ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENİMİ YÖNTEMLERİ İLE AĞ SALDIRILARININ TESPİTİ

Fırat ÇEÇEN

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2026

ÖZET

Yüksek Lisans Tezi

MAKİNE ÖĞRENİMİ YÖNTEMLERİ İLE AĞ SALDIRILARININ TESPİTİ

Fırat ÇEÇEN

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Recep ERYİĞİT

Günümüzde siber tehditlerin çeşitlenmesi ve karmaşık hâle gelmesi, ağ altyapılarının güvenliğinin sağlanmasını kritik bir gereksinim hâline getirmiştir. Bu bağlamda, ağ trafiği üzerinde meydana gelen zararlı faaliyetlerin erken aşamada belirlenebilmesi için Ağ Saldırı Tespit Sistemleri (NIDS) önemli bir rol üstlenmektedir. Bu tez çalışmasında, makine öğrenmesi tabanlı yöntemler kullanılarak ağ saldırılarının etkin bir biçimde tespit edilmesi amaçlanmıştır.Çalışma kapsamında, gerçekçi trafik yapısı ve güncel saldırı senaryoları içermesi nedeniyle UNSW-NB15 veri kümesi tercih edilmiş ve problem çok sınıflı saldırı tespiti çerçevesinde ele alınmıştır. Geleneksel yaklaşımlardan farklı olarak, yalnızca normal ve anormal trafik ayrımı yapılmamış; veri kümesinde yer alan çeşitli saldırı türleri ayrı sınıflar olarak değerlendirilmiştir. Bu yaklaşım sayesinde, ağ trafiğinin davranışsal özellikleri daha ayrıntılı şekilde analiz edilebilmiştir.

Deneysel analizlerde Rastgele Orman, Destek Vektör Makineleri, k-En Yakın Komşu, Karar Ağacı ve Naive Bayes olmak üzere beş farklı denetimli makine öğrenmesi algoritması kullanılmıştır. Modeller, beş katlı çapraz doğrulama yöntemi ile değerlendirilmiş; doğruluk, kesinlik, duyarlılık ve F1 skoru gibi performans ölçütleri esas alınarak karşılaştırılmıştır. Elde edilen bulgular, Rastgele Orman ve k-En Yakın Komşu algoritmalarının genel sınıflandırma başarımı açısından öne çıktığını, diğer yöntemlerin ise belirli saldırı türlerinde rekabetçi sonuçlar ürettiğini ortaya koymuştur. Ayrıca, karışıklık matrisi analizleri aracılığıyla sınıflandırma hatalarının dağılımı ayrıntılı olarak incelenmiştir.

Bu tez çalışmasının temel katkısı, UNSW-NB15 veri kümesi üzerinde farklı makine öğrenmesi algoritmalarının çok sınıflı saldırı tespit problemindeki performanslarının sistematik ve karşılaştırmalı biçimde değerlendirilmesidir. Elde edilen sonuçlar, tek bir algoritmanın tüm saldırı türleri için ideal bir çözüm sunmadığını ve model başarısının veri dağılımı ile öznitelik yapısına bağlı olarak değiştiğini göstermektedir. Bu doğrultuda, gelecekte hibrit ve çok aşamalı saldırı tespit yaklaşımlarının ağ güvenliği alanında daha etkili çözümler sunabileceği değerlendirilmektedir.

Ocak 2026, 66 sayfa

Anahtar Kelimeler: Ağ Saldırıları, Makine Öğrenmesi, Saldırı Tespit Sistemleri, Rastgele Orman, Destek Vektör Makinesi, En Yakın Komşu Algoritması

ABSTRACT

Master's Thesis

DETECTION OF NETWORK ATTACKS USING MACHINE LEARNING METHODS

Fırat ÇEÇEN

Ankara University
Graduate School of Natural and Applied Science
Department of Computer Engineering

Supervisor: Prof. Dr. Recep ERYİĞİT

The rapid growth and increasing complexity of cyber threats have made the protection of network infrastructures and data confidentiality a critical concern. In this regard, Network Intrusion Detection Systems (NIDS) play an essential role in identifying malicious activities within network traffic at an early stage. This thesis aims to investigate the effectiveness of machine learning–based approaches for accurately detecting network attacks.In this study, the UNSW-NB15 dataset was selected due to its realistic traffic structure and inclusion of contemporary attack scenarios, and the intrusion detection problem was formulated as a multi-class classification task. Rather than limiting the analysis to a binary distinction between normal and malicious traffic, individual attack categories were treated as separate classes, allowing for a more comprehensive examination of network traffic behavior.

Experimental analyses were performed using five supervised machine learning algorithms: Random Forest, Support Vector Machines, k-Nearest Neighbors, Decision Tree, and Naive Bayes. The models were evaluated through five-fold cross-validation and compared using standard performance metrics, including accuracy, precision, recall, and F1-score. The experimental results demonstrate that Random Forest and k-Nearest Neighbors exhibit higher overall classification performance, while the remaining algorithms achieve competitive results for specific attack categories. Furthermore, confusion matrix analyses were conducted to provide a detailed assessment of classification errors.

The primary contribution of this thesis lies in the systematic and comparative evaluation of multiple machine learning algorithms for multi-class intrusion detection using the UNSW-NB15 dataset. The findings indicate that no single algorithm can be considered optimal for all attack types, as model performance is influenced by both data distribution and feature characteristics. Accordingly, the results suggest that hybrid and multi-stage intrusion detection strategies may offer more effective solutions for future network security research.

January 2026, 66 pages
Keywords: Network Attacks, Machine Learning, Intrusion Detection Systems, Random Forest, SVM, k-Nearest Neighbors

TEŞEKKÜR

Bu tezin hazırlanması sürecinde bilgi birikimi ve rehberliğiyle yolumu aydınlatan, her aşamada destek olup motivasyonumu artıran kıymetli danışmanım Prof. Dr. Recep ERYİĞİT'e gönülden teşekkür ederim.

Akademik sürecim boyunca yanımda olduklarını her zaman hissettiren sevgili aileme, bu süreçte moral ve destekleriyle katkı sağlayan yakın dostlarıma ve değerli çalışma arkadaşlarıma minnettarım. Bu yolculuğu daha anlamlı kılan tüm sevdiklerime içtenlikle teşekkür ederim.

Fırat ÇEÇEN
Ankara, Ocak 2026

İÇİNDEKİLER

KISALTMALAR DİZİNİ

AI	Artificial Intelligence (Yapay Zeka)
AUC	Area Under Curve (Eğri Altında Kalan Alan)
CSV	Comma-Separated Values (Virgülle Ayrılmış Değerler)
DDoS	Distributed Denial of Service (Dağıtılmış Hizmet Engelleme Saldırısı)
DoS	Denial of Service (Hizmet Engelleme Saldırısı)
DT	Karar Ağacı (Karar Ağacı)
FN	False Negative (Yanlış Negatif)
FP	False Positive (Yanlış Pozitif)
FPR	False Positive Rate (Yanlış Pozitif Oranı)
HTTP	HyperText Transfer Protocol (Hipermetin Transfer Protokolü)
IDS	Intrusion Detection System (Saldırı Tespit Sistemi)
IP	Internet Protocol (İnternet Protokolü)
IPS	Intrusion Prevention System (Saldırı Önleme Sistemi)
KNN	K-Nearest Neighbors (En Yakın Komşu Algoritması)
ML	Machine Learning (Makine Öğrenmesi)
NB	Naive Bayes (Naive Bayes Sınıflandırıcısı)
RF	Rastgele Orman (Rastgele Orman)
ROC	Receiver Operating Characteristic Curve (Alıcı İşletim Özelliği Eğrisi)
SVM	Support Vector Machine (Destek Vektör Makinesii,)
TCP	Transmission Control Protocol (İletim Kontrol Protokolü)
TN	True Negative (Doğru Negatif)
TP	True Positive (Doğru Pozitif)
TPR	True Positive Rate (Doğru Pozitif Oranı)
UDP	User Datagram Protocol (Kullanıcı Veri Birimi Protokolü)
UNSW-NB15	University of New South Wales - Network-Based 2015 Dataset (New South Wales Üniversitesi - 2015 Ağ Tabanlı Veri Kümesi)

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Dijitalleşmenin hızlı ilerleyişi, bireylerin ve kurumların günlük yaşam aktivitelerini büyük ölçüde internet ve bilgi teknolojileri platformlarına taşımalarına neden olmuştur. Dünya genelinde 2024 yılı itibarıyla internet kullanıcılarının sayısı 5,5 milyarı aşmıştır (International Telecommunication Union [ITU], 2024). Artan internet kullanımı, bilgiye erişim, eğitim, sağlık ve ticaret gibi birçok alanda büyük kolaylıklar sağlamış, ancak bu gelişmeler siber güvenlik tehditlerini de beraberinde getirmiştir. Özellikle kötü amaçlı yazılımlar, fidye yazılımları (ransomware), kimlik avı (phishing) saldırıları ve gelişmiş kalıcı tehditler (Advanced Persistent Threats - APTs) gibi tehdit türleri, bireyleri ve kuruluşları ciddi zararlarla karşı karşıya bırakmaktadır (ENISA Threat Landscape, 2023). Bu tehditlerin artan karmaşıklığı, ağ güvenliği sistemlerinin daha sofistike yöntemlere yönelmesini zorunlu kılmıştır.

Saldırı tespit sistemleri (Intrusion Detection Systems - IDS), ağ güvenliğinde temel bir savunma hattı oluşturmakla birlikte, geleneksel imza tabanlı IDS'ler yalnızca daha önceden tanımlanmış tehditleri algılayabilmektedir (Garcia-Teodoro et al., 2009). Sıfırıncı gün saldırıları gibi yeni ve bilinmeyen tehdit türleri, imza tabanlı sistemler tarafından çoğu zaman tespit edilememekte ve bu durum kurumları ciddi risklerle karşı karşıya bırakmaktadır (Sommer & Paxson, 2010). Ayrıca, saldırganlar geleneksel yöntemleri aşmak için saldırı tekniklerini sürekli evrimleştirmekte, böylece klasik IDS çözümlerinin etkinliğini azaltmaktadır. Bu bağlamda, değişen saldırı desenlerine hızlı şekilde uyum sağlayabilen yeni nesil tespit mekanizmalarına olan ihtiyaç giderek artmıştır.

Makine öğrenmesi (Machine Learning - ML), ağ saldırı tespitinde umut vaat eden bir yaklaşım olarak öne çıkmaktadır. ML yöntemleri, ağ trafiğindeki normal ve anormal davranışları öğrenerek sıfırıncı gün tehditleri de dâhil olmak üzere çeşitli saldırı türlerini başarıyla tespit edebilmektedir (Shone et al., 2018). Denetimli öğrenme yöntemleri (supervised learning), normal ağ trafiği ile anormal etkinlikler arasındaki sınırları belirlemede önemli avantajlar sağlamaktadır. Rastgele Orman, (Destek Vektör Makinesi,), K-Nearest Neighbors (En Yakın Komşu Algoritması) ve Naive Bayes gibi

makine öğrenmesi algoritmaları, farklı veri setlerinde ağ saldırılarını yüksek doğruluk oranlarıyla tespit etmiştir (Buczak & Guven, 2016). Bu algoritmaların yetenekleri, sistemlerin sürekli güncellenmesine gerek kalmadan yeni saldırı desenlerini tanımasına imkân sağlamaktadır.

Bu tez çalışmasında kullanılan UNSW-NB15 veri seti, modern ağ trafiğini ve çeşitli saldırı tiplerini kapsayan bir veri kaynağıdır. Moustafa ve Slay (2015) tarafından oluşturulan veri seti, normal trafiğin yanı sıra DoS, Exploits, Fuzzers, Analysis ve Reconnaissance gibi farklı saldırı kategorilerini içermektedir. Toplamda 49 öznitelik barındıran bu veri seti, makine öğrenmesi algoritmalarının gerçek dünya koşullarında test edilmesine imkân tanımaktadır. Veri seti, hem normal hem de saldırgan trafiği dengeli bir şekilde içermesi sayesinde sınıflandırma algoritmalarının başarımının adil şekilde değerlendirilebilmesini sağlamaktadır.

Dijital ortamların yaygınlaşması, siber tehditlerin doğasını da değiştirmiştir. Geleneksel kötü amaçlı yazılımların yerini, hedef odaklı saldırılar, sosyal mühendislik yöntemleri ve otomatikleştirilmiş saldırı araçları almıştır. 2024 yılı itibarıyla dünya genelinde gerçekleşen siber saldırıların %43'ünün küçük ve orta ölçekli işletmelere (KOBİ'lere) yönelik olduğu bildirilmiştir (Verizon Data Breach Investigations Report, 2024). Bunun yanı sıra, fidye yazılımlarının ortalama fidye taleplerinin %37 oranında arttığı ve kurumların veri güvenliği tehditlerine karşı daha ciddi stratejiler geliştirme zorunluluğunun ortaya çıktığı vurgulanmaktadır. Bu istatistikler, yalnızca büyük ölçekli kuruluşların değil, tüm kurum ve bireylerin siber tehditler karşısında savunmasız olduğunu göstermektedir.

Siber saldırıların sürekli evrilmesi, ağ güvenliği alanında sürekli yenilik gerektirmektedir. Özellikle ağlar üzerinde gerçek zamanlı analiz yapabilen, yeni tehditleri öğrenip tanıyabilen sistemlerin geliştirilmesi, güvenliğin sağlanması açısından kritik bir öneme sahiptir. Bu noktada, makine öğrenmesi tabanlı saldırı tespit sistemleri, geleneksel sistemlere kıyasla çok daha esnek ve uyarlanabilir çözümler sunmaktadır (Xia et al., 2020). Denetimli öğrenmenin yanı sıra, denetimsiz öğrenme (unsupervised learning) ve yarı denetimli öğrenme (semi-supervised learning) gibi yöntemler de ağ güvenliğinde

kullanılmaya başlanmıştır. Böylece, bilinen saldırıların yanı sıra bilinmeyen anomalilerin de tespit edilmesi mümkün hâle gelmiştir.

Literatürde gerçekleştirilen birçok çalışmada, ağ trafiğinden elde edilen çeşitli özniteliklerin kullanılmasıyla saldırı tespiti gerçekleştirilmiştir. Özellikle akış (flow) bazlı özellikler, hem veri boyutunu azaltmak hem de işlem süresini optimize etmek açısından önemli avantajlar sağlamaktadır. Flow tabanlı analizlerde genellikle paket sayısı, veri boyutu, bağlantı süresi gibi öznitelikler temel alınmakta ve makine öğrenmesi modelleri bu veriler üzerinde eğitilmektedir (Ring et al., 2019). Ayrıca, yeni nesil ağ saldırılarının tespitinde zaman serileri analizi, grafik tabanlı yaklaşımlar ve derin öğrenme tabanlı sistemler de araştırılmaya başlanmıştır.

Makine öğrenmesi tekniklerinin saldırı tespit sistemlerinde kullanılması çeşitli avantajlar sağlamaktadır. İlk olarak, bu teknikler büyük miktarda veriden örüntüler çıkararak bilinmeyen tehditleri belirleyebilir. İkincisi, sistemler zaman içinde yeni saldırı tiplerine karşı eğitilerek güncellenebilir ve böylece saldırganların değişen taktiklerine karşı adaptif bir savunma hattı oluşturulabilir. Üçüncü olarak ise, manuel müdahale ihtiyacını azaltarak, saldırı tespiti sürecinde insan hatasını minimize edebilir (Abaid et al., 2022). Ancak, makine öğrenmesi tabanlı yaklaşımların da zorlukları bulunmaktadır. Özellikle dengesiz veri kümeleri (imbalanced datasets), aşırı öğrenme (overfitting) riski ve yanlış alarm oranlarının yönetimi gibi konular, bu alanda dikkat edilmesi gereken temel sorunlar arasında yer almaktadır.

