All Sciences Proceedings

1st International Conference on Pioneer and Innovative Studies

June 5-7, 2023 : Konya, Turkey

ICPIS

All Sciences Proceedings
http://as-proceeding.com/

https://as-proceeding.com/index.php/icpis

Hiperparemetreleri Ayarlanmış Makine Öğrenmesi Yöntemleri Kullanılarak Ağdaki Saldırıların Tespiti

Doğan EROL[1*], Halit BAKIR [2 I]

[1]*Savunma Teknolojileri Anabilim Dalı / Tezli Yüksek Lisans Enstitüsü, Sivas Bilim ve Teknoloji Üniversitesi, Türkiye*
[2]*Bilgisayar Mühendisliği Bölümü / Bilgisayar Yazılımı Anabilim Dalı, Üniversite, Türkiye*

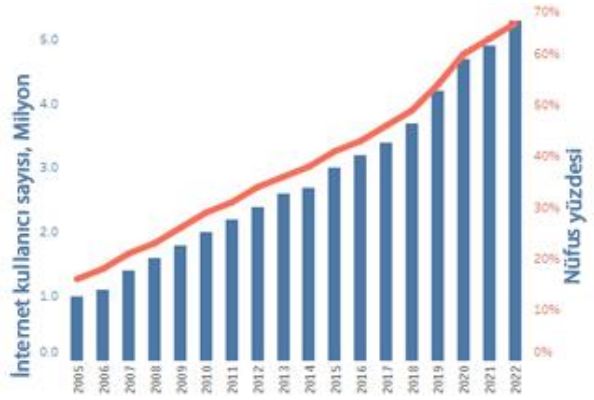
[*]*(220102001@sivas.edu.tr)*

Özet – Bilişim teknolojilerinde yaşanan hızlı gelişmeler her geçen gün bilgisayar ve internet kullanan kişi sayısını artırmaktadır. Teknolojide yaşanan yeni gelişmelerle birlikte internet kullanıcıları, bilgisayar veya ağ yapılarında güvenlik açığı olmasından dolayı çeşitli güvenlik tehditleri ile karşı karşıya gelmektedir. Bazı kötü niyetli internet kullanıcılarının veya bilgisayar korsanlarının yaptıkları saldırı denemeleri sonucunda ağ üzerinde bulunan sistemlere izinsiz girişlerin tespit edildiği gözlemlenmektedir. Maruz kalınan izinsiz girişlerin engellenmesi için internet kullanıcıları Saldırı Tespit Sistemleri, antivirüs ve trojan tespit etme ve engelleme programları kullanmaktadırlar. Bu amaçla son yıllarda, saldırıların tespit edilmesi için makine öğrenmesi algoritmalarının kullanımı bir hayli artmıştır. Bu çalışmada ağdaki trafiğin saldırı olup olmadığının tespiti için ikili sınıflandırma yapılmış olup saldırı olarak sınıflandırılan trafik, saldırı türlerine göre sınıflandırılmıştır. Bu çalışmada, Saldırı Tespit Sistemleri'nde sıkça kullanılan veri setlerinden biri olan Kaggle platformundan alınan NSL- KDD veri seti kullanılmıştır. Veri seti üzerine topluluk öğrenme algoritmalarından RandomForest, DecisionTree, ExtremeGradientBoosting (XgBoosting), GradientBoosting, KNN ve Bagging uygulanmış ve analizi yapılmıştır. Çalışmada Yapay Sinir Ağları algoritmaları ile hiperparametre optimizasyonu gerçekleştirilerek multi-class sınıflandırma da denenmiştir. Yüksek doğruluk elde edilmesine karşın klasik makine öğrenmesi metotlarından daha düşük sonuçlar elde edilmiştir. Makine öğrenmesi algoritmalarıyla elde edilen başarılı sonuçlar sayesinde Saldırı Tespit Sistemleri üzerinde iyileştirmeler ve düzenlemelere gidilebilir.

Anahtar Kelimeler – Makine öğrenmesi, Saldırı tespit sistemi, Randomforest, Gradientboosting, Hiperparametre ayarlama

I. GİRİŞ

Bilişim teknolojilerinde yaşanan hızlı gelişmeler her geçen gün bilgisayar ve internet kullanan kişi sayısını artırmaktadır. İnternet kullanımı iş, eğitim, sağlık, ulaşım gibi farklı alanlarda hızlıca yaygınlaşmakta ve internet kullanıcı sayısı bir hayli artmaktadır. Uluslararası Telekomünikasyon Birliği'nin yayınladığı rapora göre 2022 yılı sonlarında dünya genelinde 5,3 milyar internet kullanıcısının olduğu gözlemlenmektedir[25].

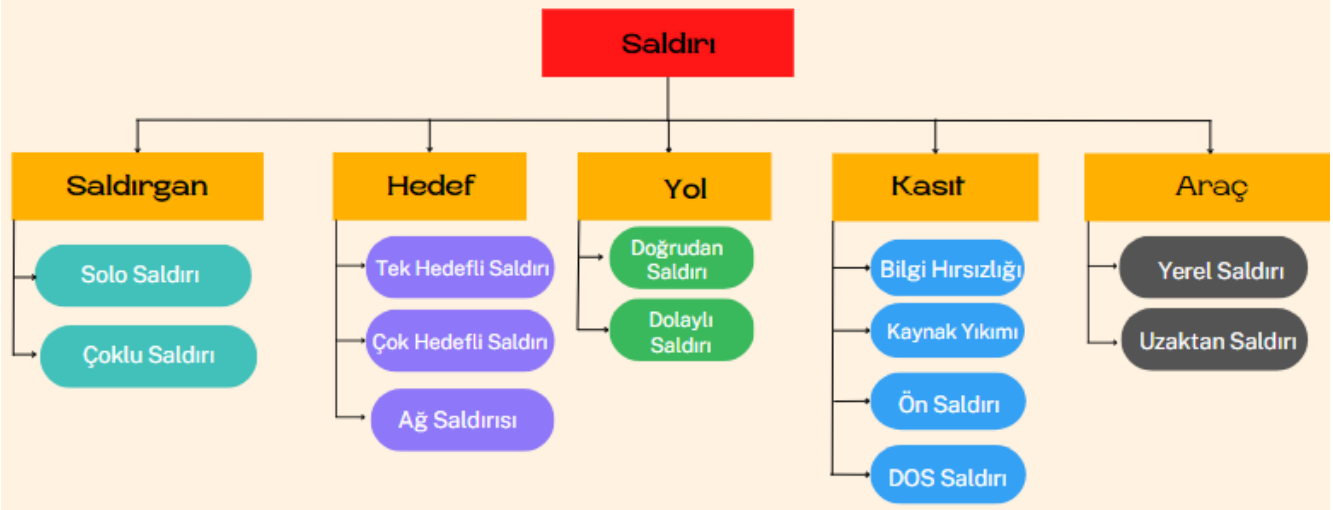


Şekil 1. İnternet kullanıcı sayısı[25]

Teknolojide yaşanan yeni gelişmelerle birlikte internet kullanıcıları, bilgisayar veya ağ yapılarında

[I] Khaled Bakour or Halit Bakır: Due to the author's dual citizenship, his name can be written in two different ways.

güvenlik açıkları olmasından dolayı bazı güvenlik tehlikeleri yaşamaktadırlar. Bazı kötü niyetli internet kullanıcıları veya bilgisayar korsanları tarafından yapılan saldırılar sonucunda ağ üzerindeki bilgisayarlar üzerinde izinsiz girişlerin tespit edildiği gözlemlenmektedir. Maruz kalınan izinsiz girişlerin engellenmesi için internet kullanıcıları Saldırı Tespit Sistemleri, antivirüs ve trojan tespit etme ve engelleme programları kullanmaktadırlar. Ayrıca izinsiz girişlerin engellenmesi için çeşitli güvenlik programları bulunmaktadır. Saldırı Tespit Sistemleri(STS) hem bilgisayar ağı içinden hem de bilgisayar ağı dışından yapılan olası saldırıların saptanması için kullanılan bir yöntemdir[24]. Saldırı sınıfları Şekil.2'de verilmiştir.



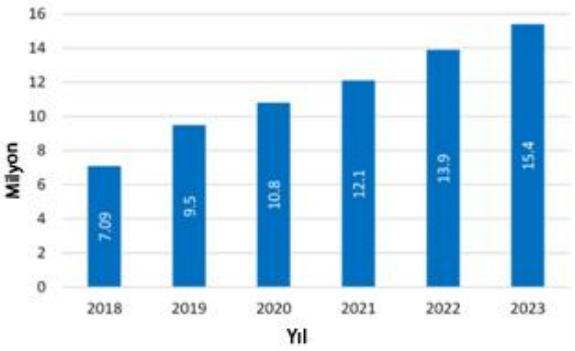
Şekil 2. Saldırı sınıfları

STS yapılan saldırıları 3 farklı yöntemle tespit etmeye çalışır. Bu yöntemler imza tabanlı, anomali tabanlı ve hibrit tabanlı saldırı tespit sistemleridir.

