NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 2022; 11(4), 862-870

Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences

Araştırma makalesi / Research article

www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

Makine öğrenme yöntemleri ile ağ trafik analizi

Network traffic analysis with machine learning methods

Bülent Tuğrul[1,*], Adil Shihab Ahmed Ahmed [2]

[1] Ankara Üniversitesi, Bilgisayar Mühendisliği Bölümü, 06830, Ankara Türkiye
[2] Ankara Üniversitesi, Adli Bilişim Bölümü, 06590, Ankara Türkiye

Öz

Saldırı Tespit Sistemleri (STS) ağa yapılan saldırıları ağ yöneticilerine bildirmek için kullanılan tekniklerden biridir. Her ne kadar çeşitli anomali tespit teknikleri geliştirilmiş olsa da, bu alanda yüksek veri boyutsallığı, hesaplama karmaşıklığı üzerindeki etki, ve hesaplama süresi gibi zorluklar ve sorunlar bulunmaktadır. Bunun yanı sıra saldırı tespit sistemlerinin yanlış alarm vermeleri de anomali trafik tespit sorunlarından biri olmakta, bu sorunları aşmak için makine öğrenme yöntemlerine başvurarak sorunların azaltılması ve saldırı tespit sistemlerinin performansını yükseltilmesi amacıyla kullanılmaktadır. Bu çalışmada saldırı tespit sistemlerinin performansını yükseltmek amacıyla belirlediğimiz makine öğrenme yöntemlerini uygulayarak en iyi performansı gösteren J48 algoritması olup %99.92 bir doğruluk oranı elde edilmiştir. Bu algoritma saldırı tespit sistemleri tarafından kullanılması için önerilen algoritma olup STS'nin çeşitli ağ trafiğini ayırt etmesine ve dışarıdan gelen trafiği saldırı olup olmadığına karar vermesinde yardımcı olacaktır.

Anahtar Kelimeler: Ağ güvenliği, Hizmet engelleme saldırısı, Makine öğrenmesi, Saldırı tespit sistemi.

Abstract

Intrusion Detection Systems (IDS) are one of the techniques used to notify network administrators of attacks on the network. Although various anomaly detection techniques have been developed, there are challenges and problems in this area, such as high data dimensionality, impact on computational complexity, and computation time. In addition, false alarms by intrusion detection systems are one of the problems in detecting anomaly traffic. Machine learning methods are used to overcome these problems, reduce the issues, and increase the performance of intrusion detection systems. In this study, the decision tree algorithm shows the best performance by applying the machine learning methods we have determined to increase the performance of intrusion detection systems, and it has demonstrated an accuracy rate of 99.92%. This algorithm is recommended for use by intrusion detection systems in our study, and it will help STS distinguish between various network traffic and decide whether the incoming traffic is an attack or not.

Keywords: Network security, Distributed denial of service attack (DDoS), Intrusion detection system, Machine learning

1 Giriş

Dünya çapında bilgi ve iletişim teknolojisi alanındaki mevcut gelişmeler, ağ yöneticileri için büyük bir zorluk teşkil etmektedir. Nitekim bilgisayar ağları çok hızlı şekilde genişlemekte ve ağ kullanıcı sayısı her geçen gün artmaktadır. Artan ağ verimi ve güvenlik tehdidi ile birlikte, izinsiz giriş tespit sistemleri (STS'ler) üzerine yapılan çalışmalar, bilgisayar bilimi alanında büyük ilgi görmektedir.

Ağlarda bulunan kötü amaçlı yazılımlar verilere ve sistemlere karşı birçok tehdit oluşturmaktadır. Bu nedenle bu tehdidin güvence altına alınması gerekmektedir. Bu noktada ise bilgisayar ağlarında bulunan kötü amaçlı yazılımlara karşı verimli ve etkili yöntemlere gereksinim vardır. Ağ üzerinden geçen çok büyük miktarda veri, verinin önem ve gizliliği, veri güvenliğin sağlanması ve kullanıcı verilerinin korunması gerekliliği ile ilgili olarak günümüz dünyasında ağı yöneten ve sistemi olası hasarlara karşı koruyan bir güvenlik sistemine ihtiyaç duyulmaktadır.

Son zamanlarda dikkat çeken araçlardan biri de saldırı tespit sistemleridir. STS'ler, ağ trafiğini şüpheli etkinliklere karşı izler ve bu tür faaliyetleri keşfettiği anda harekete geçer. Bu sistem, zararlı etkinlik veya politika ihlaline karşı yapılan hamleleri yöneticiye bildirmekle yükümlüdür. Bu sistem, izinsiz girişin türü, konumu ve kaynağı hakkında bilgi verir. Aynı zamanda bu sistem, normal bir davranışı modelleyebilir ve modelden saparsa trafiği bir saldırı olarak algılar. Bu tür bir yaklaşıma anomali tespiti denir. Ayrıca farklı saldırıların davranışlarını modelleyebilir ve modele uyuyorsa trafiği bir saldırı olarak algılayabilir. Bu tür bir yaklaşıma kötüye kullanım tespiti veya imza tespiti denir.

Bu çalışmada makine öğrenme yöntemlerini kullanarak STS'lerin performansını yükseltmek için belirlediğimiz algoritmalar arasında en iyi performans gösteren yöntem saptanmıştır.

Bunun için CICIDS2017 veri seti kullanılmıştır. Bu veri setini, veri ön işlemeden geçirip veri dengeleme işlemine tabi tutularak temiz bir veri seti elde edildikten sonra makine öğrenme yöntemleri (Naive Bayes, Destek Vektör Makinesi

* Sorumlu yazar / Corresponding author, e-posta / e-mail: btugrul@eng.ankara.edu.tr (B. Tuğrul)
Geliş / Recieved: 15.05.2022 Kabul / Accepted: 25.07.2022 Yayımlanma / Published: 14.10.2022
doi: 10.28948/ngmuh.1113956

(DVM), K en yakın komşu algoritması (KNN), J48 ve Çok Katmanlı Algılayıcı (Multilayer Perceptron-MLP)) WEKA programı aracılığı ile uygulanmıştır ve bunların arasında en iyi performans gösteren yöntem saptanmıştır.

Saldırı tespit sistemleri tarafından kullanılan metotlar (algoritmalar) kimi zaman ağ üzerinden geçen trafiği yanlış algılayıp yöneticiye alarm olarak bildirmesi örneğin normal trafikleri saldırı olarak algılaması veya saldırı trafiklerini algılamamaları gibi sorunlar ile karşı karşıya kalabilmekte. Çalışmamız bu gibi sorunların iyileştirilmesini veya giderilmesini ve çok sayıda yanlış alarmlara karşı gerekli önlemlerin alınmasını amaçlamaktadır.

Çalışmanın hedefleri:

I. Boyut azaltma yöntemini kullanarak zamana dayalı verilere ve özellik çıkarımına dayalı öğrenme doğruluğunu dinamik olarak artıran otomatik bir sistem geliştirmek
II. CICIDS2017 veri setinin temiz ve dengeli bir alt kümesini seçmek
III. Belirlediğimiz yöntemleri kullanarak seçilen veri seti üzerinde uygulamak
IV. Bütün yöntemlerin performanslarının karşılaştırılması

En iyi performans gösteren yöntemin önerilmesi ve saldırı tespit sistemlerinde kullanılması sonucunda anormal trafiğin tespit edilmesi ve daha iyileştirilmesi sonucunda operatöre bildirilip gerekli önlemlerin alınması.

