Avrupa Bilim ve Teknoloji Dergisi
Sayı 41, S. 349-356, Kasım 2022

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 41, pp. 349-356, November 2022

Research Article

Dengesiz ML-Tabanlı NIDS Veri Setlerinin Sınıflandırma Performanslarının Karşılaştırılması

Güneş Harman[1*], Emine Cengiz[2]

[1*] Yalova Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Yalova, Türkiye (ORCID: 0000-0001-5413-124X), gunes.guclu@yalova.edu.tr
[2]Yalova Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Yalova, Türkiye (ORCID: 0000-0002-6695-9500), emine.cengiz@yalova.edu.tr

(İlk Geliş Tarihi 5 Ağustos 2022 ve Kabul Tarihi 20 Kasım 2022)

(**DOI:** 10.31590/ejosat.1157441)

ATIF/REFERENCE: Harman, G., Cengiz, E. (2022). Dengesiz ML-Tabanlı NIDS Veri Setlerinin Sınıflandırma Performanslarının Karşılaştırılması. *Avrupa Bilim ve Teknoloji Dergisi*, (41), 349-356.

Öz

Ağ tabanlı Saldırı Tespit Sistemleri (NIDS), ağda bulunan tüm cihazlardan gelen trafiği izlemek ve analiz etmek için kullanılır. Makine Öğrenimi (ML) tabanlı NIDS, günümüzde bilgisayar ağlarını siber saldırılara karşı korumak için önemli araçlardan biridir. ML tabanlı NIDS'in eğitimi ve değerlendirilmesi için ağ veri özellikleri önemli bir etkiye sahiptir. Bu nedenle ML modelinin doğruluğunu ve performansını değerlendirmek için birden çok veri kümesinin ortak temel özellik kümesi içermesi gerekir. Bu çalışmada ortak NetFlow özelliklerine sahip NIDS veri setleri (NF-UNSW-NB15, NF-BoT-IoT, NF-ToN-IoT ve NF-CSE-CIC-IDS2018) kullanılarak ikili sınıflandırma yapılmıştır. Veri setlerindeki saldırı ve normal akış (saldırı yok) sınıfları dengesiz dağılım göstermektedir. Bunun üstesinden gelmek için Rastgele Alt Örnekleme yöntemi kullanılmıştır. Sınıflandırma yöntemleri olarak Rastgele Orman, K-En Yakın Komşuluk, Destek Vektör Makineleri ve Yapay Sinir Ağları algoritmaları kullanılmıştır. Farklı veri setlerinin yeniden örneklenmiş durumlarına, ML yöntemleri kullanılarak doğruluk ve performansları karşılaştırılmıştır. Bu çalışma kapsamında kullanılmış olan dört veri seti içinde en iyi sonucu Rastgele Orman algoritması vermiştir.

Anahtar Kelimeler: Ağ Saldırı Tespit Sistemleri, Makine Öğrenmesi, NetFlow

Comparison of Classification Performances of Imbalanced ML-Based Nıds Datasets

Abstract

Network Based Intrusion Detection Systems (NIDS) are used to track and analyze traffic from all devices on the network. Nowadays Machine Learning (ML) based NIDS is one of the important tools to protect computer networks against cyber attacks. Network data characteristics have a significant impact for training and evaluation of ML-based NIDS. Therefore, to evaluate the accuracy and performance of the ML model, multiple datasets must contain a common core set of features. In this study, binary classification was performed using NIDS datasets (NF-UNSW-NB15, NF-BoT-IoT, NF-ToN-IoT and NF-CSE-CIC-IDS2018) with common NetFlow features. The attack and benign classes in the datasets show an unbalanced distribution. To overcome this, the Random Undersampling method was used. Random Forest, K-Nearest Neighbors, Support Vector Machines and Artificial Neural Networks were used as classification methods. The accuracy and performance of different datasets were compared to the resampled cases using ML methods. As a result of the study, the Random Forest algorithm gave the best result for all four data sets.

Keywords: Network Intrusion Detection Systems, Machine Learning, NetFlow

[*] Sorumlu Yazar: gunes.guclu@yalova.edu.tr

1. Giriş

Günümüzde, çeşitli teknolojik uygulamaların performansını ve verimliliğini artırmak için Makine Öğrenimi (ML) yöntemleri kullanılmaktadır (Ghahramani, 2015). ML modelleri, uzmanlar tarafından gerçekleştirilemeyen karmaşık veri kalıplarını çıkarma ve öğrenme konusunda üstün yeteneklere sahiptir (Sarhan, Layeghy, Gallagher & Portmann, 2021). Bu modeller geleneksel bilgi işlem algoritmalarına göre daha iyi performans göstermektedir. Bu durum ML'nin kullanım alanını oldukça yaygınlaştırmıştır. Siber güvenlik alanında kullanılan modellerin amacı, kurumların güvenliğini geliştirmek ve güçlendirmek içindir (Buczak & Guven, 2015). Bu modeller bilgisayar ağlarının tehditlere karşı korunmasında kullanılmış (Apruzzese vd., 2018) ve gelişmiş algılama yetenekleri gerektiren karmaşık modern saldırılarını tespit etmiştir (Alrashdi vd., 2019).

Ağ tabanlı Saldırı Tespit Sistemleri (NIDS), gelen trafiği saldırı veya normal akış olarak sınıflandırarak ağ veri davranışlarını öğrenmeyi, aynı zamanda dijital ağları siber tehditlerden korumayı amaçlamıştır (Garcia-Teodoro vd., 2009). Geleneksel NIDS'ler, saldırı imzalarını gelen trafik imzalarıyla eşleştirerek, bilinen saldırılara karşı yüksek tespit doğruluğu sağlar (Garuba vd., 2008). Ancak, bu sistemler sıfır gün saldırıları olarak bilinen bilgisayar ağlarına yönelik görünmeyen tehditleri veya bilinen saldırıların yeni türevlerini tespit edemez (Garcia-Teodoro vd., 2009). Bu nedenle saldırı davranışlarını öğrenmek ve ağdaki izinsiz girişleri tespit etmek için ML yöntemlerini uygulanır (Sinclair vd., 1999). Bu alanda birçok ML tabanlı NIDS geliştirilmiş ve belirli veri kümelerine uygulandığında çoğunlukla yüksek doğruluk elde edilmiştir (Sarhan, Layeghy & Portmann, 2021). Tüm veri kümeleri için standart bir özellik kümesine sahip olması, saldırı senaryolarında önerilen ML modellerinin güvenilir bir şekilde değerlendirilmesi açısından oldukça önemlidir. Bu durum aynı zamanda modelin genellenebilirliğinin ve ağ senaryolarında performansın değerlendirilmesini sağlar.