- İmza Tabanlı STS: Kötü niyetli yazılım güvenlik programındaki veri tabanıyla karşılaştırılarak ağda oluşan anormallik tespit edilmeye çalışılır.
- Anomali Tabanlı STS: Ağdaki paketlerin başlıklarının bilgileri hafızaya alınır ve ağdaki sistemin daha önce yaptığı davranışlar ile karşılaştırılır. Karşılaştırma sonucunda anormal bir trafiğin görülmesi durumunda sistem alarm üretir.
- Hibrit Tabanlı STS: İmza Tabanlı STS, sisteme yönelik yapılan yeni saldırı tiplerini tespit edememesinden dolayı güvenlik zafiyeti oluşturmaktadır. Güvenliği en üst seviyede tutmak amacıyla hibrit tabanlı STS'lerde imza tabanlı STS ile anomali tabanlı STS'ler beraber kullanılmaktadır[34].

Veri setindeki ağ üzerindeki yapılan saldırılar dört farklı yönteme göre ayrılmıştır[31,21,42]. Bunlar:

- Bilgi Tarama-Probe**:** Bilgi tarama yönteminde hedeflenen cihaz veya cihazların işletim sistemi, açık portları, IP adresi gibi bilgilerini elde etmek amaçlanmaktadır[31,39].
- Hizmet Reddi-Denial of Service-DOS: Bu saldırı tipinde hedef sisteme, saldırgan tarafından küçük paketler(programlar) gönderilerek sistemin hizmet veremez duruma gelmesi amaçlanmaktadır[23]. Hizmet Reddi saldırıları ağ trafiğinin incelenmesi ile tespit edilmektedir. Ağdaki trafiğin saldırı mı yoksa normal trafik mi olduğunun ayırt edilmesi zor olduğundan Hizmet Reddi saldırıları ağdaki sistemler için en tehlikeli saldırı tiplerinden bir tanesidir. Hizmet Reddi saldırılarının tespit edilmesi için anomali tabanlı STS'ler kullanılabilir[28]. DDoS saldırıları, bir şirketin web sitesinin temelini oluşturan altyapı gibi ağ kaynaklarının sınırlamalarından yararlanır. Bir DDoS saldırısı, hedeflenen çevrimiçi kaynağa, kapasitesini aşırı yükleyecek ve böylece arızalanmasına neden olacak birçok istekle doldurmayı içerir. Hedefe akan trafik, farklı kaynaklardan gelir ve tek bir kaynağı engelleyerek saldırıyı durdurmayı zorlaştırır [13]. Şekil.3'te Dünya genelindeki Hizmet Reddi saldırı sayılarının yıllara göre dağılımı verilmiştir[10].



Şekil 3. 2018-2023 yılları arası DOS saldırı sayıları[10]

- Remote to Local-R2L: Bu saldırı tipinde hedef sistemde kullanıcı haklarına sahip değilken sistemde oturum açılarak bilgi elde etmek amaçlanmaktadır[21]
- User toRoot–U2R: Hedef sistemde normal kullanıcı yetkilerine sahipken kullanıcı yetkisini yönetici(administrator, root) yetkilerine yükseltmeye dayalı bir saldırı türüdür.

STS hedeflenen sisteme yapılan bu gibi izinsiz müdahalelerin önceden, olay anında veya saldırı sonrası işletim sisteminin günlüğünden veya sistemin bulunduğu ağın trafiğinden alınan bilgilerin çeşitli yöntemler ile analiz edilerek kullanıcıya bildirilmesidir. STS'nin öncelik amacı bu müdahaleleri önlemek ve olası saldırıların kayıt altına alınarak yetkili kişileri veya kullanıcıyı bilgilendirmeye dayanır[9].

STS tasarlanırken ağ trafiği incelenerek ağda yaşanan davranışlardan veriler oluşturulmaktadır. Ağdaki bu veriler kullanılarak sistem eğitildikten sonra anormal durumların belirlenmesi için bir yöntem belirlenmektedir. Yapılan eğitimler ve tasarımlar ile sistemin ve kullanıcının güvenliğinin üst düzeyde sağlanması amaçlanmaktadır. Bu amaçla son yıllarda saldırının tespit edilmesi için makine öğrenmesi algoritmalarının kullanımı bir hayli artmıştır. Makine öğrenmesi yöntemlerinde yer alan algoritmaların başarılı sonuçları STS'nin tasarımlarında kullanılan bileşenlere ışık tutmaktadır. Makine öğrenmesi algoritmalarıyla elde edilen başarılı sonuçlar sayesinde STS üzerinde iyileştirmeler ve düzenlemelere gidilebilir. Çalışma sonucunda elde edilen veriler ile saldırıların tespit edilerek sınıflandırılmasına olanak sağlayacaktır. STS geliştiricileri bu verileri kullanarak savunma sistemlerini geliştirebilir.

Yapılan literatür taramasında Makine öğrenmesi ve derin öğrenme yöntemleri ile STS ve ağda yaşanan anomalilerin sınıflandırılması üzerine yapılan benzer çalışmalardan bazıları incelenmiştir. Çavuşoğlu ve Kaçar yaptıkları çalışmada NSL-KDD veri setinde ilk defa farklı niteliklerden çıkarım yaparak yeni veri seti oluşturmuşlardır. Ağdaki trafiklerin saldırı mı ve normal bir trafik mi olduğunu analiz edip sınıflandırmıştır. Veri setine uyguladıkları farklı makine öğrenmesi yöntemlerinden en yüksek doğruluk oranı elde eden iki algoritmanın %99,33 ile KNN ve %99,38 ile J48 algoritmalarının olduğunu söylemişlerdir[15]. Cemile ve arkadaşları ağdaki davranışların normal veya saldırı şeklinde sınıflandırılması için NSL-KDD veri setine Destek vektör makinaları (DVM), rastgele orman (RO), aşırı öğrenme makineleri-Extreme Learning Machines (ELM) ve KNN (k en yakın komşu) yöntemlerini uygulayıp sonuçları doğruluk ve F1-Skor parametrelerine göre kıyaslamışlardır. Çalışmalarında en iyi performansı %99,8 doğruluk değeri, %99,9 F1-Skor değeri ile ELM göstermiştir[11]. Şahingöz ve arkadaşları NSL-KDD veri setine Decisiontree, RO, KNN, adaboost, yapay sinir ağları, gradyan artırma, doğrusal ayrımcılık analizi gibi yöntemleri uygulayarak, sonuçları doğruluk oranı, eğitim süreleri ve çalıştırma süreleri açısından değerlendirmişlerdir. Çalışma sonucunda en başarılı yöntemin %99,88 doğruluk oranıyla Adaboost yönteminin elde ettiği gözlemlenmiştir[37]. Bıçakcı ve Toklu yaptıkları çalışmada kullandıkları NSL-KDD veri seti üzerine hibrit bir öznitelik azaltma yöntemini uygulamışlardır. Çalışmalarında öznitelik çıkartma yönteminden Yığılmış Otomatik Kodlayıcı ve öznitelik seçme yönteminden de Select K-Best Features(SelectKBest) yöntemini kullanmışlardır. Bıçakcı ve Toklu öznitelik azaltma yöntemlerini veri setine uyguladıktan sonra Rastgele Orman ve Destek Vektör Makineleri yöntemleri ile ikili sınıflandırma işlemi yapmıştır. Yaptıkları çalışma sonucunda en yüksek doğruluk oranının %98,67 ile Rastgele orman modelinin elde ettiğini belirtmişlerdir. Önerdikleri model düşük frekansta olan U2R ve R2L saldırılarının tespit edememektedir[8]. Erdem ve Özgür NSL-KDD veri seti üzerinde makine öğrenmesi yöntemlerinden DecisionTree, Lojistik Regresyon (LR), Yapay Sinir Ağları (YSA), KNN ve DVM modellerini kullanarak istifleme tekniğinin sınıflandırılması için çalışma yapmışlardır. Çalışmalarında %90,57 doğruluk oranı elde eden en başarılı yöntemin DVM olduğunu belirtmişlerdir[19]. Baykan ve Khorram