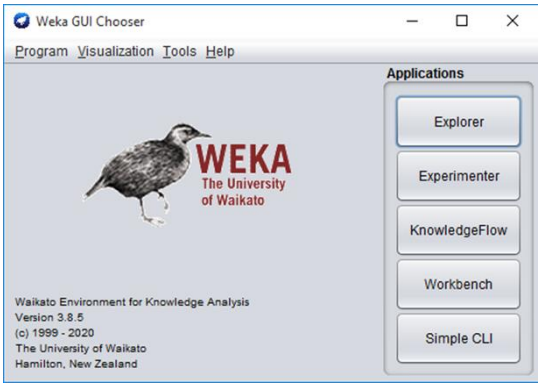
İlgili Çalışmalar: Son zamanlarda teknoloji alanındaki gelişmelere bağlı olarak ağ cihazlarının sayısı artış eğilimi göstermektedir. Bu durum güvenlik sorunu daha da önemli hale getirmektedir. Bu güvenlik problemine çözüm üretmek için KDD-Cup99 ve NSL-KDD gibi veri setleri oluşturuldu. Bu veri setlerinin eski olmaları, yeterli senaryolar içermemeleri ve barındırdıkları bazı problemler nedeniyle yeni veri setlerin oluşturulmasına gerek duyuldu. Ağa izinsiz giriş tespiti sistemlerinin geliştirilmesinde CICIDS2017 veri seti artık yaygın olarak kullanılmaktadır. Yulianto vd. [1] CICIDS2017 veri kümesi ile bir IDS sistemi tasarlamışlardır. Sistemin performansını artırmak için Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE), Temel Bileşen Analizi (PCA) ve Topluluk Özellik Seçim'ini (EFS) yöntemlerini çalışmalarında kullanmışlardır. Yapmış oldukları analizler sonucunda geliştirdikleri yöntemin SVM tabanlı yöntemden daha iyi sonuçlar ürettiğini ifade etmişlerdir. Engelen vd. [2] CICIDS2017 veri setinde tespit ettikleri trafik üretimi, akış yapısı, özellik çıkarma ve etiketleme ile ilgili bazı problemleri ortadan kaldırmak için veri seti üzerinde çeşitli ön işlem yöntemleri uygulamışlardır. Elde ettikleri yeni veri setinin %20'den fazla verisi yeniden oluşturularak etiketlenmiştir. Eski ve yeni veri seti çeşitli makine öğrenmesi yönetmeleri ile eğitilerek sonuçlar elde edilmiştir. Sonuçlar yeni oluşturulan veri setinin daha iyi performansa sahip olduğunu göstermektedir. Priyanka ve Kumar [3] diğer çalışmalardan farklı olarak bir derin ögrenme yapısı olan Evrişimli Sinir Ağlarını (CNN) geleneksel makine öğrenmesi yöntemleri ile beraber kullanmıştır. Ama CNN yöntemi rastgele ormandan daha iyi sonuç üretememiştir. Rosay vd. [4] benzer şekilde CICIDS2017 veri setinde gözlemledikleri bazı problemleri temizlemek için yeni bir özellik çıkarım aracı olan açık kaynak kodlu LycoSTand'u geliştirmişlerdir. Bu aracı ve orijinal CICIDS2017 veri setini kullanarak yeni LYCOS-IDS2017 veri setini oluşturmuşlardır. Orijinal ve yeni veri setleri arasındaki performans karşılaştırması gerçekleştirmişledir. Tüm makine öğrenimi yöntemlerinin önemli iyileşmeler gösterdiklerini savunmaktadırlar. Ayrıca, Rastgele Orman yöntemi tüm performans metrikleri dikkate alındığında diğer bütün yöntemlerden daha iyi değerler üretmiştir. Sonuç olarak yapılan tüm bu çalışmalar ışığında, gelen ağ trafiğini analiz ederek saldırıların önüne geçilmesi hususunda saldırı tespit sistemlerinin önemi vurgulanmıştır.

2 Materyal ve metot

Saldırı Tespit Sistemi veri analizi kapsamında yürütülen bu çalışma, CICIDS2017 veri seti, Windows 10 işletim sistemi, Intel(R) Core (TM) i3-3110M CPU, 4 GB Ram, 500 Gigabayt HDD nitelikli belleği olan dizüstü bilgisayarda WEKA (Waikato Environment for Knowledge Analysis) sürüm 3.8.5 uygulaması sayesinde yapılan deneyler sonucunda makine öğrenmesi ve veri madenciliği algoritmaları değerlendirilmiş ve performans ölçümleri incelenmiştir. Test çalışmalarına geçmeden önce kullandığımız metotlar WEKA programı aracılığı yapılmıştır. Bu program 2.1 başlık altında kısaca incelenmiştir.

2.1 WEKA Program

Waikato Environment for Knowledge Analysis (Weka), Yeni Zelanda Waikato Üniversitesinde Java ile yazılmış açık kaynak olarak geliştirilen bir makine öğrenimi yazılımdır. WEKA, bir dizi makine öğrenme algoritması kullanarak doğrudan verilere uygulayabilen veya Java kodundan çağırabilen bir veri madenciliği yazılımıdır. WEKA, "ARFF" (Attribute Relationship File Format) formatına sahip olmakla birlikte çıkan sonuçları görsel bir şekilde alınması da mümkündür Şekil 1'de WEKA programı giriş ekranı gösterilmiştir [5, 6].



Şekil 1. WEKA programı giriş ekranı

Explorer: Ham veriler üzerinde veri madenciliği görevlerini gerçekleştirmek için grafiksel ara yüz ile ekrandaki dosyaları açma veya alma, daha sonra çalışma hedefine göre veri işleme, çeşitli algoritmalar kullanarak (sınıflandırma,

kümeleme, ilişkilendirme algoritmalarını) bir çok analiz gerçekleştirilebilir [5].

Experimenter: Veri setleri üzerinde algoritma uygulamalarımıza deneyleri tasarlamamıza, çalıştırmamıza ve sonuçları analiz etmemize olanak tanır [6].

KnowledgeFlow: WEKA ara yüzü olan Explorer üzerinde bulunan bütün algoritmaları grafiksel olarak uygulamamızı sağlayan bir ara yüzdür, ayrıca KnowledgeFlow verileri artımlı veya toplu olarak işleyebilmektedir [6].

Workbench: Bu kullanıcı ara yüzü birçok zaman uzmanlar tarafından kullanılmaktadır [7].

Simple CLI: Weka programı bir işletim sistemi üzerinde kullanılması halinde ve o işletim sistemi komut satırı desteklemiyor ise Weka programı kendi üzerinde bulunan Simple CLI ara yüzü ile kendi komutlarını çalıştırabilir. Java tabanlı bu sürüm birçok farklı uygulama alanında, özellikle eğitim amaçlı ve araştırma amaçlı kullanılmaktadır [6].