Bir bilgisayar ağının trafiğini izlemek ve analiz etmek; normal olmayan olayların hemen fark edilmesi, tıkanıklık sorunları, donanım tarafında meydana gelen hatalar ve güvenlik olayları gibi sorunların krize dönüşmeden fark edilip çözüm üretilmesini sağlar. Ağ trafiğini izlemede kullanılan araçlardan birisi de NetFlow'dur. NetFlow, yaygın olarak kullanılan bir ağ trafiği toplama için endüstri standardı bir protokolüdür (Claise vd., 2004). 3. Katman ve sonrasında çalışır. NetFlow özellikleri, ağ saldırılarının tanımlanmasında önemli bir yere sahiptir.

Sınıflama performansını etkileyen unsurlar arasında birçok etken bulunmakla beraber; veri seti veya veri setlerinde kullanılan özellikler, kullanılan sınıflandırma algoritmaları ve sınıfların dağılımı en önemli kriterlerin başında gelmektedir. Bazı durumlarda bir veri setinde sınıflandırma algoritmalarını kullanmak tek başına yeterli olmayabilir. Bu amaçla sınıflandırma algoritmalarını daha iyi değerlendirebilmek için aynı özelliklere sahip farklı veri setleri kullanılmıştır. Ayrıca veri setlerindeki dengesiz örnek dağılımı aşırı öğrenmeye sebep olmaktadır. Bunun üstesinden gelmek için yeniden örnekleme yapılmıştır.

Bu çalışmada ortak NetFlow özelliklerine sahip NIDS veri setleri (NF-UNSW-NB15, NF-BoT-IoT, NF-ToN-IoT ve NF-CSE-CIC-IDS2018) kullanılarak ikili sınıflandırma yapılmıştır. Veri setlerindeki dengesiz sınıf dağılımını ortadan kaldırmak için Rastgele Alt Örnekleme yöntemi kullanılmıştır. Sınıflandırma yöntemleri olarak Rastgele Orman (RO), K-En Yakın Komşuluk (KNN), Destek Vektör Makineleri (DVM) ve Yapay Sinir Ağları (YSA) algoritmaları kullanılmıştır. Farklı veri setlerinin yeniden örneklenmiş durumlarına, ML yöntemleri uygulanarak doğruluk ve performansları karşılaştırılmıştır. Bu çalışma kapsamında kullanılmış olan dört veri seti içinde en iyi sonucu RO algoritması vermiştir. KNN algoritması için en iyi doğruluk oranı (%98.7), NF-UNSW-NB15 ve NF-CSE-CIC-IDS2018 veri setlerinde elde edilmiştir.

DVM algoritmasında en iyi sonuç (%96.8) NF-BoT-IoT veri setinde, YSA algoritmasında ise en iyi doğruluk oranı (%96.9) NF-CSE-CIC-IDS2018 veri setinde elde edilmiştir. Çalışma sonucunda ise en düşük performans (%60.7) NF-BoT-IoT veri setinde YSA algoritmasında elde edilmiştir. Bu çalışmanın temel amacı; aynı özelliklere sahip farklı veri setleri kullanılarak sınıflandırma algoritmalarının performanslarını karşılaştırmaktır. Kullanılan veri setleri dengesiz sınıf dağılımına sahip olduğundan, modelin verimini artırmak için dengesizlik oranı azaltılmalıdır. Bu oranı azaltmak için çoğunluk gruplarının veri büyüklüğünü azaltan bir veri örnekleme modeli kullanılmıştır.

Yapılan çalışmanın ilerleyen bölümlerinde bahsedilen konular şu şekildedir. İkinci bölümde bu alanda yapılmış olan çalışmalardan kısaca bahsedilmiştir. Üçüncü bölüm çalışma kapsamında kullanılan veri setlerinden ve ML yöntemleri hakkında bilgi vermektedir. Dördüncü bölüm sınıflandırma algoritmalarının performans değerlendirmeleri ve sonuçlarını içermektedir. Son olarak sonuç bölümünde ise kullanılan sınıflandırma algoritmalarının ve veri setlerinin değerlendirmesi yapılmıştır.

2. Literatür Taraması

Bu bölümde, NIDS özellik kümelerinin genellenebilirliğini açıklamaya ve değerlendirmeye çalışan çalışmalardan bahsedilmektedir. ML modelinin veri setleri arasında performansının değerlendirilmesinin yapılabilmesi için ortak bir temel özellik kümesinin olması gerekir. Bunun için Sarhan vd. (Sarhan, Layeghy, Moustafa & Portmann, 2020) aynı 12 NetFlow tabanlı özelliği paylaşan dört veri kümesini oluşturdu ve yayınladı. Veri setleri, mevcut NIDS veri setlerinin NetFlow formatına dönüştürülmesiyle oluşturulmuştur. NetFlow özellikleri, paket başlıklarında bulunduklarından dolayı paket incelemesi gerektiren karmaşık özelliklere oranla ağ trafiğinden çıkarılması daha kolaydır. Bu çalışmada Ekstra Ağaç sınıflandırıcı kullanılarak ikili ve çoklu sınıflandırma yapılmıştır.