yaptıkları çalışmada KNN, DVM ve RO makine öğrenme algoritmalarını kullanarak algoritmaların sınıflandırma doğruluğunu artırmak için Parçacık Sürü Optimizasyonu (PSO) ve Yapay Arı Kolonisi (YAK) tekniklerini kullanmışlardır. Çalışmalarında ağ trafiğini kötü amaçlı trafik ve normal trafik şeklinde sınıflandırmışlardır. NSL-KDD veri setine uyguladıkları makine öğrenme yöntemlerinden %99,8 doğruluk oranı ile en yüksek başarı elde eden yöntemin KNN olduğunu belirtmişlerdir[7]. Nur yaptığı çalışmada UNSW-NB15 ve NSL-KDD veri seti üzerine bulut üzerine yapılan saldırıların tespit edilerek sistemin güvenliğini artırmak için Gri Kurt Optimizasyon (GKO) ve Yapay Sinir Ağları'na dayalı yeni bir hibrit STS önermiştir. Yaptığı çalışmada NaiveBayes(NB) makine öğrenmesi yönteminin en başarılı yöntem olduğunu belirtmiştir. NB algoritması ile saldırıların tespit edilmesi için yaptığı sınıflandırmada NSL-KDD ve UNSW-NB15 veri setlerinde % 97,83 ve % 98,21 doğruluk oranı elde etmiştir[32]. Ekici ve Takcı yaptıkları çalışmada CICIDS 2017 veri setine Yinelemeli Özellik Elemesi, İleri Yönelimli Seçim, RO, DecisionTree, NB, Lojistik Regresyon ve Ekstrem Gradyan Artırma (XGB) gibi makine öğrenmesi algoritmalarını uygulamışlardır. Makine öğrenmesi yöntemleri ile elde ettikleri sonuçları Doğruluk, Duyarlık, Kesinlik ve F1-Skor gibi metrikler ile değerlendirmişlerdir. Doğruluk oranına göre en başarılı sonucun % 94 ile RF, kesinlik, duyarlılık ve F1-Skora göre en başarılı sonucun ise %93 ile RF hem de XGB algoritmalarının elde ettiğini belirtmişlerdir[18]. Aygün yaptığı çalışmada KDDTest+ veri setini iki farklı kategori altında incelenmiş ve sonuçları doğruluk oranı ile kıyaslamıştır. Deterministik otomatik kodlayıcı yönteminden %88,28 gürültü giderici otomatik kodlayıcı yönteminden ise %88,65 doğruluk oranı elde ettiğini belirtmiştir[5]. Ahmetoğlu ve Resul CIC-IDS2017 ve CSE-CICIDS2018 veri setlerini birleştirerek yeni bir veri seti oluşturmuştur. Oluşturdukları yeni veri setine ikili sınıflandırma için çeşitli makine öğrenme yöntemlerini uygulamışlardır. Uyguladıkları algoritmalar içinde en başarılı sonuçları %99'luk bir doğruluk oranı ve F1-Skor ile RF ve DNN algoritmalarının gösterdiğini belirtmişlerdir[1]. Karaman ve arkadaşları CSE-CIC-IDS2018 veri setindeki saldırıları sınıflandırırken Botnet, DDOS, DOS, BruteForce saldırılarını ele almışlardır. Veri seti üzerinde makine öğrenmesi yöntemlerinden olan yapay sinir ağlarını kullanarak %99,11 doğruluk oranı elde etmişlerdir[27]. Çakır ve Angin KDD CUP'99 veri setinde ikili sınıflandırma çalışması yapmışlardır. Yaptıkları çalışmada derin öğrenme metotlarından Uzun Kısa Vadeli Hafıza Ağları (LSTM), zamansal evrişimli ağlar (TCN) ile başarı oranlarını kıyaslamışlardır. İkili sınıflandırmada LSTM'nin %97, TCN' nin %96 doğruluk oranı elde ettiğini belirtmişlerdir[14]. Kurt yaptığı çalışmada NSL-KDD veri seti üzerinde literatürde az kullanılan CatBoost algoritmasının performansını, RandomForest ve AdaBoost algoritmalarının performanslarıyla karşılaştırmıştır. Catboost algoritması %79,43 doğruluk ve %96,95 duyarlılık oranı elde etmiştir[29]. Bakour ve arkadaşları Darpa veri seti üzerinde Tabu Araması (TS) ve Genetik Algoritma (GA) kullanarak Hibrit bir STS önermişlerdir. Tasarladıkları Genetik algoritma %98,4 tespit oranı, %99,37 tespit doğruluğu elde etmiştir[6]. Esmaeili ve arkadaşları DDOS saldırılarını belirlemek için makine öğrenmesi modellerinden Çok Katmanlı Algılayıcı, KNN, Lineer Diskriminant Analizi, Rastgele Orman, Çift Yönlü Uzun Kısa Süreli Bellek(BiLSTM) yöntemlerini kullanmışlardır. NSL-KDD veri setinde yapmış oldukları çalışmada saldırıları belirlemek için ikili sınıflandırma yapmışlardır. Oluşturdukları modellerin başarılarını ise Doğruluk oranlarına göre belirlemişlerdir. Test sonuçlarında en yüksek başarıya %82,3 doğruluk oranıyla BiLSTM ile ulaşmışlardır[22]. Varetta yaptığı çalışmada NSL-KDD veri setini kullanarak ağda yaşanan saldırıları belirlemek için çoklu sınıflandıran bir model tasarlamıştır. Model oluşturmak için Karar Ağacı, Rastgele Orman ve Destek Vektör Makinesi yöntemlerini kullanmıştır. Kullandığı makine öğrenmesi yöntemlerinin başarılarını karşılaştırırken Doğruluk oranlarını baz almıştır. Çalışmasında en yüksek doğruluk oranını Rastgele Orman yöntemi ile elde etmiştir. Oluşturduğu model ağdaki saldırılardan DoS saldırılarını %0.87 doğruluk oranı ile, U2R saldırılarını ise %0.98 doğruluk oranı ile sınıflandırmıştır[43].

Yapılan literatür taramasında NSL-KDD veri seti ile çalışılan çoğu çalışmanın ikili sınıflandırma üzerine yapıldığı gözlemlenmiştir. Çalışmalarda oluşturulan modellerin başarılarının karşılaştırılmasında Doğruluk oranı kullanılmıştır. İncelenen çalışmalarda oluşturulan modellerinin başarılarının karşılaştırılması için karışıklık

matrisleri kullanılmamıştır. Bu çalışmada ise ağ akışında yaşanan anomali olaylar makine öğrenmesi algoritmaları ile analiz edilerek ağdaki trafiğin saldırı olup olmadığının tespiti için ikili sınıflandırma yapılmıştır. Ayrıca bu çalışmada ağdaki trafiğin incelenmesi sonucu trafiğin normal trafik, DOS saldırı, probe saldırı veya remoteatack olup olmadığını çoklu sınıflandıran bir model tasarlanmıştır. İkili ve çoklu sınıflandırmalar için STS'de sıkça kullanılan veri setlerinden biri olan Kaggle platformundan alınan NSL- KDD veri seti kullanılmıştır[41]. Saldırıların tespit edilmesi için kullanılan bu veri setine topluluk öğrenme algoritmalarından RO, DecisionTree, XGB, KNN, Bagging ve Yapay Sinir Ağları uygulanarak elde edilen sonuçların karşılaştırmalı analizi yapılmıştır. Bu çalışmada ayrıca klasik makine öğrenmesi yöntemlerinden daha başarılı bir sonuç alınıp alınmayacağının belirlenmesi için Yapay Sinir Ağları(YSA) yöntemi kullanılarak bir model oluşturulmuş ve doğruluk oranları kıyaslanmıştır. Çalışmada oluşturulan modellerin başarıları karşılaştırılırken Doğruluk, F1-Skor ve Karışıklık Matrisleri sonuçları ölçüt olarak kullanılmıştır. Kullanılan makine öğrenmesi yöntemlerinin başarısını artırmak için standart parametreler kullanılmamış olup, makine öğrenmesi yöntemlerinin parametreleri değiştirilerek elde edilen sonuçlar karşılaştırılmıştır.

II. MATERYAL VE YÖNTEM

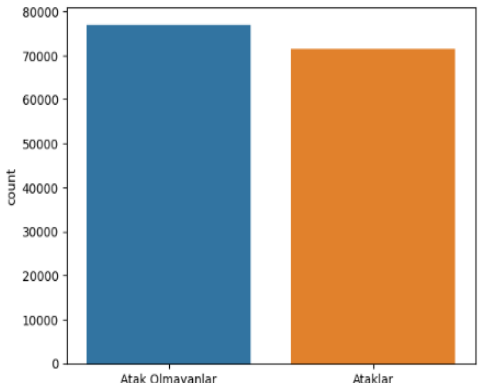
A. Kullanılan Veri Seti

Yapılan bu çalışmada Kaggle sitesindeki NSL-KDD veri setindeki KDDTrain+ ve KDDTest+ veri setlerindeki verilerin tamamı birleştirilerek yeni bir veri seti oluşturulmuştur[41]. Elde edilen veri seti 148515 satır ve 43 sütundan(öznitelik) oluşmaktadır.

