Dicle University
Journal of Engineering
https://dergipark.org.tr/tr/pub/dumf
duje.dicle.edu.tr

Araştırma Makalesi / Research Article

Öznitelik Seçme Yöntemlerinin Makine Öğrenmesi Tabanlı Saldırı Tespit Sistemi Performansına Etkileri

Effects of Feature Selection Methods on Machine Learning Based Intrusion Detection System Performance

Sura Emanet[1], Gozde Karatas Baydogmus[2*], Onder Demir[1],

[1] Marmara Üniversitesi, Bilgisayar Mühendisliği Bölümü, suraemanet@marun.edu.tr
ORCID: https://orcid.org/0000-0003-2879-9208, odemir@marmara.edu.tr ORCID: https://orcid.org/0000-0003-4540-663X

[2] Biruni Üniversitesi, Bilgisayar Mühendisliği Bölümü, gbaydogmus@biruni.edu.tr
ORCID: https://orcid.org/0000-0003-2303-9410

MAKALE BİLGİLERİ

Makale Geçmişi:

Geliş 18 Ekim 2021
Revizyon 13 Aralık 2021
Kabul 28 Aralık 2021
Online 31 Aralık 2021

Anahtar Kelimeler:

Saldırı tespit sistemi, makine öğrenmesi, öznitelik seçimi, öznitelik filtreleme ve saldırı tespiti

ÖZ

Artan İnternet tabanlı teknolojilerin kullanımı insanlara ve kurumlara önemli avantajlar sağlamanın yanı sıra bir takım dezavantajları da beraberinde getirmiştir. Bunlardan en önemlisi siber saldırılardır. Siber saldırıların çeşitlenmesi ve artmasıyla, büyük miktarlara ulaşan kritik verilerin silme, değiştirilme, ifşa edilme gibi eylemlere karşı korunması her geçen gün daha zor hale gelmektedir. Bu sebeple bilgi sistemlerinin güvenliğinin sağlanması amaçlı geliştirilen araçlardan biri olan Saldırı Tespit Sistemleri çok önemli yere sahip bir çalışma alanı olmuştur. Bu çalışmada, CSE-CIC-IDS2018 veri kümesi üzerinde literatürde önerilen çeşitli öznitelik seçim yöntemleri ve makine öğrenmesi teknikleri kullanılarak, öznitelik seçiminin Saldırı Tespit Sistemi başarım ve performansı üzerindeki etkisi incelenmiştir. Orijinal veri kümesini temsil edebilecek en iyi alt kümeyi belirlemek için Ki-Kare Testi, Spearman'ın Sıralama Korelasyon Katsayısı ve Özyinelemeli Öznitelik Eliminasyonu yöntemleri kullanılmıştır. Yeni veri kümeleri Adaptif Yükseltme, Karar Ağacı, Lojistik Regresyon, Çok Katmanlı Algılayıcı, Ekstra Ağaçlar, Pasif-Agresif ve Gradyan Artırma makine öğrenmesi yöntemleri ile sınıflandırılarak performans sonuçlarının karşılaştırmalı bir analizi yapılmıştır. Performansların objektif değerlendirilebilmesi için K-Fold kullanılmıştır. K-Fold işleminin hesaplama ve zaman yönünden maliyetli olması sebebiyle paralleştirme uygulanarak işlem süresi düşürülmüştür. Elde edilen deneysel sonuçlara göre Ki-Kare Testi ve Spearman'ın Sıralama Korelasyon Katsayısı öznitelik seçim yöntemleri veri boyutunun indirgenmesinden dolayı işlem yükünü azaltarak işlem süresini %45 oranında kısaltmış fakat hata oranını sırasıyla %14,46 ve %10,52 artırmıştır. Ayrica, Özyinelemeli Öznitelik Eliminasyonu yönteminin uygun ayar parametreleri kullanıldığında, işlem süresini %38 oranında kısaltması ile birlikte sistemin hata oranını da %2,95'e kadar düşürdüğü görülmüştür.

ARTICLE INFO

Article history:

Received 18 October 2021
Received in revised form 13 December 2021
Accepted 28 December 2021
Available online 31 December 2021

Keywords:

Intrusion detection system, machine learning, feature selection, feature filtering and intrusion detection

Doi: 10.24012/dumf.1051340

* *Sorumlu Yazar*
Gözde Karataş Baydoğmuş
abaydoamus@biruni.edu.tr

ABSTRACT

The increasing use of the Internet-based technologies has brought along some disadvantages as well as providing significant advantages to people and institutions. The most important of these disadvantages is cyber-attacks. With the variety and increase of cyber-attacks, it becomes more and more difficult to protect large amounts of critical data against actions such as deletion, modification and disclosure. For this reason, Intrusion Detection Systems, one of the tools developed to ensure the security of information systems, has become a very important study area. In this study, the effect of feature selection on Intrusion Detection System performance and success, was investigated. The study was developed on the CSE-CIC-IDS2018 dataset by using various feature selection methods and machine learning techniques suggested in the literature. Chi-Square Test, Spearman's Ranking Correlation Coefficient and Recursive Feature Elimination methods were used to determine the best subset that could represent the original dataset. The new datasets created with the features determined by each feature selection method were classified using Adaptive Boosting, Decision Tree, Logistic Regression, Multilayer Perceptron, Extra Trees, Passive-Aggressive and Gradient Boosting machine learning methods, and a comparative analysis of the obtained performance results was made. K-Fold was used to evaluate the performances objectively. Since the K-Fold process is costly in terms of computation and time, the processing time is reduced by applying parallelization. According to the experimental results obtained, Chi-Square Test and Spearman's Ranking Correlation Coefficient feature selection methods reduced the processing load due to the reduction of the data size and shortened the processing time by 45%, but increased the error rate by 14.46% and 10.52% respectively. On the other hand, it has been observed that the Recursive Feature Elimination method reduces the processing time by 38% and the error rate of the system up to 2.95% when appropriate setting parameters are used.

Giriş

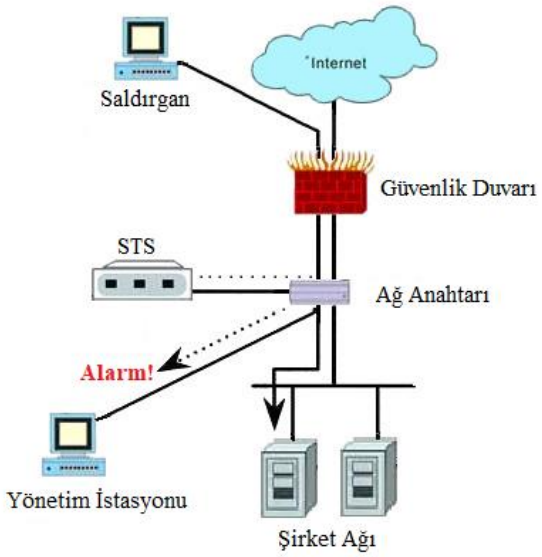
Saldırı tespit sistemleri (STS), ağ güvenliği altyapısında yaygın olarak kullanılan; anomali ve imza tabanlı saldırıları tespit ederek ağları korumak için geliştirilmiş sistemlerdir. Saldırı tespit yöntemleri temelinde, STS'ler üç kategoriye ayrılabilir: kötüye kullanım, anomali ve spesifikasyon tabanlı [1]. Kötüye kullanım veya imza tabanlı bir STS, saldırı özellikleri ile önceden depolanan saldırı imzaları veya modelleri arasında bir eşleşme arayarak saldırıları tespit edebilir ve bilinen saldırıları tespit etmek için uygundur; ancak yeni veya bilinmeyen saldırıları tespit etme noktasında zayıf kalır. Anomali tabanlı STS'lerin saldırı tespiti noktasındaki dayanağı, saldırı sürecinin normal kullanıcı davranışından farklı davranışlar üretebilmesidir [2].

Ağı veya sistemleri kötü amaçlı faaliyet veya politika ihlalleri açısından izleyen bir güvenlik teknolojisi olan STS'ler, ağda dolaşan veri paketlerini izler ve şüpheli etkinlik algılandığında alarm verir. STS'ler çoğunlukla güvenlik duvarından sonra, bir anahtara veya bir ağ TAP (Terminal Erişim Noktası)'ye bağlanır ve trafiğin STS'ye yayıldığı (veya TAP aracılığıyla gönderildiği) Inline olmayan bir modda kullanılır. Bu açıklamalar doğrultusunda Şekil 1 bir STS'nin internetteki varlığını göstermektedir.

Bir STS'nin saldırıları tespit etme noktasındaki kabiliyetini geliştirmek için öğrenme yetenekleri nedeniyle genellikle makine öğrenimi teknikleri kullanılır. Bu yüzden çalışmalar, özellikle en yüksek doğruluk ve en düşük yanlış alarm oranlarının belirlenmesi olmak üzere, sistemlerin performansını iyileştirmek için makine öğrenimi yöntemlerine odaklanmıştır [1]. Son yıllarda bu sistemler üzerinde karar ağacı, rastgele orman, destek vektör makinesi ve yapay sinir ağları gibi birçok makine öğrenimi algoritması uygulanmış ve çeşitli iyileştirmeler yapılmıştır. Bununla birlikte, her algoritmanın her türden saldırıyı tespit etmede avantajları ve dezavantajları bulunabilir. Bazı algoritmalar, yalnızca belirli saldırı türlerini tespit etmede yüksek oranda etkili olabilir [3].

STS'nin verimliliği, doğrudan öğrenme modeli ve veri kümesinin kalitesi ile ilişkilidir. Birçok çalışma bilinen eksiklikleri olan veri kümelerine dayanmaktadır. Güncel olmayan saldırı trafiği, anonimlik (gizlilik veya etik nedeniyle), simüle edilmiş trafik (gerçek bir üretim ağından değil) ve trafik çeşitliliğinin olmaması, bunlarla sınırlı olmamak üzere eksiklikler arasındadır [4].