Yazarlar (Sarhan, Layeghy & Portmann, 2021) önerilen özellik setini (Sarhan, Layeghy, Moustafa & Portmann, 2020) genişleterek toplam 43 NetFlow tabanlı özellik çıkarmışlardır. Oluşturulan ve etiketlenen veri setleri: NF-UNSW-NB15-v2, NF-BoT-IoT-v2, NF-ToN-IoT-v2 ve NF-CSE-CIC-IDS2018-v2 dir. Ortak özellik kümeleri, gelecekteki NIDS veri kümelerinde kullanılmak üzere önerilmektedir. Özellik kümesini değerlendirmek için Ekstra Ağaç sınıflandırıcısı kullanılmıştır. Bu çalışmalarında yazarlar, daha önce oluşturdukları (Sarhan, Layeghy, Moustafa & Portmann, 2020) temel NetFlow veri kümelerini ve veri kümelerinin orijinal özellik kümelerini karşılaştırmışlardır. Önerilen NetFlow özellik seti, tüm veri setlerinde saldırı tespit doğruluğu açısından kıyaslandığında diğer özellik setlerinden daha iyi performans göstermektedir.

Sarhan vd. sıfır gün olarak da bilinen görünmeyen saldırıların tespitinde ML tabanlı NIDS'in performansını değerlendirmek için Sıfır Atış Öğrenmesi tabanlı bir yöntem önermişlerdir. ML modeli olarak Çok Katmanlı Algılayıcı ve Rastgele Orman (RO)

algoritmaları kullanılmıştır. NIDS veri setlerinden UNSW-NB15 ve NF-UNSW-NB15-v2 kullanılmıştır. Performans değerlenmesinde standart metriklere ek olarak, Sıfır Gün Tespit Oranı (Zero-Day Detection Rate) adı verilen yeni bir değerlendirme metriği tanımlamışlardır. Sonuçlar, ML tabanlı NIDS'in, sıfırıncı gün saldırılarına karşı önemli ölçüde koruma yeteneğine sahip olduğunu göstermiştir (Sarhan, Layeghy, Gallagher & Portmann, 2021).

Sarhan vd. ortak bir özelliğin farklı ağ ortamlarına ve saldırı senaryolarına genelleştirilebilirliğini üzerine çalışmışlardır. İki özellik setini (NetFlow ve CICFlowMeter), üç temel veri setinde (CSE-CIC-IDS2018, BoT-IoT ve ToN-IoT) doğruluk açısından değerlendirilmiştir. Veri setlerinde bulunan ağ veri akışlarını sınıflandırmak için Derin İleri Besleneli Sinir Ağı ve RO sınıflandırıcıları kullanılmıştır. Sonuçlar, ML modelleri algılama doğruluğunu geliştirmede NetFlow özelliğinin üstünlüğünü göstermektedir. Ayrıca, öğrenme modellerinin karmaşıklığı nedeniyle, ML modellerinin sınıflandırma kararlarını açıklamak ve yorumlamak için Yapay Zeka metodolojisi olan SHapley Additive ExPlanations (SHAP) yöntemi kullanılmıştır. İki ortak özellik setinin Shapley değerleri, her bir özelliğin ML tahminine katkısını belirlemek için çoklu veri setlerinde analiz edilmiştir (Sarhan, Layeghy & Portmann, 2021).

Meftah vd. UNSW-NB15 veri setini kullanarak iki aşamalı anomali tabanlı bir ağ saldırı tespit sistemi üzerine çalışmışlardır. En iyi veri kümesi özelliklerini seçmek için Özyinelemeli Özellik Yok Etme ve RO kullanılmıştır. Logistik Regression, Gradyan Artırma ve Destek Vektör Makinesi (DVM) yöntemleri kullanılarak izinsiz trafiği ve normal trafiği belirlemek için ikili bir sınıflandırma yapılmıştır. Sınıflandırmasının sonuçlarından en iyi performansı DVM (%82.11) vermiştir. Ardından, doğruluk oranını artırmak için DVM'nin çıktısı bir dizi çok terimli sınıflandırıcıya verilmiştir. Daha sonra, saldırıların türünü tahmin etmek için Karar Ağaçları (C5.0), Naive Bayes ve DVM yöntemleri kullanılarak çoklu sınıflandırma yapılmıştır. DVM'nin çıktısı saldırı türünün tahmin etme doğruluğunu artırmak için kullanılmıştır. DVM ile entegre edildikten C5.0 uygulanması en yüksek doğruluk oranını %12 kadar artırdığı gözlenmiştir (Meftah vd., 2019).

Jing ve Chen çalışmalarında izinsiz giriş tespiti için doğrusal olmayan bir ölçekleme yöntemine sahip DVM önermişlerdir. UNSW-NB15 veri seti, önerilen DVM tabanlı modelin etkinliğini göstermek için kullanılır. Hem ikili sınıflandırma hem de çoklu sınıflandırma üzerinde çalışma yapılmıştır. Önerilen yöntemin ikili sınıflandırması için doğruluğu %85.99 ve çoklu sınıflandırma için %75.77 bulunmuştur (Jing ve Chen, 2019).

Karataş vd. CSE-CIC-IDS2018 veri setinin dengesizlik oranını incelemişleridir. Sistemin verimliliğini artırmak için Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE) adı verilen sentetik bir veri üretme modeli kullanılarak dengesizlik oranı düşürülmüştür. ML modeli olarak Karar Ağacı, RO, K-En Yakın Komşu, Adaboost, Gradyan Artırma ve Lineer Diskriminant Analizi kullanılmıştır. Örneklenmiş bir veri kümesinin kullanılması sonucunda modellerin ortalama doğruluğunun %4.01 ile %30.59 arasında arttığı gözlenmiştir (Karataş vd., 2020).