2.2 Veri seti açıklaması ve veri işleme

Bu çalışmada kullanılan veri seti Brunswick'in (UNB) Kanada Siber Güvenlik Enstitüsü (CIC) tarafından oluşturulmuştur. Bu saldırı tespit sistemi veri tabanı serbest bir veri tabanından ibarettir ve araştırmacılar tarafından kullanılmıştır. CICIDS2017 Veri seti 3,8 milyon veri kaydı barındırır [8]. Veritabanı kayıt işlemi 2017 yılında beş gün sürmüştür. Veritabanı bulunan trafik türleri ise Benign, DoS Hulk Saldırısı, Port Tarama, DDoS Saldırısı, Dos Golden Eye saldırısı, FTP Patator, SSH Patator, Dos Slow Loris, DoS Slow HTTP Testi, Botnet, kaba kuvvet (brute-force), XSS Web Saldırısı, Infiltration SQL Enjeksiyon ve Heart Bleed saldırıları bulunmaktadır.

2.2.1 Veri ön işleme

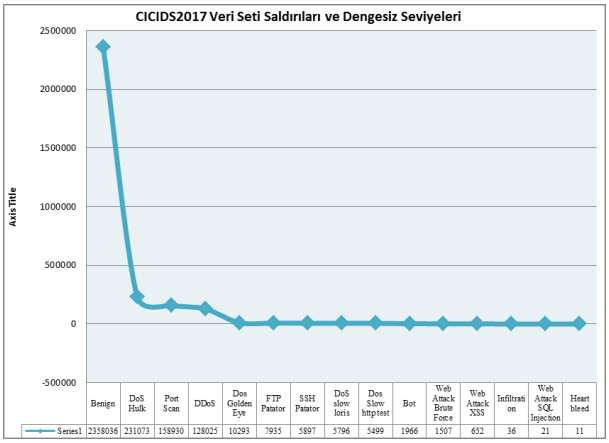
Veri ön işleme, ham verilerin hazırlanması ve bir makine öğrenimi modeline uygun hale getirilmesi sürecidir. Veri ön işleme bir makine öğrenimi modeli oluştururken ilk ve en önemli adımlarından biridir çünkü verileri, veri madenciliğinde fiilen uygulamak için uygun bir şekilde veya biçimde hazırlar. Bazen veriler, bir makine öğrenmesi projesi oluştururken ön işleme sırasında her zaman tek bir dosya halinde temiz ve formatlanmış verilere rastlamamız söz konusu olmayabilir. Bu verileri farklı dosyalar içerisinden toplayarak tek bir dosya haline getirilmesi sonucunda uygun bir format biçiminde işlenmesi sağlanır. Çünkü bu verilerle herhangi bir işlem yapılırken, verilerin temiz ve formatlı bir şekilde olması zorunludur.

Veri ön işlemeye ihtiyaç duyulmasının temel nedeni ise, gerçekte veriler genellikle gürültü ve karmaşık değerler içerir ve makine öğrenimi için uygun bir biçimde değildir. Dolaysıyla veri ön işlemenin görevi, verileri temizlemek ve bir makine öğrenim modelinin uygulanmasına hazır hale getirilmesidir sonuç olarak makine öğrenim algoritmalarının doğruluğunu ya da verimliliğini artıran ve sağlıklı bir yöntem seçilmesine yardımcı olan bir işlemdir. Bu yüzden verilerin temizlenmesini ve formatlanmasını veri ön işleme üstlenerek yapmaktadır.

Şekil 2. CICIDS2017 veri seti üzerinde veri ön işleme şeması

2.2.2 Dengesiz veriler (Imbalanced data)

Veri dengesizliği, genellikle bir veri kümesi içerisinde bulunan sınıfların eşit olmayan dağılımını yansıtır. Bu çalışmada kullandığımız CICIDS2017 veri setinde, STS'ler tarafından tespit edilen trafiğin çoğu saldırı değildir ancak tespiti yapılan atakların pek azı saldırı sınıfı olarak nitelendirilmektedir. Bu da bize veri setinde bulunan saldırı sınıfları hakkında saldırı ve saldırı olmayan sınıflar arasında 80:20 gibi bir oranın belirlenmesini ve fikir sahibi olmamızı sağlar. Dolayısıyla bu bilgiler ışığından yola çıkarak ilk olarak, verilerdeki dengesizliği görmek için aşağıda (Şekil 3) bir sınıf dağılımı yapılmış ve ardından da hafiften aşırıya doğru dengesiz verilerin aralığı Tablo 1'de açıklanmıştır.



Şekil 3. CICIDS2017 veri seti saldırıları ve dengesiz seviyeleri

Tablo 1. Verilerin dengesizlik derecesi ve % düzeyi

NO	Dengesizlik Derecesi	% Düzeyi
1	Hafif	Veri setinin %20-40'ı
2	Orta	Veri setinin %1-20'si
3	Fazla (Aşırı)	Veri setinin <%1'i

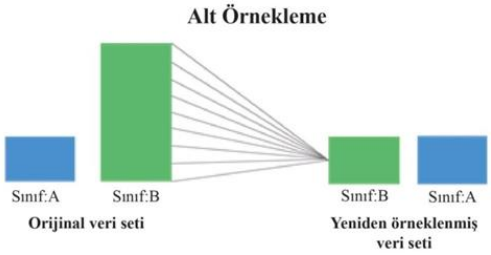
Bu derece sınıflandırmasına göre, veri kümelerde mevcut olan saldırı sınıfını değerlendireceğiz ve modelimizde kalacak sınıf yalnızca orta veya daha büyük olan verilerdir. Ancak veri setinde gözlenen herhangi bir aşırı dengesiz veri bulunması halinde, veri dengeleme sürecini engellememek için kaldırılacaktır [9].

2.2.3 *Verileri yeniden örnekleme*

Yeniden örnekleme tekniği, dengesiz veri kümeleriyle başa çıkmak için yaygın olarak benimsenen ve kullanılan bir tekniktir. Genellikle bu tekniğin uygulanması çok kolaydır ve çalıştırılması hızlıdır. Ayrıca başlangıç için ideal bir tekniktir.

Yeniden örnekleme tekniğinin çalışma mekanizması ise, azınlık sınıfına örnek ekleyerek veya çoğunluk sınıfından örnek silerek veri kümesinin daha dengeli bir hale gelmesini sağlar. Bu sayede de daha iyi makine öğrenimi modelleri oluşturur.

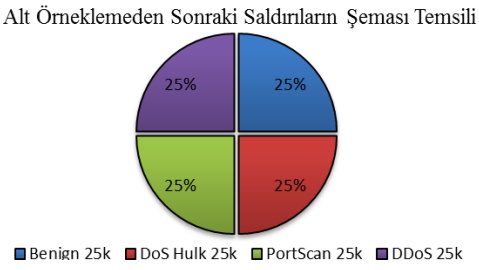
Bu değişiklikleri belirli bir veri kümesinde tanıtmanın yolu iki ana yöntemle elde edilir: Aşırı Örnekleme ve Alt Örnekleme. Bu çalışmada veri setini dengelemek için alt örnekleme yöntemi kullanılacaktır.



Şekil 4. Veri alt örnekleme şeması temsili

Şekil 3'e göre, örneklerin sayısı bir etiket türünden diğerine değişkenlik gösterdiğinden dolayı, veri kümesi örnekleri arasında bir dengeleme söz konusu olmadığı için bu da öğrenme modellerinin performansını etkileyecektir. Çalışmamız sırasında makine öğrenme yöntemlerini denemek üzere en üstten 4 etiket türü için filtrelenen CICIDS2017 verilerinin bir alt kümesi seçildi. Filtreleme sonucunda, Şekil 5'te gösterilen (Benign, Dos Hulk, Port Tarama ve DDoS) türler elde edilmiştir.