Veri kümesindeki öznitelikler sınıflandırma performansını etkileyen en önemli unsurlardan biridir [5]. Öznitelik sayısının az olması sınıfların düzgün ayrışamamasına, fazla olması ise eğitim süresinin artması, gürültüsü fazla olan özniteliklerin doğruluk oranını düşürmesi gibi problemlere neden olabilmektedir. Bu nedenle eğitim süresini azaltacak, veri kalitesini geliştirecek ve modelin başarısını artıracak orijinal veri kümesini temsil edebilen yeterli sayıda özniteliğin doğru bir şekilde belirlenmesi gerekmektedir.



Şekil 1. İnternette Saldırı Tespit Sistemi

Çalışmada, 2018 yılında hazırlanmış, saldırı çeşitliliği yüksek, gerçek ağ trafiğine sahip bir veri kümesi olan CSE-CIC-IDS2018 üzerinde literatürde önerilen çeşitli öznitelik seçim yöntemleri ve makine öğrenmesi teknikleri kullanılarak performanslı bir STS elde etme üzerinde durulmuştur. Modelin başarısını artırmak ve saldırı tespit süresini azaltmak için öznitelikler Ki-Kare (Chi-Square) Testi, Spearman 'ın Sıralama Korelasyon Katsayısı (Spearman's Rank Correlation Coefficient) ve Özyinelemeli Öznitelik Eliminasyonu (Recursive Feature Elimination) yöntemleri uygulanarak incelenmiş ve belirlenmiştir. Her bir yöntem için belirlenen öznitelikler ile oluşturulmuş yeni veri kümesi Adaptif Yükseltme (AdaBoost), Karar Ağacı (Decision Tree), Lojistik Regresyon (Logistic Regression), Çok Katmanlı Algılayıcı (Multi-Layer Perceptron), Ekstra Ağaçlar (Extra Trees), Pasif-Agresif (Passive Aggressive) ve Gradyan Artırma (Gradient Boosting) makine öğrenmesi yöntemleri ile sınıflandırılarak elde edilen performans sonuçlarının karşılaştırmalı bir analizi yapılmıştır.

Gelecek çalışmalarda öznitelik seçimi ve kolektif öğrenme yöntemleri kullanılarak yeni bir hibrit model önerilmesi planlanmaktadır. Bu sebeple farklı makine öğrenimi yöntemlerinin tahmin performanslarını görmek verimli bir model oluşturabilmesi adına önemlidir.

Elde edilen deneysel sonuçlar incelendiğinde Ki-Kare (Chi-Square) Testi ve Spearman 'ın Sıralama Korelasyon Katsayısı öznitelik seçim yöntemleri, sistemin başarımını düşürmüş olsa da veri boyutunun indirgenmesinden dolayı işlem yükünü azaltmış ve işlem süresini kısaltmıştır. Diğer taraftan Özyinelemeli Öznitelik Eliminasyonu yönteminin uygun ayar parametreleri kullanıldığında, işlem süresini %38 oranında kısaltması ile birlikte sistemin hata oranını da %2,95 oranında düşürdüğü görülmüştür.

Literatür Taraması

Bir makine öğrenimi algoritmasının performansı büyük ölçüde eğitildiği veri kümesine bağlıdır [6]. Makine öğrenimi tabanlı STS ile ilgili mevcut araştırmaların çoğunda, eğitim DARPA, KDD Cup 99 ve NSL-KDD veri kümeleri ile gerçekleştirilmiştir. Ancak bazı araştırmacılar popüler fakat güncelliğini kaybetmiş bu veri kümelerinin yeni çalışmalar için kullanılmamasını önermiştir [6, 7].

CSE-CIC-IDS2018, Kanada Siber Güvenlik Enstitüsü (CIC) ve İletişim Güvenliği Kurumu (CSE) tarafından Amazon Web Servisleri LAN ağının bir bölümü üzerinden toplanarak oluşturulmuştur [4]. Çalışmada kullanılan CSE-CIC-IDS2018, CICIDS2017 veri kümesinin güncel halidir ve saldırı çeşitliliği yüksek, bilinen en yeni saldırı trafiği veri kümesidir. CICIDS2017, CSE-CIC-IDS2018 ve popüler diğer veri kümelerindeki anomali temelli saldırı tespitine yönelik bulguları bildiren bazı çalışmalar şunlardır:

Sharafaldin ve arkadaşları, saldırı tespiti için oluşturulan veri kümelerinin güncel saldırıları kapsamaması, kullanımlarının güvenli olmaması ve saldırı çeşitliliklerinin yetersiz olması gibi sebepler dolayısıyla CICIDS2017 veri kümesini oluşturmuştur. Bu veri kümesi üzerinde 6 farklı makine öğrenimi yöntemi karşılaştırılmıştır; K-En Yakın Komşu, Rastgele Orman, ID3, Adaptif Yükseltme, Naive Bayes ve Karesel Ayrım Analizi. %98 F1-Skoru ile ID3 algoritmasının en yüksek başarımı gösterdiği bildirilmiştir [4].

Wankhede ve Kshirsagar, belirli bir günde yapılan DoS saldırılarını tespit etmek için CICIDS2017 veri kümesi üzerinde iki farklı makine öğrenimi yöntemini uygulamıştır, bunlar; Rastgele Orman ve Yapay Sinir Ağı. Ek olarak, veri kümesinin farklı bölümlenmesinin saldırı tespitinin başarısına yönelik etkisini incelemek amacıyla eğitim veri kümesi %20-%80 arasında bölümlenerek Rastgele Orman ve Çok Katmanlı Algılayıcı algoritmalarının başarımları karşılaştırılmıştır. %99,95 doğruluk oranı ile Rastgele Orman yönteminin en yüksek başarımı gösterdiği ve Rastgele Orman yöntemi için %50, MLP yöntemi için %30 bölümlenmenin optimum olduğu bildirilmiştir [8].

Zhou ve Pezaros, CSE-CIC-IDS2018 veri kümesi kullanılarak eğitilen bir modelin sıfır gün (Zero-Day) saldırıları üzerindeki başarımını incelemiştir. Çalışmada 10-Katlamalı Çapraz Doğrulama yöntemi ile altı makine öğrenimi sınıflandırıcısı karşılaştırılmıştır; Rastgele Orman, Naive Bayes, Karar Ağacı, Çok Katmanlı Algılayıcı, K-En Yakın Komşu ve Karesel Ayrım Analizi. Denemeler her saldırı tipi için normal trafik ile ikili karşılaştırılarak yapılmıştır. Karar Ağacının en yüksek saldırı tespit doğruluğunu sergilediği bildirilmiştir [9]. Sonrasında, eğitim veri kümesi üzerinde Normal ve Saldırı olmak üzere etiketlemeler oluşturulup model eğitilmiştir. Test veri kümesi için 1 haftalık normal trafik ve 8 farklı yeni saldırı trafiği oluşturulmuştur; ZeroAccess, DDoS Bot'a Darkness, Google Doc Macadocs, Bitcoin Miner, Drowor Worm, Nuclear Ransomware, False Content Injection, Ponmocup Trojan. Çalışma sonucunda Karar Ağacı modeli kullanılarak %96 doğruluk oranı ile saldırı tespiti yapılabildiği bildirilmiştir.

Kanimozhi ve Jacob, CSE-CIC-IDS2018 veri kümesindeki Botnet saldırılarını tespit etmek için Çok Katmanlı Algılayıcı yöntemini uygulamıştır. Çalışmada, varsayılan hiper-parametreler ile modelin aşırı uyum (overfitting) durumuna düşmesi sebebiyle hiper parametre optimizasyonu yapılmış ve %99,97 doğruluk oranına ulaşılmıştır [10].

Yulianto ve arkadaşları, Adaptif Yükseltme tabanlı STS'nin performansını iyileştirmek için CICIDS2017 veri kümesi üzerinde Temel Bileşen Analizi (Principal Component Analysis-PCA), Sentetik Azınlık Aşırı Örnekleme (Synthetic Minority Oversampling Technique-SMOTE) ve Topluluk Öznitelik Seçimi (Ensemble Feature Selection-EFS) yöntemlerini kullanmıştır. Değerlendirme sonuçları, %90,01 F1-Skoru ile SMOTE ve EFS yöntemlerinin birlikte kullanımlarının en iyi performans iyileştirmesini sağladığını göstermiştir [11].

Wani ve arkadaşları, Bulut Bilişim Ortamı üzerinde Destek Vektör Makineleri, Rastgele Orman ve Naive Bayes yöntemlerini kullanarak DDoS saldırı tespiti yapmıştır. Çalışma sonucunda, oluşturulan yeni veri kümesi üzerinden 9 öznitelik kullanılarak %99,80 F1-Skoru ile Destek Vektör Makine yönteminin en yüksek başarımı gösterdiği bildirilmiştir [12].

McKay ve arkadaşları, CICIDS2017 veri kümesindeki Botnet saldırılarını tespit etmek için Rastgele Orman, OneR, K-En Yakın Komşu, J48, Çok Katmanlı Algılayıcı ve NaiveBayes yöntemlerini uygulamıştır.

Çalışmada veri kümesi dengeli ve normal olmak üzere iki farklı şekilde bölünmüştür. Dengeli veri kümesi kullanılarak eğitilen modellerin tümü ile daha başarılı sonuçlar elde edilmiştir. %98,73 doğruluk oranı ile J48 yönteminin en yüksek başarımı gösterdiği bildirilmiştir [13].