3. Materyal ve Metod

3.1 Veri Seti

Ağları izlemek ve analiz etmek için ağ trafiğindeki bilgileri toplamak gereklidir (Sarhan, Layeghy, Moustafa & Portmann, 2020). Bu durum ya ağ trafiğindeki paketleri yakalayarak ya da akışlar biçiminde ağ paketlerinin bir özetini yakalayarak gerçekleşir. Paket yakalama, ağ ve güvenlik analizi için trafik geçmişine tam erişim sağlar. Bu durum veri depolama için büyük kapasite gerektirebilir. Veri setinin büyük hacimli olması gizlilik ve güvenlik sorunlarını da beraberinde getirir. Diğer bir yöntem ise ağ trafiği özetini akışlar olarak yakalamaktır. Ağ akışı tarafından sağlanan bilgiler hem ağ güvenliği hem de uygun bir ağ planlaması için de bir gereklidir. NetFlow, Cisco tarafından 1996 yılında geliştirilmiştir ve ağ akışlarını temsil etmek için kullanılmıştır (Kerr & Bruins, 2021).

Bu çalışmada kullanılan veri setleri ağ yapılarını temsil eden sanal ağ test yatakları aracılığıyla (virtual network testbeds) oluşturulmuştur. Bu tür veri setlerinin oluşturulmasında farklı saldırı senaryoları yürütülür, ilgili ağ trafiği yakalanır ve ilgili saldırı türü ile etiketlenir. Ek olarak, saldırı olmayan trafiği temsil eden normal ağ trafiği oluşturulur, ağ trafiği yakalanır ve etiketlenir. Hem kötü amaçlı hem de kötü amaçlı olmayan ağ trafiği, yerel paket yakalama (pcap) biçiminde yakalanır. Daha sonra veri akışıyla ilgili bilgileri temsil etmek için 12 veri özelliği nProbe aracı kullanılarak çıkarılmıştır. Tablo 1' de bu çıkarılan özellikler gösterilmiştir.

Tablo 1. NetFlow özellikleri (Table 1. NetFlow features)

Özellik	Açıklama
IPV4_SRC_ADDR	IPv4 kaynak adresi
IPV4_DST_ADDR	IPv4 hedef adresi
L4_SRC_PORT	IPv4 kaynak bağlantı noktası numarası
L4_DST_PORT	IPv4 hedef bağlantı noktası numarası
PROTOCOL	IP protokolü tanımlayıcı baytı
TCP_FLAGS	Tüm TCP bayraklarının kümülatifi
L7_PROTO	Katman 7 protokolü (sayısal)
IN_BYTES	Gelen bayt sayısı
OUT_BYTES	Giden bayt sayısı
IN_PKTS	Gelen paket sayısı
OUT_PKTS	Giden paket sayısı
FLOW_DURATION_MILLISECONDS	Milisaniye cinsinden akış süresi

NF-UNSW-NB15: UNSW-NB15 veri kümesinin NetFlow tabanlı formatı olarak geliştirilmiştir. Toplam veri akışı sayısı 1.623.118'dir. Bunun 72.406'sı (%4.46) saldırı örnekleri ve 1.550.712'si (%95.54) normal akıştır.

NF-BoT-IoT: BoT-IoT veri kümesinin NetFlow tabanlı formatı olarak geliştirilmiştir. Toplam veri akışı sayısı 600.100'dür. Bunun 586.241'i (%97.69) saldırı örneği ve 13.859'u (%2.31) normal akıştır.

NF-ToN-IoT: ToN-IoT veri kümesinin NetFlow tabanlı formatı olarak geliştirilmiştir. Toplam veri akışı sayısı

1.379.274'dür. Bunun 1.108.995'i (%80.4) saldırı örneği ve 270.279'u (%19.6) normal akıştır.

NF-CSE-CIC-IDS2018: CSE-CIC-IDS2018 veri kümesinin NetFlow tabanlı formatı olarak geliştirilmiştir. Toplam akış sayısı 8.392.401'dir. Bunun 1.019.203'ü (%12.14) saldırı örneği ve 7.373.198'i (%87.86) normal akıştır.

3.2. Veri Ön İşleme

Makine öğrenmesi modellerinin performanslarını etkileyen en önemli kriterlerden biri kullanılan veri seti veya veri setlerinin kullanışlı ve belirli bir formatta olmasıdır. Veri ön işleme; eksik değerleri doldurma, aykırı değerleri belirleme ve temizleme, tekrar eden verileri silme, dönüştürme, veri birleştirme, boyut azaltma gibi yöntemleri kullanarak veriyi daha kullanışlı bir hale getirmektedir. Bu çalışmada veri ön işleme iki aşamada gerçekleştirilmiştir.

1. Veri temizleme
2. Yeniden örnekleme

Veri Temizleme

Veri temizleme, veri analizine başlamadan önce verinin doğru olmasını sağlama işlemidir. Verilerin eksik, yanlış veya ilgisiz bölümlerinin belirlenmesi, sonrasında bu kısımların değiştirilmesi veya silinmesi işlemine dayanır. Amaç veri setlerinin bilgi içeriklerinin en iyi şekilde analiz edilmesini sağlamaktır. Bu çalışmanın veri temizleme aşamasında veri setindeki tüm girişlerden ağ saldırılarını sınıflandırırken kullanışlı olmayan (Söderström, 2021), (Wang vd., 2021) dört özellik kaldırılmıştır. Bunlar IPV4_SRC_ADDR, L4_SRC_PORT, IPV4_DST_ADDR ve L4_DST_PORT dir.