Ayrıca, örnek işleme sınırlamaları nedeniyle verilerin alt kümesi azaltılmıştır. Yani etiket türlerinin her biri 25K örneğe indirgenmiştir. Bu da bize kabul edilebilir bir hesaplama süresi veren toplam 100K satırlık bir veri örneği sonucunu elde etmemizi sağladı. Veri dağılımı Şekil 5'te gösterilmiştir.



Şekil 5. Alt örneklemeden sonraki her biri saldırının şeması temsili

2.3 *Boyut indirgeme*

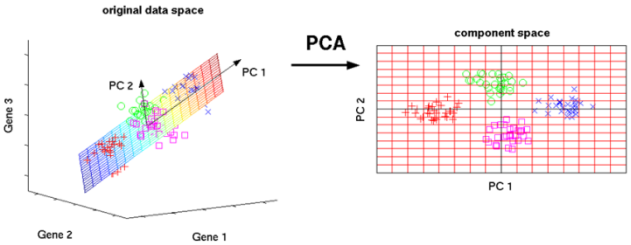
Yüksek boyutlu verilerin işlenmesi ve bu veriler içerisinde en etkili güce sahip olan önemli özellikli verileri bulmak zordur. Ayrıca büyük veriler önemli sayıda öznitelik öğrenme modelini yavaşlatmakta ve doğruluk oranını etkilemektedir. Böyle bir durumla başa çıkmanın en pratik yolu, verileri makine öğrenimi ve model üretimi için kaynak veri olarak kullanmadan önce öznitelik sayısını en aza indirmektir [10]. Şekil 6, veri boyutunu azaltırken izlenecek genel fikri temsil etmektedir. STS alanlarında daha önce araştırılmış ve test edilmiş, iyi bilinen bazı boyutsal küçülme (azaltma) yöntemleri vardır. Her boyut indirgeme metodunun, bu çalışmada inceleyeceğimiz kendi veri projeksiyonu ve görselleştirme yolu vardır.

Şekil 6. Veri azaltma şeması

2.3.1 *Temel bileşenler analizi (Principle components analysis- PCA)*

Temel Bileşen Analizi, makine öğreniminde verilerin boyutunu azaltmak için kullanılan denetimsiz bir öğrenme algoritmasıdır. PCA, boyut azaltmak için kullanılan istatistiksel bir yöntemdir. PCA, eğilimleri ve kalıpları korurken, veri özellikleri açısından yüksek boyutlu verileri basitleştiren bir boyut azaltma tekniğidir. Bu, verilerin özelliklerin açıklaması olarak işlev gören daha küçük boyutlara çevrilmesiyle elde edilir. PCA, her bir özelliğin varyansını dikkate alarak çalışır, çünkü yüksek nitelik, sınıflar arasındaki iyi ayrımı gösterir ve dolayısıyla boyutluluğu azaltır. PCA, veri noktası ile ana bileşen arasında olan mesafeyi azaltır. PCA, m<n olduğunda; n-boyutlu uzayı m-boyutlu uzaya eşler. Burada n ana bileşenleri temsil eder [11]. PCA algoritması, aşağıdaki gibi bazı matematiksel kavramlara dayanmaktadır:

- Varyans ve Kovaryans
- Öz değerler ve Öz faktörleri



Şekil 7. PCA temsili

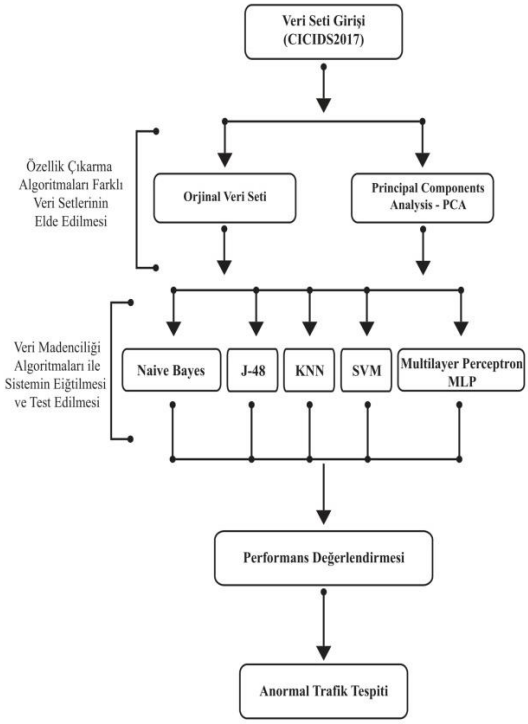
2.4 *Makine öğrenme yöntemleri*

Makine Öğrenimi (MÖ), cihazları bilinen kurallara göre programlamak yerine deneyimlerinden öğrenmelerine yardımcı olan yapay zekâ tekniklerinden biridir [12]. MÖ, insan yardımına, karmaşık matematiksel denklemlere ihtiyaç duymaz ve dinamik ağlarda çalışabilir. Son zamanlarda IoT güvenliğini sağlamak için makine öğrenim tekniklerini dikkate değer bir şekilde geliştirildi [13, 14].

Bu nedenle, cihazların davranışlarını analiz ederek çeşitli saldırıları erken bir aşamada tespit etmek için MÖ yöntemleri kullanılabilir. Buna ek olarak, bazı cihazlar için

çeşitli MÖ algoritmaları kullanılarak uygun çözümler sunulabilir.

Makine Öğrenme Yöntemleri ise, verileri ön işleme aşamasından geçirerek, boyutsallığını küçülttükten ve etiketlerle birleştirdikten sonra öğrenme ve test etme için geçen süreçtir. Kullanılacak sınıflandırıcılar, (Naive Bayes, Destek Vektör Makinesi (DVM), K en yakın komşu algoritması (KNN), J48 ve Çok Katmanlı Algılayıcı (Multilayer perceptron-MLP)) yöntemlerini kullanarak, makine öğrenmesi yöntemi sayesinde data analizi için önerilen en iyi model seçimi hedeflenmiştir. Şekil 8'de izlenecek metodoloji kapsamlı bir biçimde ele alınmıştır.



Şekil 8. Performans testleri blok diyagramı

- ***Karar ağacı***

Karar ağacı, uygulaması basit ve sonuçları herkes tarafından kolayca yorumlanabilir bir yöntemdir. Temsili yapısı, herkes tarafından anlaşılabilecek (IF-THEN) kuralları olarak dışa aktarabilir. Karar ağacı algoritmaları, hiyerarşik yapıları bazı karar düğümlerinin ortadan kaldırılmasına izin verdiği için diğer öğrenme algoritmalarına göre daha hızlı çalışır. Karar ağacından kusursuz bir model öğrenmek yerine, en basit ağaç ile modeli bulmak, test verileri üzerindeki performansın iyileştirilebilmesi için önemlidir (Alpaydin, 2010, s:185). Karar ağaçları, verileri analiz edebilir ve ağdaki kötü niyetli faaliyetleri gösteren önemli özellikleri belirleyebilir. Üstelik karar ağacı yöntemi çok sayıda izinsiz giriş tespit verilerini analiz edip güvenlik sistemlerinin performansını artırır [15].