Kanimozhi ve Jacob, Botnet saldırılarının tespiti için CSE-CIC-IDS2018 veri kümesi üzerinde altı makine öğrenimi sınıflandırıcısını karşılaştırmıştır; K-En Yakın Komşu, Naive Bayes, Destek Vektör Makinesi, Rastgele Orman, Adaptif Yükseltme ve Çok Katmanlı Algılayıcı Ağı. Performanslar kalibrasyon eğrileri üzerinden değerlendirilmiştir. Kalibrasyon eğrisi, mükemmel eğriye en yakın olan sınıflandırıcının MLP olduğu bildirilmiştir [14].

Ferrag ve Maglaras, Brute-Force, Web, DoS, DDoS, Botnet ve Infilteration saldırılarını tespit etmek için CSE-CIC-IDS2018 veri kümesi üzerinde dört farklı makine öğrenimi yöntemi uygulamıştır, bunlar; Destek Vektör Makinesi, Tekrarlayan Sinir Ağları, Evrişimsel Sinir Ağları ve Rastgele Orman. Elde edilen en yüksek doğruluk oranlarının sırasıyla %92,19, %96,12, %96,18, %98,55, %98,71 ve %96,23 olduğu bildirilmiştir [15].

Pehlivanoğlu ve arkadaşları, Tek Seviyeli ve İki Seviyeli Hibrit Yöntem olmak üzere iki farklı yöntemin CSE-CIC-IDS2018 veri kümesi üzerinde saldırı tespit başarısını test etmiştir. Çalışmada Evrişimsel Sinir Ağı, Rastgele Orman, Hafif Gradyan Artırma, Evrişimsel Rastgele Orman, Hafif Gradyan Rastgele Orman ve Rastgele Orman-Rastgele Orman makine öğrenimi yöntemleri uygulanmıştır. Sonuçlar, %98,00 doğruluk oranı ve %86,00 makro F1-Skoru ile Evrişimsel Rastgele Orman hibrit modelinin en iyi saldırı tespitini yaptığını göstermiştir [16].

Filho ve arkadaşları, CIC-DOS, CICIDS2017, CSE-CIC-IDS2018 ve kendi oluşturdukları veri kümeleri üzerinde DoS saldırılarının tespiti için Rastgele Orman makine öğrenimi yöntemini uygulamıştır. Elde edilen F1-Skor değerleri sırasıyla %99,00, %99,00, %100,00 ve %99,00 çıkmıştır [17].

Zhou ve arkadaşları, CICIDS2017 veri kümesini kullanarak yüksek doğruluk oranı ile performanslı bir şekilde saldırı tespiti yapılabilmesi için öznitelik seçimi ve toplu öğrenme yöntemlerini uygulamıştır. Çalışmada, Korelasyon Tabanlı Öznitelik Seçimi ve Yarasa Algoritmasının (CFS-BA) faydalarını C4.5, Rastgele Orman ve ForestPA'ya dayalı bir topluluk sınıflandırıcısı ile birleştiren yeni bir yöntem önerilmiştir. Önerilen yöntemin %96,76 doğruluk oranı ve %98,10 F1-Skoru ile en yüksek başarımı gösterdiği bildirilmiştir. Sonuçlar, tek sınıflandırıcının bulunduğu bireysel yaklaşımlardan önemli ölçüde daha iyi performans gösterdiğini ortaya koymaktadır [5].

Fitni ve arkadaşları, çalışmalarında her bir sınıflandırma algoritmasının faydalarını bütünleştiren oylama adı verilen toplu öğrenme yaklaşımını gerçeklemiştir. Toplu öğrenme için en uygun temel sınıflandırıcıları belirlemek amacıyla CSE-CIC-IDS2018 veri kümesi üzerinde 7 farklı tek sınıflandırıcı ile karşılaştırmalar yapılmıştır. Karşılaştırma sonuçlarına göre bir topluluk modeli oluşturulmuştur. Topluluk modeli için seçilen sınıflandırıcılar; Lojistik Regresyon, Karar Ağacı ve Gradyan Artırma yöntemleridir. En önemli veri özniteliklerini belirlemek için Spearman korelasyon analizinden faydalanılmıştır. Sonuçlar, 80 öznitelikten 23'ünün seçildiğini ve modelin şu başarı oranlarını aldığını göstermiştir; Doğruluk %98,80; Kesinlik %98,80; Duyarlılık %97,10 ve F1-Skor %97,90 [18].

2021 yılında araştırmacılar İnternet üzerinden verilen hizmetlerin artması sonucunda ağ altyapısının siber saldırılara daha fazla maruz kaldığını tespit ederek, ağ trafiğinden yakalanan paket örnekleri üzerinde DDoS saldırılarını tespit eden bir derin öğrenme modeli önermiştir [19]. Çalışmada, CIC-DDoS2019 veri kümesi üzerinde inceleme yapılmış ve popüler bir derin öğrenme yaklaşımı olan Derin Sinir Ağları kullanılmıştır. Derin Sinir Ağlarının tercih edilmesinin sebebi eğitildikçe kendini güncellemesi, öznitelik çıkarma ve sınıflandırma işlemlerini içeren katmanlara sahip olmasıdır. Sonuçlar, ağ trafiğine yapılan saldırıların %99,99 başarı ile tespit edildiğini ve saldırı türlerinin %94,57 doğruluk oranı ile sınıflandırıldığını göstermektedir.

Arslan, çalışmasında internet trafik verilerini daha kolay işlenebilir hale getirmek için bir veri ön işleme önermiş ve makine öğrenmesi teknikleri ile ağ analizi yaparak sınıflandırmayı hedeflemiştir [20]. Önerdiği veri ön işleme trafik analiz süresini önemli ölçüde kısaltmış ve başarım oranını artırmıştır. Çalışmanın eğitim ve testi için güncel bir veri kümesi olan CSE-CIC-IDS2018 tercih edilmiştir. Sonuçlar, ikili sınıflandırma için Ekstra Ağaçlar algoritması ile %99,0 ve çoklu sınıflandırma için Rastgele Orman algoritması ile %98,5 başarım oranı elde edildiğini göstermektedir.

Emhan ve Akın, anomali tespiti için kullanılan makine öğrenmesi algoritmalarının daha verimli hale getirilmesini sağlamak üzere bir çalışma yapmıştır [21]. Buna göre filtreleme tabanlı öznitelik seçimi yöntemlerinin anomali tabanlı saldırı tespit etmedeki başarısını gösteren araştırmalar gerçeklenmeye çalışılmıştır. Araştırmacılar çalışmayı geliştirmek için popüler bir saldırı tespit veri kümesi olan NSL-KDD veri kümesini tercih etmiştir. Çalışmada öncelikle filtreleme tabanlı Korelasyon Tabanlı, Simetrik Belirsizlik Katsayısı, Kazanç Oranı, Bilgi Kazancı, One-R, ve Ki-Kare Öznitelik Seçimi yöntemleri ile veri kümesinin boyutu azaltılmış ve 8 öznitelik seçilmiştir.

Öznitelik seçiminden sonra K-en yakın komşu ve Rastgele Orman algoritmaları ile ayrı ayrı saldırı tespiti gerçekleştirilmiştir. Çalışma sonucunda Ki-Kare, Bilgi Kazancı ve One-R ile öznitelik seçimi yapılmış veri kümesinde sınıflandırma yapmanın en iyi sonucu verdiği görülmüştür.

Önerilen Sistem

STS'ler ilerleyen teknoloji ile beraber, artan ihtiyaçlar doğrultusunda istekleri karşılamaya uygun olmalıdır [22]. Hala gelişmekte ve bir çok araştırmaya konu olan Makine Öğrenmesi bu anlamda başvurulan yöntemlerdendir [23]. STS'lerde bu yöntemlerin kullanılmasının amacı; sistemin, hakkında bilgisi olmadığı bir veriyi hızlı ve yüksek doğruluk oranı ile tahmin edebilmesidir [24].

STS'ler çoğunlukla artan işlem süresi ve düşük tespit oranı ile sonuçlanan çeşitli alakasız ve gereksiz öznitelikler içeren büyük miktarda veriyle ilgilenir [25]. Bu nedenle öznitelik seçimi, makine öğrenimi tabanlı STS'lerde performans iyileştirmeleri elde etmek için önemlidir. Bu yöntem, doğruluk tespitini geliştirmek ve sınıflandırma eğitim süresini azaltmak için en önemli veri kümesi özniteliklerini seçmek için kullanılır [5].

Bu çalışmada, doğru ve verimli sınıflandırma sonuçlarına sahip bir STS elde etmek için literatürde önerilen farklı öznitelik seçim yöntemleri ile çeşitli makine öğrenmesi tekniklerinin kullanılmasına odaklanılmıştır. STS için geliştirilen makine öğrenimi modeli yaklaşımı Şekil 2'de verilmiştir. Model iki aşamadan oluşmaktadır; Veri ön işleme ve Tabakalı 5-Katlamalı Çapraz Doğrulama. İlk aşamada veri temizleme ve dönüştürme işlemlerinden sonra öznitelik seçimi uygulanmış sonraki aşamada elde edilen yeni veri kümesi üzerinde 5-Katlamalı Çapraz Doğrulama kullanılarak seçilen makine öğrenmesi modelleri uygulanmıştır.

Veri ön işleme aşaması, CSE-CIC-IDS2018 veri kümesi üzerinde; eksik öznitelik değerlerinin tamamlanması, hatalı verinin düzeltilmesi, tutarsızlıkların saptanması ve temizlenmesi, ölçeklendirme, normalizasyon ve çeşitli öznitelik seçim yöntemlerinin uygulanmasından oluşmaktadır. Veri kümesinin oldukça fazla örnek içermesi ve sadelik ihtiyacı sebebiyle, öznitelik seçim yöntemleri uygulanmadan veri kümesi atak dağılımları değişmeyecek şekilde %50 oranında küçültülmüştür.