Yeniden örnekleme

Algoritmaların performansına etki eden ve genellikle göz ardı edilen en önemli noktalardan biri veri setindeki sınıflar arası dengedir. Veri kümesi dengesizliği, modelin çoğunluk sınıfına önyargılı olacağından dolayı sınıflandırma problemlerinde hatalara yol açabilir. Dengesiz bir veri kümesi üzerinde eğitimin olumsuz etkisine karşı koymak için kullanılan üç ana yöntem vardır. Bu yöntemler; Aşırı Örnekleme, Alt Örnekleme ve Sentetik Veri Üretilmesidir. Aşırı Örnekleme sınıf dağılımları eşit oluncaya kadar azınlık sınıfına ait verilerden rastgele seçilen bir bölümünün örnekleri çoğaltılarak bu sınıfa ait veri sayısı artırılır. Yüksek düzeyde dengesiz sınıf dağılımı olan büyük bir veri setinin hesaplama maliyeti yüksek olabilir. Alt Örneklemede ise sınıf dağılımları eşit olana kadar çoğunluk olan sınıfın örneklerinden rastgele seçilen bir bölümünün silinmesiyle gerçekleşir. Bu yöntem veri boyutu çok büyük olduğu durumlarda kullanışlıdır. Az sayıda örneğe sahip durumlarda ise örnek uzayın rastgeleliği zarar görebilir. Bu yöntem hem veriyi daha dengeli hale getirir hem de veri boyutunu azalttığı için sınıflandırma yönteminin çalışma süresini kısaltabilir. Sentetik Veri Üretilmesi aşırı örnekleme yöntemine benzer yapıdadır. Farkı ise azınlık sınıfına ait verilerin bir bölümünün rastgele üretilmesi yerine yeni verilerin belli bir algoritma ile üretilmesidir. Bu çalışmada ise Rastgele Alt Örnekleme yöntemi kullanılarak sınıf dengesizliği problemi giderilmiştir.

Tablo 2'de Rastgele Alt Örnekleme yapılmadan önce ve yapıldıktan sonra veri setlerinin saldırı ve normal akış sınıflarının dağılımları verilmiştir.

Tablo 2. Veri setlerinin saldırı ve normal akış sınıfları (Table 2. Attack and normal flow classes of datasets)

	Rastgele Alt Örnekleme yapılmadan önce veri dağılımı			**Rastgele Alt Örnekleme yapıldıktan sonra veri dağılımı**		
Veri Setleri	**Toplam veri akış sayısı**	**Saldırı örnek sayısı**	**Normal Akış**	**Toplam veri akış sayısı**	**Saldırı örnek sayısı**	**Normal Akış**
NF-UNSW-NB15	1.623.118	72.406	1.550.712	144.812	72.406	72.406
NF-BoT-IoT	600.100	586.241	13.859	27.718	13.859	13.859
NF-ToN-IoT	1.379.274	1.108.995	270.279	414.558	270.279	270.279
NF-CSE-CIC-IDS2018	8.392.401	1.019.203	7.373.198	2.038.406	1.019.203	1.019.203

3.3 Makine Öğrenmesi Yöntemleri

Saldırı tespit sistemleri için öncelikle ağ akışının normal davranışı belirlenmelidir. Bunu gerçekleştirebilmek için sistemin bir öğrenme algoritması kullanılarak eğitilmesi gerekir. Literatürde birçok ML algoritması bulunmaktadır. Bu çalışmada Rastgele Orman, K-En Yakın Komşuluk, Destek Vektör Makineleri ve Sinir Ağları algoritmaları kullanılmıştır.

3.3.1 Rastgele Orman

Rastgele Orman (RO), sınıflandırma ve regresyon problemleri için kullanılabilen denetimli ML yöntemidir (Belgiu & Drăguţ, 2016), (Kuş vd., 2021). Karar Ağaçlarını kullanarak bir karar ormanı oluşturur. Karar ağaçlarının en büyük problemlerinden biri aşırı öğrenmedir. RO bu problemin üstesinden gelmek için hem veri setinden hem de özellik setinden rastgele alt örnekler seçer ve bunları eğitir. Bu yöntemle karar ağaçları oluşturulur. Daha sonra karar ağaçları bireysel olarak tahminde bulunur. Regresyon probleminde karar ağaçlarının tahminlerinin ortalaması, sınıflandırma probleminde ise tahminler arasında en çok oy alan seçilir.

3.3.2 K-en yakın komşuluk

K-En Yakın Komşu (KNN) Algoritması, denetimli bir ML algoritmasıdır. KNN algoritması eğitim aşamasına sahip değildir (Zhang vd., 2017). KNN iki değer üzerinden tahmin yapar. Bunlar 'uzaklık' ve 'k komşuluk' sayısıdır. Uzaklık değeri, tahmin edilecek noktanın diğer noktalara uzaklığıdır. Uzaklık hesaplanırken genellikle Öklid fonksiyonu kullanılır. Ayrıca Manhattan, Minkowski ve Hamming fonksiyonları da kullanılabilir. Uzaklık hesaplandıktan sonra sıralama yapılır ve bu değer uygun olan sınıfa atanır. En yakın kaç komşu üzerinden hesaplama yapılacağını ise 'k komşuluk' sayısı belirler (Baykan & Khorram, 2021). K=1 olarak seçilirse aşırı öğrenme problemi ortaya çıkar. Çok büyük seçilirse de genel sonuçlar verecektir. Bu yöntemde optimum K değerini belirlemek problemin asıl konusu olarak düşünülebilir.