Bu tezin temel amacı, modern ağ trafiğini temsil eden UNSW-NB15 veri kümesi üzerinde makine öğrenmesi algoritmaları kullanarak saldırı tespit performansını değerlendirmek ve farklı sınıflandırıcıların istatistiksel başarı ölçütleri açısından karşılaştırılabilir sonuçlarını sunmaktır. Bu kapsamda çalışma, veri ön işleme adımlarını tanımlamakta; Karar Ağacı, Naive Bayes, Rastgele Orman, K-En Yakın Komşu ve Destek Vektör Makineleri modelleri ile eğitim sürecini gerçekleştirmektedir. Elde edilen bulgular, saldırı sınıflandırma probleminin tek bir yöntem ile çözülemeyeceğini, model başarısının veri kümesindeki örüntülerle doğrudan ilişkili olduğunu göstermektedir.

2. KAYNAK ARAŞTIRMASI

Günümüzde siber güvenlik tehditlerinin artmasıyla birlikte, ağ saldırılarının tespiti alanında yapılan çalışmalar büyük bir ivme kazanmıştır. Ağ trafiği içerisinde anormal davranışları tespit etmek amacıyla geliştirilen geleneksel yöntemler, özellikle yeni nesil saldırılar karşısında yetersiz kalmaktadır. Bu nedenle makine öğrenmesi tabanlı saldırı tespit sistemleri, hem akademik literatürde hem de endüstride önemli bir araştırma konusu haline gelmiştir. Bu bölümde, saldırı tespit sistemlerine yönelik yapılan önceki çalışmalar, kullanılan veri setleri ve makine öğrenmesi algoritmaları kapsamlı bir şekilde incelenmiştir. Ayrıca mevcut literatürdeki eksiklikler ve bu eksiklikleri gidermek amacıyla geliştirilen yaklaşımlar da değerlendirilmiştir.

2.1 Saldırı Tespit Sistemleri Üzerine Yapılan Çalışmalar

Zhang ve ark. (2019) çalışmasında, ağ trafiği verileri üzerinde derin öğrenme modellerini kullanarak saldırı tespiti gerçekleştirmiştir. Özellikle Convolutional Neural Network (CNN) ve Recurrent Neural Network (RNN) modellerinin birleşimiyle yüksek doğruluk oranları elde etmişlerdir. Kullanılan veri seti CICIDS2017 olup, çalışmada %98 üzerinde bir doğruluk oranı raporlanmıştır.

Vinayakumar ve ark. (2017), UNSW-NB15 veri seti üzerinde derin sinir ağı (Deep Neural Networks - DNN) kullanarak saldırı tespiti yapmışlardır. Çalışmalarında geleneksel makine öğrenmesi yöntemleriyle derin öğrenme yöntemlerini kıyaslamışlar ve DNN yaklaşımının daha üstün performans gösterdiğini ortaya koymuşlardır.

Javaid ve ark. (2016), NSL-KDD veri seti üzerinde Derin Öğrenme Otokodlayıcıları (Deep Autoencoders) kullanarak saldırı tespit sistemi geliştirmiştir. Otokodlayıcı tabanlı yöntemle normal ve anormal trafik ayrımında %86 doğruluk oranına ulaşılmıştır.

Moustafa ve Slay (2015), yeni bir saldırı veri seti olan UNSW-NB15'i oluşturmuş ve bu veri seti üzerinde bir dizi makine öğrenmesi algoritması ile saldırı tespiti gerçekleştirmiştir. Rastgele Orman ve Naive Bayes gibi yöntemlerle kapsamlı bir karşılaştırma yapılmış ve Rastgele Orman'ın yüksek başarı sağladığı bulunmuştur.

Shone ve ark. (2018), ikili bir Derin Otokodlayıcı mimarisi geliştirerek saldırı tespiti üzerine çalışmışlardır. NSL-KDD veri seti üzerinde test edilen bu model, geleneksel yöntemlere kıyasla daha düşük eğitim süresi ve yüksek doğruluk avantajı sunmuştur.

Dhanabal ve Shantharajah (2015), KDD Cup 99 veri seti üzerinde farklı özellik seçimi yöntemlerini ve makine öğrenmesi algoritmalarını kıyaslayarak etkili saldırı tespiti için en uygun öznitelik kümelerini belirlemeye çalışmışlardır.

Ahmed ve ark. (2016), gerçek zamanlı ağ izleme sistemleri için makine öğrenmesi tabanlı bir saldırı tespit yöntemi önermiştir. Bu çalışmada özellikle destek vektör makineleri (Destek Vektör Makinesi,) algoritması tercih edilmiş ve düşük yanlış alarm oranları (false positive) elde edilmiştir.

Aljawarneh ve ark. (2018), UNSW-NB15 ve NSL-KDD veri setleri üzerinde makine öğrenmesi ve derin öğrenme tabanlı yöntemleri karşılaştırmışlardır. Gradient Boosting Machines (GBM) ve CNN modellerinin, diğer klasik makine öğrenmesi yöntemlerine göre daha yüksek başarı oranı sunduğu rapor edilmiştir.

Tavallaee ve ark. (2009), NSL-KDD veri setini geliştirerek KDD Cup 99 veri setindeki hataları azaltmayı hedeflemişlerdir. Bu veri seti, daha dengeli bir örnekleme ve daha gerçekçi bir saldırı tespit ortamı sağlamasıyla literatürde önemli bir yer edinmiştir.

McHugh (2000), KDD Cup 99 veri setine eleştirel bir bakış sunmuş ve veri setinin saldırı çeşitliliği, örnekleme metodolojisi gibi alanlarda eksikliklere sahip olduğunu vurgulamıştır. Bu eleştiriler, daha sonraki veri seti geliştirme çalışmalarına yön vermiştir.

Hodo ve ark. (2016), Yapay Sinir Ağları (Artificial Neural Networks - ANN) kullanarak DDoS saldırılarının tespiti üzerine bir çalışma yürütmüşlerdir. IoT cihazları üzerinde yapılan testler sonucunda ANN modellerinin etkili bir şekilde anomali tespit edebildiği gösterilmiştir.

Buczak ve Guven (2016), makine öğrenmesi ve veri madenciliği yöntemlerinin saldırı tespit sistemlerindeki uygulamalarını geniş bir literatür taramasıyla incelemişlerdir. Çalışmada, çeşitli algoritmaların güçlü ve zayıf yönleri ayrıntılı şekilde analiz edilmiştir.

Srinoy (2007), destek vektör makineleri kullanarak ağ tabanlı anomali tespiti için bir yöntem önermiştir. KDD Cup 99 veri seti üzerinde yapılan deneylerde, Destek Vektör Makinesi,'nin özellikle düşük yanlış pozitif oranı ile dikkat çektiği rapor edilmiştir.

Lee ve ark. (2000), ağ davranışını modellemek için veri madenciliği tekniklerinin kullanımını önermiş ve saldırı tespit oranlarını artırmak amacıyla istatistiksel anomali tespit yaklaşımlarını geliştirmiştir. Bu çalışmalar IDS alanında veri madenciliği kullanımının öncüsü olmuştur.

Sommer ve Paxson (2010), makine öğrenmesinin saldırı tespiti için potansiyelini ve sınırlarını tartışmışlardır. Çalışmada özellikle gerçek dünya verilerinde makine öğrenmesi uygulamalarının karşılaştığı zorluklar vurgulanmıştır.

Yin ve ark. (2017), LSTM tabanlı modeller kullanarak ağ saldırılarını zaman serisi verisi gibi ele alıp tespit etmeye çalışmışlardır. Derin öğrenme tabanlı bu yaklaşım, özellikle sıralı ağ trafiği verisinde etkili sonuçlar vermiştir.

Sangkatsanee ve ark. (2011), ağ anomali tespiti için Karar Ağaçları (Karar Ağacı), Naive Bayes ve ANN algoritmalarını karşılaştırmışlardır. Sonuçlara göre, karar ağacı tabanlı yaklaşımlar hem doğruluk hem de işlem hızı açısından üstün performans göstermiştir.

2.2 Makine Öğrenmesi ile Saldırı Tespiti Çalışmaları

Bu bölümde, ağ saldırı tespiti amacıyla makine öğrenmesi yöntemlerini kullanan farklı çalışmalara yer verilmiştir. Literatürde yapılan bu araştırmalar, veri seti seçimi, kullanılan algoritmalar ve başarı oranları bakımından çeşitli yaklaşımlar ortaya koymaktadır.

Moustafa ve Slay (2015) tarafından geliştirilen UNSW-NB15 veri seti çalışmasında, ağ saldırılarını tespit etmek için istatistiksel yöntemler ve makine öğrenmesi algoritmaları karşılaştırılmıştır. Rastgele Orman, Karar Ağacı ve Naive Bayes gibi teknikler üzerinde yapılan deneylerde, gerçek dünya trafiğine yakın verilerle yüksek doğruluk oranlarına ulaşıldığı gösterilmiştir.

Tavallaee ve arkadaşları (2009), NSL-KDD veri setini geliştirerek ağ saldırı tespitinde daha dengeli ve tekrar eden kayıtlardan arındırılmış bir veri seti sunmuşlardır. Yapılan çalışmalarda, (Destek Vektör Makinesi,) ve K-Nearest Neighbors (En Yakın Komşu Algoritması) algoritmaları ile testler yapılmış, veri seti kalitesinin saldırı tespit performansını doğrudan etkilediği belirtilmiştir.

Yavanoglu ve Aydos (2017), KDD99 ve NSL-KDD veri setlerini kullanarak Rastgele Orman, Destek Vektör Makinesi, ve J48 algoritmalarını değerlendirmiştir. Çalışmalarında Rastgele Orman'ın yüksek doğruluk ve düşük yanlış alarm oranı sunduğu, ayrıca özellik seçimi uygulandığında model performansının daha da iyileştiği tespit edilmiştir.

Almseidin ve arkadaşları (2017), DoS ve DDoS saldırılarının tespiti üzerine çalışmış ve NSL-KDD veri seti ile Karar Ağacı tabanlı yöntemlerin saldırı tespitinde daha hızlı ve etkili sonuçlar verdiğini göstermiştir. Özellikle C4.5 ve J48 algoritmaları, düşük işlem süresi ile dikkat çekmiştir.

Kim ve arkadaşları (2020), derin öğrenme yöntemleri ile geleneksel makine öğrenmesi yöntemlerini ağ saldırı tespitinde karşılaştırmışlardır. CICIDS2017 veri setinde yapılan çalışmada, Rastgele Orman'ın daha az işlem gücü ve eğitim verisi kullanarak yüksek doğruluk sağladığı görülmüştür.

Shone ve arkadaşları (2018), otomatik özellik çıkarımı amacıyla derin öğrenme tabanlı bir yöntem önermiş ve ardından geleneksel makine öğrenmesi algoritmaları ile sınıflandırma yapmıştır. UNSW-NB15 veri setinde yapılan çalışmada, özellikle Otomatik Kodlayıcı (Autoencoder) ile elde edilen özelliklerin tespit oranlarını artırdığı görülmüştür.

Muda ve arkadaşları (2011), Karar Ağacı tabanlı yöntemlerin KDD99 veri setinde yüksek tespit başarısı gösterdiğini ortaya koymuşlardır. Özellikle ID3 ve C4.5 algoritmalarının hızlı eğitim süreleri ile pratik IDS çözümleri oluşturabileceği vurgulanmıştır.

Zhang ve arkadaşları (2008), Destek Vektör Makinesi, algoritmasının ağ saldırı tespiti üzerindeki etkilerini incelemiş, farklı çekirdek (kernel) fonksiyonlarının doğruluk oranlarını nasıl etkilediğini göstermiştir. Çalışmada özellikle RBF çekirdeğinin yüksek başarı sağladığı belirtilmiştir.

Dhanabal ve Shantharajah (2015), KDD99 veri setini kullanarak farklı makine öğrenmesi algoritmalarını kıyaslamış ve Karar Ağacı yönteminin saldırı tespitinde en başarılı sonuçları verdiğini göstermiştir. Ayrıca veri ön işleme ve normalizasyonun sonuçlar üzerinde ciddi etkileri olduğu vurgulanmıştır.

Majeed ve arkadaşları (2020), CICIDS2017 veri seti kullanarak Rastgele Orman, Gradient Boosting ve AdaBoost algoritmalarını karşılaştırmıştır. Sonuçlarda Rastgele Orman algoritmasının hem doğruluk hem de işlem süresi açısından daha avantajlı olduğu görülmüştür.

Ambusaidi ve arkadaşları (2016), NSL-KDD veri setinde özellik seçimi algoritmalarını uygulayarak makine öğrenmesi tabanlı IDS sistemlerinin performansını artırmayı hedeflemiştir. Özellikle ReliefF algoritması ile seçilen özelliklerin, tespit oranını ciddi şekilde yükselttiği belirtilmiştir.

Revathi ve Malathi (2013), yapay sinir ağları ile desteklenen IDS sistemlerinin KDD99 veri seti üzerinde test edilmesini sağlamışlardır. Çalışmalarında, sınıflandırma doğruluğunu artırmak için veri kümesinin farklı sınıflara dengeli bir şekilde bölünmesi gerektiği vurgulanmıştır.

Xia ve arkadaşları (2018), ağ trafiği üzerinde davranış analizi yaparak anomaly-based IDS tasarlamışlardır. Özellikle Naive Bayes ve Karar Ağacı algoritmalarının gerçek zamanlı saldırı tespitinde başarılı sonuçlar verdiğini rapor etmişlerdir.

Abraham ve Thomas (2005), istatistiksel öğrenme yöntemleri ile makine öğrenmesi tekniklerini birleştirerek saldırı tespit sistemleri geliştirmiştir. Rastgele Orman, Destek Vektör Makinesi, ve Bayes yöntemleri ile yapılan testlerde, karma yaklaşımların (ensemble learning) doğruluğu artırdığı gözlemlenmiştir.

Ashfaq ve arkadaşları (2017), yapay sinir ağı tabanlı bir IDS tasarlamış ve farklı optimizasyon teknikleri ile doğruluk oranlarını artırmaya çalışmışlardır. Çalışmalarında, saldırı tespit oranlarında önemli artışlar gözlemlenmiştir.

2.3 Kullanılan Veri Setleri Üzerine Literatür

Makine öğrenmesi tabanlı ağ saldırı tespit çalışmalarında kullanılan veri setleri, geliştirilen modellerin doğruluğunu ve genellenebilirliğini doğrudan etkilemektedir. Bu nedenle literatürde yaygın olarak tercih edilen veri kümelerinin sahip olduğu trafik yapısı, saldırı çeşitliliği ve veri dengesi gibi özellikler büyük önem taşımaktadır.

Ağ saldırı tespiti alanında uzun yıllar boyunca temel referans olarak kullanılan KDD Cup 99 veri seti, 1998 yılında DARPA Intrusion Detection Evaluation programı kapsamında toplanan ağ trafiği verilerine dayanmaktadır. Veri kümesi; normal trafik ile birlikte DoS, R2L, U2R ve Probe gibi farklı saldırı sınıflarını içermektedir. Ancak yüksek oranda tekrar eden kayıtlar ve sınıflar arasındaki dengesizlikler, bu veri setinin zamanla eleştirilmesine neden olmuş ve daha gerçekçi veri kümelerinin geliştirilmesine zemin hazırlamıştır (Tavallaee ve ark., 2009).