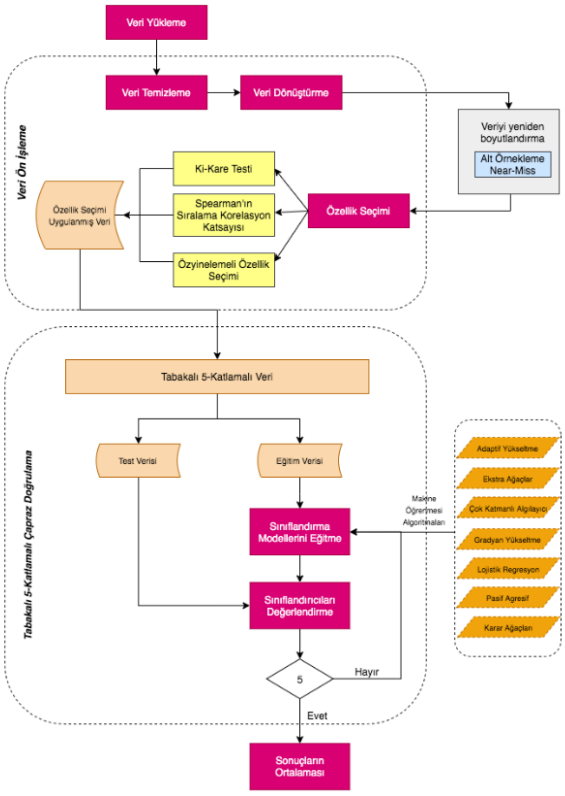
Uygulanan her bir öznitelik seçim yöntemi için, eğitim veri kümesi ile Adaptif Yükseltme (AdaBoost), Karar Ağacı, Lojistik Regresyon, Çok Katmanlı Algılayıcı (MLP), Ekstra Ağaçlar, Pasif-Agresif ve Gradyan Artırma makine öğrenme algoritmaları kullanılarak oluşturulan modeller eğitilmiştir. Test veri kümesi ile, oluşturulan saldırı tespit modellerinin bir değerlendirmesi ve performanslarının karşılaştırmalı analizi yapılmıştır.

Performanslar Tabakalı 5-Katlamalı Çapraz Doğrulama (Stratified 5-Fold Cross Validation) tekniği kullanılarak doğruluk (accuracy), kesinlik (precision), duyarlılık (recall), F1-Skoru (F1-Score) ve hesaplama zamanı metrikleri üzerinden değerlendirilmiştir. Kullanılan veri kümesi, öznitelik seçim yöntemleri ve modeller ile ilgili bilgiler devam eden kısımda detaylandırılmıştır.

Veri Kümesi

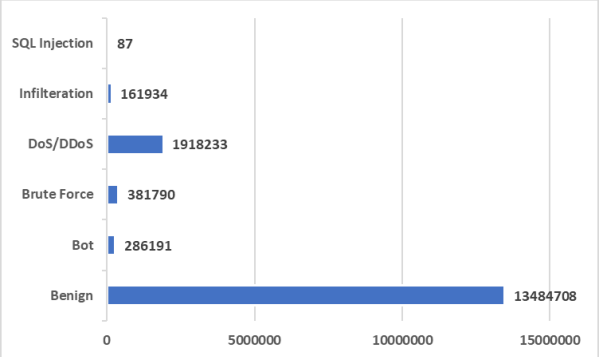
Çalışmada Kanada Siber Güvenlik Enstitüsü (Canadian Institute for Cybersecurity) ve İletişim Güvenliği Kuruluşu (Communications Security Establishment) iş birliği ile üretilmiş ve herkese kullanım imkânı sunulmuş, güncel bir veri kümesi olan CSE-CIC-IDS2018 tercih edilmiştir [26].

Veri kümesinde BruteForce (Web, XSS, FTP, SSH), Botnet, DoS (Hulk, SlowHTTPTest, GoldenEye, Slowloris), DDoS (HOIC, LOIC-UDP, LOIC-HTTP), Web saldırıları (SQL Injection) ve Ağa içeriden sızma (Infilteration) olmak üzere 6 tipte 14 farklı saldırı türü (2,748,235 saldırı) vardır. CICFlowMeter-V3 [27] kullanılarak elde edilen paketler ağ trafik akışlarına dönüştürülmüş ve 80 öznitelik sunulmuştur.

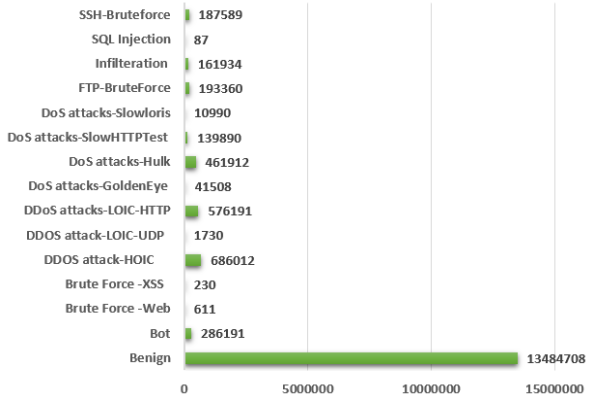


Şekil 2. Geliştirilen makine öğrenmesi modeli

Şekil 3, normal ve çeşitli saldırı türlerinin veri kümesindeki dağılımlarını, Şekil 4'de her etiket için örnek dağılımları göstermektedir. Her iki şekilde de Benign tipli veri sayısının oldukça fazla olduğu görülmektedir bu da Yanlış Negatif sayısının fazla olmasına sebep olacak ciddi bir hata ile sonuçlanabilir.



Şekil 3. Etiket kategori dağılımı



Şekil 4. Veri kümesindeki etiket sayısı

Tablo 1 CSE-CIC-IDS2018 veri kümesinde yer alan bazı öznitelikler için kısa açıklamalar içermektedir. Öznitelikler ve ayrıntılı açıklamaları için referans [26] incelenebilir.

Tablo 1. CSE-CIC-IDS2018 öznitelikler

Öznitelik	**Kısa Açıklama**
Dst Port	Hedef Bağlantı Noktası
Totfwd/bwdpkts	İleri ve geri yönlerde toplam paket sayısı
Fwd/bwdpkts	Saniyedeki ileri/geri paket sayısı
Flowduration	Akış süresi
Idl_min	Akışın aktif hale gelmeden önce boşta kaldığı süre

Öznitelik Seçimi

Öznitelik seçimi, tahmine veya beklenen çıktı değişkenlerine katkıda bulunabilecek önemli özniteliklerin seçilme sürecidir [7]. Öznitelik seçiminde kullanılan yöntemler, istatistiksel bilgiye dayalı olan filtreleme (filter) yöntemleri, öznitelikler üzerinde arama işlemleri gerçekleştiren sarmal (wrapper) yöntemler ve en iyi bölen ölçütünü bulmaya dayalı olan gömülü (embedded) yöntemler olmak üzere genel olarak üç grupta toplanmaktadır [28].

Eğitim ve test modelinde kullanılacak öznitelikler 3 farklı yöntem uygulanarak belirlenmeye çalışılmıştır; Ki-Kare (Chi-Square) Testi, Spearman'ın Sıralama Korelasyon Katsayısı (Spearman's Rank Correlation Coefficient) ve Özyinelemeli Öznitelik Eliminasyonu (Recursive Feature Elimination-RFE).

Ki-Kare istatistiksel testi, öznitelikler ve cevap değişkenleri arasındaki ilişkinin gücünü göstererek en iyi özniteliklerin seçimini kolaylaştırır [29].

Spearman'ın Sıralama Korelasyon Katsayısı, yüksek korelasyonlu öznitelikleri tanımlamak için kullanılır [30].

Özyinelemeli Öznitelik Eliminasyonu (RFE), en zayıf özniteliği (veya öznitelikleri) belirtilen öznitelik sayısına ulaşılana kadar özyinelemeli olarak ortadan kaldırarak, orijinal veri kümesini temsil edebilecek en iyi alt kümenin belirlenmesini sağlar.

Belirtilen üç yöntem için CSE-CIC-IDS2018 veri kümesindeki her bir özelliğin puan hesaplamaları yapılmıştır. Yapılan puan hesaplamaları sonucunda eşiği geçen öznitelikler ile sınıflandırıcılar üzerinde performans karşılaştırmaları yapılmıştır.

Ki-Kare Testi, Spearman'ın Korelasyon Sıralama Katsayısı ve RFE yöntemleri uygulanarak sırasıyla 31, 25 ve 40 öznitelik belirlenmiştir. Belirlenen öznitelikler ile oluşturulan yeni veri kümeleri çeşitli makine öğrenimi algoritmaları kullanılarak sınıflandırılmıştır. Belirlenen öznitelikler ile ilgili tüm detaylar Uygulama ve Başarımlar bölümünde yer almaktadır.

Sonuçlar, sistem başarımının Ki-Kare testi ve Spearman'ın korelasyon analizi uygulandığında düştüğünü, RFE yöntemi uygulandığında ise arttığını göstermiştir. En yüksek başarımı veren model %98,76 doğruluk oranı ile Ekstra Ağaçlar modeline ait olsa da süre metriği dikkate alındığında sırasıyla %98,65 ve %95,15 doğruluk oranları ile Karar Ağacı ve Lojistik Regresyon modelleri de ön plana çıkmıştır.

Kullanılan Makine Öğrenmesi Yaklaşımları

Bu kısımda çalışmada kullanılan makine öğrenmesi yaklaşımlarından bahsedilmiştir.

Karar Ağacı Algoritması (DT), sınıflandırma ve regresyon için kullanılan parametrik olmayan denetimli bir öğrenme yöntemidir. Amaç, veri özniteliklerinden çıkarılan basit karar kurallarını öğrenerek, hedef değişkenin değerinin tahmin edildiği bir model oluşturmaktır.