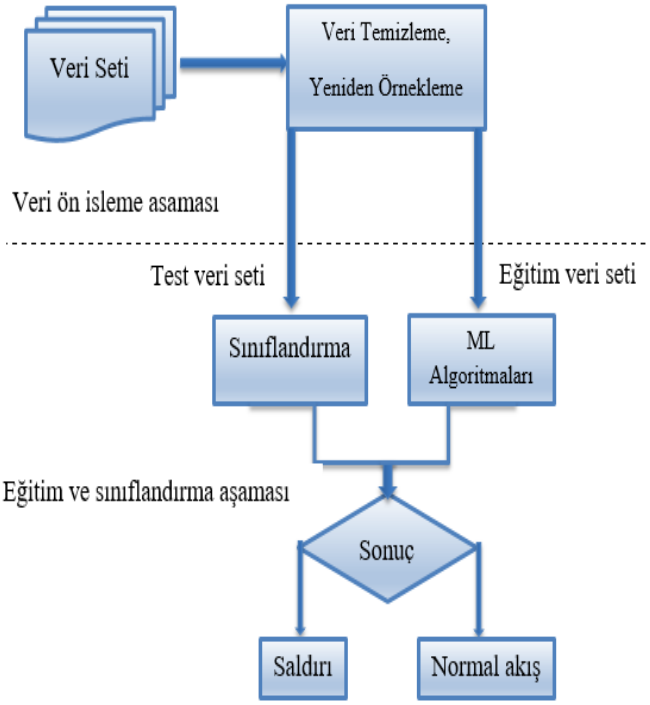
3.3.3 Destek vektör makineleri

Destek Vektör Makineleri (DVM) ilk olarak Vapnik (1995) tarafından sınıflandırma ve regresyon problemlerini çözmek için önerilmiştir (Bamakan vd., 2016). Genellikle sınıflandırma problemlerinde kullanılır. Hem doğrusal hem de doğrusal olmayan veri sınıflandırmada kullanılabilir. Bir düzlem üzerine yerleştirilmiş noktaları birbirinden ayırmak için bir doğru çizer. Bu doğrunun, iki sınıfın noktaları için de maksimum uzaklıkta olması hedeflenir. Karmaşık ama küçük ve orta ölçekteki veri setleri için uygundur. Son zamanlarda, DVM'lere olan yoğun ilgi nedeniyle araştırmacılar tarafından birçok uygulama geliştirilmiştir (Ahmad vd., 2018). DVM, görüntü işleme ve örüntü tanıma uygulamalarında yaygın olarak kullanılmaktadır.

3.3.4 Yapay sinir ağları

Günümüzde birçok makine öğrenmesi modeli mevcuttur. Yapay Sinir Ağları (YSA) bunlardan sadece bir tanesidir. İnsan beyni ve sinir sisteminden esinlenerek tasarlanmış bir modelidir. YSA, katmanlar şeklinde kurulmuş bir yapıdır. İlk katman giriş, son katman çıkış ve orta kısımda bulunan katmanlar ise gizli katman olarak adlandırılır (Çakır & Angin, 2021). YSA'nın yapısında sisteme belli girdiler ve çıkması gereken değerler verilir. Bu girdiler katsayılar ile çarpılıp toplanır. Sonra bu toplam bir aktivasyon fonksiyonuna verilerek bir çıktı elde edilir. Beklenen değer ile sonuç değerinin farkı alarak hata oranı bulunur. Bu hata oranına göre katsayılar güncellenir ve bu işlem döngü şeklinde devam eder. Bu kısımda öğrenme işlemini gerçekleşmiş olur.

Şekil 1' de önerilen modelin aşamaları gösterilmiştir.



Şekil 1. Önerilen modelin akış şeması (Figure 1. Flow chart of the proposed model)

4. Sonuçlar ve Tartışma

Bu bölümde çalışmada kullanılan NetFlow veri setlerine uygulanan ML yöntemlerinin performans sonuçları verilmiştir. Eğitim ve test verilerinin rastgele oluşturulmasından dolayı performans sonuçlarının değişkenliğini azaltmak için K katlamalı Çapraz Doğrulama (K-Fold CrossValidation) yöntemi kullanılmıştır. K değeri 10 olarak seçilmiştir. Çalışmada çapraz doğrulama sırasında yeniden örnekleme yapılmıştır. 10 eşit parçaya bölünen veri kümesindeki bir parça test verisi olarak ayrılmıştır. Geriye kalan 9 parçada yeniden örnekleme yapılmış ve dengeli veride ML yöntemleri eğitilmiştir. Oluşturulan modellerin performans değerlendirilmesi test verisinde yapılmıştır. ML algoritmalarının performans ölçüsünü hesaplamak için Doğruluk, Duyarlılık, Kesinlik, F-Ölçütü ve RocAuc değerleri kullanılmıştır (Sokolova & Lapalme, 2009). Bu değerler denklem [1-4]'e göre hesaplanır.

Doğru Pozitif (True positive - TP): Saldırı örneği olduğu düşünülen durumun gerçekte de saldırı örneği olması.

Yanlış Pozitif (False positive - FP): Saldırı örneği olduğu düşünülen durumun gerçekte normal akış olması.

Yanlış Negatif (False negative -FN): Normal akış olduğu düşünülen durumun gerçekte saldırı örneği olması.

Doğru Negatif (True negative -TN): Normal akış olduğu düşünülen durumun gerçekte de normal akış olması.

Doğruluk (Accuracy): Doğru olarak sınıflandırılan örneklerin yüzdesidir.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

Kesinlik (Precision): Pozitif olarak tahmin edilen değerlerin gerçekte kaç tanesinin pozitif olduğunu göstermektedir.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (2)$$

Duyarlılık (Recall): Pozitif olarak tahmin etmemiz gereken değerlerin ne kadarını pozitif olarak tahmin ettiğimizi gösteren bir metriktir.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (3)$$

F1 Skor: Kesinlik ve Duyarlılık değerlerinin harmonik ortalamasını göstermektedir.

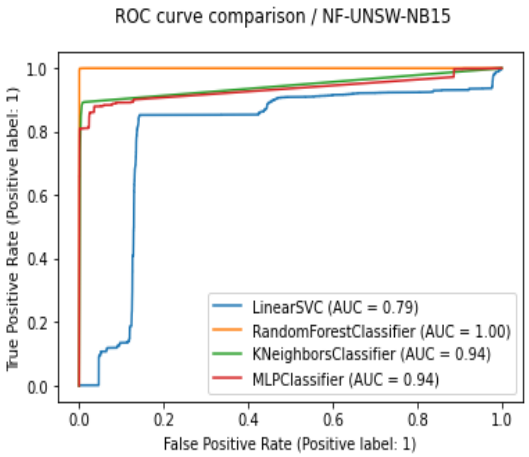
$$F_1 = 2 * \frac{\text{kesinlik} * \text{duyarlılık}}{\text{kesinlik+duyarlılık}} \quad (4)$$

Tablo 3' de dört farklı veri setine uygulanan ML yöntemlerinin doğruluk ve performans değerleri gösterilmiştir.