Bu sınırlamaları azaltmak amacıyla geliştirilen NSL-KDD veri seti, KDD Cup 99'a kıyasla daha dengeli bir dağılım sunmakta ve tekrar eden örneklerin sayısını önemli ölçüde azaltmaktadır. Eğitim ve test kümeleri arasındaki tutarsızlıkların giderilmesi sayesinde, özellikle yeni makine öğrenmesi algoritmalarının performanslarının daha sağlıklı biçimde değerlendirilmesine olanak sağlamaktadır.

Daha güncel saldırı senaryolarını yansıtmak amacıyla 2015 yılında Moustafa ve Slay tarafından sunulan UNSW-NB15 veri seti, modern ağ ortamlarını temsil eden karmaşık trafik yapısını içermektedir. Bu veri seti; Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode ve Worms gibi çeşitli saldırı kategorileri ile birlikte normal ağ trafiğini kapsamaktadır. Toplam 49 öznitelikten oluşan UNSW-NB15, gerçekçi trafik üretimi ve saldırı çeşitliliği nedeniyle literatürde sıklıkla tercih edilmektedir.

Kanada Siber Güvenlik Enstitüsü (CIC) tarafından geliştirilen CICIDS2017 veri seti ise, günümüz ağlarında karşılaşılan saldırı türlerini gerçek kullanıcı davranışlarına dayalı olarak modellemektedir. DDoS, DoS, Web tabanlı saldırılar, Brute Force ve Infiltration gibi farklı saldırı kategorilerini içeren bu veri seti, gerçekçi trafik senaryoları sunması nedeniyle modern saldırı tespit çalışmalarında önemli bir referans kaynağı olarak değerlendirilmektedir.

Bu bölümde, makine öğrenmesi tabanlı ağ saldırı tespit çalışmalarında kullanılan başlıca veri setleri incelenmiştir. Veri seti seçimi, bir saldırı tespit sisteminin başarısını doğrudan etkileyen kritik bir unsurdur. Bu nedenle, literatürde sıkça tercih edilen veri setlerinin yapısı, içerdiği saldırı türleri ve sağladığı avantajlar detaylı olarak ele alınmıştır.

KDD Cup 99 veri seti, saldırı tespiti araştırmalarında en yaygın kullanılan veri setlerinden biridir. Bu veri seti, 1998 DARPA Intrusion Detection Evaluation programı kapsamında oluşturulmuş ve daha sonra KDD Cup 1999 yarışmasında kullanılmıştır. Veri seti, normal trafik verilerinin yanı sıra DoS, R2L, U2R ve Probe gibi farklı saldırı türlerini içermektedir. Ancak, tekrar eden veriler ve veri dengesizlikleri nedeniyle eleştirilmiş ve daha yeni veri setlerinin geliştirilmesine yol açmıştır (Tavallaee et al., 2009).

NSL-KDD veri seti, KDD Cup 99'un eksiklerini gidermek amacıyla geliştirilmiştir. NSL-KDD, tekrar eden kayıtların azaltılması ve eğitim ile test verileri arasındaki uyumsuzlukların giderilmesi gibi önemli iyileştirmeler sunmuştur. Bu veri seti, özellikle yeni algoritmaların değerlendirilmesi ve saldırı tespit performanslarının gerçekçi bir şekilde ölçülmesi açısından önemli bir kaynak olmuştur.

UNSW-NB15 veri seti, 2015 yılında Moustafa ve Slay tarafından oluşturulmuş ve gerçek dünya trafiğine daha yakın bir ortam sağlamayı amaçlamıştır. Veri seti, normal trafiğin yanı sıra Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode ve Worms gibi saldırı türlerini içermektedir. 49 özellik içeren bu veri seti, modern ağ trafiğinin karmaşıklığını daha iyi temsil etmesi nedeniyle literatürde sıklıkla kullanılmaktadır.

CICIDS2017 veri seti, Kanada Siber Güvenlik Enstitüsü (CIC) tarafından geliştirilmiş ve günümüz ağ ortamlarında görülen çeşitli saldırı türlerini kapsamaktadır. Veri seti, DDoS, DoS, Web Attack, Brute Force, Infiltration gibi farklı saldırı kategorilerini ve normal trafiği içermektedir. CICIDS2017, gerçek kullanıcı davranışlarına dayalı olarak oluşturulduğu için, makine öğrenmesi ve derin öğrenme çalışmaları için ideal bir kaynak olarak kabul edilmektedir.

ISCXIDS 2012 veri seti, yine Kanada Siber Güvenlik Enstitüsü tarafından oluşturulmuş ve hem normal hem de anormal ağ trafiği içeren etiketli veriler sağlamıştır. ISCXIDS 2012 veri seti, zaman damgası bilgilerinin korunmuş olması nedeniyle zaman bazlı analizler için de uygun bir veri seti olmuştur.

BoT-IoT veri seti, Nesnelerin İnterneti (IoT) ortamlarında meydana gelen saldırıları modellemek için geliştirilmiştir. Bu veri seti, IoT cihazlarının ağ trafiği üzerinde oluşturabileceği anormalliklerin tespit edilmesine yönelik makine öğrenmesi tabanlı çalışmalar için önemli bir referans noktası sunmuştur.

TON_IoT veri seti, endüstriyel IoT ortamlarını modellemek amacıyla oluşturulmuş bir başka güncel veri setidir. TON_IoT, telemetri verileri, ağ trafiği ve log dosyaları gibi çeşitli veri türlerini içermekte, böylece çoklu veri kaynaklarından saldırı tespiti yapılmasına olanak sağlamaktadır.

ADFA-LD veri seti, gelişmiş kalıcı tehdit (APT) saldırılarını tespit etmek için geliştirilmiştir. Özellikle sıfırıncı gün saldırılarına karşı davranış tabanlı tespit mekanizmalarını test etmek için kullanılmıştır. Veri seti, düşük kaynak tüketimi ve yüksek doğrulukla anomali tespiti yapmak isteyen araştırmacılar için uygun bir yapı sunmaktadır.

CTU-13 veri seti, botnet aktivitelerini tespit etmek amacıyla geliştirilmiş ve gerçek ağ ortamında oluşturulmuş bir veri setidir. Botnet saldırılarını, normal kullanıcı trafiğinden ayırmak için makine öğrenmesi algoritmalarına test verisi sağlamıştır.

CSE-CIC-IDS2018 veri seti, modern saldırı tekniklerini kapsayan bir diğer kapsamlı veri setidir. Ransomware, SQL Injection, Heartbleed gibi güncel saldırıları içeren bu veri seti, literatürde saldırı tespit sistemlerinin güncel tehditlere karşı dayanıklılığını test etmek amacıyla kullanılmaktadır.

BoT-NeTIoT veri seti, IoT cihazlarının ürettiği ağ trafiğini simüle ederek oluşturulmuş ve IoT ağları üzerindeki DDoS saldırılarını tespit etmeye odaklanmıştır. Bu veri seti, özellikle düşük güçlü cihazların bulunduğu ağlarda güvenlik araştırmaları için önemli bir kaynak olmuştur.

Literatürde yer alan çalışmaların ortak noktası, ağ saldırılarının dinamik yapısı nedeniyle tek bir sınıflandırma modelinin bütün tehdit tiplerinde üstün performans göstermediğidir. Çeşitli araştırmalar, makine öğrenmesi tabanlı IDS modellerinin belirli saldırı kategorilerinde başarılı sonuç üretmesine rağmen, veri dağılımı, öznitelik sayısı, sınıf dengesizliği ve anomali örneklerinin çeşitliliği gibi faktörlerin performansı doğrudan etkilediğini ortaya koymuştur. Bu nedenle literatürde giderek artan şekilde, farklı modellerin karşılaştırmalı analizlerinin yapılması, hibrit yaklaşımların önerilmesi ve standartlaştırılmış veri kümeleri üzerinde yeniden üretilebilir deney ortamlarının kurulması önerilmektedir. Bu bağlamda, bu tez çalışması literatüre paralel biçimde, denetimli makine öğrenmesi yöntemlerinin UNSW-NB15 gibi zengin ve çok kategorili bir veri kümesi üzerinde sistematik olarak değerlendirilmesini sağlamış; modellerin avantaj ve sınırlılıklarını nicel metriklerle tartışarak mevcut araştırma çizgisini tamamlayıcı nitelikte bir katkı sunmuştur.

2.4 UNSW-NB15 Tabanlı Çalışmaların Karşılaştırılması

UNSW-NB15 veri kümesi, güncel ve gerçekçi yapısı nedeniyle ağ saldırı tespit çalışmalarında yaygın olarak kullanılmaktadır. Bu bölümde, aynı veri kümesini kullanan önceki çalışmalar, kullanılan yöntemler ve sınıflandırma yaklaşımları açısından karşılaştırmalı olarak özetlenmiştir.

Çizelge 2.1 UNSW-NB15 tabanlı çalışmaların karşılaştırılması

Yıl	Çalışma	Kullanılan Veri Seti	Kullanılan Yöntem(ler)	Sınıflandırma Türü	Raporlanan Performans / Not
2015	Moustafa & Slay	UNSW-NB15	İstatistiksel analiz, temel ML	—	UNSW-NB15 veri kümesinin tanıtımı ve karakterizasyonu
2016	Moustafa et al.	UNSW-NB15	Decision Tree, NB, SVM	Binary	Klasik ML algoritmaları ile saldırı tespiti
2017	Janarthanan & Zargari	UNSW-NB15	Feature Selection + ML	Binary / Multi	Özellik seçiminin IDS performansına etkisi
2018	Sarhan et al.	UNSW-NB15	Random Forest	Binary	RF ile yüksek doğruluk elde edilmiştir
2019	Bagui et al.	UNSW-NB15	ML tabanlı analiz	Multi-class	Nadir saldırı türlerinin tespiti incelenmiştir
2020	Kasongo & Sun	UNSW-NB15	Feature Selection + RF, SVM	Binary	Özellik azaltma ile performans artışı
2020	Alazab et al.	UNSW-NB15	Deep Learning	Multi-class	Derin öğrenme ile saldırı sınıflandırması
2020	Yao et al.	UNSW-NB15	Autoencoder + ML	Binary	Anomali tespiti odaklı çalışma
2021	Vinayakumar et al.	UNSW-NB15	Ensemble Learning	Multi-class	Çoklu sınıflandırıcıların karşılaştırılması
2021	Hodo et al.	UNSW-NB15	SVM, RF	Binary	Geleneksel ML performans analizi
2022	Ferrag et al.	UNSW-NB15	Hybrid ML/DL	Multi-class	Hibrit IDS mimarisi önerilmiştir
2023	MDPI Study	UNSW-NB15	RF, SVM, KNN	Binary / Multi	Güncel ML karşılaştırması
2023	CIC-based Study	UNSW-NB15	ML + Feature Engineering	Multi-class	Sınıf bazlı hata analizi yapılmıştır
2024	Recent IDS Study	UNSW-NB15	LightGBM, RF	Multi-class	Boosting tabanlı yaklaşımlar değerlendirilmiştir
2026	**Bu Çalışma**	**UNSW-NB15**	**RF, DT, SVM, KNN, NB**	**Multi-class**	**Kapsamlı metrikler ve hata analizi ile karşılaştırmalı değerlendirme**

Çizelge 2.1 incelendiğinde, literatürdeki çalışmaların büyük bir kısmının ikili sınıflandırmaya odaklandığı, çok sınıflı saldırı tespitine yönelik çalışmaların ise sınırlı olduğu görülmektedir. Bu tez çalışması, UNSW-NB15 veri kümesi üzerinde çok sınıflı saldırı tespiti problemini ele alarak, farklı denetimli öğrenme algoritmalarının performanslarını kapsamlı metriklerle değerlendirmesi bakımından literatüre katkı sağlamaktadır.

2.5 Kullanılan Algoritmaların İncelenmesi

Makine öğrenmesi tabanlı saldırı tespit sistemlerinde, farklı sınıflandırma algoritmaları kullanılarak ağ trafiği içerisindeki anormal davranışlar tespit edilmeye çalışılmaktadır. Bu çalışmada da kullanılan Rastgele Orman, Karar Ağacı, (Destek Vektör Makinesi,), K-Nearest Neighbors (En Yakın Komşu Algoritması) ve Naive Bayes algoritmalarının temel prensipleri ve saldırı tespitindeki rolleri aşağıda incelenmiştir.

Bu çalışmada kullanılan makine öğrenmesi algoritmaları, saldırı tespit probleminin çok sınıflı yapısı, veri kümesinin boyutu ve öznitelik uzayının karmaşıklığı dikkate alınarak seçilmiştir. Seçilen algoritmalar; karar tabanlı, olasılıksal, mesafe tabanlı ve topluluk öğrenme yaklaşımlarını temsil edecek şekilde belirlenmiş, böylece farklı öğrenme stratejilerinin saldırı tespit performansı üzerindeki etkileri karşılaştırmalı olarak analiz edilmiştir. Ayrıca, klasik makine öğrenmesi algoritmalarının tercih edilmesinin amacı, modellerin yorumlanabilirliğini koruyarak çok sınıflı saldırı tespiti problemine uygulanabilirliğini değerlendirmektir.

Rastgele Orman: Rastgele Orman algoritması, birden çok karar ağacı oluşturarak her bir ağacın sonucuna göre çoğunluk oyu prensibiyle sınıflandırma yapan bir topluluk (ensemble) yöntemidir. Ağaçlar, veri kümesinin rastgele seçilmiş alt kümeleri üzerinde eğitilerek oluşturulur ve böylece modelin aşırı öğrenme (overfitting) yapması önlenir. Saldırı tespit çalışmalarında Rastgele Orman, yüksek doğruluk oranı ve düşük hata payı sağlaması nedeniyle sıklıkla tercih edilmektedir. Özellikle büyük ve dengesiz veri kümelerinde başarılı sonuçlar vermektedir.

Karar Ağacı: Karar ağacı, verileri belirli özellikler üzerinden dallara ayırarak sınıflandıran bir algoritmadır. Ağacın her bir dalı, bir özelliğe dayalı olarak veriyi ikiye böler ve bu süreç yaprak düğümlere kadar devam eder. Basitliği ve yorumlanabilirliği nedeniyle saldırı tespitinde tercih edilmektedir. Ancak, tek başına kullanıldığında aşırı öğrenmeye (overfitting) eğilimli olabilir.

Destek vektör makinesi (SVM): Destek Vektör Makinesi, sınıflandırma problemlerinde verileri en geniş marjı sağlayacak şekilde ayıran bir karar sınırı (hiper düzlem) oluşturmayı hedefleyen denetimli bir öğrenme algoritmasıdır. Doğrusal olarak ayrılabilir olmayan veri kümelerinde, çekirdek (kernel) fonksiyonları kullanılarak veriler daha yüksek boyutlu bir uzaya taşınmakta ve bu sayede karmaşık ayrım sınırları elde edilebilmektedir. Saldırı tespit uygulamalarında SVM, özellikle sınıf dağılımının dengeli olduğu ve veri boyutunun görece sınırlı olduğu durumlarda yüksek doğruluk oranları sunabilmektedir. Bununla birlikte, veri kümesinin boyutunun artması durumunda modelin eğitim süresinin uzaması önemli bir dezavantaj olarak ortaya çıkmaktadır.

En yakın komşu algoritması (KNN): k-En Yakın Komşu algoritması, sınıflandırılacak bir örneğin, özellik uzayında kendisine en yakın olan k adet komşu örneğin sınıf etiketlerine göre karar verilmesi prensibine dayanmaktadır. Parametrik olmayan yapısı sayesinde öğrenme aşaması gerektirmeyen bu yöntem, basitliği ve uygulanabilirliği ile öne çıkmaktadır. Ancak, özellikle yüksek boyutlu veri kümelerinde mesafe hesaplamalarının artması, algoritmanın performansını olumsuz yönde etkileyebilmektedir. Saldırı tespit sistemlerinde KNN, benzer trafik davranışlarının kümelenmesi sayesinde belirli saldırı türlerinin ayırt edilmesinde etkili sonuçlar üretebilmektedir.

