Ö.K., 2018. Neural Network Based Intrusion Detection Systems with Different Training Functions, 2018 6th International Symposium on Digital Forensic and Security (ISDFS), 22-25 March 2018, 1-6, Antalya-Türkiye.

[16]Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression*. John Wiley & Sons.

[17]Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth International Group.

[18]Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

[19]Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems (NeurIPS 2018)*, 6638-6648.

[20]Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.

[21]Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.