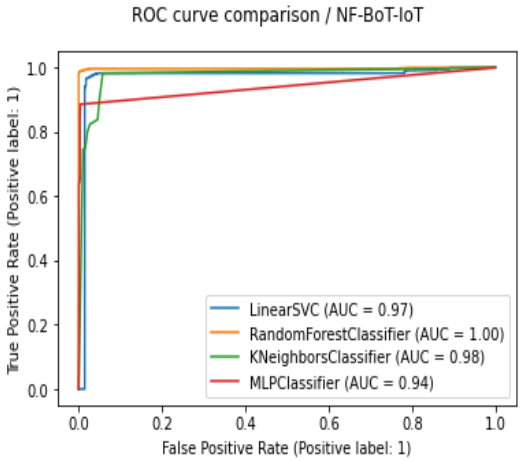
Tablo 3. ML yöntemlerinin doğruluk ve performans sonuçları (Table 3. Accuracy and performance results of ML method)

ML Yöntem	Veri Seti	Doğruluk (%)	Kesinlik (%)	Duyarlılık (%)	F1 Skor (%)
RO	NF-UNSW-NB15	99.9	99.2	99.1	99.2
	NF-BoT-IoT	99.4	99.9	99.4	99.7
	NF-ToN-IoT	99.5	99.9	99.4	99.7
	NF-CSE-CIC-IDS2018	99.9	99.1	98.5	98.8
KNN	NF-UNSW-NB15	98.7	83.6	88.9	86.2
	NF-BoT-IoT	82.7	99.9	82.3	90.3
	NF-ToN-IoT	90.4	99.8	88.2	93.6
	NF-CSE-CIC-IDS2018	98.7	84.5	88.9	86.6
DVM	NF-UNSW-NB15	87.9	23.9	78.4	36.6
	NF-BoT-IoT	96.8	99.9	96.7	98.3
	NF-ToN-IoT	79.7	96.0	78.0	86.1
	NF-CSE-CIC-IDS2018	85.9	20.5	74.9	32.2
YSA	NF-UNSW-NB15	79.4	13.9	69.7	23.2
	NF-BoT-IoT	60.7	99.9	59.8	74.8
	NF-ToN-IoT	81.0	97.9	78.0	86.8
	NF-CSE-CIC-IDS2018	96.9	62.0	82.3	70.7

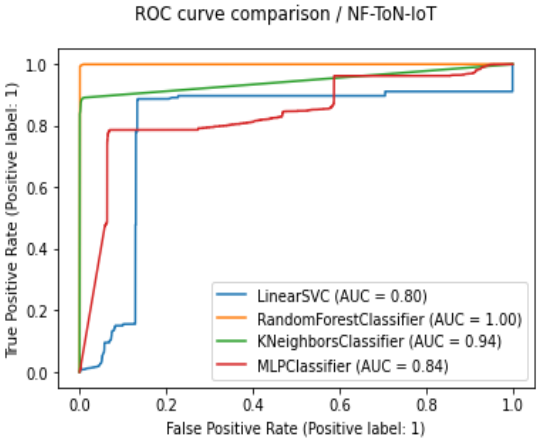
ROC (Receiver Operating Characteristic) eğrisi, ikili sınıflandırma işlemlerinde gerçek pozitiflerin sayısının, yanlış pozitiflerin bir fonksiyonu olarak çizilmesiyle oluşan bir performans ölçümüdür. ROC eğrisi altında kalan alan (AUC), modelin sınıfları ne kadar iyi ayırt edebildiğini gösterir. AUC, 0 ve 1 arasında bir değer alır. Bu değer büyüdükçe makine öğrenmesi modellerinin sınıfları ayırt etme başarısıda artar. Şekil 2, 3, 4 ve 5 da veri setlerine ait ROC eğrileri verilmiştir.



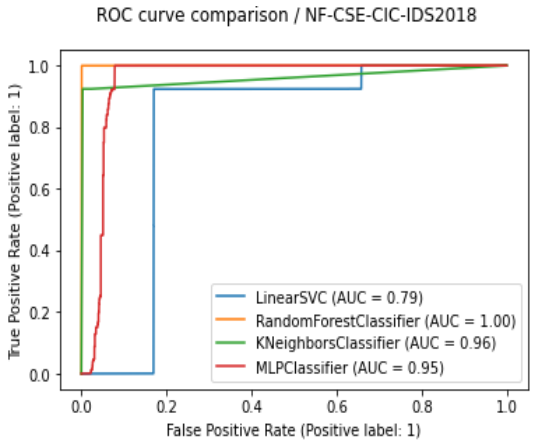
Şekil 2. NF-UNSW-NB15 veri setine ait ROC eğrisi (Figure 2. ROC curve of the NF-UNSW-NB15 dataset)



Şekil 3. NF-Bot-IoT veri setine ait ROC eğrisi (Figure 3. ROC curve of the NF-Bot-IoT dataset)



Şekil 4. NF-ToN-IoT veri setine ait ROC eğrisi (Figure 4. ROC curve of the NF-ToN-IoT dataset)



Şekil 5. NF-CSE-CIC-IDS2018 veri setine ait ROC eğrisi (Figure 5. ROC curve of the NF-CSE-CIC-IDS2018 dataset)

5. Sonuç

Günlük faaliyetlerimizin büyük ölçüde ağlar ve bilgi sistemlerine bağlı olması izinsiz giriş tespiti ve önlenmesini önemli bir konu haline getirmiştir. Saldırı tespit sistemlerinde çeşitli teknikler kullanılmıştır. Ancak son literatür çalışmalarına bakıldığında en yaygın olanı ML teknikleri olduğu gözlenmiştir. Bu çalışmada ortak özelliklere sahip NetFlow tabanlı NIDS veri setleri kullanılarak ikili sınıflandırma yapılmıştır. Modelin verimini artırmak için dengesizlik oranı azaltılmalıdır. Bu oranı azaltmak için çoğunluk gruplarının veri büyüklüğünü azaltan bir veri örnekleme modeli olan Rastgele Alt Örnekleme yöntemi kullanılmıştır. Sınıflandırma yöntemleri olarak Rastgele Orman, K-En Yakın Komşuluk, Destek Vektör Makineleri ve Yapay Sinir Ağları kullanılmıştır. Yapılan çalışmanın sonuçlarının doğruluk ve AUC değerlerine bakıldığında tüm veri setleri için en iyi sonucu RO algoritmasının verdiği gözlenmiştir. NF-UNSW-NB15 ve NF-BoT-IoT veri setleri için doğruluk değerinde en düşük performansı sırasıyla %79.4 ve %60.7 olarak YSA algoritmasının verdiği gözleniyor. Ancak NF-UNSW-NB15 veri seti için AUC değeri DVM sınıflandırıcı için en düşük değerdedir (0.79). NF-ToN-IoT ve NF-CSE-CIC-IDS2018 veri setlerinde ise doğruluk ve AUC değerlerinde düşük performansı DVM algoritmasının verdiği gözlenmiştir. Çalışma sonucunda veri setleri aynı özelliklere sahip olsa bile sınıflandırma algoritmalarının farklı performans gösterdiği gözlenmiştir.