Adaptif Yükseltme Algoritması (ADA), "Adaptive Boosting" yaklaşımının kısaltması olan AdaBoost, prestijli Gödel ödülüne layık görülmüş ilk başarılı boosting algoritmasıdır. Amaç, sınıflandırma problemlerine odaklanarak bir takım zayıf sınıflandırıcıları güçlü olana dönüştürmektir. Algoritmada başlangıçta her bir örnek için eşit bir dağılım ile başlanır ve sınıflandırma performansına göre en iyi zayıf sınıflandırıcı bulunur. Ardından ağırlıklar güncellenerek yanlış sınıflandırılan örneklere odaklanılır. Böylelikle belirli sayıdaki iterasyon sonucunda en güçlü zayıf sınıflandırıcılar bir araya getirilerek güçlü bir

sınıflandırıcı oluşturulur ve sınıflandırma başarısı arttırılır [31, 32].

Lojistik Regresyon (LR), bağımlı değişkeni ikili (binary) yapıda olan veri kümeleri üzerinde uygulanan bir regresyon analizidir. Diğer tüm regresyon analizlerinde olduğu gibi, lojistik regresyon da bir tahmin analizidir. Bu tür analizlerde temel amaç bağımlı ve bağımsız değişkenler arasındaki ilişkiyi, en az değişken ile en iyi uyuma sahip olacak biçimde tanımlayabilen, kabul edilebilir bir model kurmaktır.

Ekstra Ağaçlar (ET), Rastgele Orman sınıflandırıcısının farklı bir versiyonudur. Rastgele orman metodunda olduğu gibi veri kümesinin kopyaları kullanılarak model eğitilir, ancak düğümlerin dallara ayrılma aşamasında karar kriteri kullanarak optimum ayrılmayı yapmak yerine rastgele dallanma yoluna gidilir. Bu metot, bazı veri analizi problemlerinin çözümünde karmaşıklığı ve işlem yükünü azaltmasına rağmen yüksek gürültü barındıran büyük veri kümelerinin analizinde performansı düşüktür. İstatistiksel açıdan değerlendirildiğinde bu yöntem genellikle bias artışına sebep olurken varyansı düşürür [33].

Pasif-Agresif (PA) Algoritmalar, çevrimiçi öğrenme algoritmalarıdır. Genellikle büyük ölçekli veriler için kullanılır. Tüm eğitim veri kümesinin bir anda kullanıldığı toplu öğrenmenin aksine çevrimiçi öğrenme algoritmalarında, giriş verileri sırayla gelir ve model adım adım güncellenir.

Çok Katmanlı Algılayıcılar (MLP), günümüzde birçok problemin çözümünde kullanılmaktadır. Bugün özellikle sınıflandırma işlemlerinde en çok kullanılan yöntemlerin başında gelmektedir. MLP' de Delta öğrenme kuralı denilen bir öğrenme yöntemini kullanılmaktadır. Bu kuralın amacı; ağın istenen çıktı ile ürettiği çıktı arasındaki hatayı minimum yapmaktır. MLP'ler; girdi katmanı, gizli katmanlar ve çıktı katmanı olmak üzere 3 katmandan oluşmaktadır. Bilgiler girdi katmanından ağa tanıtılır, gizli katmanlardan çıktı katmanına ulaşır ve çıktı katmanından dış dünyaya aktarılır. MLP'lerde; eğiticili öğrenme yöntemi kullanılmaktadır. Ağa hem örnekler hem de bu örneklerden oluşturulması gereken çıktılar sunulmaktadır. Ağ; örneklere bakarak problem uzayında bir çözüm üretir, bu genellemeye bağlı olarak gelecek yeni örnekler için de çözüm üretebilmektedir [34].

Gradyan Artırma Algoritması (GB), sınıflandırma ve regresyon için kullanılan denetimli bir makine öğrenme yöntemidir. Adaptif Yükseltme algoritmasına benzer şekilde, zayıf sınıflandırma modellerinin bir kombinasyonu, genellikle bir karar ağacı, modeli oluşturulur. Bu yöntem, her adımda yinelemeli olarak kayıp fonksiyonunu en iyi azaltan yeni bir karar ağacı ekleyerek, yüksek tahmin doğruluğuna sahip güçlü bir sınıflandırıcı elde etmeyi amaçlar.

Geliştirilen sistemde "scikit-learn" kütüphanesi içerisinde yer alan sınıflandırıcılar diğer çalışmalar ile karşılaştırma yapılabilmesi adına varsayılan parametreler ile kullanılmıştır.

Değerlendirme Metrikleri

Makine öğrenimi sınıflandırıcılarının performanslarını değerlendirmek için yaygın olarak birkaç ölçüm kullanılır. Önerilen modeli değerlendirmek için aşağıdaki performans ölçütleri kullanılmıştır [35]:

Denklemlerdeki TP, FP, TN ve FN sırasıyla doğru pozitif, yanlış pozitif, doğru negatif ve yanlış negatifi temsil etmektedir.

Doğruluk (Accuracy); Doğru şekilde sınıflandırılan örneklerin toplam örnek sayısına oranıdır.

$$Doğruluk = \frac{TP + TF}{TP + TF + FP + FN} \quad (1)$$

Kesinlik (Precision); Doğru sınıflandırılmış pozitif örnek sayısının, toplam pozitif tahminlenmiş örnek sayısına oranıdır.

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

Duyarlılık (Recall); Doğru sınıflandırılmış pozitif örnek sayısının, toplam pozitif örnek sayısına oranıdır.

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

Kesinlik ve duyarlılık ölçütleri tek başına anlamlı bir karşılaştırma sonucu çıkarmamız için yeterli değildir. Kesinlik, Tip 2 Hata (False Negative) değerini, duyarlılık ise Tip 1 Hata (False Positive) değerini dikkate almaz. Bu sebeple, her iki ölçütün beraber değerlendirildiği tüm hata maliyetlerini içeren F1-Skor tanımlanmıştır. F1-Skor, kesinlik ve duyarlılığın harmonik ortalamasıdır.

$$F1 - Skor = \frac{2 * Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık} \quad (4)$$

Uygulama ve Başarımlar

Önerilen sistemi gerçeklemek için Python programlama dili, makine öğrenimi ile veri işleme araçlarından Sklearn [36], Numpy [37] ve Pandas [38] kütüphaneleri ile birlikte kullanılmıştır. Aşağıdakiler değerlendirmeler için kullanılan 64-bit Microsoft Windows işletim sistemli bilgisayara ait teknik özelliklerdir:

- CPU: Intel Core i7-7700K @ 4.2 GHz
- RAM: 32 GB

Veri Önişleme

Veri kümesinden CICFlowMeter-V3 ile öznitelikler çıkarılmış ve Flow ID, Source IP, Source Port, Destination IP ve Destination Port öznitelikleri veri kümesinden silinmiştir. Ek olarak bir saldırı için saldırı zamanı bilgisi önemsiz olduğundan ve saldırı zamanının saldırı durumu veya tipi ile herhangi bir ilişkisi bulunmadığından Timestamp özniteliği de veri kümesinden silinmiştir. Hataları önlemek için 'Infinity' ve 'NaN' değerleri uygun değerler ile değiştirilmiştir. InitFwd Win Byts ve InitBwd Win Bytes sütunları bazı örneklerde -1 değerini içermektedir. Bu sebeple InitFwd Win BytsNeg ve InitBwd Win BytsNeg şeklinde iki yeni sütun oluşturulmuştur. Oluşturulan sütunlara orijinal öznitelikler göz önüne alınarak; -1 değerini içeren veri ile karşılaştığında 1, aksi durumda 0 değerleri atanmıştır. Son olarak veri ölçeklendirilmiş ardından normalize edilmiştir.

Veri kümesinde 16.232.943 adet veri olduğundan sadelik ihtiyacı ve hesaplama süresinin azaltılması açısından Near-Miss alt örnekleme algoritması kullanılarak veri kümesi %50 oranında küçültülmüştür. Bu küçültme sadece veri sayısı üzerinde yapılmış öznitelik sayısında herhangi bir değişiklik yapılmamıştır. Veri kümesi %50 oranında küçüldükten sonra veri sayısı 8.116.473 olmuştur. Küçültme işlemi, en yakın üç azınlık sınıfı örneğine minimum ortalama mesafeye sahip çoğunluk sınıf örnekleri üzerinden eleme yapılarak, atak dağılımları ve orijinal veri kümesi ile elde edilen doğruluk ve F1-Skor değerleri yaklaşık eşit olacak şekilde yapılmıştır. Tablo 2 ve Tablo 3 orijinal ve küçültülmüş veri kümesi için Doğruluk, F1-Skor ve Süre değerlerinin karşılaştırmalarını içermektedir.

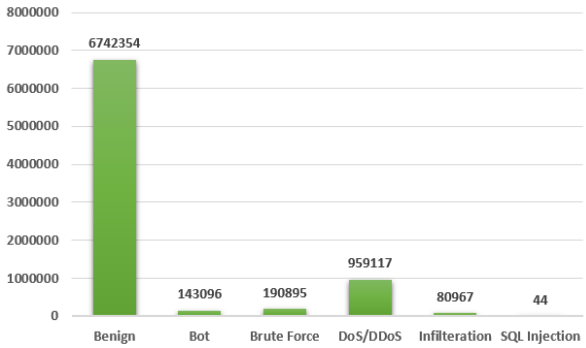
Tablo 2. Lojistik regresyon tabanlı 5-katlamalı çapraz doğrulama sonuçları

Veri kümesi	Doğruluk (%)	F1-Skor (%)	Süre (dk)
Orijinal	95,13	90,80	14,50
Küçültülmüş	95,15	90,79	08,05

Tablo 3. Karar ağaçları tabanlı 5-katlamalı çapraz doğrulama sonuçları

Veri kümesi	Doğruluk (%)	F1-Skor (%)	Süre (dk)
Orijinal	98,75	97,33	78,29
Küçültülmüş	98,65	97,34	4930

Şekil 5 küçültülmüş veri kümesindeki her etiket için atak dağılımlarını göstermektedir. Çalışmanın bundan sonraki kısmında küçültülmüş veri kümesi ile işlem yapılmıştır.