B. Veri Setinde Yapılan İşlemler

a. İkili sınıflandırma

Veri setinde yapılan analizlerde boş(null) değerlerin olmadığı gözlemlenmiştir. Veri setindeki kategorik özniteliklerden attack_state özniteliğine saldırı durumlarını 1, saldırı olmayan durumlarını 0 ile gösteren labelencoding işlemi yapılmıştır. Saldırı durumlarının sayısını gösteren grafik Şekil.4'te verilmiştir.



Şekil 4. Ağdaki normal trafik ve saldırı trafiği

Veri dağılımı yakınsak olduğundan veri dengeleme işleminin uygulanmasına gerek duyulmamıştır. Oluşturulan modelin Doğruluk oranını artırmak için veri setindeki özniteliklere normalizasyon yöntemi uygulanmıştır. Veri setindeki özniteliklerin korelasyon matrisi oluşturularak incelenmiştir. Model oluşturmak için öznitelik seçerken en yüksek korelasyona sahip öznitelikler seçilmiştir.

b. Multi-class sınıflandırma

Veri setindeki R2L ve U2R saldırıları birleştirilerek RemoteAttack isminde yeni bir öznitelik tanımlanmış ve veri setine eklenmiştir. Hazırlanan veri seti DoS/DDoS saldırıları sınıfı altında 11 farklı saldırı, Probe saldırı sınıfı altında 6 farklı saldırı, Remote saldırı sınıfının altında ise 22 farklı saldırı türü içermektedir. Bu işlem sonucu ile biri normal diğeri saldırılar olmak üzere dört sınıf içeren bir veri seti oluşturulmuştur. Veri setindeki saldırı sınıflarının isimleri Tablo.1' de verilmiştir.

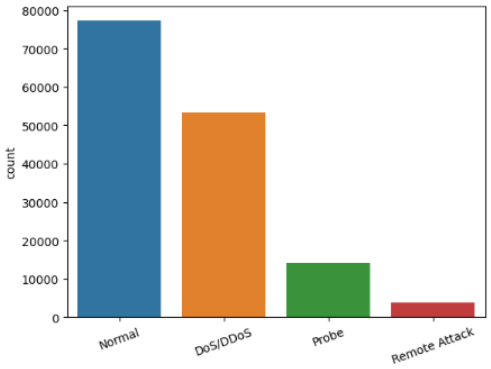
Veri setindeki attack_class özniteliği kategorik veri olduğu için Integer Encoding yöntemiyle saldırı sınıflarına göre integer değerlere çevrilmiştir. 0 değeri normal trafiği, 1 değeri Dos saldırıları, 2 değeri probe saldırıları, 3 değeri ise Remote saldırıları temsil etmektedir. Saldırı sınıflarının kategorilere göre dağılımları Şekil.5'te verilmiştir. Ayrıca, veri setindeki protokol türlerine göre saldırı tipleri incelenmiş ve saldırı sayıları Şekil.6'da verilmiştir.

Veri setindeki protocol_type, service ve flag öznitelikleri kategorik değerlere sahip olduğundan Integer Encoding işlemi ile Integer değerlere dönüştürülmüştür. Öznitelikler arasında en yüksek korelasyona sahip olanların korelasyon matrisi Şekil.7'de verilmiştir. Modellerin başarısını artırmak için veri setindeki öznitelikler korelasyon matrisine göre seçilmiştir. Aralarında düşük korelasyon olan öznitelikler veri setinden çıkarılmış

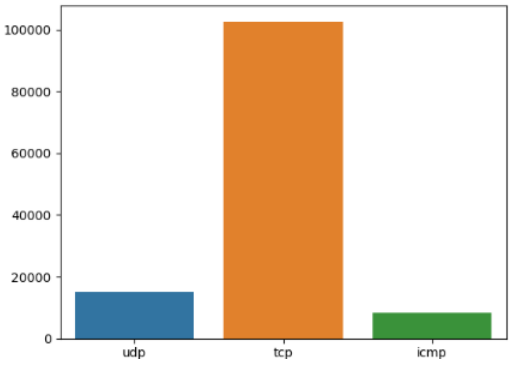
veri seti üzerindeki özniteliklere min-max normalizasyon yöntemi de uygulandıktan sonra veri setine makine öğrenmesi metotları uygulanmıştır.

Tablo 1. Veri setindeki saldırı sınıflarının isimleri

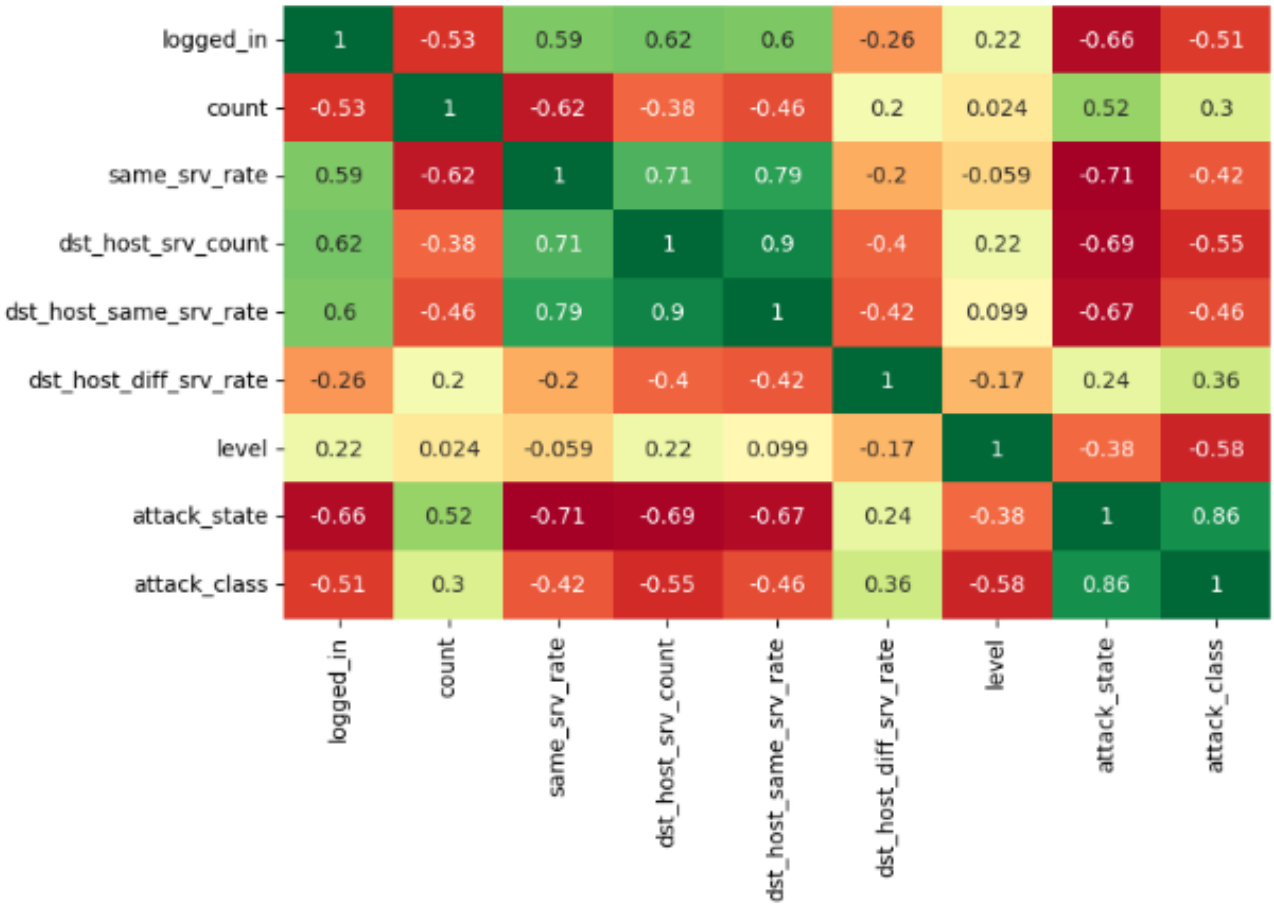
DOS	Probe	Remote Attack	
apache2 back land neptunemailbomb pod processtable smurf teardropudpstorm worm	Ipsweep Mscan Nmap Portsweep Saint Satan	buffer_overflow loadmdoule perl ps rootkit sqlattack xterm ftp_write guess_passwd http_tunnel imap	multihop named phf sendmail snmpgetattack snmpguess spy warezclient warezmaster xclock xsnoop



Şekil 5. Saldırı sınıfları dağılımı



Şekil 6. Protokol tipine göre saldırı sayıları



Şekil 7. En yüksek korelasyona sahip öznitelikler

C. Makine Öğrenmesi

Makine öğrenmesi ile bilgisayarlar insanlar gibi yeni bir şeyi öğrenebilirler. Makine öğrenmesi çeşitli öğrenme algoritmaları veya modelleri kullanılarak bilgisayarlara öğrenme becerisi katan bilimsel bir çalışma alanıdır. Bilgisayarların analitik veri modelleri oluşturmasına ve net bir şekilde kodlanmadan otomatik olarak gizli iç görüler bulmasına izin verir. Veriler üzerine matematiksel ve istatistiksel işlemler uygulayarak tahminlerden çıkarımlar yaparak öğrenme metotları oluşturulur[4]. Makine öğrenmesinde çözüme ulaşmak için farklı matematiksel ve istatiksel yöntemler kullanılır. Her bir yöntem makine öğrenmesinde kullanılan veri kümesine uygun olarak seçilmelidir[20].