Karar ağacı algoritmalarının çalışma mekanizmaları ise, karar ağacı algoritmaları, giriş özelliklerini ağaç düğümlerine eşler. Özellikler, bilgi kazanımı yöntemiyle önemlerine göre sıralanır. Aralarından en önemli özellikli olan kök düğüm olarak seçilir. Kök düğümü seçtikten sonra algoritma özyinelemeli olarak çalışmaya başlar. Her düğüm için, karşılık gelen özelliklerin her bir değeri için yeni dallar oluşturulur. Daha sonra eğitim örnekleri, o özellik için değerlerine göre bu dallara ayrılır. Her dal için düğüm olarak yeni bir özellik seçilir ve özyinelemeli algoritma bu düğümler üzerinde çalışmaya devam eder. Algoritma, tüm yaprak düğümleri aynı örnek etikete sahip olana kadar çalışır.

- ***Naive bayes***

Bu makine öğrenimi sınıflandırıcı modeli Bayes sınıflandırması veya daha çok Naive Bayes sınıflandırması olarak bilinir. Sonuçları tahmin etmek için olasılıklara ve koşullu olasılıklara dayanan istatistiksel bir sınıflandırmadır. Gerçek olan veriler üzerinde denendiği zaman yüksek performans göstermiştir. Naive Bayes yöntemi Bayes teoremine dayanmaktadır [16].

Naive Bayes sınıflandırıcısının çeşitli sınırlamaları olmasına rağmen, özellikler gerçek sınıfa göre koşullu olarak bağımsız ise, optimal bir sınıflandırıcıdır. Genellikle, daha karmaşık algoritmalarla karşılaştırılan ilk sınıflandırıcılardan biridir. Buna ek olarak Naive Bayes, belirli kullanıcı türleri, diğer karmaşık sınıflandırıcılara (örn., DVM) kıyasla sınıflandırma modelini daha sezgisel olarak anladıklarını ifade etmiştir. Naïve Bayes sınıflandırıcısının en büyük avantajlarından biri, çevrimiçi bir algoritma olması ve eğitiminin doğrusal zamanda tamamlanabilmesi, ayrıca NB'nin kolay anlaşılması, sınıflandırmalar için daha az veri gerektirmesi, uygulanmasının kolay olması, çok aşamalı sınıflandırma için uygulanabilir olması gibi avantajları vardır.

NB özellikler arasındaki etkileşimlere ve doğru sonuca ulaşmaya direnebilecek ön bilgilere bağlıdır. Son olarak NB genellikle ağ katmanında ve anomali tespitinde izinsiz giriş tespitini yakalamak için kullanılır [17].

- ***K- En yakın komşuluk algoritması (K-NN)***

K en yakın komşuluk algoritması, belirli bir veri örneğinin sınıfını tahmin etmek için "özellik benzerliği" fikrini kullanan en basit denetimli makine öğrenme algoritmalarından biridir. Bir örneği komşularına olan mesafesini hesaplayarak komşularına göre tanımlar. KNN algoritmasında, K parametresi modelin performansını etkiler. K değeri çok küçük ise model aşırı uyuma duyarlı olabilir. K değeri çok büyük ise örneğinin yanlış sınıflandırılmasına neden olabilir. Ayrıca bu algoritma sınıflandırmanın yanı sıra regresyon için de kullanılmaktadır [18].

KNN, genellikle Öklid mesafesini kullanan denetimli öğrenme tekniğine dayalı en basit makine öğrenimi algoritmalarından biridir. K-NN, istatistiksel olarak parametrik olmayan bir algoritmadır [19], yani temel veriler üzerinde herhangi bir varsayımda bulunmaz.

KNN'nin kullandığı Öklid uzaklığı, bilinmeyen düğümlere ortalama bir değer belirler [20]. Örneğin, verilerde bulunan herhangi bir düğüm eksik ise, bu eksik düğümü en yakın komşunun değerinden tahmin edilir. Bu değer kesin değildir, ancak olası bir eksik düğümü ortaya çıkar ise tanımlamaya yardımcı olur.

- ***Destek Vektör Makinesi (DVM)***

Destek Vektör Makinesi (DVM), çeşitli özelliklere sahip bir model sınıflandırma ve regresyon türü olup verileri analiz etmek için kullanılır. Esas olarak ikili sınıflandırma (lineer ve non lineer) tekniği olarak kullanılmıştır. Bu teknik, ilk olarak 1998 yılında bilim adamı Vladimir Vapnik tarafından önerilen ve temeli istatistiksel metotlara dayanan kuvvetli bir sınıflandırıcı tekniğidir [21].

Destek Vektör Makinesi (DVM), sınıflandırıcı, istatistiksel öğrenme teorisine dayanır ve yapısal risk minimizasyon kurallarını kullanarak bir pozitif örnek sınıfını bir negatif örnek sınıfından izole etmek için bir hiper düzlem üretir. DVM, veri noktalarını bir hiper düzlemle ayırmayı ve her veri noktasının hangi sınıfa ait olduğunu belirlemeyi amaçlar. DVM, tüm sınıfları ayırmak gerektiğinden destek vektörleri arasındaki marjı maksimize eder. DVM, yüksek sınıflandırma doğruluğu ve regresyon ve sınıflandırma görevlerini çözmedeki performansı nedeniyle popüler bir makine öğrenme tekniğidir. SVM başlangıçta ikili sınıflandırma için tasarlanmıştır. Daha sonra çok sınıflı senaryolara genişletilmiştir [22].

STS'ler açısından bir destek vektör makinesi kullanmanın ana avantajı, izinsiz girişleri gerçek zamanlı olarak tespit etme yeteneği sahiptir. Dolayısı ile DVM'ler, iyi dairesel yapıları ve boyutsallık problemin üstesinden gelme yetenekleri nedeniyle anormal izinsiz girişleri tespit etmek için popüler bir teknik haline gelmiştir. Ayrıca, küresel (global) olarak minimum düzeydeki gerçek riskleri bulmak için de faydalıdırlar. Çünkü küçük eğitim-örnek koşulları altında yüksek boyutlu alanlardaki çekirdek hilelerini iyi bir şekilde genelleştirebildikleri için, yapısal riskleri azaltmada kullanabilirler.

- ***Çok katmanlı algılayıcı- Multilayer perceptron (MLP)***

Çok katmanlı algılayıcı, yapay sinir ağı türlerinden biri olarak bilinmektedir. MLP ileri besleme olarak tanımlanır yani girdileri çıktı vektörlerine eşleme yapan bir ağ türüdür. MLP, girdi katmanının alt seviyede olduğu, çıktı katmanının en üst seviyede olduğu ve ortadaki diğer katmanların gizlendiği, birden fazla düğüm ve seviye katmanına sahip bir grafik olarak karakterize edilebilir bir tekniktir. Bu yüzden bu sinir ağı türünde bir veya daha fazla gizli katman oluşturmak mümkündür. MLP ağının bağlantısı, üst seviyedeki düğümlerin alt seviyedeki tüm düğümlere bağlantı sağlandığı yerdir. Çok Katmanlı Algılayıcı genellikle denetimli öğrenme zorlukları için kullanılır. MLP, doğrusal olmayan verileri çözme sınırlamasını düzelten tekil bir katman öncüsünün uzantısıdır. Girdi katmanında iletilen veriler daha sonra gizli katmanda sabit sayıda yineleme ve katman için ileri geri akar sonuçlar çıktı katmanından elde edilir. Verileri eğitirken uygulanan mimariyi çalıştıran modeli temsil eder [23].