Şekil 5. Etiket kategori dağılımı

'Label' sütunu verinin hangi saldırı türünde olduğunu göstermektedir. Bu sütun ikili sınıflandırmaya uygun olacak şekilde sayısallaştırılmıştır. Tablo 4 veri kümesindeki normal ve kötü amaçlı ağ trafik yüzdesini göstermektedir.

Tablo 4. Normal ve kötü amaçlı ağ trafik yüzdesi

	Sayı	Yüzde (%)
Normal	6.742.354	83,07
Saldırı	1.374.119	16,93

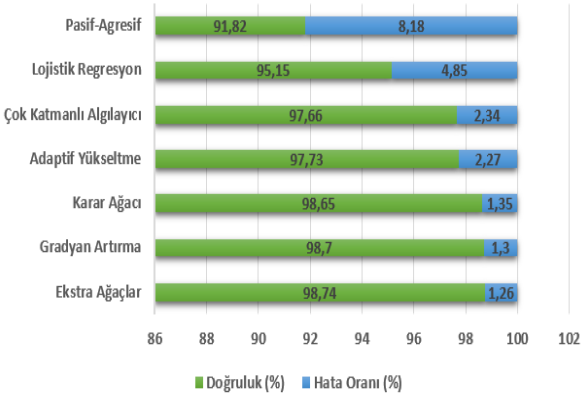
Öznitelik Seçimi

Çapraz doğrulama işleminin hesaplama ve zaman yönünden maliyetli olması sebebiyle *joblib* kütüphanesi kullanılarak çok-çekirdekli paralelleştirme (multi-core parallelism) uygulanmış, çalışma zamanı düşürülmüştür. Makine Öğrenmesi modellerinin veri kümesinde bulunan tüm öznitelikler ile çalıştırılması sonucu elde edilen doğruluk değeri başarımları ve işlem süreleri Tablo 5 de gösterilmektedir. Elde edilen sonuçlar ağaç tabanlı algoritmaların başarımlarının yüksek olduğunu göstermektedir. En yüksek doğruluk değeri 98,74 ile Ekstra Ağaçlar modeline aittir. Doğruluk değeri ve işlem süresi birlikte değerlendirildiğinde Karar Ağacı ve Lojistik Regresyon modellerinin de ön plana çıktığı görülmektedir.

Tablo 5. Tüm öznitelikler ile model başarımları

Model	Doğruluk (%)	Hata (%)	Süre (dk)
ET	98,74	1,26	20,53
GB	98,70	1,30	157,55
DT	98,65	1,35	14,04
ADA	97,73	2,27	35,36
MLP	97,66	2,34	270,38
LR	95,15	4,85	3,09
PA	91,82	8,18	1,58

Şekil 6 verilen algoritmaların tüm özniteliklerle çalıştırılması sonucu oluşan doğruluk ve hata oranlarının değerlerini göstermektedir. Bu değerler incelendiğinde en optimum sonucu Ekstra Ağaçlar algoritmasının sağladığı görülmektedir.



Şekil 6. Tüm öznitelikler ile ulaşılan doğruluk ve hata oranı

STS için en uygun veri yapısı ve içeriğini belirlemek için öznitelik seçim yöntemlerinin sonuçları karşılaştırılmıştır. Öznitelikler 3 yaklaşım ile belirlenmiştir.

1. Her bir özniteliğin puanı Ki-Kare testi uygulanarak hesaplanmış, düşük puanlı öznitelikler kaldırılmıştır.
2. Yüksek korelasyona sahip öznitelikler Spearman korelasyon analizi ile belirlenerek kaldırılmıştır.
3. Özyinelemeli Öznitelik Eliminasyonu ile, tüm öznitelikler sıralanarak en zayıf öznitelikler belirtilen öznitelik sayısına ulaşılıncaya kadar kaldırılmıştır.

Ki-Kare test sonuçları Tablo 6'da sunulmuştur. Tablodaki hücreler modelde kullanılacak yüksek puanlı öznitelikleri göstermektedir.

Tablo 6. Ki-Kare öznitelik seçimi

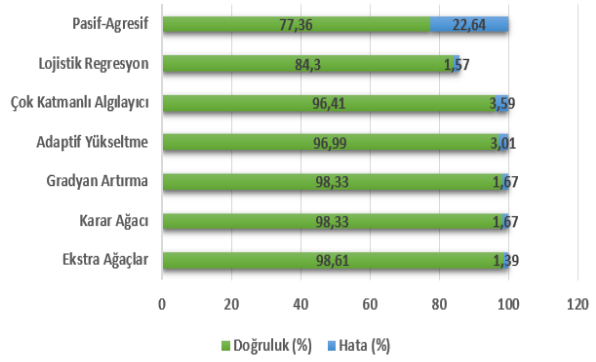
Öznitelik ismi	**Puan**	**Öznitelik İsmi**	**Puan**
FlowDuration	6,37	Bwd IAT Max	6,07
TotBwdPkts	5,92	BwdPkts/s	5,52
TotLenBwdPkts	5,69	PktLenMin	5,06
FwdPktLenMax	7,21	PktLenMax	9,00
FwdPktLenMin	6,15	PktLenMean	6,16
FwdPktLenStd	8,00	PktLenStd	7,52
BwdPktLenMax	6,36	Down/UpRatio	9,23
BwdPktLenMin	7,96	Pkt Size Avg	6,27
BwdPktLenStd	6,74	SubflowBwdPkts	5,92
Flow IAT Mean	6,38	SubflowBwdByts	5,69
Flow IAT Std	6,31	FwdAct Data Pkts	5,80
Flow IAT Min	6,38	Active Std	9,82
Fwd IAT Tot	6,37	IdleMean	5,16
Fwd IAT Mean	6,38	IdleMin	8,39
Fwd IAT Std	7,35	InitFwd Win BytsNeg	5,84
Fwd IAT Min	6,38		

Tablo 7 belirlenen öznitelikler ile modellerin doğruluk değer karşılaştırmalarını içermektedir. Sonuçlar ağaç tabanlı algoritmaların başarımlarının yüksek olduğunu fakat tüm özniteliklerle elde edilen sonuçlar ile karşılaştırıldığında doğruluk değerlerinin az miktarda düştüğünü göstermektedir. Ek olarak, veri boyutunun indirgenmesi işlem yükünü azalttığından dolayı işlem süresi kısalmıştır.

Tablo 7. Ki-Kare öznitelik seçim yöntemi ile model başarımları

Model	**Doğruluk (%)**	**Hata (%)**	**Süre (dk)**
ET	98,61	1,39	16,26
DT	98,33	1,67	3,39
GB	98,33	1,67	75,55
ADA	96,99	3,01	16,43
MLP	96,41	3,59	239,29
LR	84,30	1,57	1,06
PA	77,36	22,64	0,25

Şekil 7 verilen algoritmaların Ki-Kare öznitelik seçimi ile belirlenen öznitelikler kullanılarak çalıştırılması sonucu oluşan doğruluk ve hata oranlarının değerlerini göstermektedir. Şekil 6'ya benzer olarak en başarılı algoritmanın Ekstra Ağaçlar olduğu görülmüştür.



Şekil 7. Ki-Kare özellik seçimi ile ulaşılan doğruluk ve hata oranı

Spearman korelasyon analizine ait sonuçlar Tablo 8'de gösterilmektedir. Bu çalışmada, yüksek korelasyonları belirlemek için 0,8 eşik değeri kullanılmıştır. Uygulama sonrası 77 öznitelik içerisinden 25 öznitelik belirlenmiştir.

Tablo 8. Spearman'ın sıralama korelasyon katsayısı öznitelik seçimi

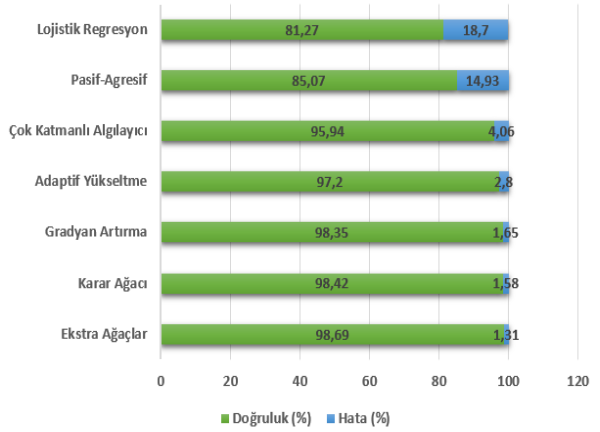
	Öznitelik ismi		**Öznitelik ismi**
1	Protocol	14	CWE FlagCount
2	TotFwdPkts	15	Down/UpRatio
3	FwdPktLenMin	16	FwdByts/b Avg
4	FlowPkts/s	17	FwdPkts/b Avg
5	Fwd PSH Flags	18	FwdBlk Rate Avg
6	Bwd PSH Flags	19	BwdByts/b Avg
7	Fwd URG Flags	20	BwdPkts/b Avg
8	Bwd URG Flags	21	BwdBlk Rate Avg
9	BwdPkts/s	22	Active Mean
10	FIN FlagCnt	23	Active Std
11	RST FlagCnt	24	IdleMean
12	PSH FlagCnt	25	InitFwd Win BytsNeg
13	URG FlagCnt		

Tablo 9 belirlenen öznitelikler ile modellerin doğruluk değer karşılaştırmalarını içermektedir. Sonuçlar ağaç tabanlı modellerin işlem süresi ve doğruluk değerleri yönlerinden Ki-Kare testine ait sonuçlara kıyasla daha başarılı olduğunu göstermektedir. Uygulanan her iki öznitelik seçim yöntemi tüm öznitelikler ile elde edilen sonuçlar ile karşılaştırıldığında işlem süresi yönünden olumlu bir etki oluştursa da doğruluk değerlerinde bir artış oluşturmamıştır. Özellikle lineer modellerin başarımı düşmüştür.