a. K-en yakın komşu (KNN)

KNN algoritması 1950 yılında ortaya çıkmıştır[16]. KNN algoritması örnek tabanlı olup sınıflandırma veya regresyon problemleri için kullanılabilir[45]. KNN algoritması nesnelerin birbirleri arasındaki yakınlık ilişkilerine göre kümeleme işlemi yapılmasına dayanmaktadır. Yani "örnek bir olayın sonucunun kendisine en yakın olan komşusunun olayın sonucu ile aynı olur" ilkesine dayanmaktadır. Eldeki vaka sonuçları eğitim grubundaki en yakın komşularındaki vaka sonuçlarının ortalamasına eşit olacaktır. Tahminleme veya sınıflandırma problemlerinde tahmin edilen veya sınıflandırılan bağımlı değişken değeri hesaplanırken, bağımsız değişkenin *k* sayıdaki komşu bağımsız değerlerinin aritmetik ortalaması alınarak hesaplanır[3]. Algoritma, iki noktayı birleştiren düz bir yol olan Öklid mesafesini kullanır. KNN algoritmasını bir veri setine uygulamadan önce, veri setinin hazırlanması, yani veri setinin parametrelerinin normalize edilmesi gerekir. A ve B noktaları arasındaki Öklid mesafesi, onları birleştiren doğru parçasının uzunluğudur[35]. Öklid mesafesi için kullanılan formül denklem 1'de verilmiştir. Noktalar arası uzaklıklar hesaplandıktan sonra, her bir eleman birbirlerine göre sıralanır ve gelen değer, en uygun sınıfa atanır.

$$d(A,B)=\sqrt{(x1-x2)^2+(y2-y1)^2} \quad (1)$$

b. Decisiontree

Karar ağacı algoritmasında kullanılan yöntem böl ve fethet yöntemidir. Aynı zamanda karar ağacı algoritması seçim ağacı olarak bilinmektedir. Bu yöntemde, en iyi tahmin yapılırken bağımlı ve bağımsız değişkenler arasındaki bütün ilişkiler araştırılır. En güçlü ilişkiye sahip değişken belirlendiği zaman veri seti bu değişkenin değerine göre ayrılır ve olası bölünmeler bitene kadar devam edilir. Karar ağacı, dalları üzerinde genişleyen ağaç yapısı oluşturmaktadır. Ağaç temel bir düğümle başlar ve bu düğüme kök düğüm denir. Karar ağacı yönteminde kök düğüm belirlenir. Kök düğüm belirlendikten sonra en yukarıdan en aşağıya doğru düğümler oluşturulur[38].

c. Bagging(torbalama)

BaggingBootstrapaggregation kelimelerinin kısaltması olup ağaç topluluğu algoritmalarından biridir. Bu yöntemde veri seti içerisinden tekrarlı bir şekilde örnekler çekilerek yeni ağaçlar oluşturulur. Oluşturulan bu ağaçlardan ortaya yeni bir ağaç topluluğu ortaya çıkartılır. Bagging yöntemi veri setindeki rastgele seçtiği verilerden alt kümeler oluşturur, oluşturduğu bu alt kümelerden de bir model oluşturur. Modeller birbirinden bağımsızdır ve birbirine paralel olarak çalışır. Tüm modellerden gelen tahminler birleştirilerek nihai tahmin yapılır. Bagging artık optimizasyonuna dayalı performans arttırmaya çalışır[12].

d. Randomforest(RO)

Rastgele orman yöntemi çok sayıda karar ağacı oluşturma ilkesine dayanır. Oluşturulan bu ağaçlar üzerinden değerlendirme yapılabilmektedir. RO yönteminde karar ağaçlarının oluştuğu yapılara orman (forest) denmektedir. Orman içindeki karar ağaçları, veri setinden örneklem gruplarının seçilmesi ve her düğüm noktasındaki değişkenlerin rastgele belirlenmesiyle oluşturulmaktadır[33]. Yani veri seti rassal küçük gruplara ayrılarak karar ağaçları oluşturulur. Tahminleme veya sınıflandırma aşamalarında veri setinden oluşturulan karar ağaçlarının tahminlerinin ortalaması alınır[2].

e. Gradientboosting

GradientBoosting hem sınıflandırma hem de tahminleme problemleri için çalışan bir topluluk makine öğrenme algoritmasıdır. Zayıf sınıflandırıcı gruplar birleştirilerek güçlü bir sınıflandırıcı oluşturmak için yükseltme tekniği kullanılır. Temel sınıflandırıcı için kullanılan sınıflandırma ağaçları önceki ağaçlarda hesaplanan hatalar üzerine kuruludur[17].

f. Parametre optimizasyonu

Makine öğrenmesi yöntemleri kullanılırken yüksek başarı elde etmek için girilmesi gereken her bir parametreye hiper-parametrelerin optimizasyonu denmektedir. Bu optimizasyon yöntemi çalışmanın maliyetini düşürmekte ve makine öğrenmesi metotlarının performansını pozitif yönde artırmaktadır. Parametreler öğrenilebilir parametre ve hiper parametre olmak üzere ikiye ayrılmaktadır. Modelin öğrenilebilir parametreleri veri setindeki verilerden tahmin edilip öğrenilebilen değerlerdir. Bu yüzden tasarımcı model parametrelerini ayarlamaz. Hiper-parametreler ise verilerden değil tasarımcı tarafından ayarlanan parametrelerdir[40]. Makine öğrenmesi yöntemlerinde GridSearch veya RandomSearch gibi yöntemler hiper-parametre değerini belirlemek için kullanılmaktadır[36].

g. Yapay sinir ağları

Yapay sinir ağları (YSA), insan beyninden esinlenerek öğrenme aktivitelerini taklit etmektedir. YSA İnsan beyninin öğrenme, hatırlama, muhakeme etme yolu ile topladığı bilgilerden yeni bir bilgi üretebilmesi gibi temel işlemlerin gerçekleştirildiği bilgisayar yazılımlarıdır. Yeni bir bilginin/davranışın öğrenilmesi sürecinde matematik işlemlerinin modellenmesi ile ortaya çıkmıştır[26]. Örneğin, beyindeki nesne sınıflandırması, birçok karmaşık doğrusal-doğrusal olmayan işlem katmanı aracılığıyla gerçekleştirilir. Derin öğrenme, sinir ağı modellerini karmaşık görevler ve davranışlar üzerinde doğrudan eğiterek, başka türlü modellemesi neredeyse imkânsız olabilecek beyin işlevleri için aday modelleri verimli bir şekilde oluşturmanın bir yolunu sağlar[46]. Yapay Sinir Ağları (YSA), girdi ve çıktı arasında gizli bir katmanın bulunduğu ve derin sinir ağının (diğer adıyla derin öğrenme), katmanların üst üste istiflendiği birden fazla gizli katman kullandığı bir mimaridir. Bir katman düğümlerden oluşur. Her düğüm beyindeki gibi önceki ve sonraki katmanlardaki düğümlere bağlıdır. Her girdi, her nörondaki bir ağırlıkla çarpılır. Çarpılan çıktılar bir sonraki katman için girdidir[44].

III. BULGULAR

Deneysel çalışmalarda kullanılan makine öğrenmesi yöntemlerinin değerlendirilmesi Python programlama dili ile gerçekleştirilmiştir. Yöntemlerin kullanıldığı donanım olarak Intel Core i7 7500U işlemci ve 8 GB DDR4 belleğe sahip bir bilgisayar kullanılmıştır. Veri setindeki verilerin %70'i Train, %30'u Test verisi olarak ayrılmıştır. Test verilerinin olduğu veri seti ve train verilerinin olduğu veri setine DecisionTree, GradientBoosting, Bagging, KNN, XgBoosting ve Randomforest makine öğrenmesi yöntemleri uygulanmıştır. Makine öğrenmesi yöntemleri değerlendirilirken doğruluk, kesinlik ve F1-Skor değerleri ve karışıklık matrisleri sonuçları kullanılmıştır.