Kaynakça

Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE access*, *6*, 33789-33795. DOI: 10.1109/ACCESS.2018.2841987

Akhan Baykan, N. & Khorram, T. (2021). Network Intrusion Detection using Optimized Machine Learning Algorithms . Avrupa Bilim ve Teknoloji Dergisi , (25) , 463-474 . DOI: 10.31590/ejosat.849723

Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M., & Ming, H. (2019, January). Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 0305-0310). IEEE. DOI: 10.1109/CCWC.2019.8666450

Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., & Marchetti, M. (2018, May). On the effectiveness of machine and deep learning for cyber security. In *2018 10th international conference on cyber Conflict (CyCon)* (pp. 371-390). IEEE. DOI: 10.23919/CYCON.2018.8405026

Bamakan, S. M. H., Wang, H., Yingjie, T., & Shi, Y. (2016). An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. *Neurocomputing*, *199*, 90-102. https://doi.org/10.1016/j.neucom.2016.03.031

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, *114*, 24-31. https://doi.org/10.1016/j.isprsjprs.2016.01.011

Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, *18*(2), 1153-1176. DOI: 10.1109/COMST.2015.2494502

Claise, B. (2004). Cisco systems netflow services export version 9 (No. rfc3954).

Çakır, B. & Angın, P. (2021). Zamansal Evrişimli Ağlarla Saldırı Tespiti: Karşılaştırmalı Bir Analiz . Avrupa Bilim ve Teknoloji Dergisi , Ejosat 2021 Ocak , 204-211 . DOI: 10.31590/ejosat.848784

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, *28*(1-2), 18-28. doi:10.1016/j.cose.2008.08.003

Garuba, M., Liu, C., & Fraites, D. (2008, April). Intrusion techniques: Comparative study of network intrusion detection systems. In *Fifth International Conference on Information Technology: New Generations (itng 2008)* (pp. 592-598). IEEE. DOI: 10.1109/ITNG.2008.231

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, *521*(7553), 452-459. https://doi.org/10.1038/nature14541

Jing, D., & Chen, H. B. (2019, October). SVM based network intrusion detection for the UNSW-NB15 dataset. In *2019 IEEE 13th international conference on ASIC (ASICON)* (pp. 1-4). IEEE. DOI: 10.1109/ASICON47005.2019.8983598

Karatas, G., Demir, O., & Sahingoz, O. K. (2020). Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. *IEEE Access*, *8*, 32150-32162. DOI: 10.1109/ACCESS.2020.2973219

Kerr DR, Bruins B L, (2021) U.S. Patent No. 6,243,667. Washington, DC: U.S. Patent and Trademark Office.

Kuş, İ. , Bozkurt Keser, S. & Yolaçan, E. (2021). Saldırı Tespit Sistemlerinde Topluluk Öğrenme Yöntemlerinin Kıyaslanması . Avrupa Bilim ve Teknoloji Dergisi , Ejosat 2021 Supplement 1 , 725-734 . DOI: 10.31590/ejosat.971875

Meftah, S., Rachidi, T., & Assem, N. (2019). Network based intrusion detection using the UNSW-NB15 dataset. *International Journal of Computing and Digital Systems*, *8*(5), 478-487. DOI: http://dx.doi.org/10.12785/ijcds/080505

Sarhan, M., Layeghy, S., & Portmann, M. (2021). Evaluating Standard Feature Sets Towards Increased Generalisability and Explainability of ML-based Network Intrusion Detection. *arXiv preprint arXiv:2104.07183*. https://doi.org/10.48550/arXiv.2104.07183

Sarhan, M., Layeghy, S., & Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications*, *27*(1), 357-370. https://doi.org/10.1007/s11036-021-01843-0

Sarhan, M., Layeghy, S., Gallagher, M., & Portmann, M. (2021). From Zero-Shot Machine Learning to Zero-Day Attack Detection. *arXiv preprint arXiv:2109.14868*. https://doi.org/10.48550/arXiv.2109.14868

Sarhan, M., Layeghy, S., Moustafa, N., & Portmann, M. (2020). Netflow datasets for machine learning-based network intrusion detection systems. In *Big Data Technologies and Applications* (pp. 117-135). Springer, Cham. DOI: 10.1007/978-3-030-72802-1_9

Sinclair, C., Pierce, L., & Matzner, S. (1999, December). An application of machine learning to network intrusion detection. In *Proceedings 15th annual computer security applications conference (ACSAC'99)* (pp. 371-377). IEEE. DOI: 10.1109/CSAC.1999.816048

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427-437. https://doi.org/10.1016/j.ipm.2009.03.002

Söderström, A. (2021). Anomaly-based Intrusion Detection Using Convolutional Neural Networks for IoT Devices. MSc Thesis, Blekinge Institute of Technology, Karlskrona, Sweden.

Wang, C., Wang, B., Sun, Y., Wei, Y., Wang, K., Zhang, H., & Liu, H. (2021). Intrusion Detection for Industrial Control Systems Based on Open Set Artificial Neural Network. *Security and Communication Networks*, *2021*. https://doi.org/10.1155/2021/4027900

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient kNN classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems*, *29*(5), 1774-1785. DOI: 10.1109/TNNLS.2017.2673241