Tablo 9. Spearman'ın sıralama korelasyon katsayısı öznitelik seçim yöntemi ile model başarımları

Model	**Doğruluk (%)**	**Hata (%)**	**Süre (dk)**
ET	98,69	1,31	12,10
DT	98,42	1,58	1,32
GB	98,35	1,65	28,23
ADA	97,20	2,80	7,53
MLP	95,94	4,06	194,53
PA	85,07	14,93	0,18
LR	81,27	18,7	0,53

Şekil 8 verilen algoritmaların Spearman'ın Sıralama Korelasyon Katsayısı ile seçilen özniteliklerle çalıştırılması sonucu oluşan doğruluk ve hata oranlarının değerlerini göstermektedir. 25 öznitelik ile çalışan makine öğrenmesi algoritmalarının hata oranlarının tüm öznitelikler kullanılarak elde edilenlere göre daha yüksek olduğu görülmektedir.



Şekil 8. Spearman'ın sıralama korelasyon katsayısı özellik seçimi ile ulaşılan doğruluk ve hata oranı

RFE yönteminde ilk aşama tüm öznitelik kümesinin kullanılarak bir model oluşturulması ve her öznitelik için bir önem puanının hesaplanmasıdır. Sonraki aşamada en az önem puanına sahip öznitelikler ortadan kaldırılarak model yeniden oluşturulur ve önem puanları tekrar hesaplanır. Bu işlem öznitelik kümesinde istenilen sayıda öznitelik kalana kadar devam ettirilir. Dolayısıyla, seçim sonunda istenilen öznitelik alt kümesi bir ayar parametresidir.

Bu yöntemde belirlenmesi gereken bir diğer parametre ise özniteliklerin önem puanlarının belirleneceği makine öğrenme yöntemidir. Tüm öznitelikler ile elde edilen Karar Ağacı model başarımının doğruluk ve süre metrikleri yönünden diğer modellere kıyasla daha başarılı olması sebebiyle önem puanlarının belirlenmesinde kullanılacak makine öğrenimi yöntemi Karar Ağacı olarak belirlenmiştir. Model başarımları 60, 50, 40 ve 30 öznitelik sayısı için incelenmiş ve 40 öznitelik sayısının model başarımını artırdığı ve işlem süresini azalttığı tespit edilmiştir. Tablo 10 Özyinelemeli Öznitelik Eliminasyonuna ait sonuçları göstermektedir.

Tablo 10. Özyinelemeli öznitelik eliminasyonu

	Öznitelik ismi		**Öznitelik ismi**
1	Protocol	21	Bwd IAT Min
2	FlowDuration	22	FwdHeaderLen
3	FwdPktLenMax	23	BwdHeaderLen
4	FwdPktLenMean	24	FwdPkts/s
5	FwdPktLenStd	25	BwdPkts/s
6	BwdPktLenMax	26	PktLenMax
7	BwdPktLenStd	27	PktLenStd
8	FlowByts/s	28	RST FlagCnt

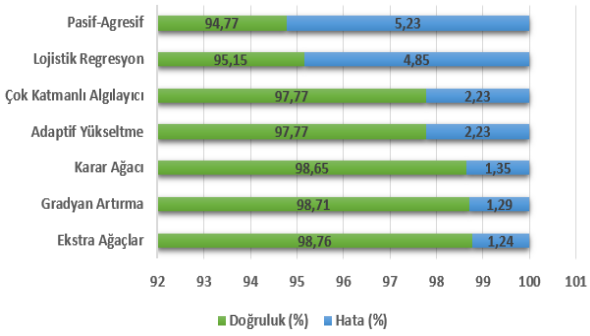
9	FlowPkts/s	29	PSH FlagCnt
10	Flow IAT Mean	30	ACK FlagCnt
11	Flow IAT Max	31	URG FlagCnt
12	Flow IAT Min	32	ECE FlagCnt
13	Fwd IAT Tot	33	Pkt Size Avg
14	Fwd IAT Mean	34	BwdSeg Size Avg
15	Fwd IAT Max	35	FwdAct Data Pkts
16	Fwd IAT Min	36	FwdSeg Size Min
17	Bwd IAT Tot	37	Active Max
18	Bwd IAT Mean	38	Active Min
19	Bwd IAT Std	39	IdleMin
20	Bwd IAT Max	40	InitBwd Win BytsNeg

Tablo 11 belirlenen öznitelikler ile modellerin doğruluk değer karşılaştırmalarını içermektedir.

Tablo 11. RFE yöntemi ile model başarımları

Model	Doğruluk (%)	Hata (%)	Süre (dk)
ET	98,76	1,24	8,19
GB	98,71	1,29	117,04
DT	98,65	1,35	6,34
ADA	97,77	2,23	24,24
MLP	97,77	2,23	177,19
LR	95,15	4,85	1,20
PA	94,77	5,23	0,35

Şekil 9 verilen algoritmaların RFE yöntemi ile seçilen özniteliklerle çalıştırılması sonucu oluşan doğruluk ve hata oranlarının değerlerini göstermektedir. Bu tabloya göre RFE uygulanmış veri kümesi ile çalışan makine öğrenmesi algoritmalarının hata oranı düşmüş veya sabit kalmıştır.



Şekil 9. RFE ile özellik seçimi ile ulaşılan doğruluk ve hata oranı

Tablo 11 de yer alan sonuçlar ve tüm öznitelikler ile elde edilen sonuçlar doğruluk ve süre metrikleri üzerinden karşılaştırıldığında özyinelemeli sistemin başarımı artırdığı görülmüştür. Ancak yalnızca doğruluk oranı CSE-CIC-IDS2018 gibi dengesiz bir veri kümesinde yanıltıcı bir metrik olabileceğinden kesinlik, duyarlılık, F1-Skor değerleri de incelenmiştir. Tablo 12 tüm öznitelikleri kullanılarak elde edilen başarımları içermektedir.

Tablo 12. Tüm öznitelikler - kesinlik, duyarlılık, F1-skor değerleri

Model	Kesinlik (%)	Duyarlılık (%)	F1-Skor (%)
ADA	96,44	94,47	95,42
DT	97,40	97,28	97,34
LR	89,35	92,42	90,79
MLP	95,36	96,04	95,61
ET	97,82	97,19	97,50
PA	90,07	87,46	90,47
GB	98,51	96,32	97,38

Tablo 13 RFE yöntemi ile belirlenen öznitelikleri içeren veri kümesine ait sonuçları göstermektedir.

Tablo 13. RFE yöntemi ile belirlenen öznitelikler – kesinlik, duyarlılık, F1-skor değerleri

Model	Kesinlik (%)	Duyarlılık (%)	F1-Skor (%)
ADA	96,58	94,49	95,50
DT	97,40	97,28	97,34
LR	89,35	92,41	90,79
MLP	95,51	95,80	95,68
ET	97,86	97,21	97,53
PA	87,03	86,19	79,62
GB	98,52	96,34	97,39

Elde edilen sonuçlar, RFE yöntemi kullanıldığında Pasif-Agresif sınıflandırma modelinin doğruluk değerlerinin arttığını fakat saldırı tespit başarısının düştüğünü göstermiştir. Diğer modeller için kesinlik ve duyarlılık değerlerinin genel olarak iyileştiği, doğruluk değeriyle birlikte saldırı tespit başarısının da arttığı görülmüştür. Kesinlik ve duyarlılık metrikleri tek başlarına model başarımını değerlendirme için yeterli metrikler değildir. Kesinlik False Negative (normal trafik olarak yanlış niteleme) değerini, duyarlılık ise False Positive (saldırı olarak yanlış niteleme) değerini dikkate almaz. Bu yüzden F1-Skor metriği model başarımının değerlendirilmesi için en anlamlı ölçüm yöntemidir. En yüksek doğruluk ve saldırı tespit başarımı Ekstra Ağaçlar modeline ait olsa da süre metriği dikkate alındığında 98,65 oranı ile Karar Ağacı modelinin daha başarılı olduğu söylenebilir.