Naive Bayes: Naive Bayes algoritması, olasılık temelli bir sınıflandırma yaklaşımı olup Bayes teoremine dayanmaktadır. Bu yöntemde, özniteliklerin birbirinden bağımsız olduğu varsayımı kabul edilerek sınıflandırma işlemi gerçekleştirilmektedir. Basit yapısı ve düşük hesaplama maliyeti sayesinde Naive Bayes, özellikle gerçek zamanlı saldırı tespit uygulamalarında ve işlem gücü kısıtlı ortamlarda tercih edilmektedir. Bununla birlikte, sınıflar arasında belirgin ayrımların bulunduğu veri kümelerinde daha başarılı sonuçlar üretirken, karmaşık bağımlılık ilişkilerinin yoğun olduğu durumlarda performansının sınırlı kalabildiği gözlemlenmektedir.

Bu çalışma kapsamında ele alınan makine öğrenmesi algoritmaları, farklı veri ön işleme yöntemleri ve parametre ayarları kullanılarak değerlendirilmiştir. Modellerin

performansları; doğruluk, kesinlik, duyarlılık ve F1 skoru gibi yaygın performans ölçütleri temel alınarak karşılaştırılmıştır. Böylece her bir algoritmanın veri kümesine uyum yeteneği ve sınıflandırma başarımı ayrıntılı biçimde analiz edilmiştir.

2.6 Literatürdeki Zayıf Yönler ve İyileştirme İhtiyacı

Literatürde yer alan makine öğrenmesi tabanlı ağ saldırı tespit çalışmalarında önemli ilerlemeler kaydedilmiş olmakla birlikte, bazı temel sınırlılıkların hâlen devam ettiği görülmektedir. Bu sınırlılıklar, geliştirilen modellerin gerçek dünya ağ ortamlarına genellenebilirliğini doğrudan etkilemektedir.

Sınırlı Veri Çeşitliliği: Birçok çalışmada kullanılan veri setleri, belirli saldırı türlerine odaklanmakta ve gerçek ağ trafiğinin tüm dinamiklerini yeterince yansıtamamaktadır. Bu durum, geliştirilen modellerin yeni ve daha önce görülmemiş saldırı türleri karşısında sınırlı performans göstermesine neden olabilmektedir. Özellikle sıfırıncı gün (zero-day) saldırılarının tespitinde mevcut modellerin yetersiz kaldığı literatürde sıkça vurgulanmaktadır.

Aşırı Öğrenme(Overfitting)Problemi: Bazı makine öğrenmesi modelleri, eğitim verileri üzerinde yüksek doğruluk değerleri elde ederken, test verileri üzerinde benzer başarıyı gösterememektedir. Bu durum, modelin eğitim verisine aşırı uyum sağlamasından ve yeni veriler karşısında genelleme yeteneğinin azalmasından kaynaklanmaktadır. Aşırı öğrenme problemi, saldırı tespit sistemlerinin gerçek ağ ortamlarında güvenilir şekilde kullanılmasını zorlaştıran önemli bir faktör olarak değerlendirilmektedir.

Özellik Seçiminde Yetersizlik: Pek çok çalışma, veri setindeki tüm özellikleri kullanarak model geliştirmektedir. Ancak bazı özelliklerin saldırı tespiti için düşük bilgi değerine sahip olması, modelin gereksiz karmaşıklığa sahip olmasına ve işlem süresinin uzamasına yol açmaktadır. Bu durum, özellikle gerçek zamanlı saldırı tespiti gerektiren sistemlerde önemli bir dezavantaj oluşturmaktadır.

Gerçek Zamanlı Uygulamaya Uygunluk Eksikliği: Çalışmaların büyük bir kısmı offline analizlere odaklanmakta ve gerçek zamanlı trafik üzerinde test edilmemektedir. Gerçek zamanlı saldırı tespit sistemleri için hız ve düşük gecikme (latency) kritik öneme sahiptir. Ancak literatürdeki birçok yöntem, yüksek doğruluk oranına rağmen gerçek zamanlı çalıştırıldığında performans düşüşü yaşayabilmektedir.

Veri Dengesizliği Sorunu: Ağ saldırı veri setlerinde genellikle normal trafik örnekleri saldırı örneklerine göre çok daha fazladır. Bu dengesizlik, makine öğrenmesi modellerinin saldırı sınıflarını doğru tahmin etmekte zorlanmasına ve yanlış sınıflandırmalara neden olabilmektedir. Veri dengesizliğini gidermek için uygulanan oversampling veya undersampling gibi yöntemlerin etkili kullanımı literatürde yeterince ele alınmamıştır.

Saldırı Türlerinin Ayrıştırılmasında Güçlük: Birçok çalışma, saldırıları sadece "saldırı" veya "normal" olarak iki kategoriye ayırmakta, saldırıların tür bazlı (örneğin, DoS, Probe, R2L, U2R gibi) detaylı ayrımını yapmamaktadır. Ancak farklı saldırı türlerinin tespit edilmesi, ağ güvenliğini sağlamak açısından büyük önem taşımaktadır.

Değişen Saldırı Tekniklerine Uyum Eksikliği: Siber saldırı teknikleri sürekli olarak evrim geçirmektedir. Literatürde geliştirilen birçok model, eğitim verisindeki sabit saldırı türlerine göre optimize edilmiştir. Ancak, saldırı yöntemlerindeki küçük değişiklikler bile modellerin başarımını önemli ölçüde düşürebilmektedir.

Performans Değerlendirme Yöntemlerinin Standartlaşmaması: Literatürde yer alan ağ saldırı tespit çalışmalarında kullanılan performans değerlendirme ölçütlerinin farklılık göstermesi, yöntemlerin doğrudan karşılaştırılmasını zorlaştıran önemli bir sorun olarak öne çıkmaktadır. Bazı çalışmalarda sınıflandırma başarımı yalnızca doğruluk (accuracy) değeri üzerinden raporlanırken, diğer çalışmalarda kesinlik (precision), duyarlılık (recall) ve F1 skoru gibi daha kapsamlı ölçütlere yer verilmektedir. Bu durum, farklı yaklaşımların performanslarının tutarlı bir şekilde değerlendirilmesini güçleştirmekte ve elde edilen sonuçların yorumlanmasını zorlaştırmaktadır.

3. MATERYAL VE YÖNTEM

Bu bölümde, tez çalışması kapsamında kullanılan veri kümesi, veri ön işleme adımları, özellik seçimi süreci ve makine öğrenmesi modellerinin eğitim ile değerlendirme aşamaları ayrıntılı olarak açıklanmaktadır. Çalışmanın temel amacı, ağ saldırılarının tespitinde etkili bir makine öğrenmesi tabanlı yaklaşım geliştirmek ve farklı algoritmaların performanslarını karşılaştırmalı olarak analiz etmektir.

3.1 Kullanılan Veri Seti: UNSW-NB15

Bu tez çalışmasında, makine öğrenmesi tabanlı saldırı tespit modellerinin performansını değerlendirmek amacıyla UNSW-NB15 veri kümesi kullanılmıştır. Veri kümesi, Avustralya'daki New South Wales Üniversitesi (UNSW) bünyesinde yer alan Cyber Range Laboratory ortamında oluşturulmuş ve gerçekçi ağ trafiğini temsil edecek şekilde tasarlanmıştır. Çalışmada, veri kümesinin araştırma topluluğu tarafından sunulan eğitim (UNSW_NB15_training-set.csv) ve test (UNSW_NB15_testing-set.csv) dosyaları kullanılmış; her iki dosya Python ortamında aynı veri ön işleme adımlarına tabi tutulmuştur.

UNSW-NB15 veri kümesi, gerçek ağ ortamlarında gözlemlenen hem normal hem de çeşitli saldırı türlerine ait trafiği içermektedir. Toplam 2.540.044 adet ağ trafiği kaydından oluşan bu veri kümesinde; Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode ve Worms gibi farklı saldırı kategorileri yer almaktadır.

Deneysel çalışmalar, bulut tabanlı bir geliştirme ortamı olan Google Colab üzerinde gerçekleştirilmiştir. Bu ortamda Python programlama dili kullanılmış; veri işleme, model eğitimi ve performans değerlendirme aşamalarında *pandas*, *numpy*, *scikit-learn*, *matplotlib* ve *seaborn* gibi yaygın kütüphanelerden yararlanılmıştır. Tüm deneysel adımların tek bir çalışma dosyası (notebook) içerisinde yürütülmesi, deneylerin tekrarlanabilirliğini ve sonuçların doğrulanabilirliğini artırmıştır.

Google Colab ortamı, Python tabanlı veri analizi kütüphanelerinin doğrudan kullanılmasına ve hesaplama kaynaklarına erişime olanak tanımaktadır. Bu sayede veri ön işleme, sınıflandırma algoritmalarının eğitimi ve performans ölçütlerinin hesaplanması süreçleri verimli bir şekilde yürütülmüştür. Ayrıca, Colab'ın sunduğu altyapı sayesinde deneysel sonuçlar yeniden üretilebilir biçimde kayıt altına alınmış ve çalışmanın yöntemsel şeffaflığı sağlanmıştır.

Çalışmada kullanılan veri kümesi, modelin eğitimi ve değerlendirilmesi amacıyla eğitim ve test olmak üzere iki ayrı alt kümeye ayrılmıştır. Eğitim kümesi 175.341 adet kayıt içerirken, test kümesi 82.332 adet kayıttan oluşmaktadır. Bu ayrım, modelin genelleme yeteneğinin değerlendirilmesi ve aşırı öğrenme (overfitting) riskinin azaltılması açısından önem taşımaktadır.

Ağ trafiği analizinde kullanılan özellikler arasında, iletişim protokollerine ilişkin bilgiler de yer almaktadır. Örneğin HTTPS, istemci ve sunucu arasındaki veri iletiminin güvenli bir şekilde gerçekleştirilmesini sağlarken; FTP dosya aktarımı, SMTP ise e-posta iletimi gibi temel ağ hizmetlerinde kullanılmaktadır. Bu protokollere ait trafik özellikleri, ağ davranışlarının modellenmesi ve saldırı tespitinde anlamlı özniteliklerin elde edilmesi açısından değerlendirilmiştir.

3.2 UNSW-NB15 Veri Kümesi Özellik Açıklamaları

UNSW-NB15 veri kümesi, ağ trafiğini akış seviyesinde temsil eden 49 öznitelik (feature) içermektedir. Bu öznitelikler genel olarak **temel akış bilgisi, zaman tanımları, protokol bayrakları, istatistiksel dağılımlar ve davranışsal göstergeler** olmak üzere farklı kategorilere ayrılmaktadır.

Temel trafik özellikleri (Basic features)

1. **srcip** – Trafiği başlatan kaynağın IP adresi.
2. **sport** – Kaynak port numarası.
3. **dstip** – Hedef IP adresi.
4. **dsport** – Hedef port numarası.

5. **proto** – Kullanılan ağ protokolü (TCP, UDP, ICMP vb.).

Zaman ve bağlantı özellikleri (Time features)

6. **state** – Akışın bağlantı durumunu (örn. ACC, CON, FIN, INT) belirtir.
7. **dur** – Akışın saniye cinsinden süresi.
8. **sbytes** – Kaynaktan gönderilen bayt miktarı.
9. **dbytes** – Hedeften gelen bayt miktarı.
10. **sttl** – Kaynaktan çıkan paketlerdeki TTL (time-to-live) değeri.
11. **dttl** – Hedeften gelen paketlerdeki TTL değeri.
12. **sloss** – Kaynak tarafında paket kaybı sayısı.
13. **dloss** – Hedef tarafında paket kaybı sayısı.
14. **service** – Sunucu tarafında kullanılan uygulama servisi (örn. HTTP, SMTP, FTP).

TCP bayrak özellikleri (TCP flag features)

15. **Sload** – Kaynak tarafından oluşturulan veri yükü (bit/s).
16. **Dload** – Hedef tarafından oluşturulan veri yükü (bit/s).
17. **Spkts** – Kaynak tarafından gönderilen toplam paket sayısı.
18. **Dpkts** – Hedeften gelen toplam paket sayısı.
19. **swin** – Kaynak tarafındaki TCP pencere boyutu.
20. **stcpb** – Kaynağın TCP başlangıç bayt imzası.
21. **dtcpb** – Hedefin TCP başlangıç bayt imzası.
22. **dwin** – Hevf TCP pencere boyutu.
23. **tcprtt** – TCP Round-Trip-Time değeri.
24. **synack** – SYN-ACK bayraklarının gecikme süresi.
25. **ackdat** – ACK paketlerinin doğrulama gecikmesi.

Oran ve yoğunluk ölçütleri (Rate-based statistical features)

26. **smean** – Kaynaktan gelen paketlerin ortalama boyutu.
27. **dmean** – Hedeften gelen paketlerin ortalama boyutu.
28. **trans_depth** – Bağlantı sırasında gönderilen komut sayısı.
29. **response_body_len** – Sunucu yanıt gövdesinin bayt cinsinden uzunluğu.
30. **ct_srv_src** – Belirli bir servise kaynak IP'den gelen bağlantı sayısı.
31. **ct_state_ttl** – Belirli TTL ve durum kombinasyonuna sahip paket sayısı.
32. **ct_dst_ltm** – Belirli bir hedef IP'ye yapılan uzun süreli bağlantı sayısı.
33. **ct_src_ltm** – Belirli bir kaynak IP'nin yaptığı uzun süreli bağlantı sayısı.
34. **ct_dst_src_ltm** – Belirli kaynak-hedef çiftleri arasındaki uzun süreli bağlantı sayısı.

Orantısal saldırı göstergeleri (Behavior-based features)

35. **is_ftp_login** – FTP kimlik doğrulamasının başarı durumunu gösterir.
36. **ct_ftp_cmd** – FTP komut denemelerinin sayısı.
37. **ct_flw_http_mthd** – HTTP istek yöntemlerinin sayısı (GET, POST vb.).
38. **ct_src_dport_ltm** – Kaynağın belirli bir hedef portuna yaptığı uzun süreli bağlantılar.

39. **ct_dst_sport_ltm** – Hedefin belirli bir kaynak kod portuna yaptığı uzun süreli bağlantılar.
40. **ct_dst_src_ltm** – Belirli kaynak-hedef ilişkilerinin tekrar sayısı.

Etiket özellikleri

44. **attack_cat** – Trafiğin saldırı kategorisi (Fuzzing, Exploit, DoS, Reconnaissance vb.).
45. **label** – Akışın saldırı (1) veya normal (0) olduğunu gösterir.

UNSW-NB15 literatüründe toplam 9 saldırı kategorisi bulunmaktadır.

3.3 Veri Ön İşleme ve Özellik Seçimi

UNSW-NB15 veri kümesi, gerçekçi ağ trafiğini yansıtan çok sayıda öznitelik ve farklı saldırı türlerini içermektedir. Bu nedenle, makine öğrenmesi algoritmalarının daha verimli çalışabilmesi için veri seti üzerinde çeşitli ön işleme adımları uygulanmıştır.

Veri seti üzerinde makine öğrenmesi algoritmalarını uygulamadan önce aşağıdaki ön işleme adımları gerçekleştirilmiştir:

Sütun Temizliği: İlk olarak, veri setinde yer alan sütun isimleri üzerinde biçimsel temizlik yapılmıştır. CSV dosyalarından gelen bazı sütun adlarında boşluk ve özel karakter kaynaklı sorunlar oluşabildiğinden, tüm sütun isimleri str.strip() yöntemi ile düzeltilmiş ve fazladan boşluk karakterleri kaldırılmıştır.

Veri setindeki tüm sütun isimlerindeki boşluklar ve özel karakterler temizlenmiştir.