A. İkili Sınıflandırma

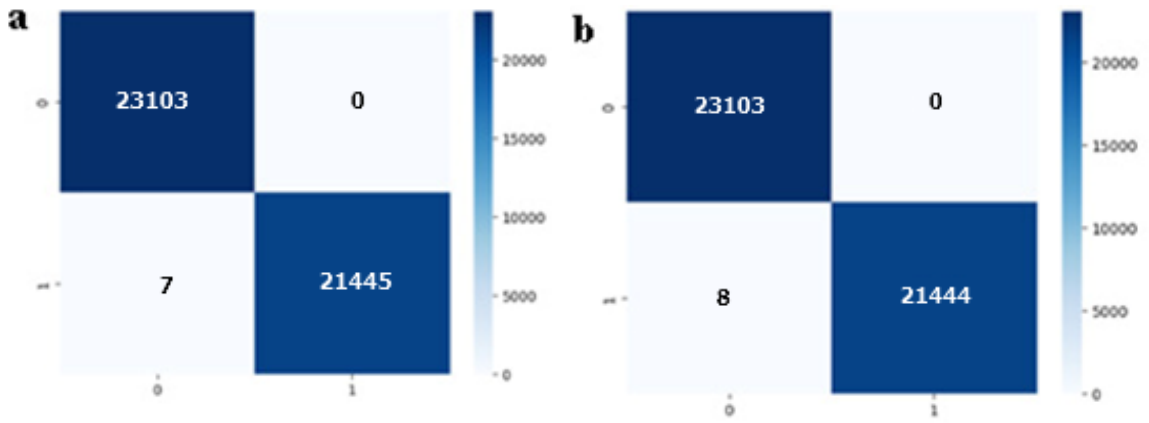
Çalışmanın bu aşamasında ağdaki trafiğin saldırı mı yoksa normal bir trafik mi olduğu analiz edilerek ikili sınıflandırma yapılmıştır. İkili sınıflandırma için makine öğrenmesi yöntemlerinden elde edilen sonuçların bilgisi Tablo.2' de verilmiştir.

DecisionTree, RandomForest, GradientBoosting, Bagging, xGBoost ve KNN yöntemlerinin sonuçları doğruluk ve F1-Skorları açısından incelendiğinde %99,984 doğruluk ve % 99,984 F1-Skor sonucu elde eden Bagging yöntemi ikili sınıflandırmada en yüksek başarıyı elde eden yöntem olmuştur. En başarılı iki yöntemin karışıklık matrisleri Şekil.9'da verilmiştir.

Tablo 2. Veri seti için ikili sınıflandırma performansları

Makine Öğrenmesi Yöntemleri	***Doğruluk***	***Hassasiyet***	***Duyarlılık***	***F1-Score***
Decision Tree	0.99977	0.99978	0.99977	0. 99977
Random Forest	0.99979	0.99980	0.99979	0.99979
K Neighbors	0.99937	0. 99937	0. 99936	0. 99937
Gradient Boosting	0.99982	0.99982	0.99981	0.99982
Bagging	*0.99984*	*0. 99984*	*0. 99983*	*0. 99984*
Xg Boosting	0.99979	0.99980	0. 99979	0. 99979

Veri setine optuna kütüphanesi ile gözetimli öğrenme algoritmalarından ArtificialNeural Network (Yapay Sinir Ağları) uygulanmış ve ikili sınıflandırma yapılmıştır. Neural network için parametre optimizasyonları belirlenmiş ve çalışma sonunda elde edilen en iyi sonuçlar Tablo.4'te verilmiştir.



Şekil 9. En başarılı 2 yöntem için karışıklık matrisi sonuçları (a) Bagging (b) GradientBoosting

Tablo 3. Yapay sinir ağları ikili sınıflandırma sonuçları

Makine Öğrenmesi Yöntemleri	Denenen Parametreler	Kullanılan Parametreler	Doğruluk	Hassasiyet	Duyarlılık	F1-Score
Yapay Sinir Ağları	hidden_layer_sizes= [(50,50,50), (50,100,50), (100,50)] activation=['relu','tanh','logistic'] learning_rate=['constant','adaptive'] solver= ['adam', 'sgd'] max_iter=[200,500, 1000, 5000] learning_rate_init=[0.1,0.01, 0.001, 0.0001]	hidden_layer_sizes= (50, 50, 50) activation=tanhlearning_rate=adaptive solver=sgdmax_iter=500 learning_rate_init= 0.001	0.999483	0.999485	0.999480	0.999483

Yapay Sinir Ağlarından(YSA) elde edilen ikili sınıflandırma sonuçları literatürdeki çalışmalardan daha yüksek doğruluk oranı ve F-1 skor elde etmiştir. YSA sonuçları klasik makine öğrenmesi algoritmaları ile doğruluk ve F1-Skorlar açısından karşılaştırıldığında daha düşük doğruluk oranı ve F1-Skoru elde etmiştir.

B. Çoklu(Multi-Class) Sınıflandırma

Çalışmanın bu aşamasında ağda yaşanan saldırıların türünü belirlemek için çoklu sınıflandırma yapılmıştır. Veri seti dosyası %70 eğitim ve %30 test verisi olarak ayarlandıktan sonra makine öğrenmesi yöntemleri uygulanarak çoklu sınıflandırma yapılmış ve Tablo.4'teki sonuçlar elde edilmiştir.

Yapay sinir ağlarından elde edilen sonuçlar klasik makine öğrenmesi algoritmaları ile doğruluk ve F1-Skorlar açısından karşılaştırıldığında daha düşük doğruluk oranı ve F1-Skoru elde etmiştir.

Tablo 4.Veri setine uygulanan multi-class sınıflandırma sonuçları

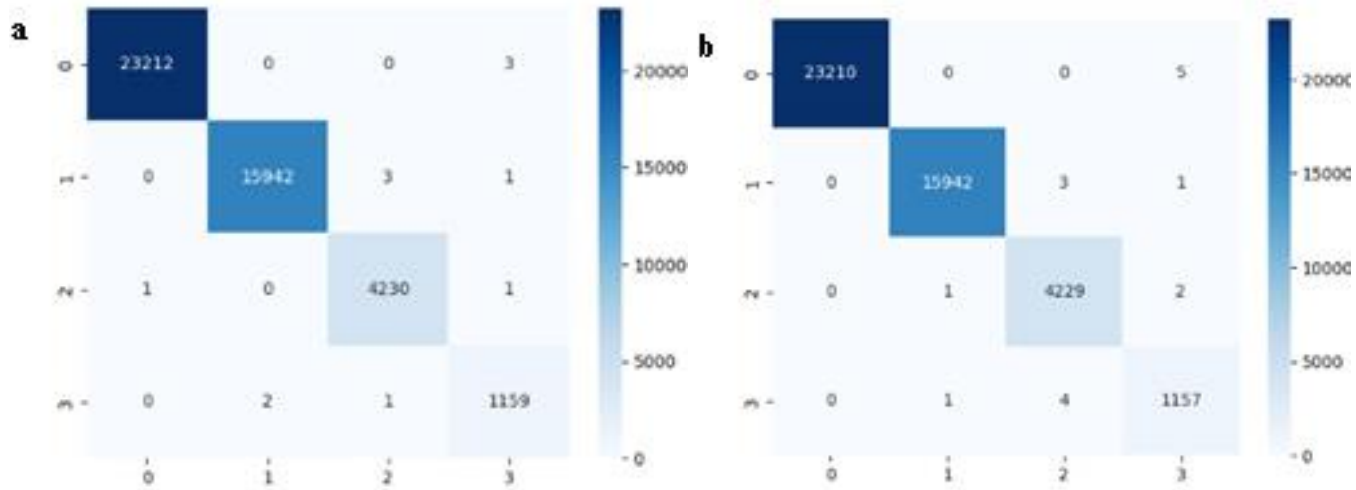
Makine Öğrenmesi Yöntemleri	Doğruluk	Hassasiyet	Duyarlılık	F1-Score
Decision Tree	0.9991	0.9967	0.9973	0.9970
Random Forest	0.9996	0.9978	0.9986	0.9982
K Neighbors	0.9979	0.9953	0.9933	0.9943
Gradient Boosting	0.9991	0.9966	0.9958	0.9962
Bagging	0.9992	0.9968	0.9973	0.9971
xGBoost	**0.9997**	**0.9986**	**0.9991**	**0.9988**

IV. TARTIŞMA

Yöntem sonuçları makine öğrenmesi yöntemleri kullanılarak ağda yaşanan trafiğin saldırı mı yoksa normal bir trafik mi olduğunun başarılı bir şekilde analiz edilebildiğini ve belirlendiğini göstermiştir. Tablo.5'teki veriler incelendiğinde %99,97 doğruluk ve %99,88 F1-Skor oranıyla xGBoost yöntemi en başarılı sonuçları elde etmiştir. xGBoost

yönteminin karışıklık matrisi sonuçlarına bakıldığında saldırıların neredeyse tamamını doğru sınıflandırmıştır. Doğruluk oranlarına bakıldığında xGBoost yöntemini %99,96 doğruluk oranıyla RandomForest, %99,92 doğruluk oranıyla Bagging yöntemleri takip etmiştir. En başarılı iki yöntemin karışıklık matrisleri Şekil.10'da verilmiştir. Oluşturulan modellerin karışıklık matrisleri sonuçlarına bakıldığında modellerin saldırıları sınıflandırırken çok az hata yaptığı görülmektedir. Yapılan saldırı sayılarının fazla olması göz önünde bulundurulduğunda saldırıların çoğunu doğru tespit edebildiğinden, modelin yaptığı bu küçük hatalar veri ve sistem güvenliğinin sağlanması hususunda görmezden gelinebilir. Sonuç olarak eğitilen model saldırıların çoğunu tespit edebilmiştir.