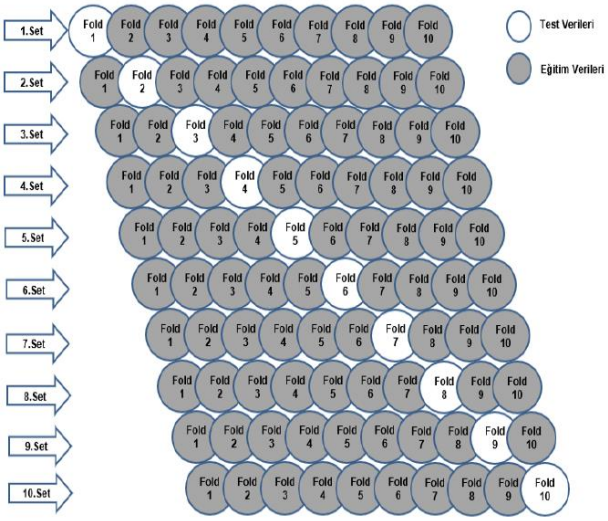
Sonuç olarak MLP eğitim aşamasında çok uzun zaman almasına rağmen verileri test etmek için çok az zaman harcayan bir sinir ağı sınıflandırıcıdır. Sınıflandırma, regresyon ve tahmin gibi çeşitli görevler için kullanılır. MLP üç katmana dayalı olarak çalışır: sınıf sayısı (çıktı), veri kümesi (giriş) ve gizli katmanlar.

3 Bulgular ve tartışma

Brunswick'in (UNB) Kanada Siber Güvenlik Enstitüsü (CIC) tarafından oluşturulmuş CICIDS2017 veri seti üzerinde uygulanan her bir algoritmanın ağırlıklı ortalaması Tablo 2'de sunulmuştur. Bunun yanı sıra verilmiş olan algoritmaları uygularken sınama seçeneği olarak Çapraz Doğrulama kullanılıp kat sayısı ise 10 olarak belirlenmiştir. İlk adımın birinci kısmı test verisi geri kalan 9'u ise eğitim verisidir. Ardından ikinci adımda ise verinin 2. kısmı test verisi ve geri kalan 9 kısmı ise eğitim verisi olarak analiz edilmiştir. Bütün adımlar (10 adım) aynı şekilde devam etmektedir. Çıkan analiz sonucunda %94.13 Naive Bayes, %96.58 Destek vektör makinesi (DVM), %99.87 K en yakın komşu algoritması (KNN) ve bu yöntemde K sayısı 1 olarak en iyi sonuç ortaya çıkmıştır, ardından %99.92 doğruluk sonucunu J48 vermiştir, son olarak %98.50 doğruluk sonucunu MLP algoritması kayıt vermiştir.

Tablo 2. Çapraz Doğrulama Kat=10 kullanarak alınan algoritma sonuçları

	Naive Bayes	SVM	KNN	J48	MLP
Doğruluk	94.14	96.58	99.88	99.92	98.50
Hassasiyet	0.95	0.96	0.99	0.99	0.98
Duyarlılık	0.94	0.96	0.99	0.99	0.98
F-Skor	0.94	0.96	0.99	0.99	0.98
ROC Alan	0.98	0.98	0.99	1.00	0.99



Şekil 9. 10 katlı çapraz doğrulama yapısı

Tablo 3. Çapraz Doğrulama Kat=10 ve boyutsal indirgeme yöntemi PCA kullanarak alınan algoritma sonuçları

	PCA Naive Bayes	PCA SVM	PCA KNN K=1	PCA J48	PCA MLP
Doğruluk%	82.42	95.07	99.88	99.76	98.23
Hassasiyet	0.83	0.95	0.99	0.99	0.98
Duyarlılık	0.82	0.95	0.99	0.99	0.98
F-Skor	0.81	0.95	0.99	0.99	0.98
ROC Alan	0.94	0.97	0.99	0.99	0.99

Tablo 3'te CICIDS2017 veri seti üzerinde boyutsal indirgeme algoritması olan PCA uygulanıp veri setinin boyutu küçültülmüştür ve test seçeneği olarak Çapraz Doğrulama kat sayısı 10 olup ardından belirlediğimiz algoritmalar uygulanarak her bir algoritma için doğruluk elde edilmiştir. Bunlar %82.42 Naive Bayes, %95.07 DVM, %99.88 k-NN, ardından %99.76 J48 ve son olarak %98.23 MLP olarak sonuçlar elde edilmiştir, bunun yanı sıra (Hassasiyet, Duyarlılık, F-Skor, ROC Alan) değerlerinin ağırlıklı ortalaması alınarak Tablo 3'te gösterilmiştir.

Veri seti üzerinde uygulamış olunan boyutsal indirgeme algoritması olan PCA ardından test seçeneği olarak Çapraz Doğrulama kat sayısı 10 olarak algoritmalar arasında en iyi performansı gösteren algoritma K-NN ve bu yöntemde k sayısı 1 olarak en iyi sonuç elde edilerek doğruluk değeri %99.88 olarak bu aşamada en yüksek performansı göstermiştir.

Tablo 4. Bölünme yüzdesi 80% kullanarak alınan algoritma sonuçları

	Naive Bayes	SVM	KNN	J48	MLP
Doğruluk%	93.94	96.57	99.87	99.92	98.74
Hassasiyet	0.83	0.955	0.99	0.99	0.98
Duyarlılık	0.94	0.968	0.99	0.99	0.98
F-Skor	0.81	0.951	0.99	0.99	0.98
ROC Alan	0.93	0.96	0.99	0.99	0.98

Tablo 4'teki sonuçlar makine öğrenimine sunulan CICIDS2017 100 bin satır, 70 nitelik ve 4 sınıftan oluşan veri seti üzerinde uygulanan algoritmalar neticesinde diğer tablolarda uygulandığı gibi ağırlıklı ortalaması alınmıştır. Bu aşamada kullanılan test seçeneği olarak bölünme yüzdesi % 80 olarak belirlenmiştir. Bu değerin %80'i eğitim seti ve %20'de test seti olarak her bir algoritmayı verilen veri seti üzerinde uygulayarak ayrı ayrı analiz edilecektir. CICIDS2017 veri setinin %20'i üzerinde yaklaşık 20K satırlık bir veri seti uygulanmış olan algoritmalar sonucunda %93.94 Naive Bayes, %96.57 DVM, %99.87 k-NN algoritması ve bu yöntemde k sayısı 1 olarak en iyi performansı göstermiş, %99.92 J48 ve son olarak %98.74 MLP olarakta görülmüştür.

Sonuç olarak uygulamış olunan bölünme yüzdesi %80 değeriyle algoritmalar arasında en iyi performansı gösteren algoritma J48 olup ve doğruluk değeri %99.92 olarak bu aşamada en yüksek performansı göstermiştir.