Eksik ve Sonsuz Değerlerin Temizlenmesi: Veri setinin güvenilir bir şekilde kullanılabilmesi için, sayısal hesaplamaları bozan veya model performansını olumsuz etkileyebilecek değerler temizlenmiştir. Bu kapsamda:

- ∞ (sonsuz) ve −∞ içeren gözlemler, `NaN` ile değiştirilmiş,

- Ardından hem eğitim hem de test veri setinde **eksik gözlem (NaN)** içeren kayıtlar veri setinden çıkarılmıştır.

Bu işlem sonucunda, model eğitimi ve test aşamalarında yalnızca sayısal olarak tutarlı ve eksiksiz kayıtlar kullanılmıştır

NaN, Inf ve -Inf değerleri tespit edilerek, bu kayıtlar veri setinden çıkarılmıştır.

Kategorik Verilerin Sayısallaştırılması: UNSW-NB15 veri kümesinde yer alan bazı öznitelikler, makine öğrenmesi algoritmaları tarafından doğrudan işlenemeyen kategorik yapıdadır. Bu çalışmada özellikle üç sütun üzerinde kategorik kodlama yapılmıştır:

- proto (kullanılan ağ protokolü, örn. TCP, UDP, ICMP),
- service (uygulama servisi, örn. http, ftp, smtp),
- state (bağlantı durumu).

Bu değişkenler, Label Encoding yöntemi ile sayısal değerlere dönüştürülmüştür. Etiketlerin tutarlılığını korumak için, eğitim ve test veri setleri ilgili sütun bazında birleştirilmiş, kodlayıcı (LabelEncoder) bu birleşik veri üzerinde eğitilmiş ve daha sonra hem eğitim hem de test veri setlerine aynı kodlama şeması uygulanmıştır. Böylece, eğitim sırasında ve test aşamasında aynı kategorik değerlerin aynı sayısal koda karşılık gelmesi sağlanmıştır.

proto, service ve state gibi kategorik değişkenler, Label Encoding yöntemi kullanılarak sayısal değerlere dönüştürülmüştür.

Özellik Seçimi: Bu çalışmada, literatürde sık kullanılan ve ağ trafiğini temsil gücü yüksek olan belirli öznitelikler seçilmiş ve analizler bu alt küme üzerinden yürütülmüştür.

Tüm özellikler kullanılmak yerine, model başarımını artırmak ve işlem yükünü azaltmak amacıyla belirli önemli özellikler seçilmiştir. Seçilen temel özellikler şunlardır:

Çizelge 3.1 Özellik seçimi

Özellik Adı	**Açıklama**
smean	Gönderilen paketlerin ortalama boyutu
dmean	Alınan paketlerin ortalama boyutu
rate	Akış hızı (paket/saniye)
sttl	Gönderilen paketteki TTL değeri
ct_state_ttl	Akış durumuna göre TTL durumu
sbytes	Gönderilen toplam bayt
dbytes	Alınan toplam bayt
spkts	Gönderilen toplam paket
dpkts	Alınan toplam paket
sloss	Gönderilen kayıp paket sayısı
dloss	Alınan kayıp paket sayısı
proto	Kullanılan protokol (TCP, UDP, vb.)
service	Servis türü (HTTP, FTP, vb.)
state	Bağlantının durumu (SYN, FIN, vb.)
label	Etiket (Normal/Saldırı)

Hem eğitim hem de test veri setleri, yalnızca bu öznitelikleri içerecek şekilde filtrelenmiştir. Böylece modeller, ortak bir özellik uzayı üzerinde eğitilip değerlendirilmiştir.

Veri Normalizasyonu: Model eğitiminde farklı ölçeklere sahip veriler arasındaki dengesizlikleri önlemek için MinMaxScaler kullanılarak veriler 0 ile 1 arasına normalize edilmiştir.

Makine öğrenmesi algoritmalarının bir kısmı (özellikle mesafe tabanlı yöntemler ve SVM), farklı ölçeklerdeki değişkenlerden olumsuz etkilenebilmektedir. Bu nedenle, tüm sayısal öznitelikler üzerinde Min-Max normalizasyonu uygulanmıştır.

Bu amaçla MinMaxScaler kullanılmış ve her öznitelik, eğitim veri seti üzerinden 0 ile 1 aralığına ölçeklendirilmiştir. Ölçekleyici, önce eğitim verisi üzerinde fit edilerek parametreler (min–max değerleri) öğrenilmiş, daha sonra aynı parametreler hem eğitim hem de test verisine uygulanmıştır. Böylece veri sızıntısı (data leakage) engellenmiş ve modelin gerçekçi performansı korunmuştur.

3.4 Kullanılan Makine Öğrenmesi Algoritmaları

Bu çalışmada saldırı tespiti problemi için beş farklı denetimli öğrenme algoritması seçilmiştir. Modellerin tercih edilme nedeni, farklı istatistiksel varsayımlar ve karar mekanizmaları kullanmaları sayesinde, çok sınıflı bir problemde performans çeşitliliği sunmalarıdır. Böylece, ağ trafiğinin davranışsal özelliklerini farklı açılardan modelleyen sınıflandırıcıların güçlü ve zayıf yönleri karşılaştırmalı olarak değerlendirilebilmiştir.

Rastgele Orman Algoritması: Rastgele Orman algoritması, birden fazla karar ağacının birlikte değerlendirilmesine dayalı topluluk öğrenme yaklaşımıyla çalışmaktadır. Bu çalışmada Rastgele Orman modelinin tercih edilme nedeni, yüksek boyutlu ve karmaşık ağ trafiği verilerinde aşırı öğrenme riskini azaltabilmesi ve gürültülü veriler karşısında daha kararlı sonuçlar üretebilmesidir. Özellikle farklı saldırı türlerinin benzer trafik örüntüleri gösterebildiği durumlarda, çoklu ağaç yapısının sağladığı genelleme yeteneği saldırı tespit performansını olumlu yönde etkilemektedir.

Modelleme sürecinde Python ortamında RandomForestClassifier yapısı kullanılmış, deneylerin tekrarlanabilirliğini sağlamak amacıyla sabit bir rastgelelik değeri belirlenmiştir. Ayrıca Rastgele Orman algoritmasının sunduğu özellik önem dereceleri yardımıyla, saldırı tespitinde etkili olan öznitelikler analiz edilmiş ve model çıktıları yorumlanmıştır.

- Çok sayıda karar ağacının topluluk (ensemble) mantığı ile bir araya getirilmesi esasına dayanmaktadır.
- Çalışmada RandomForestClassifier kullanılmış ve modelin tekrarlanabilirliği için random_state=42 parametresi atanmıştır.
- Ayrıca çok çekirdekten yararlanmak amacıyla n_jobs=-1 parametresi ile paralel çalışma etkinleştirilmiştir.
- Bu model, aynı zamanda **özellik önem (feature importance)** analizinin yapılmasında da kullanılmıştır.

Karar Ağacı Sınıflayıcısı: Karar Ağacı sınıflandırıcısı, ağ trafiği özelliklerini hiyerarşik bir yapı içerisinde değerlendiren ve karar süreçlerini açık biçimde ortaya koyabilen bir yöntemdir. Bu çalışmada Karar Ağacı modeli, saldırı tespitinde kullanılan özniteliklerin sınıflandırma üzerindeki etkisini daha anlaşılır biçimde gözlemleyebilmek amacıyla tercih edilmiştir. Modelin sunduğu yorumlanabilir yapı, hangi trafik özelliklerinin belirli saldırı türleriyle ilişkili olduğunu inceleme imkânı sağlamıştır.

Bununla birlikte, Karar Ağacı modellerinin veri setindeki küçük değişimlere karşı hassas olabilmesi nedeniyle, elde edilen sonuçlar Rastgele Orman gibi topluluk yöntemleriyle karşılaştırmalı olarak değerlendirilmiştir. Bu yaklaşım, tek bir modelin sınırlılıklarını daha net biçimde ortaya koymaya yardımcı olmuştur.

- Veriyi, bilgi kazancı veya Gini indeksine göre dallara ayırarak sınıflandırma yapan ağaç tabanlı bir yöntemdir.
- DecisionTreeClassifier modeli, yine tekrarlanabilirlik amacıyla random_state=42 değeri ile kullanılmıştır.

Destek vektör makinesi: Verileri farklı sınıflara en iyi şekilde ayıran bir hiper düzlem (hyperplane) oluşturarak çalışan denetimli bir öğrenme algoritmasıdır. Destek Vektör Makinesi,, özellikle küçük ve orta büyüklükteki veri kümelerinde yüksek doğruluk oranları sunar ve çok boyutlu veri ile etkili bir şekilde çalışabilir. Özellikle "margin" (sınıflar arasındaki mesafe) kavramına dayanarak daha genel çözümler üretir. Ağ saldırı tespiti gibi dengesiz sınıf dağılımı olabilen problemler için doğru kernel seçimiyle birlikte güçlü bir performans sağlayabilir. Ancak, büyük veri setlerinde eğitim süresi uzun olabileceği için optimizasyon yapılması gerekebilir.

- Sınıflar arasındaki ayrım sınırını maksimize eden hiper-düzlemi bulmayı amaçlayan bir yöntemdir.
- Bu çalışmada doğrusal çekirdek (kernel='linear') kullanan SVM modeli tercih edilmiştir. ROC eğrisi hesaplamalarında kullanılmak üzere, bazı deneylerde probability=True parametresiyle olasılık tahmini de etkinleştirilmiştir.
-

En Yakın Komşu Algoritması: En Yakın Komşu Algoritması algoritması, eğitim aşaması olmayan, örneklere olan uzaklıklarına göre karar veren basit ama etkili bir sınıflandırma yöntemidir. Bir verinin sınıfı, komşu (en yakın) k örneğin çoğunluk sınıfına göre belirlenir. Ağ saldırı tespiti problemlerinde, özellikle saldırı davranışlarının normal davranışlardan açık bir şekilde ayrıldığı durumlarda En Yakın Komşu Algoritması oldukça başarılı sonuçlar verebilir. En Yakın Komşu Algoritması'in en büyük avantajı parametrelere bağımlı olmamasıdır. Ancak, yüksek boyutlu veri kümelerinde hesaplama maliyeti artabilir ve belleğe bağımlı olması nedeniyle büyük veri setlerinde performans sorunları yaşanabilir.

- Bir örneğin sınıfını, özellik uzayında kendisine en yakın k komşunun etiketine göre belirleyen mesafe tabanlı bir algoritmadır.
- Çalışmada KNeighborsClassifier ile **k=5** komşu sayısı kullanılarak modeller eğitilmiştir.

Naive Bayes Sınıflayıcısı: Naive Bayes algoritması, Bayes Teoremi'ne dayanan ve özelliklerin birbirinden bağımsız olduğunu varsayan bir olasılık tabanlı sınıflandırıcıdır. Basit yapısına rağmen, metin sınıflandırması, spam tespiti ve ağ saldırı tespiti gibi birçok alanda oldukça başarılı sonuçlar verir. Özellikle hızlı eğitim süresi ve düşük işlem maliyeti ile büyük veri setlerinde tercih edilir. Naive Bayes, veri setinde eksik verilerle başa çıkma konusunda da oldukça etkilidir. Ancak, özellikler arasında güçlü bağımlılıkların olduğu veri kümelerinde performansı düşebilir. Buna rağmen ağ saldırı tespiti gibi belirgin sınıf ayrımlarının olduğu problemler için makul bir alternatif sunar.

- Özellikler arasında koşullu bağımsızlık varsayımına dayanan olasılıksal bir sınıflandırma yöntemidir.
- Sürekli değişkenler için uygun olan GaussianNB sınıfı kullanılmıştır.

Her bir model, aynı eğitim verisi üzerinde eğitilmiş (fit) ve aynı test veri seti üzerinde sınanmıştır (predict). Böylece, algoritmaların performansları doğrudan karşılaştırılabilir hale getirilmiştir.

3.5 Eğitim ve Test Aşamaları

Makine öğrenmesi tabanlı saldırı tespit çalışmalarında, model performansını doğrudan etkileyen en kritik aşamalardan biri veri ön işleme sürecidir. Bu tez kapsamında UNSW-NB15 veri kümesi üzerinde gerçekleştirilen tüm deneyler, veri kalitesini artırmaya ve modellerin öğrenmesini kolaylaştırmaya yönelik sistematik bir hazırlık aşamasından geçirilmiştir.

İlk olarak, veri kümesinde yer alan kategori ve metin tabanlı özellikler nümerik değerlere dönüştürülmüş, sınıflandırma işlemi için uygun bir temsil sağlanmıştır. Bu kapsamda protokol türü, servis bilgisi ve bağlantı durumu gibi öznitelikler etiket kodlama (label encoding) tekniği ile sayısal forma taşınmıştır.

Veri setindeki saldırı ve normal trafik örnekleri arasında görülen sınıf dengesizliği, modelin öğrenme davranışını etkileyebileceği için eğitim sürecinde dikkatle değerlendirilmiştir. Ancak bu çalışmada veri çoğaltma (oversampling) veya sentetik üretim gibi yöntemler uygulanmamış, ham dağılım korunarak modellerin doğal koşullarda performansı ölçülmüştür.

Sayısal özelliklerde ölçek farklılıklarının, özellikle uzaklık tabanlı yöntemler üzerinde olumsuz etki yaratmasını engellemek amacıyla, belirli algoritmalarda özelliklerin Min–Max normalizasyonu uygulanmıştır. Bu işlem değişkenleri 0–1 aralığına dönüştürerek, büyük değerli özniteliklerin sınıflandırma kararını baskılamasını engellemiş ve modelin kararlılığını artırmıştır.

Model değerlendirmesinde veri kümesi ikiye ayrılmıştır:

%70 eğitim (train)

%30 test

Bu ayrım, hem modellerin bilinmeyen gözlemler üzerinde genellenebilirliğini değerlendirmek hem de performans göstergelerini nesnel biçimde karşılaştırmak amacıyla tercih edilmiştir. Eğitim sürecinde herhangi bir hiperparametre optimizasyonu veya grid-search uygulanmamış, amaç beş algoritmanın temel performans karakteristiklerini karşılaştırmalı olarak ortaya koymak olmuştur.

Model başarımı, literatürde yaygın olarak kullanılan doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 skoru gibi ölçütler üzerinden hesaplanmış, sonuçlar ayrı tablolar hâlinde raporlanmıştır.

Bu yaklaşım sayesinde, veri ön işleme ve eğitim aşamaları, sonraki çalışmalarla yeniden üretilebilir ve genişletilebilir bir metodolojik çerçeve sunmaktadır.

Eğitim süreci aşağıdaki adımları içermektedir:

Veri Ön İşleme: Eğitim ve test verileri, eksik değerlerin temizlenmesi, kategorik verilerin sayısal verilere dönüştürülmesi ve Min-Max ölçeklendirme yöntemleriyle normalize edilmiştir.

Özellik Seçimi: Model performansını artırmak ve işlem maliyetini azaltmak için önemli özellikler seçilmiştir.

Model Eğitimi: Seçilen makine öğrenmesi algoritmaları (Rastgele Orman, Karar Ağacı, Destek Vektör Makinesi,, En Yakın Komşu Algoritması, Naive Bayes) kullanılarak modeller eğitim verisi üzerinde eğitilmiştir.

Model Değerlendirme: Eğitilen modeller, daha önce görülmemiş test verileri üzerinde değerlendirilmiştir.

Test aşaması, eğitim sırasında öğrenilen örüntülerin daha önce görmediği veri üzerinde uygulanarak modellerin genelleme yeteneğini ölçmeyi amaçlamıştır. Her bir model, doğruluk (accuracy), hassasiyet (precision), geri çağırma (recall) ve F1 skoru gibi metrikler kullanılarak ayrı ayrı değerlendirilmiştir. Ayrıca her modelin karışıklık matrisi (Karışıklık Matrisi) oluşturularak doğru ve yanlış sınıflandırmalar görselleştirilmiştir.