Veri setine optuna kütüphanesi ile gözetimli öğrenme algoritmalarından ArtificialNeural Network (Yapay Sinir Ağları) uygulanmış ve çoklu sınıflandırma yapılmıştır. Neural network için parametre optimizasyonları belirlenmiş ve çalışma sonunda elde edilen en iyi sonuçlar Tablo.5'te verilmiştir.



Şekil 10. En başarılı iki yöntemin karışıklık matrisleri sonuçları (a) XGBoost (b) RandomForest

Tablo 5. Yapay sinir ağları multi-class sınıflandırma sonuçları

Makine Öğrenmesi Yöntemleri	Denenen Parametreler	Kullanılan Parametreler	Doğruluk	Hassasiyet	Duyarlılık	F1-Score
Yapay Sinir Ağları	hidden_layer_sizes= [(50,50,50), (50,100,50), (100,50)] activation=['relu','tanh','logistic'] learning_rate=['constant','adaptive'] solver= ['adam', 'sgd'] max_iter=[200,500, 1000, 5000] learning_rate_init=[0.1,0.01, 0.001, 0.0001]	hidden_layer_sizes= (50, 100, 50) activation=tanhlearning_rate=constantsolver=adam max_iter=200 learning_rate_init= 0.01	0.9971	0.9894	0.99	0.99

V. SONUÇLAR

Ağ akışında yaşanan anomali olaylar, makine öğrenmesi algoritmaları ile analiz edilerek ağdaki trafiğin saldırı olup olmadığının tespiti için ikili sınıflandırılmış olup saldırı olarak sınıflandırılan trafik saldırı türlerine göre sınıflandırılma yapılmıştır. Sınıflandırmalar için STS'de sıkça kullanılan veri setlerinden biri olan Kaggle platformundan alınan NSL-KDD veri seti kullanılmıştır. Saldırıların belirlenmesi için en yaygın kullanıma sahip bu veri seti üzerinde makine

öğrenmesi algoritmalarından RO, GradientBoosting, DecisionTree, XGB, KNN, Bagging ve Yapay Sinir Ağları yöntemleri denenerek karşılaştırmalı analizleri yapılmıştır. Yapılan bu çalışmada Kaggle sitesindeki NSL-KDD verisetindeki KDDTrain+ ve KDDTest+ verisetlerinin tamamı kullanılarak yeni bir veri seti oluşturulmuştur[41]. Veri setinde ağ trafiğinin saldırı mı yoksa normal trafik mi olduğu ikili sınıflandırma ile yapılmış saldırı tespitinde en yüksek başarıyı gösteren yöntem %0.99984 doğruluk oranı ile Bagging yöntemi olmuştur. Saldırı türlerinin belirlenmesi için ise Multi Class sınıflandırma yapılmış, sınıflandırmada en başarılı yöntem %0.9997doğruluk oranı ile xGBoost Machine olmuştur. Multi Class sınıflandırmada Yapay Sinir Ağları algoritması kullanılmış ve YSA literatürdeki çalışmalardan yüksek doğruluk oranı elde etmesine karşın çalışmadaki klasik makine öğrenmesi modellerinden düşük başarı elde edilmiştir. NSL-KDD veri seti kullanılarak saldırıların tespit edilmesi ve sınıflandırılması ile yapılan çalışmalar incelenmiştir. Literatürdeki yapılan çalışmalar incelendiğinde ağdaki trafiğin ikili sınıflandırma yapılarak, saldırı olup olmadığının belirlenmesindeki en başarılı sonuç ELM yöntemi ile elde edilmiştir. ELM yöntemi ile %99,9 doğruluk oranının elde edildiği gözlemlenmiştir. Bu çalışmada ise Bagging yöntemi kullanılarak ikili sınıflandırma için %0.99984 doğruluk oranı elde edilmiştir. Bagging yönteminde doğruluk oranını artırmak çeşitli parametreler denenmiş ve Bagging yönteminden optimum doğruluk oranı elde etmek için kullanılmıştır. Bu çalışmada doğruluk oranını artırmak için veri setindeki özniteliklerin korelasyon matrisi incelenerek öznitelik seçimi yapılmış ve veri setinde bulunan 43 öznitelik 26'ya düşürülmüştür. Düşürülen öznitelik sayısı doğruluk oranının yükselmesini sağlamıştır. Yapılan litretatür taramasında ağdaki saldırıların tespit edilmesi için çoklu sınıflandırma yapılan az sayıda çalışmaya rastlanmıştır. Yapılan bu çalışmalarda makine öğrenmesi yöntemlerinin klasik parametreleri kullanılmıştır. Bu çalışma hem ikili sınıflandırma hem de multi-class sınıflandırma içerdiğinden STS tasarımı için önemli bilgiler ve modeller içermektedir. Çalışmada kullanılan tüm makine öğrenmesi yöntemlerinde K-fold cross validasyon parametresi olarak 2 ve 10 arasında CV değerleri denenmiş ve en ideal doğrulama validasyonu olarak 10 seçilmiştir. Çalışmada saldırıların tespitinde en başarılı yöntem olan Bagging ile oluşturulan model STS'nin tasarımlarında kullanılan bileşenlere ışık tutacaktır. Makine öğrenmesi algoritmaları, veri setlerinde yer alan verileri kullanarak, saldırıların sınıflandırılması için anlamlı ve işe yarayacak bilgiler sunmaktadır. Makine öğrenmesi algoritmalarıyla elde edilen başarılı sonuçlar sayesinde STS üzerinde iyileştirmeler ve düzenlemelere gidilebilir. Çalışma sonucunda elde edilen veriler ile saldırıların tespit edilerek sınıflandırılmasına olanak sağlayacaktır. STS geliştiricileri oluşturulan modeli kullanarak savunma sistemlerini geliştirebilecektir. Bu çalışmada saldırıların tespit edilmesindeki başarıyı artırmak için klasik makine öğrenmesi ve Yapay Sinir Ağı yöntemleri kullanılarak hem ikili hem de çoklu sınıflandırmalar yapılmış, makine öğrenmesi yöntemleri için standart parametreler kullanılmamış farklı parametreler denenerek en iyi parametreler belirlenmiştir. Makine öğrenmesi yöntemleri için belirlenen en uygun parametreler oluşturulan modelin doğruluk oranını ve F-1 skor değerlerini olumlu ölçüde artırdığı ve sonuç olarak ağdaki saldırıları en iyi şekilde tespit ettiği gözlemlenmiştir. Karışıklık matrisi sonuçlarına bakıldığında STS için yüksek hassasiyetli bir model oluşturulduğu sonucuna ulaşılabilir. Yapılan bu çalışma ileride bu tür çalışma yapacak kişiler için önemli veriler içermektedir. Sonuç olarak makine öğrenmesi yöntemleri kullanılarak oluşturulan modellerin başarılarını artırmak için standart parametreler kullanmak yerine parametre değerleri ile oynanarak en iyi modeller oluşturulabilir.

TEŞEKKÜR

Çalışmamda kaynak olarak kullandığım NSL-KDD veri setini hazırlayıp Kaggle platformunda paylaşan M.Tavallaee ve arkadaşlarına[41] sonsuz teşekkürler!

SON NOT

Bu çalışma "Derin öğrenme teknikleri" dersi kapsamında yapılmıştır.