Sonuç ve Gelecek Çalışmalar

Bu çalışmada CSE-CIC-IDS2018 veri kümesi üzerinde farklı öznitelik seçim yöntemleri kullanılarak çeşitli STS modelleri geliştirilmiştir. Ki-Kare Testi, Spearman'ın Sıralama Korelasyon Katsayısı ve Özyinelemeli Öznitelik Eliminasyonu (RFE) olmak üzere 3 farklı öznitelik seçim yöntemi kullanılarak elde edilen yeni veri kümelerinin, orijinal boyuttaki veri kümesi ile karşılaştırılması için Adaptif Yükseltme, Karar Ağacı, Lojistik Regresyon, Çok Katmanlı Algılayıcı, Ekstra Ağaçlar, Pasif-Agresif ve Gradyan Artırma sınıflandırma algoritmaları kullanılmıştır. Tüm deneylerin Tabakalı 5-Katlamalı Çapraz Doğrulama ile gerçekleştirilmesi sebebiyle oluşan hesaplama ve zaman maliyeti çok-çekirdekli paralelleştirme (multi-core parallelism) uygulanarak düşürülmüştür. Ki-Kare istatistiksel testi ve Spearman korelasyon analizi uygulanarak oluşturulan yeni veri kümeleri ile elde edilen model başarımlarında, ağaç tabanlı yöntemlerin başarımlarının %97'nin üzerinde olduğu görülmektedir. Spearman korelasyon analizine ait sonuçlar ağaç tabanlı modellerin işlem süresi ve doğruluk değerleri yönlerinden Ki-Kare testine ait sonuçlara kıyasla daha başarılı olduğunu göstermektedir. Uygulanan her iki öznitelik seçim yöntemi tüm öznitelikler ile elde edilen sonuçlar ile karşılaştırıldığında işlem süresi yönünden olumlu bir etki oluştursa da doğruluk değerlerinde bir artış oluşturmamıştır. Özellikle lineer modellerin başarımı düşmüştür. RFE yöntemi kullanılarak elde edilen sonuçlar diğer iki yöntemdeki gibi doğruluk ve süre metrikleri yönlerinden incelendiğinde tüm modeller için sistemin başarımının arttığı görülmüştür.

Bu aşamada yalnızca doğruluk oranı CSE-CIC-IDS2018 gibi dengesiz bir veri kümesinde yanıltıcı bir metrik olabileceğinden kesinlik, duyarlılık, F1-Skor değerleri de incelenmiştir. Elde edilen sonuçlar, RFE yöntemi kullanıldığında Pasif-Agresif sınıflandırma modelinin doğruluk değerlerinin arttığını fakat saldırı tespit başarısının düştüğünü göstermiştir. Diğer modeller için doğruluk değeriyle birlikte saldırı tespit başarısının da arttığı görülmüştür. En yüksek doğruluk ve saldırı tespit başarımı %98,76 oranı ile Ekstra Ağaçlar modeline ait olsa da süre metriği dikkate alındığında %98,65 oranı ile Karar Ağacı ve %95,15 oranı ile Lojistik Regresyon modelleri de ön plana çıkmaktadır. Bu bağlamda, doğru ayar parametreleri ile kullanılan RFE öznitelik seçim yöntemi ile makine öğrenimi tabanlı STS'lerin başarım ve performanslarının artırabileceği görülmüştür. İleriki çalışmalarda, -saldırı tespitinin başarımının ve performansının artırılması amaçlı- bir makine öğrenimi yönteminin tahmin performansından ziyade, birden fazla makine öğrenimi yönteminin tahminlerini birleştiren toplu bir öğrenme modeli önerilmesi ve önerilen modelin Derin Öğrenme yaklaşımları kullanılarak karşılaştırmalı bir analizinin yapılması planlanmaktadır.

Kaynaklar

[1] Kim, K., Aminanto, M.E., Tanuwidjaja, H.C. (2018). Network Intrusion Detection Using Deep Learning SpringerBriefs on Cyber Security Systems and Networks.

[2] Mishra, P., Varadharajan, V., Tupakula, U., Pilli, E.S. (2018). A Detailed Investigation and Analysis of Using Machine Learning Technique for Intrusion Detection. IEEE, 2018.

[3] Gao, X., Shan, C., Hu, C., Niu, Z., Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. IEEE, 2019.

[4] Sharafaldin, I., Lashkari A.H., Ghorbani A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterizaion. ICISSP.

[5] Zhou, Y., Cheng, G., Jiang, S., Dai, M. (2019). An Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier.

[6] Sommer, R., Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection", IEEE Symposium on security and Privacy.

[7] Aljawarneh, S., Aldawairi, M., Yassein, M. B. (2018) Anomaly-based Intrusion Detection System Through Feature Selection Analysis and Build Hybrid Efficient Model. Journal of Computational Science.

[8] Wankhede, S., Kshirsagar, D. (2018). DoS Attack Detection Using Machine Learning and Neural Network. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, pp. 1-5. Conference on Information Systems Security and Privacy (ICISSP), Portugal.

[9] Zhou, Q., Pezaros, D., (2019). Evaluation of Machine Learning Classifier for Zero-Day Intrusion detection-An Analysis on CIC AWS 2018 Dataset. School of Computing Science, University of Glasgow.

[10] Kanimozhi, V., Jacob, T.P. (2019). Artificial Intelligence based Network Intrusion Detection with Hyper-Parameter Optimization Tuning on The Realistic Cyber Dataset CSE-CIC-IDS2018 Using Cloud Computing". International Conference on Communication and Signal Processing.

[11] Yulianto, A, Sukarno, P., Suwastika, N. A. (2017). Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset. Journal of Physics: Conference Series, 1192.

[12] Wani, A.R., Rana, Q. P., Saxena, U., Pandey, N. (2019). Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques," 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, pp. 870-875.

[13] McKay, R., Pendleton, B., Britt, J., Nakhavanit, B. (2019). Machine Learning Algorithms on Botnet Traffic: Ensemble and Simple Algorithms. The International Conference on Compute and Data Analysis 2019 (ICCDA).

[14] Kanimozhi, V., Jacob, T.P. (2019). Calibration of Various Optimized Machine Learning Classifiers in Network Intrusion Detection System on the Realistic Cyber Dataset CSE-CIC-IDS2018 Using Cloud Computing. International Journal of Engineering AppliedSciencesandTechnology,2019 Vol.4, Issue 6, ISSN No. 2455-2143, Pages 209-213.

[15] Ferrag, M.A., Maglaras, L. (2019). DeliveryCoin: An IDS and Blockchain-Based Delivery Framework for Drone-Delivered Services. Computers 2019, 8, 58.

[16] Atay, R., Odabaş, D. E., Pehlivanoğlu, M. K. (2019). İki Seviyeli Hibrit Makine Öğrenmesi Yöntemi ile Saldırı Tespiti. Dergipark, 258-272.

[17] De Lima Filho, F.S., Silveira, F.A.F., De Medeiros Brito Junior, A., Vargas-Solar, G., Silveira, L.F. (2019). Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning," Security and Communication Networks, vol. 2019, Article ID 1574749, 15 pages.

[18] Fitni, Q.R.S., Ramli, K. (2020). Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems", Proc. IEEE Int. Conf. Ind. 4.0 Artif. Intell. Commun. Technol. (IAICT), pp. 118-124.

[19] Cil, A. E., Yildiz, K., Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. Expert Systems with Applications, 169, 114520.

[20] Arslan, R. S. (2021). FastTrafficAnalyzer: An Efficient Method for Intrusion Detection Systems to Analyze Network Traffic. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 12(4), 565-572.

[21] Emhan, Ö., Akın, M. (2019). Filtreleme tabanlı öznitelik seçme yöntemlerinin anomali tabanlı ağ saldırısı tespit sistemlerine etkisi. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(2), 549-559.

[22] Thomas, R., Pavithran, D. (2018). A Survey of Intrusion Detection Models based on NSL-KDD Data Set. 2018 Fifth HCT Information Technology Trends (ITT), Dubai, United Arab Emirates, 286-291.

[23] Athmaja, S., Hanumanthappa, M., Kavitha, V. (2017). A Survey of Machine Learning Algorithms for Big Data Analytics. 2017 International Conference on Innovations in Information, Communication Coimbatore, 1-4.

[24] Sahingoz, O, Çebi, C, Bulut, F, Fırat, H, Karataş, G. (2019). Saldırı Tespit Sistemlerinde Makine Öğrenmesi Modellerinin Karşılaştırılması. Erzincan Üniversitesi Fen Bilimleri Enstitüsü Dergisi 12 (2019): 1513-1525

[25] Amrita, M.A. (2013) Performance Analysis of Different Feature Selection Methods in Intrusion Detection. Int J Sci Technol Res 2(6):225–231

[26] CSE-CIC-IDS-2018 dataset from University of NewBrunswick, available online: https://www.unb.ca/cic/datasets/ids-2018.html

[27] CICFlowMeter: Network Traffic Flow Analyzer,http://netflowmeter.ca/netflowmeter.html, Accessed 28 July 2018.

[28] Saeys, Y., Inza, I., Larranaga, P. (2007). A Review of Feature Selection Techniques in Bioinformatics, Bioinformatics, 23(19), 2507-2517.

[29] Bisyron, W., Kalamullah, R., Hendri, M. (2018). Implementation and Analysis of Combined Machine Learning Method for Intrusion Detection System. International Journal of Communication Networks and Information Security.

[30] Zhang W.Y., Wei Z.W, Wang, B.H., Han, X.P. (2016). Measuring Mixing Patterns in Complex Neteorks by Spearman Rank Correlation Coefficient, Physica A 451.

[31] Solomatine, D.P., Shrestha, D.L. (2004). AdaBoost. RT: A Boosting Algorithm for Regression Problems, Neural Networks, Vol 2, 1163 – 1168.

[32] Bauer, E., Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants, Machine Learning., Volume 36, Issue 1, pp 105-139.

[33] Geurts, P., Ernst, D., Wehenkel, L. (2006). Extremely randomized trees. Machine learning 63(1): 3-42.

[34] Çatal, Ç., Özyılmaz, L. (2005). Analysis of Multiple Myeloma Gene Expression Data by Multilayer Perceptron, National Conference on Biomedical Engineering.

[35] Sokolova, M., Lapalme, G. (2009). A Systematics Analysis of Performance Measures for Classification Tasks. Information processing and management.

[36] Scikit-learn.org. scikit-learn: machine learning in Python — scikit-learn 1.0.1 documentation. [online] https://scikit-learn.org/stable/, 2021.

[37] NumPy. Numpy documentation. [online] https://numpy.org/, 2021.

[38] Pandas. Pandas Python Data Analysis Library documentation. [online] https://pandas.pydata.org/, 2021.