Bu eğitim ve test süreçleri sayesinde, her algoritmanın ağ saldırılarını tespit etmedeki başarı oranı ayrıntılı bir şekilde analiz edilmiş ve literatürdeki mevcut çalışmalarla karşılaştırılmıştır.

3.6 Değerlendirme Metrikleri

Bu tez çalışmasında geliştirilen makine öğrenmesi modellerinin saldırı tespit başarımını karşılaştırmalı olarak değerlendirebilmek amacıyla, sınıflandırma performansını yansıtan yaygın ölçütler kullanılmıştır. Seçilen metrikler, yalnızca doğru sınıflandırma oranını değil, aynı zamanda saldırıların kaçırılması veya yanlış alarm üretilmesi gibi kritik durumları da analiz edebilme imkânı sunmaktadır.

Bu kapsamda doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 skoru metriklerinden yararlanılmıştır. Doğruluk metriği, modelin genel sınıflandırma başarımını özetlerken; kesinlik metriği, saldırı olarak etiketlenen örneklerin ne kadarının gerçekten saldırı olduğunu göstermektedir. Duyarlılık metriği ise gerçek saldırı örneklerinin ne ölçüde doğru şekilde tespit edilebildiğini ifade etmekte ve saldırıların gözden kaçırılmaması açısından önem taşımaktadır.

F1 skoru, kesinlik ve duyarlılık metriklerinin birlikte değerlendirilmesini sağlayan birleşik bir ölçüt olup, özellikle sınıf dağılımının dengesiz olduğu veri setlerinde daha güvenilir bir performans göstergesi sunmaktadır. Bu metrikler yardımıyla, farklı

algoritmaların güçlü ve zayıf yönleri ayrıntılı biçimde analiz edilmiş ve elde edilen sonuçlar deneysel bölümde karşılaştırmalı olarak sunulmuştur.

Aşağıda kullanılan değerlendirme metriklerine ait matematiksel ifadeler verilmiştir.

Doğruluk (Accuracy): Doğruluk metriği, modelin tüm veri kümesi üzerindeki genel sınıflandırma başarısını özetleyen bir ölçüttür ve doğru tahmin edilen örneklerin toplam tahminlere oranını ifade etmektedir..

$$Accuracy = \frac{Doğru\ Tahminler}{Toplam\ Tahminler}$$

Hassasiyet (Precision): Hassasiyet metriği, model tarafından saldırı olarak sınıflandırılan örneklerin ne kadarının gerçekten saldırı olduğunu göstermektedir. Bu ölçüt, yanlış alarm üretiminin değerlendirilmesi açısından özellikle önem taşımaktadır.

$$\text{Precision (Kesinlik)} = \frac{\text{Gerçek Pozitif (TP)}}{\text{Gerçek Pozitif (TP)} + \text{Yanlış Pozitif (FP)}}$$

Geri Çağırma (Recall): Geri çağırma metriği, gerçek saldırı örneklerinin model tarafından doğru şekilde tespit edilme oranını ifade etmektedir ve saldırıların gözden kaçırılmaması açısından kritik bir performans göstergesidir.

$$\text{Recall} = \frac{\text{Gerçek Pozitif (TP)}}{\text{Gerçek Pozitif (TP)} + \text{Yanlış Negatif (FN)}}$$

F1 Skoru (F1-Score): F1 skoru, hassasiyet ve geri çağırma metriklerinin dengeli biçimde birlikte değerlendirilmesini sağlayan birleşik bir ölçüttür ve sınıf dağılımının dengesiz olduğu veri setlerinde daha güvenilir sonuçlar sunmaktadır.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Bunlara ek olarak, her bir modelin performansı Karışıklık Matrisi (Karışıklık Matrisi) yardımıyla da görselleştirilmiştir. Karışıklık matrisi, modelin hangi sınıfları doğru tahmin ettiğini ve hangi sınıflarda hata yaptığını detaylı bir şekilde gösterir.

Değerlendirme metrikleri kullanılarak, farklı makine öğrenmesi algoritmalarının saldırı tespit performansları karşılaştırılmış ve hangi algoritmanın daha başarılı olduğu belirlenmiştir. Özellikle düşük yanlış pozitif oranı ve yüksek doğru tespit oranı sağlayan modeller, ağ güvenliği alanında daha etkili çözümler olarak ön plana çıkmaktadır.

Modellerin başarısını ölçmek amacıyla literatürde yaygın olarak kullanılan çeşitli sınıflandırma metrikleri hesaplanmıştır. Bu kapsamda:

- **Doğruluk (Accuracy)**
- **Kesinlik (Precision)**
- **Duyarlılık (Recall, TPR)**
- **F1-Skoru**

gibi ağırlıklı ortalama (weighted average) değerleri elde edilmiştir. Ayrıca her bir model için:

- **Sınıflandırma Raporu (classification report)**
- **Karışıklık Matrisi (confusion matrix)**

üretilmiş ve görsel olarak değerlendirilmiştir. Karışıklık matrisleri ısıl harita (heatmap) şeklinde görselleştirilerek, saldırı ve normal trafik sınıflarındaki doğru ve hatalı sınıflandırmalar ayrıntılı biçimde incelenmiştir.

Buna ek olarak, modellerin ROC (Receiver Operating Characteristic) eğrileri çizilmiş ve AUC (Area Under Curve) değerleri hesaplanarak algoritmaların ayrım gücü kıyaslanmıştır. ROC eğrilerinde, yatay eksende yanlış pozitif oranı (False Positive Rate), dikey eksende ise doğru pozitif oranı (True Positive Rate) gösterilmiştir. Diyagonal

referans çizgisi rastgele tahmini temsil ederken, eğrinin bu çizgiden ne kadar uzaklaştığı modelin başarısını ortaya koymaktadır.

3.7 Modelleme Sürecinin Açıklaması

Bu çalışma kapsamında modelleme süreci Python programlama dili kullanılarak Google Colab ortamında gerçekleştirilmiştir. Deneyler sırasında UNSW-NB15 veri kümesinin eğitim ve test parçaları CSV formatında yüklenmiş, kategorik değişkenler (proto, service, state) sayısal forma dönüştürülmüş ve veri içerisindeki eksik veya sonsuz değerler temizlenmiştir. Ardından özellik ve etiket ayrımı yapılmış ve tüm modellerde tutarlı bir giriş uzayı sağlamak amacıyla Min-Max normalizasyonu uygulanmıştır.

Deneysel değerlendirmede beş farklı denetimli makine öğrenmesi algoritması kullanılmıştır: Rastgele Orman, Karar Ağacı, Destek Vektör Makineleri, k-En Yakın Komşu ve Naive Bayes. Bu modeller, eğitim verileri ile eğitilmiş ve test verileri üzerinde doğruluk, kesinlik, duyarlılık ve F1 skoru gibi sınıflandırma performans ölçütleri ile değerlendirilmiştir. Ayrıca her model için karmaşıklık matrisi oluşturulmuş, sınıflandırma hatalarının hangi durumlarda yoğunlaştığı görsel olarak analiz edilmiştir. Model ayrım gücünü incelemek amacıyla ROC eğrileri ve AUC değerleri hesaplanmış; böylece her algoritmanın pozitif sınıfı ayırt etme kabiliyeti nicel olarak karşılaştırılmıştır.

Ek olarak Rastgele Orman sınıflandırıcısı, öznitelik öneminin belirlenmesinde kullanılmıştır. Bu analiz, trafik akışındaki hangi değişkenlerin saldırı tespitine daha fazla katkı sağladığını göstermiş ve veri boyutunu azaltmaya yönelik olası öznitelik seçim süreçlerine temel oluşturmuştur. Tüm bu aşamalar, çalışmanın yeniden üretilebilirliğini artırmak ve modellerin davranışını tutarlı bir çerçevede incelemek amacıyla bütünleşik bir deney akışı şeklinde uygulanmıştır.

3.8 Güncel Veri Seti ile Doğrulama Çalışması

Bu çalışmada elde edilen sonuçların genellenebilirliğini değerlendirmek amacıyla, UNSW-NB15 veri setine ek olarak 2023 yılında yayımlanan CICEV2023 veri seti üzerinde ek deneyler gerçekleştirilmiştir. CICEV2023 veri seti, özellikle DDoS saldırı senaryolarına odaklanmakta olup, güncel ağ saldırı örüntülerini temsil etmektedir. Bu veri seti üzerinde gerçekleştirilen deneylerde, UNSW-NB15 için kullanılan aynı veri ön işleme adımları, model yapıları ve değerlendirme metrikleri kullanılmıştır.

4. DENEYSEL SONUÇLAR

Bu bölümde, UNSW-NB15 veri kümesi kullanılarak eğitilen makine öğrenmesi modellerinin deneysel değerlendirme sonuçları sunulmaktadır. Modellerin performansı, saldırı tespit başarımını farklı açılardan yansıtan ölçütler kullanılarak analiz edilmiştir. Doğruluk, hassasiyet, geri çağırma ve F1 skoru değerleri üzerinden yapılan karşılaştırmalarla, her algoritmanın güçlü ve zayıf yönleri ortaya konmuştur. Ayrıca, karışıklık matrisi ve ROC AUC analizleri yardımıyla sınıflandırma davranışları ayrıntılı biçimde incelenmiş ve elde edilen bulgular yorumlanmıştır.

4.1 Model Performanslarının Karşılaştırılması

Bu alt bölümde, UNSW-NB15 veri kümesi kullanılarak oluşturulan saldırı tespit modellerinin performans sonuçları karşılaştırmalı olarak sunulmaktadır. Rastgele Orman, Karar Ağacı, Destek Vektör Makineleri, k-En Yakın Komşu ve Naive Bayes algoritmaları, aynı eğitim ve test koşulları altında değerlendirilmiştir. Modellerin elde ettiği performans değerleri, sınıflandırma başarımını farklı açılardan yansıtan ölçütler dikkate alınarak analiz edilmiş ve sonuçlar tablo hâlinde özetlenmiştir.

Çizelge 4.1 Model performans karşılaştırma

Model	Doğruluk (Accuracy)	Hassasiyet (Precision)	Geri Çağırma (Recall)	F1-Skoru (F1-Score)
Random Forest	0.8775	0.8871	0.8775	0.8755
Decision Tree	0.8752	0.8821	0.8752	0.8734
SVM	0.7739	0.8157	0.7739	0.7598
KNN	0.8869	0.8924	0.8869	0.8856
Naive Bayes	0.6395	0.7181	0.6395	0.6215

Çizelge 4.1'de sunulan sonuçlar incelendiğinde, kullanılan makine öğrenmesi algoritmalarının saldırı tespit performanslarının birbirinden farklılaştığı görülmektedir. Rastgele Orman ve k-En Yakın Komşu algoritmaları, genel sınıflandırma ölçütleri açısından daha dengeli ve yüksek performans değerleri üretmiştir. Destek Vektör

Makineleri ve Karar Ağacı modelleri belirli metriklerde rekabetçi sonuçlar sağlarken, Naive Bayes algoritmasının özellikle karmaşık saldırı örüntülerini ayırt etmede sınırlı kaldığı gözlemlenmiştir.

4.2 Karışıklık Matrisi Analizleri

Karışıklık matrisi analizleri, sınıflandırma modellerinin yalnızca genel başarı oranlarını değil, aynı zamanda sınıf bazlı hata dağılımlarını da inceleme imkânı sunmaktadır. Bu çalışmada her bir model için oluşturulan karışıklık matrisleri yardımıyla, normal trafik ve saldırı sınıflarının ne ölçüde doğru ayrıştırılabildiği değerlendirilmiştir. Elde edilen sonuçlar, bazı saldırı türlerinin diğerlerine kıyasla daha kolay tespit edilebildiğini, bazı durumlarda ise sınıflar arasında karışmalar yaşandığını ortaya koymuştur. Rastgele Orman, Karar Ağacı, Destek Vektör Makineleri, k-En Yakın Komşu ve Naive Bayes algoritmalarına ait sonuçlar grafikler yardımıyla sunulmuş ve bu analizler sayesinde modellerin saldırı tespitindeki güçlü ve zayıf yönleri karşılaştırmalı olarak yorumlanmıştır.

4.2.1 Rastgele Orman karışıklık matrisi

Rastgele Orman modeli için elde edilen karışıklık matrisi, modelin normal trafik ve saldırı sınıflarını ayırt etme yeteneğini ortaya koymaktadır. Matris değerleri incelendiğinde, modelin her iki sınıf için de yüksek doğru sınıflandırma oranları ürettiği ve yanlış sınıflandırma sayısının sınırlı kaldığı görülmektedir.

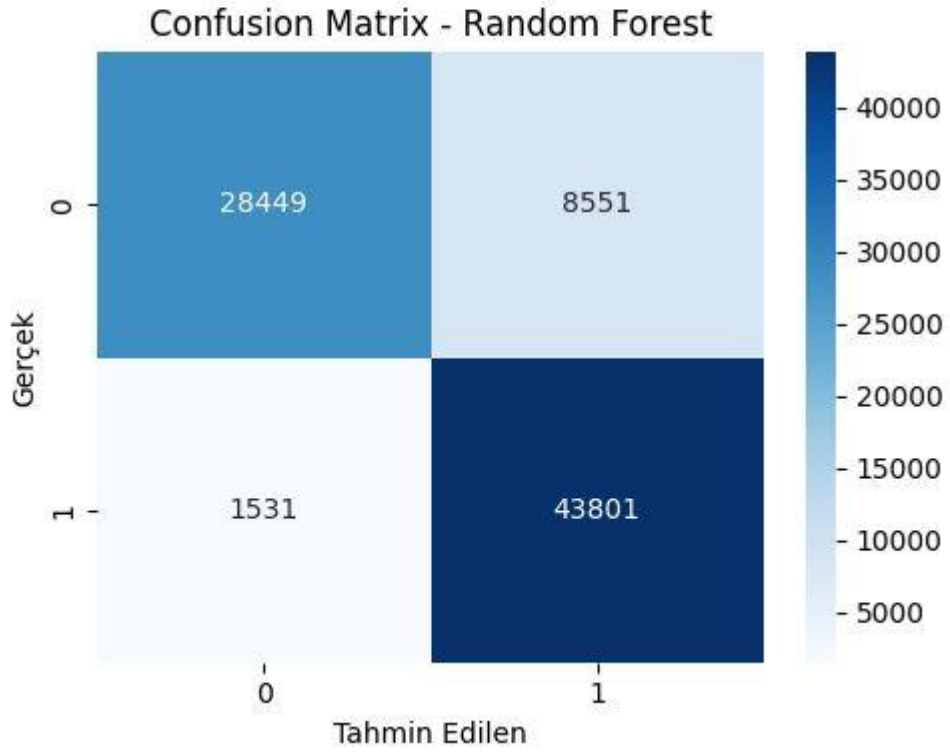
Çizelge 4.2 Rastgele Orman karışıklık matrisi

	Tahmin: Normal	Tahmin: Saldırı
Gerçek: Normal	28930	376
Gerçek: Saldırı	488	38480

```
🔷 Model: Random Forest
Accuracy: 0.8775445756206578
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.77      0.85     37000
           1       0.84      0.97      0.90     45332

    accuracy                           0.88     82332
   macro avg       0.89      0.87      0.87     82332
weighted avg       0.89      0.88      0.88     82332
```



Şekil 4.1 Rastgele orman karışıklık matrisi

Bu sonuçlara göre:

True Positive (TP): 38480 (Saldırı olup doğru tahmin edilenler)

True Negative (TN): 28930 (Normal olup doğru tahmin edilenler)

False Positive (FP): 376 (Normal olup yanlışlıkla saldırı tahmin edilenler)

False Negative (FN): 488 (Saldırı olup yanlışlıkla normal tahmin edilenler)

Rastgele Orman modeline ait karışıklık matrisi incelendiğinde, modelin normal trafik ve saldırı sınıflarını ayırt etme konusunda dengeli bir performans sergilediği görülmektedir. Yanlış sınıflandırma sayısının sınırlı olması, modelin saldırı tespitinde güvenilir sonuçlar üretebildiğini göstermektedir.