KAYNAKLAR

[1] Ahmetoğlu, H.,& Resul, D. (2021). Makine Öğrenmesi Yöntemleri Kullanarak Web Uygulama Saldırılarının

Tespitinde Genetik Öznitelik Seçimi Yaklaşımı. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, *14*(2), 109-119.
[2] Akşehir, Z. D., & Kılıç, E. (2019). Makine Öğrenmesi Teknikleri ile Banka Hisse Senetlerinin Fiyat Tahmini. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi.
[3] Altunkaynak, A., Başakın, E. E., & Kartal, E. (2020). Dalgacık k-en yakın komşuluk yöntemi ile hava kirliliği tahmini. *Uludağ University Journal of The Faculty of Engineering*, *25*(3), 1547-1556.
[4] E. Alpaydın, Introduction to machine learning. The MIT Press, 3-6, 2004.
[5] Aygün, R. C. (2017). *Derin öğrenme yöntemleri ile bilgisayar ağlarında güvenliğe yönelik anormallik tespiti* (Doctoral dissertation).
[6] Bakour, K., Daş, G. S., & Ünver, H. M. (2017, October). An intrusion detection system based on a hybrid Tabu-genetic algorithm. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 215-220). Ieee.
[7] Baykan, N. A.,& Khorram, T. (2021). Network Intrusion Detection using Optimized Machine Learning Algorithms. *Avrupa Bilim ve Teknoloji Dergisi*, (25), 463-474.
[8] Bıçakcı, M. S.,& Toklu, S. (2022). Bilgisayar Ağı Güvenliği için Hibrit Öznitelik Azaltma ile Makine Öğrenmesine Dayalı Bir Saldırı Tespit Sistemi Tasarımı. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, *11*(3), 203-220.
[9] Can, E. (2007). *Gerçek zamanlı veriler yardımı ile karar veren bir bilgisayar ağı saldırı tespit sisteminin tasarlanması ve gerçeklenmesi*(Yüksek lisans tezi, Yıldız Teknik Üniversitesi).
[10] Cisco Annual Internet Report (2018–2023) White Paper,(accessedJune 11, 2020). https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html?dtid=osscdc000283 Erişim tarihi: 02.01.2023
[11] Cemile, İ., Kenan, İ., & Hanbay, D. (2021). Saldırı Tespit Sistemlerinde Sınıflandırma Yöntemlerinin Kıyaslanması. *Computer Science*, *6*(1), 1-10.
[12] Chen, T., & Ren, J. (2009). Bagging for Gaussian process regression. *Neuro computing*, *72*(7-9), 1605-1610.
[13] Y. Cui, Q. Qian, C. Guo et al., "Towards DDoS detection mechanisms in software-defined networking," Journal of Network and Computer Applications, vol. 190, p. 103156, 2021.
[14] Çakır, B.,&Angin, P. (2021). Zamansal Evrişimli Ağlarla Saldırı Tespiti: Karşılaştırmalı Bir Analiz. *Avrupa Bilim ve Teknoloji Dergisi*, (22), 204-211.
[15] Çavuşoğlu, Ü.,& Kaçar, S. (2019). Anormal trafik tespiti için veri madenciliği algoritmalarının performans analizi. *Academic Platform-Journal of Engineering and Science*, *7*(2), 205-216.
[16] Dilki G. ve Ö. Deniz Başar. 2020, İşletmelerin finans tahmininde k – en yakın komşu algoritması üzerinden uzaklık ölçütlerinin karşılaştırılması. İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 19(38), 224-233, 2020.
[17] Einziger, G., Goldstein, M., Sa'ar, Y., &Segall, I. (2019, July). Verifying robustness of gradient boosted models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, No. 01, pp. 2446-2453).
[18] Ekici, B.,& Takcı, H. (2022). Bilgisayar Ağlarında Anomali Tespiti Yaklaşımı ile Saldırı Tespiti. *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi*, *22*(5), 1016-1027.
[19] Erdem, H.,&Ozgur, A. (2018). Feature selection and multiple classifier fusion using genetic algorithms in intrusion detection systems.
[20] Özkaya, U., Öztürk, Ş., & Barstugan, M. (2020). Coronavirus (COVID-19) classification using deep features fusion and ranking technique. Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach, 281-295.
[21] Erol, M. (2005). Saldırı Tespit Sistemlerinde İstatistiksel Anormallik Belirleme Kullanımı. *İTÜ Bilgisayar Müh. Bölümü, İstanbul*.
[22] Esmaeili, M., Goki, S. H., Masjidi, B. H. K., Sameh, M., Gharagozlou, H., & Mohammed, A. S. (2022). ML-DDoSnet: IoT Intrusion Detection Based on Denial-of-Service Attacks Using Machine Learning Methods and NSL-KDD. *Wireless Communications and Mobile Computing*, *2022*.
[23] A. Hussain, J. Heidemann, C. Papadopoulos, 2004, "Disting ushing between Singnal and Multisource Attacks Using Signal Processing", Computer Networks, vol. 46.
[24] İlya, K., Keser, S. B., & Yolaçan, E. (2021). Saldırı Tespit Sistemlerinde Topluluk Öğrenme Yöntemlerinin Kıyaslanması. *Avrupa Bilim ve Teknoloji Dergisi*, (31), 725-734.
[25] Individuals using the Internet: ITU,(2021). https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx, erişim tarihi : 29/12/2022
[26] Kabalcı, E. (2014). Yapay Sinir Ağları. *Ders notları*.
[27] Karaman, M. S., Turan, M., & Aydın, M. A. (2020). Yapay sinir ağı kullanılarak anomali tabanlı saldırı tespit modeli uygulaması. *Avrupa Bilim ve Teknoloji Dergisi*, (Ejosat Ek Özel Sayı (HORA)), 10-17.
[28] K. Kendall, Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, MIT Department of Electrical Engineering and Computer Science, 1999.
[29] Kurt, A. (2021). *Ağ Tabanlı Saldırı Tespit Sistemlerinde Topluluk Öğrenme Yöntemlerinin Karşılaştırmalı Performans Analizi* (Master'sthesis, Sakarya Üniversitesi).
[30] Maltarollo, V. G., Honório, K. M. ve Ferreira da Silva, A. B.,"Applications of Artificial Neural Networks in Chemical Problems", In: Suziki, K. (eds), Intech, (2013).
[31] Niu, Z., Guo, W., Xue, J., Wang, Y., Kong, Z. ve Huang, L. (2023). Topluluk yarı denetimli aktif öğrenmeye (ADESSA) dayalı yeni bir anormallik algılama yaklaşımı. *Bilgisayar ve Güvenlik* , *129* , 103190.
[32] Nur, I. M. (2021). *Gri kurt optimizasyon (GKO) algoritması ve yapay sinir ağı (YSA) kullanılarak hibrit bulut tabanlı saldırı tespit ve yanıt sistemi* (Master'sthesis, Konya Teknik Üniversitesi).
[33] Özdemir, S. (2018). RandomForest Yöntemi kullanılarak potansiyel dağılım modellemesi ve haritalaması: Yukarıgökdere Yöresi örneği. *Turkish Journal of Forestry*, *19*(1), 51-56.
[34] Özgür, A., & Erdem, H. (2012). Saldırı tespit sistemlerinde kullanılan kolay erişilen makine öğrenme algoritmalarının karşılaştırılması. *Bilişim Teknolojileri Dergisi*, *5*(2), 41-48.

[35] Pandey, A., & Jain, A. (2017). Comparative analysis of KNN algorithm using various normalization techniques. *International Journal of Computer Network and Information Security*, *11*(11), 36.
[36] Pulat, M., &Kocakoç, İ. D. (2021). Türkiye'de Makine Öğrenmesi ve Karar Ağaçları Alanında Yayınlanmış Tezlerin Bibliyometrik Analizi. *Yönetim ve Ekonomi Dergisi*, *28*(2), 287-308.
[37] Sahingoz, O. K., Çebi, C. B., Bulut, F. S., Fırat, H., & Karataş, G. (2019). Saldırı Tespit Sistemlerinde Makine Öğrenmesi Modellerinin Karşılaştırılması. *Erzincan Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *12*(3), 1513-1525.
[38] Şentürk A., Veri Madenciliği Kavram ve Teknikler, Ekin Yayınevi, 2006
[39] Tanrıkulu, H., & Sazlı, M. H. (2009). *Saldırı tespit sistemlerinde yapay sinir ağlarının kullanılması* (Doctoral dissertation, Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara).
[40] Tanyıldızı, E., ve Demirtaş, F. (2019). Hiper Parametre Optimizasyonu. In 2019 1st International Informatics and Software Engineering Conference (UBMYK) (pp. 1-5). IEEE.
[41] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligencefor Security and Defense Applications (CISDA), 2009.)
[42] Toprak, H. (2021). *Akıllı saldırı tespit sistemleri* (Master'sthesis, Batman Üniversitesi Fen Bilimleri Enstitüsü).
[43] Venkatesan, S. (2023). Design an Intrusion Detection System based on Feature Selection Using ML Algorithms. *Mathematical Statistician and Engineering Applications*, *72*(1), 702-710.
[44] Woldseth, R. V., Aage, N., Bærentzen, J. A., & Sigmund, O. (2022). On the use of artificial neural networks in topology optimisation. *Structural and Multidisciplinary Optimization*, *65*(10), 294.
[45] Y. Kırelli, E-Ticaret siteleri için sahtekarlık tespit sistemleri. (Yüksek Lisans Tezi), İstanbul Ticaret Üniversitesi, Bilgisayar Mühendisliği Ana Bilim Dalı, İstanbul, 2016.

[46] Yang, G. R., & Wang, X. J. (2020). Artificial neural networks for neuroscientists: a primer. *Neuron*, *107*(6), 1048-1070.