Tablo 5. Bölünme yüzdesi 80% ve boyutsal indirgeme yöntemi PCA kullanarak alınan algoritma sonuçları

	PCA Naive Bayes	PCA SVM	PCA KNN k=2	PCA J48	PCA MLP
Doğruluk%	82.45	95.07	99.86	99.69	98.94
Hassasiyet	0.83	0.95	0.99	0.99	0.99
Duyarlılık	0.82	0.95	0.99	0.99	0.98
F-Skor	0.81	0.95	0.99	0.99	0.98
ROC Alan	0.94	0.92	0.99	0.99	0.99

Tablo 5'te verilmiş olan sonuçlar veri seti üzerinde uygulanan boyutsal indirgeme algoritması olan PCA uygulanıp veri setinin boyutu küçültülmüştür, ardından algoritmaları uygulama sırasında test seçeneği olarak bölünme yüzde değeri %80 olarak seçilmiştir. Belirlediğimiz algoritmalar ve MLP uygulanarak her bir algoritma için doğruluk elde edilmiştir. Bunlar %82.45 Naive Bayes, %95.07 destek vektör makinesi, %99.86 k-NN, ardından %99.69 J48 ve son olarakta %98.94 Multilayer perceptron olarak sonuçlar elde edilmiştir, bunun yanı sıra (Hassasiyet, Duyarlılık, F-Skor, ROC Alan) değerlerinin ağırlıklı ortalaması alınarak Tablo 4'te gösterilmiştir.

Uygulamış olunan veri seti üzerinde boyutsal indirgeme algoritması olan PCA ardından test seçeneği olarak bölünme yüzdesi %80 olup algoritmalar arasında en iyi performansı gösteren algoritma k-NN ve bu yöntemde k sayısı 2 olarak en iyi sonuç elde edilerek doğruluk %99.86 olarak bu aşamada en yüksek performansı göstermiştir.

3.1 En iyi performans gösteren algoritma

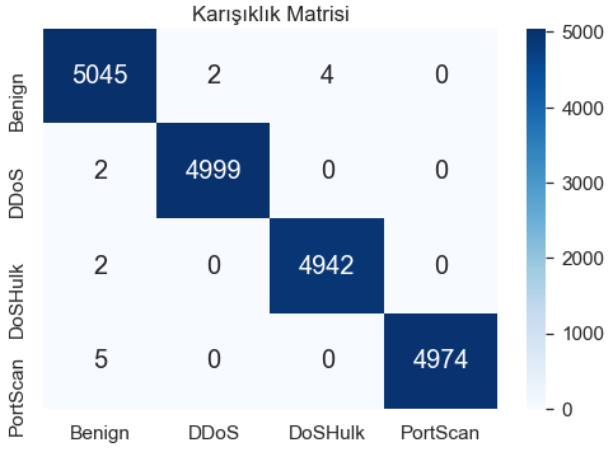
Bu çalışmada CICIDS2017 veri seti üzerinde 5 ayrı algoritma uygulanarak sonuçlar elde ettik, bu sonuçlar içerisinde kullandığımız 2 ayrı veri seti ve 2 ayrı test seçeneği bulunmakta, bu veri setinin 1. Veri Ön işlemeden geçen orijinal veri seti 2. ise aynı şekilde veri önişlemden geçerek boyutsal indirgeme işlemine yani PCA tabii tutularak yeni bir veri seti elde edilmiştir.

Diğer yandan 2 ayrı test seçeneği kullanılmakta, bunların 1. Çapraz Doğrulama kat sayısı 10 olup diğeri ise Bölünme yüzdesi 80 olarak algoritmalar ile uygulanmıştır.

Uygulanan Çapraz Doğrulama kat=10 test seçeneğinde veri setinin tamamı kullanarak, diğeri ise Bölünme yüzdesi %80 olarak veri setinin 20K'sı kullanılıp yukarıda bulunan tablolardaki sonuçlar elde edilmiştir.

Elde edilen sonuçları karşılaştırarak en iyi performansı gösteren algoritma Bölünme yüzdesi 80 test seçeneğini ve orijinal veri seti üzerinde uygulanan J48 olarak kayıt edilmiştir. Çıkan sonuca göre doğruluk değeri %99.92 olarak (Tablo 4) verilen sonuçlar en iyi performans gösteren algoritma olarak belirlenmiştir.

Tablo 6. J48 algoritması seçerek bölünme yüzdesi 80% uygulanan karışıklık matrisi sonucu



Tablo 6'da en iyi performans gösteren J48 Matrisi, veri seti üzerinde uygulanan bölünme yüzdesi 80 olup sonucu ortaya çıkmıştır, bu matrisin aa kısmı BENIGN, bb kısmı DdoS, cc kısmı DoSHulk ve dd kısmıda PortScan değerleri uygulanmış olan algoritma sunucunda belirlenmiştir.

Ancak matrikste olan DDos (ab) kısım, DoSHulk olan (ac) kısmı ve PortScan olan (ad) kısımlar algoritma tarafından BENIGN olarak algılanmıştır, aynı şekilde BENIGN olan (ba) kısımıda Ddos olarak algılanmıştır ve BENIGN olan (ca) kısmı, DoSHulk olarak algılanmıştır, ve bu kısımlarda bulunan değerler hata payı kabul edilip % 0.0751 oluşturmaktadır.

Tablo 7. Çalışma sonuçlarının benzer çalışmalar ile karşılaştırılması

Makale	PCA	En iyi algoritma	Doğruluk
Yulianto vd. [1]	Evet	Adaboost	81,83
Engelen vd. [2]	Hayır	Rasgele Orman	99,00
Priyanka ve Kumar [3]	Evet	Rasgele Orman	99,90
Rosay vd. [4]	Hayır	Rasgele Orman	99,90
Çalışmamız	Evet	J48	99,92

Sonuç olarak bu çalışmada keşfedilen en iyi performanslı makine öğrenme algoritması J48 saldırı tespit sistemlerinde üzerinde kullanılması uygun görülerek STS'ler tarafından trafik analizi ve gelen paketlerin saldırı olup olmadığını tespit ederek bulunduğu sistemi kötücül trafikten korumakla STS'lere önerilmektedir.

4 Sonuçlar

Saldırı tespit sistemleri (STS) çalışmalarında doğru bir model kullanılması çok önemlidir. Aksi halde, ağ trafiğinde dolaşan paketlerin kötü sınıflandırılması nedeniyle kaynaklanan sorunlarının olma olasılığı yüksektir. Saldırı tespit sistemleri için bir model tasarım sureci esnasında kullanılan veri seti oldukça önemlidir. Bu çalışmada kullanılan CICIDS2017 herkese açık ve araştırmacılar tarafından sık sık kullanılan bir veri setidir.

Bilinmekte olan ağ iletişim protokolleri (IPv4, IPv6, TCP, UDP, FTP, ARP, ICMP, DNS, SNMP, HTTP, SMTP ve POP), ağ güvenlik tehditleri (paket koklama ve aldatma), Hizmet Engelleme Saldırıları (SYN, ACK/FIN/PUSH, UDP, DNS, MAC ve HTTP taşmaları) ve buna ek olarak Ağ Keşfi yöntemleri (IP tarama, port tarama, işletim sistemi tarama) saldırganlar tarafından sık sık kullanılmaktadır.

Saldırganlara karsı önlem alabilmek için genellikle makine öğrenmesi tekniklerini kullanarak ağ trafik analizinde saldırı tespit sistemlerine yardımcı olabilmekledirler.

Bu araştırmada Kullandığımız CICIDS2017 veri seti Brunswick'in (UNB) Kanada Siber Güvenlik Enstitüsü (CIC) tarafından oluşturulmuştur. Bu veri seti üzerinde veri temizleme işlemi yapılarak gürültülü verilerden kurtulup boyutsal indirgeme algoritması olan temel bileşenler analizi (PCA) uygulanarak yeni bir veri seti elde edilmiştir.