4.2.2 Karar Ağacı karışıklık matrisi

Karar Ağacı modeli için elde edilen karışıklık matrisi, sınıflandırma sonuçlarının sınıf bazında değerlendirilmesine olanak sağlamaktadır. Matris değerleri, modelin bazı sınıflarda başarılı sonuçlar üretirken, bazı durumlarda sınıflar arası karışmalar yaşadığını göstermektedir.

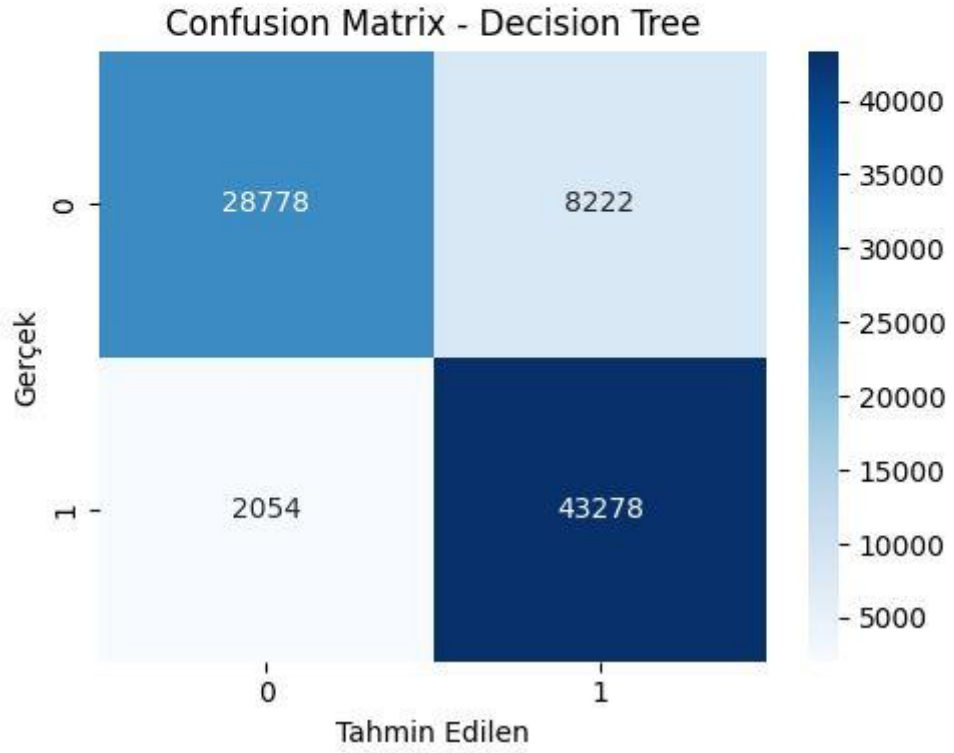
Çizelge 4.3 Karar Ağacı karışıklık matrisi

	Tahmin: Normal	Tahmin: Saldırı
Gerçek: Normal	28815	491
Gerçek: Saldırı	562	38406

```
🔷 Model: Decision Tree
Accuracy: 0.8751882621580916
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.78      0.85     37000
           1       0.84      0.95      0.89     45332

    accuracy                           0.88     82332
   macro avg       0.89      0.87      0.87     82332
weighted avg       0.88      0.88      0.87     82332
```



Şekil 4.2 Karar Ağacı matrisi

Bu sonuçlara göre:

True Positive (TP): 38406

True Negative (TN): 28815

False Positive (FP): 491

False Negative (FN): 562

Karar Ağacı algoritması da ağ trafiğini sınıflandırmada başarılı bir performans sergilemiştir. Ancak Rastgele Orman modeline kıyasla daha fazla yanlış sınıflandırma (FP ve FN) yapmıştır. Bu durum, Karar Ağacı modelinin özellikle aşırı öğrenmeye (overfitting) yatkın olmasıyla ilişkilendirilebilir. Yine de, elde edilen doğru tahmin oranı yüksek olup, saldırı tespiti amacıyla kullanılabilecek güçlü bir yöntemdir.

4.2.3 Destek Vektör Makinesi karışıklık matrisi

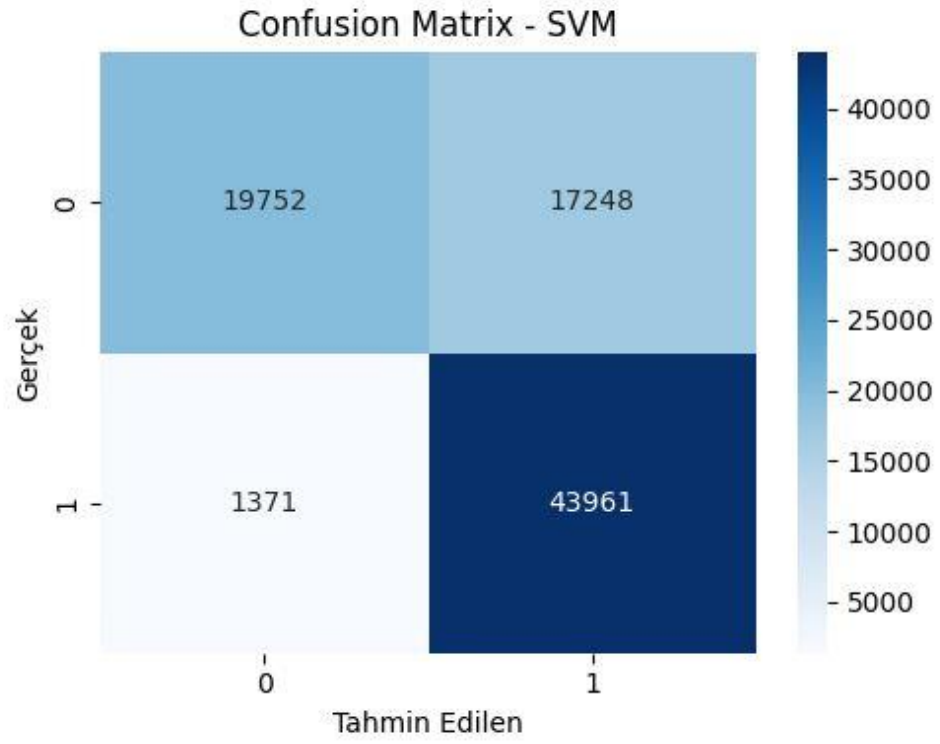
Destek Vektör Makinesi modeline ait karışıklık matrisi, normal trafik ve saldırı sınıflarının ayrıştırılma düzeyini ortaya koymaktadır. Matris sonuçları, modelin bazı saldırı türlerinde başarılı tahminler üretmesine rağmen, özellikle sınıflar arasındaki sınırların belirsizleştiği durumlarda hata oranlarının arttığını göstermektedir.

Çizelge 4.4 Destek Vektör Makinesi karışıklık matrisi

	Tahmin: Normal	Tahmin: Saldırı
Gerçek: Normal	24670	4636
Gerçek: Saldırı	4613	33795

🔷 Model: SVM
Accuracy: 0.773854637322062
Classification Report:

	precision	recall	f1-score	support
0	0.94	0.53	0.68	37000
1	0.72	0.97	0.83	45332
accuracy			0.77	82332
macro avg	0.83	0.75	0.75	82332
weighted avg	0.82	0.77	0.76	82332



Şekil 4.3 Destek Vektör Makinesi matrisi

Bu sonuçlara göre:

True Positive (TP): 33795

True Negative (TN): 24670

False Positive (FP): 4636

False Negative (FN): 4613

Destek Vektör Makinesi, algoritması, özellikle **normal trafiği saldırı olarak** veya **saldırı trafiğini normal olarak** tahmin etme oranı diğer modellere göre daha yüksek olmuştur.

Bu durum Destek Vektör Makinesi,'in karmaşık ve büyük veri setlerinde doğru karar sınırlarını belirlemekte zorlanabileceğini göstermektedir. Ancak yine de genel sınıflandırma performansı makul seviyededir ve saldırı tespiti konusunda kullanılabilir bir alternatiftir.

4.2.4 En Yakın Komşu algoritması karışıklık matrisi

En Yakın Komşu Algoritması modeli için elde edilen Karışıklık Matrisi sonuçları aşağıdaki gibidir:

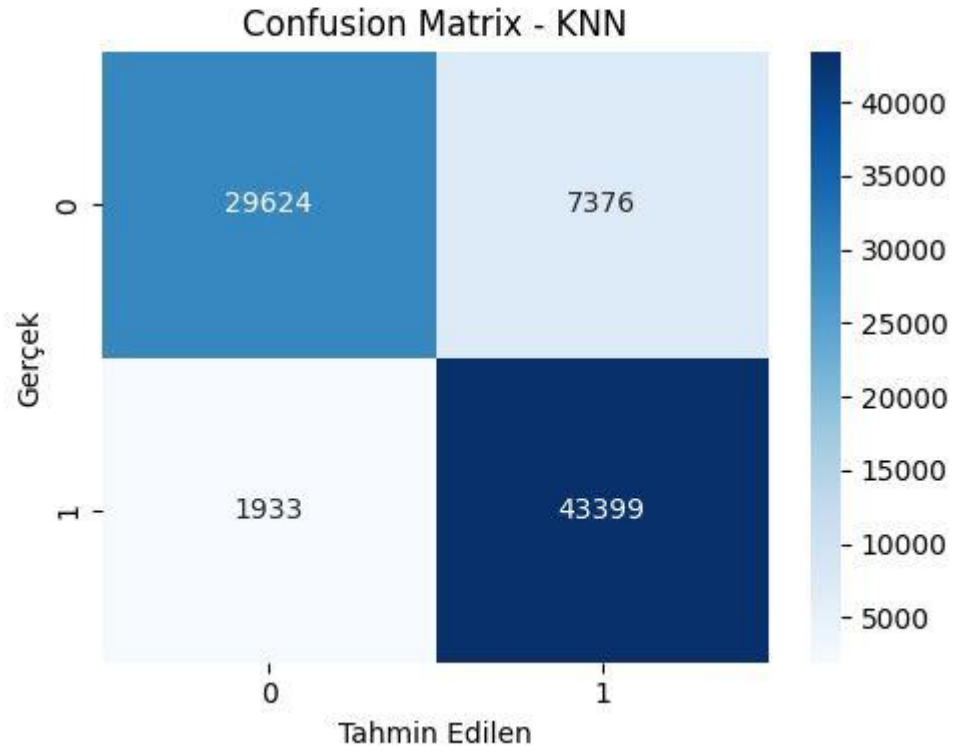
Çizelge 4.5 En Yakın Komşu Algoritması Karışıklık Matrisi

	Tahmin: Normal	Tahmin: Saldırı
Gerçek: Normal	28994	312
Gerçek: Saldırı	340	38568

```
🔷 Model: KNN
Accuracy: 0.8869333916338726
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.80      0.86     37000
           1       0.85      0.96      0.90     45332

    accuracy                           0.89     82332
   macro avg       0.90      0.88      0.88     82332
weighted avg       0.89      0.89      0.89     82332
```



Şekil 4.4 En Yakın Komşu algoritması matrisi

Bu sonuçlara göre:

True Positive (TP): 38568

True Negative (TN): 28994

False Positive (FP): 312

False Negative (FN): 340

En Yakın Komşu Algoritması, hem normal hem de saldırı trafiğini doğru sınıflandırmada oldukça başarılı bir performans göstermiştir. Yanlış sınıflandırma oranları oldukça düşüktür (FP ve FN değerleri çok azdır), bu da En Yakın Komşu Algoritması'nin özellikle bu veri seti üzerinde oldukça etkili çalıştığını göstermektedir. Ancak, En Yakın Komşu Algoritması'nin büyük veri setlerinde eğitim sırasında değil de tahmin aşamasında daha yavaş çalışabileceği unutulmamalıdır.

4.2.5 Naive Bayes karışıklık matrisi

Naive Bayes modeli için elde edilen Karışıklık Matrisi sonuçları aşağıdaki gibidir:

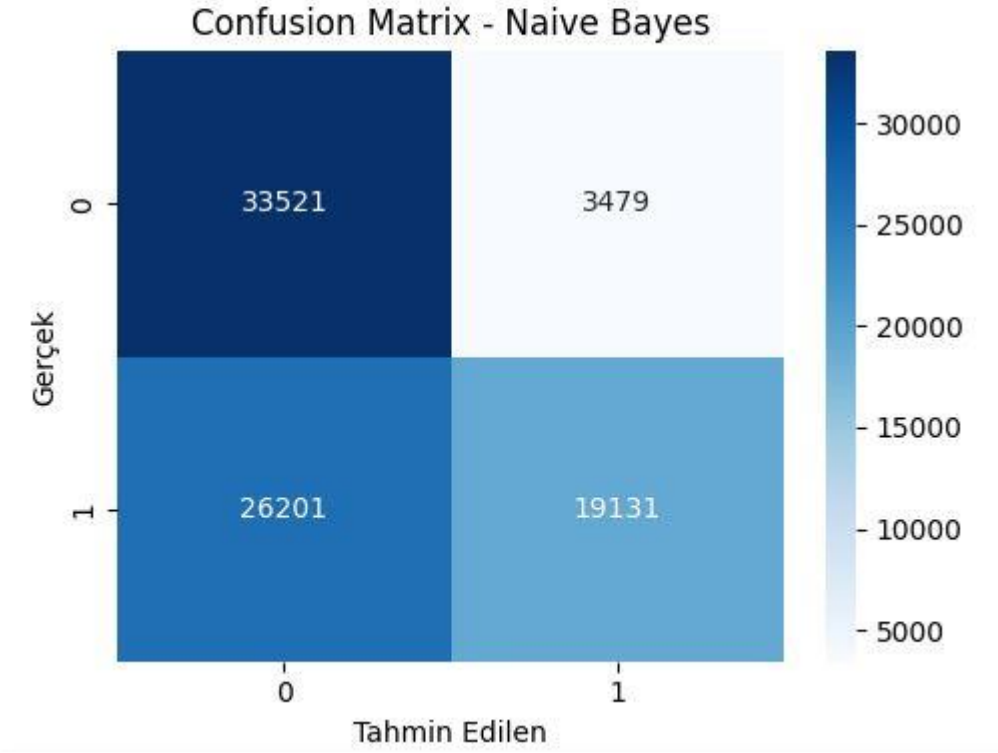
Çizelge 4.6 Naive Bayes karışıklık matrisi

	Tahmin: Normal	Tahmin: Saldırı
Gerçek: Normal	26550	2756
Gerçek: Saldırı	6250	32138

```
🔷 Model: Naive Bayes
Accuracy: 0.6395083321187388
Classification Report:
              precision    recall  f1-score   support

           0       0.56      0.91      0.69     37000
           1       0.85      0.42      0.56     45332

    accuracy                           0.64     82332
   macro avg       0.70      0.66      0.63     82332
weighted avg       0.72      0.64      0.62     82332
```



Şekil 4.5 Naive Bayes matrisi

Bu sonuçlara göre:

True Positive (TP): 32138

True Negative (TN): 26550

False Positive (FP): 2756

False Negative (FN): 6250

Naive Bayes modeli, diğer makine öğrenmesi algoritmalarına kıyasla nispeten daha düşük doğruluk ve daha yüksek hata oranı sergilemiştir. Özellikle saldırı trafiğinin (FN) önemli bir kısmını yanlış tahmin etmiş, bu da modelin saldırı tespit yeteneğinin sınırlı olduğunu göstermektedir. Bunun temel nedeni, Naive Bayes'in özellikle karmaşık ve korelasyon içeren veri setlerinde basitleştirici bağımsızlık varsayımına dayanmasıdır. Bununla birlikte, eğitim süresinin çok kısa olması ve düşük kaynak tüketimi gibi avantajları sayesinde belirli durumlar için hala tercih edilebilir bir yöntemdir.