Bu işlemler yapıldıktan sonra elde edilen her iki veri seti üzerinde belirlemiş olduğumuz makine öğrenme yöntemleri (Naive Bayes, Destek vektör makinesi (DVM), K en yakın komşu algoritması (KNN), J48 ve çok katmanlı algılayıcı (multilayer perceptron-MLP)) uygulanmıştır. Bütün bu yöntemler veri madenciliğinde kullanılan WEKA programı üzerinde gerçekleştirilmiştir.

Tablo 8. Makina öğrenme sınıflandırıcılarının başarı değerleri

Makine Öğrenme Algoritmaları	Çapraz Doğrulama kat= 10		Persentage Split % = 80	
	Normal	PCA	Normal	PCA
Naive Bayes	94.13	82.42	93.94	82.45
DVM	96.58	95.07	96.57	95.07
KNN	99.87	99.88	99.87	99.86
J48	99.92	99.76	99.92	99.69
MLP	98.50	98.23	98.74	98.94

Bu çalışma sonucunda makine öğrenimi teknolojisi kullanılarak ve daha önceden saldırı tespit sistemleri tarafından tespit edilen izinsiz girişler ve anormal trafiklerin bulunduğu bir veri setini analiz edip kullanılan modellerin başarı oranı değerlendirmesi gerekirse en yüksek performans gösteren algoritma J48 olduğu ve %99.92 bir doğruluk oranı kayıt etmiştir. Diğer algoritmaların doğruluk oranları Tablo 8'de verilmiştir. Sonuç olarak en iyi performansı gösteren özellik seçimi kullanıldığında, kötü amaçlı yazılım algılama sonuçlarının daha iyi hale getirilmesi sonucuna ulaşılmıştır.

Çıkar çatışması

Yazarlar çıkar çatışması olmadığını beyan etmektedir.

Benzerlik oranı (iThenticate): %10

Kaynaklar

[1] A. Yulianto, P. Sukarno, N.A. Suwastika, Improving adaboost-based intrusion detection system (IDS) performance on CICIDS 2017 dataset, Journal of Physics: Conference Series, 1192(1), 012018, 2019. https://doi.org/10.1088/1742-6596/ 1192/ 1/ 012018.

[2] G. Engelen, V. Rimmer, W. Joosen, Troubleshooting an intrusion detection dataset: the CICIDS2017 case study. IEEE Security and Privacy Workshops (SPW), pp. 7-12, 2021. https://doi.org/10.1109/SPW53761. 2021.00009.

[3] V. Priyanka and T. G. Kumar, Performance assessment of IDS based on CICIDS-2017 dataset. Information and Communication Technology for Competitive Strategies (ICTCS 2020), pp. 611-621, 2022. https://doi.org/10.1007/978-981-16-0739-4_58.

[4] A. Rosay, E. Cheval, F. Carlier and P. Leroux, Network intrusion detection: a comprehensive analysis of CIC-IDS2017. 8th International Conference on Information Systems Security and Privacy, pp. 25-36, 2022. https://doi.org/10.5220/0010774000003120.

[5] R.R. Boukaert, E. Frank, M. Hall, R. Kirby, P. Reutemann, A. Seewald, D. Scuse, Weka manual for version 3-7-3, The University of Waikato, 327, 2010.

[6] S. Singhal and M. Jena, A study on WEKA tool for data preprocessing, Classification and Clustering.

International Journal of Innovative Technology and Exploring Engineering (IJItee), 2(6), 250-253, 2013.

[7] C. Gürmen, Saldırı tespit sistemleri için makine öğrenme yöntemlerinin performans karşılaştırması. Yüksek Lisans Tezi, Harran Üniversitesi, Türkiye, 2020.

[8] C. Kruegel and G. Vigna, Anomaly detection of web-based attacks. Proceedings of the 10th ACM Conference on Computer and Communications Security, 251-261, Washington D.C., USA, 2003. https://doi.org/10.1145/948109. 948144.

[9] A.A. Abdulrahman and M.K. Ibrahem, Toward constructing a balanced intrusion detection dataset based on CICIDS2017, Samarra Journal of Pure and Applied Science, 2(3), 132-142, 2020.

[10] A.N. Bhagoji, D. Culina, C. Sitawarin, P. Mittal, Enhancing robustness of machine learning systems via data transformations. 52nd Annual Conference on Information Sciences and Systems (CISS), pp. 1-5, 2018. https://doi.org/10.1109/CISS.2018.8362326.

[11] K.K. Vasan and B. Surendiran, Dimensionality reduction using principal component analysis for network intrusion detection, Perspectives in Science, 8, 510-512, 2016. https://doi.org/10.1016/j.pisc.2016.05.010.

[12] M.I. Jordan and T.M. Mitchell, Machine learning: trends, perspective, and prospects, Science, 349(6245), 255-260 2015. https://doi.org/10.1126/science.aaa8415.

[13] M.A. Alsheikh, S. Lin, D. Niyato, H.P. Tan, Machine learning in wireless sensor networks: algorithms, strategies, and applications, IEEE Communications Surveys & Tutorials, 16(4), 1996-2018, 2014. https://doi.org/10.1109/COMST.2014.2320099.

[14] I. Butun, S. D. Morgera, R. Sankar, A survey of intrusion detection systems in wireless sensor networks, IEEE Communications Surveys & Tutorials, 16(1), 266-282, 2013. https://doi.org/10.1109/SURV.2013.050113.00191

[15] K. Rai, M.S. Devi, A. Guleria, Decision tree-based algorithm for intrusion detection, International Journal of Advanced Networking and Applications, 7(4), 2828-2834, 2016.

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

[17] S.M. Tahsien, H. Karimipour, P. Spachos, Machine learning based solutions for security of internet of things (Iot): a survey. Journal of Network and Computer Applications, 161, 102630, 2020. https://doi.org/10.1016/j.jnca.2020.102630.

[18] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, C. Wang, Machine learning and deep learning methods for cybersecurity. IEEE Access, 6, 35365-35381, 2018. https://doi.org/10.1109/ACCESS.2018.2836950.

[19] F. Chen, P. Deng, J. Wan, Dd Zhang, A.V. Vasilakos, X. Rong, Data mining for the internet of things: literature review and challenges. International Journal of Distributed Sensor Networks, 431047, 2015. https://doi.org/10.1155/2015/431047.

[20] Z. Deng, X. Zhu, D. Cheng, M. Zong, S. Zhang, Efficient knn classification algorithm for big data. Neurocomputing, 195, 143-148, 2016. https://doi.org/10.1016/j.neucom.2015.08.112

[21] A. Aldallal, F. Alisa, Effective intrusion detection dystem to secure data in cloud using machine learning. Symmetry, 13(12), 1-26, 2021. https://doi.org/10.3390/sym13122306.

[22] M. Al-Qatf, Y. Lasheng, M. Al-Habib, K. Al-Sabahi, Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. IEEE Access. 6, 52843-52856, 2018. https://doi.org/10.1109/ACCESS.2018.2869577.

[23] M. Alkasassbeh, M. Almseidin, Machine learning methods for network intrusion detection. arXiv preprint, arXiv:1809.02610, 2018. https://doi.org/10.48550/arXiv.1809. 02610.