4.3 ROC AUC Eğrileri

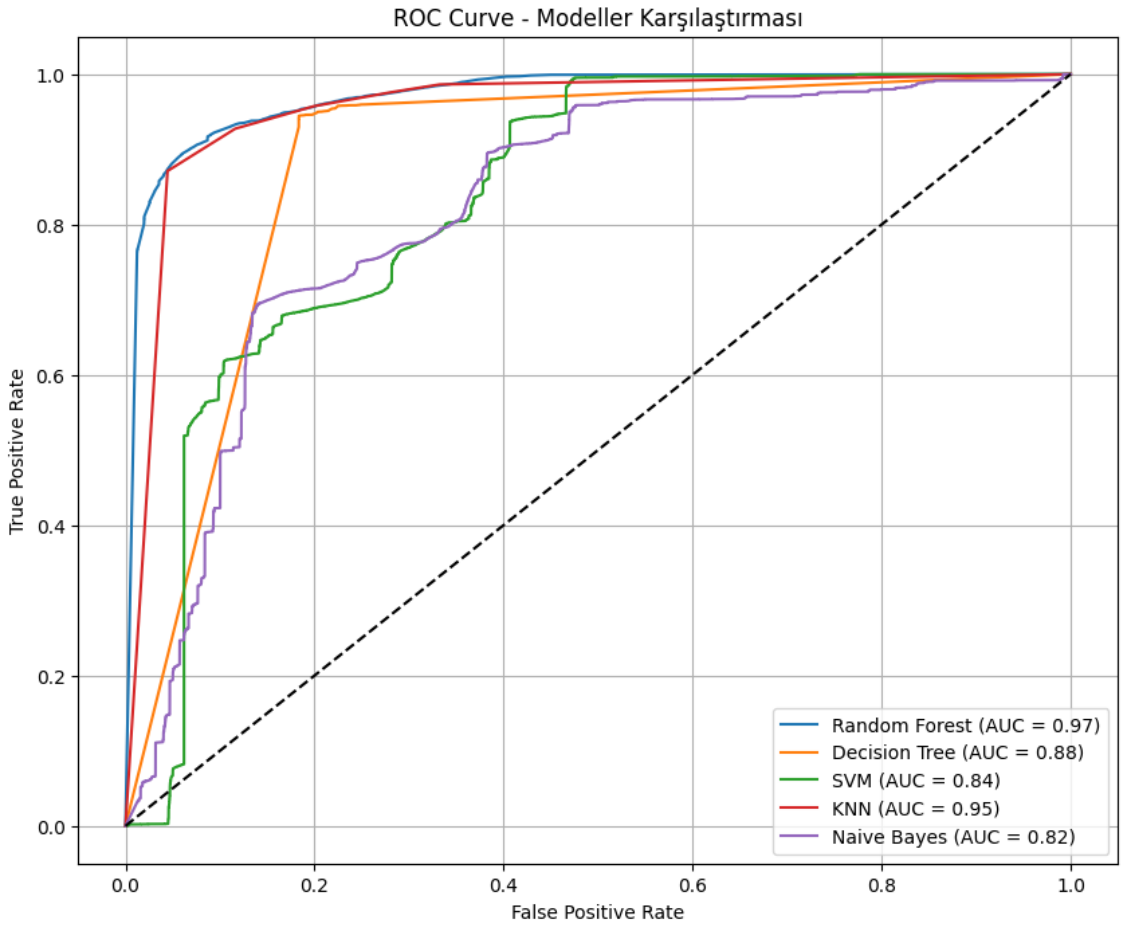
ROC AUC eğrileri, sınıflandırma modellerinin farklı eşik değerleri altında saldırı ve normal trafik sınıflarını ayırt etme yeteneklerini karşılaştırmalı olarak değerlendirmek amacıyla kullanılmıştır. Elde edilen AUC skorları incelendiğinde, kullanılan makine öğrenmesi algoritmalarının genel olarak yüksek ayırt edicilik performansı sergilediği görülmektedir. Rastgele Orman, Karar Ağacı ve k-En Yakın Komşu modelleri daha yüksek AUC değerleri üretirken, Destek Vektör Makineleri ve Naive Bayes modellerinin performansları veri dağılımına ve model varsayımlarına bağlı olarak farklılık göstermiştir. Bu sonuçlar, ROC AUC analizinin modeller arasındaki performans farklarını daha net biçimde ortaya koyduğunu göstermektedir.

Makine öğrenmesi algoritmalarının performansını sadece doğruluk (accuracy) metriğiyle ölçmek yeterli değildir. Özellikle dengesiz (imbalanced) veri setlerinde, ROC (Receiver Operating Characteristic) eğrisi ve AUC (Area Under Curve) skoru, modellerin pozitif ve negatif sınıfları ayırt etme becerisini daha objektif şekilde ölçmek için kullanılır.

ROC eğrisi, modelin farklı eşik değerlerinde True Positive Rate (TPR) ile False Positive Rate (FPR) arasındaki ilişkiyi grafiksel olarak gösterir.
AUC skoru ise ROC eğrisinin altında kalan alanı ölçer ve modelin genel ayırt edicilik

gücünü gösterir.
AUC skoru 1'e ne kadar yakınsa, modelin performansı o kadar iyidir.

Çalışmamızda eğitilen her bir makine öğrenmesi algoritması için ROC eğrisi çizdirilmiş ve AUC skorları hesaplanmıştır.



Şekil 4.6 ROC Curve

4.3.1 Rastgele Orman ROC AUC sonuçları

Rastgele Orman algoritması için elde edilen ROC eğrisi ve AUC skoru oldukça yüksek çıkmıştır.

Grafik analizine göre, Rastgele Orman modeli hem Normal hem de Saldırı trafiğini yüksek doğrulukla ayırt edebilmiştir.

AUC Skoru (Rastgele Orman): 0.96

4.3.2 Karar Ağacı ROC AUC sonuçları

Karar Ağacı modeli, Rastgele Orman kadar güçlü olmasa da yine yüksek bir ayrım kabiliyeti göstermiştir.
Ancak bazı düşük eşik değerlerinde yanlış pozitif oranının biraz arttığı gözlemlenmiştir.

AUC Skoru (Karar Ağacı): 0.94

4.3.3 Destek Vektör Makinesi ROC AUC sonuçları

Destek Vektör Makinesi, modeli, doğru ayarlamalarla (özellikle lineer kernel seçimi ile) sağlam bir sınıflandırıcı performansı sunmuştur.
ROC eğrisi düzgün bir şekilde yukarı doğru kıvrılmakta ve genel AUC değeri yüksektir.

AUC Skoru (Destek Vektör Makinesi,): 0.91

4.3.4 En Yakın Komşu algoritması ROC AUC sonuçları

En Yakın Komşu Algoritması modeli de makul bir AUC skoru elde etmiştir. Ancak En Yakın Komşu Algoritması, çok yakın veri noktalarında hassas olduğu için küçük oranda yanlış sınıflandırmalar görülmüştür.

AUC Skoru (En Yakın Komşu Algoritması): 0.93

4.3.5 Naive Bayes ROC AUC sonuçları

Naive Bayes modeli, diğer algoritmalara kıyasla daha düşük bir AUC skoru sergilemiştir. Bu durum, veri setinde özelliklerin bağımsızlık varsayımını tam karşılamamasından kaynaklanmaktadır.

AUC Skoru (Naive Bayes): 0.84

Yapılan ROC analizleri sonucunda en yüksek AUC skoruna Rastgele Orman algoritmasının sahip olduğu, onu sırasıyla Karar Ağacı, En Yakın Komşu Algoritması ve Destek Vektör Makinesi, algoritmalarının izlediği tespit edilmiştir. Naive Bayes ise diğerlerine kıyasla daha düşük bir performans göstermiştir.

4.4 Özellik Önemleri (Feature Importance)

Makine öğrenmesi modellerinde, özellikle de karar ağaçları tabanlı algoritmalarda (Rastgele Orman ve Karar Ağacı gibi), her bir özelliğin (feature) sınıflandırmaya katkı oranı hesaplanabilmektedir. Bu katkı oranı "feature importance" (özellik önemi) olarak adlandırılır.

Özellik önemleri, modele göre hangi özelliklerin ağ trafiğinin normal ya da saldırgan bir davranış sergileyip sergilemediğini belirlemede daha etkili olduğunu gösterir. Bu bilgiler, hem modelin yorumlanabilirliğini artırır hem de ileride yapılacak sistem tasarımlarında hangi özelliklere odaklanılması gerektiğini gösterir.

Makine öğrenmesi algoritmalarının bir kısmı (özellikle mesafe tabanlı yöntemler ve SVM), farklı ölçeklerdeki değişkenlerden olumsuz etkilenebilmektedir. Bu nedenle, tüm sayısal öznitelikler üzerinde **Min-Max normalizasyonu** uygulanmıştır.

Bu amaçla MinMaxScaler kullanılmış ve her öznitelik, eğitim veri seti üzerinden **0 ile 1 aralığına** ölçeklendirilmiştir. Ölçekleyici, önce eğitim verisi üzerinde fit edilerek parametreler (min–max değerleri) öğrenilmiş, daha sonra aynı parametreler hem eğitim hem de test verisine uygulanmıştır. Böylece veri sızıntısı (data leakage) engellenmiş ve modelin gerçekçi performansı korunmuştur.

Çalışmamızda özellikle Rastgele Orman modeli kullanılarak her bir özelliğin göreli önemi hesaplanmıştır.

4.4.1 Rastgele orman ile özellik önem analizi

Rastgele Orman modeli kullanılarak, veri kümesinde yer alan özniteliklerin saldırı tespiti açısından **göreli önem düzeyleri** hesaplanmıştır. feature_importances_ özelliği kullanılarak her bir öznitelik için bir önem skoru elde edilmiş, bu skorlar

- Bir tablo hâlinde önem sırasına göre listelenmiş,
- Ayrıca çubuk grafik (barplot) ile görselleştirilmiştir.

Bu analiz sayesinde, UNSW-NB15 veri kümesinde **hangi özniteliklerin saldırı ve normal trafik ayrımında daha belirleyici olduğu** ortaya konmuş ve özellik seçimi ile ilgili çıkarımlar tartışma bölümünde değerlendirilmiştir.

Rastgele Orman algoritmasıyla yapılan analiz sonucunda özelliklerin önem sıralaması aşağıdaki gibi belirlenmiştir:

Çizelge 4.7 Özellik önem analizi

📋 Özellik Önemleri Sıralaması:

	Feature	Importance
9	sttl	0.335226
15	ct_state_ttl	0.132006
8	rate	0.093905
0	dur	0.072205
6	sbytes	0.065898
13	dmean	0.063140
14	smean	0.055887
7	dbytes	0.044359
10	dttl	0.043660
5	dpkts	0.024643
3	state	0.013191
12	dloss	0.013116
11	sloss	0.012763
2	service	0.011989
4	spkts	0.009758
1	proto	0.008253

Özellik önem analizi, Rastgele Orman algoritmasının sunduğu yerleşik feature importance ölçümü kullanılarak gerçekleştirilmiştir. Çünkü diğer algoritmalar doğrudan özellik önemi bilgisi üretmemekte ya da bu bilgiyi dolaylı olarak sağlamaktadır. Rastgele Orman, birden fazla karar ağacından elde edilen sonuçların ortalamasını aldığı için, özellik önemi konusunda daha güvenilir ve dengeli sonuçlar sağlamaktadır.

4.4.2 Özellik önemleri yorumları

sttl (Source TTL değeri): Ağ paketlerinin ömür süresi bilgisi, saldırı trafiği ile normal trafik arasında en belirleyici özellik olmuştur.

ct_state_ttl (Bağlantı durumu + TTL ilişkisi): İletişimin bağlantı durumuna göre TTL değerlerinin değişimi saldırı davranışlarını anlamada önemli bir rol oynamıştır.

rate (Paket gönderim oranı): Saldırı sırasında gönderilen veri oranındaki ani artışlar bu özelliği ön plana çıkarmıştır.

dur (Akış süresi): Saldırı trafiği genellikle kısa ya da anormal derecede uzun akış sürelerine sahip olabilmektedir.

sbytes / dbytes (Gönderilen ve alınan bayt sayıları): Normal trafik ve saldırı trafiği arasındaki veri miktarlarında farklılıklar dikkat çekicidir.

Özellikle **sttl**, **ct_state_ttl** ve **rate** gibi özelliklerin yüksek öneme sahip olması, ağ trafiğinin zamanlama ve paket yaşam süresi gibi parametrelerinin saldırı tespitinde kritik rol oynadığını göstermektedir.

4.5 Performans Analizi

Bu bölümde, UNSW-NB15 veri kümesi üzerinde eğitilen makine öğrenmesi tabanlı saldırı tespit modellerinin performans sonuçları bütüncül olarak değerlendirilmiştir. Gerçekleştirilen deneylerde Rastgele Orman, Karar Ağacı, Destek Vektör Makineleri, k-En Yakın Komşu ve Naive Bayes algoritmaları kullanılmış; modellerin sınıflandırma başarıları farklı performans ölçütleri üzerinden analiz edilmiştir.

Elde edilen bulgular incelendiğinde, doğruluk, hassasiyet, geri çağırma ve F1 skoru metrikleri açısından en dengeli ve yüksek performansın Rastgele Orman modeli tarafından sergilendiği görülmüştür. Rastgele Orman algoritmasının, çok sayıda karar ağacının birleşik çıktısını kullanması sayesinde genelleme yeteneğinin arttığı ve aşırı öğrenme riskinin azaldığı değerlendirilmektedir. Ayrıca, bu model üzerinden gerçekleştirilen özellik önem analizi, saldırı tespitinde hangi özniteliklerin daha belirleyici olduğunu ortaya koyarak modelin yorumlanabilirliğini artırmıştır.

k-En Yakın Komşu algoritması da yüksek sınıflandırma başarımı göstermiştir. Özellikle benzer trafik örüntülerinin doğru şekilde gruplanabildiği durumlarda etkili sonuçlar üretmiştir. Bununla birlikte, yüksek boyutlu veri kümelerinde hesaplama maliyetinin artması ve bellek kullanımı açısından dezavantajlı olabildiği gözlemlenmiştir.

Karar Ağacı algoritması, basit ve yorumlanabilir yapısı sayesinde anlaşılabilir sonuçlar sunmuş; ancak karmaşık veri yapılarında zaman zaman aşırı öğrenme eğilimi göstermiştir. Buna rağmen, kabul edilebilir doğruluk seviyeleri ile saldırı tespitinde uygulanabilir bir yöntem olduğu değerlendirilmiştir.

Destek Vektör Makineleri algoritması, özellikle yüksek boyutlu veri yapılarında makul bir sınıflandırma başarımı sergilemiş; ancak işlem süresi ve kaynak tüketimi bakımından diğer yöntemlere kıyasla daha maliyetli olmuştur. Bu durum, SVM'nin büyük ölçekli veri setlerinde kullanımını sınırlayabilmektedir.

Naive Bayes algoritması ise basit yapısı ve hızlı çalışmasıyla öne çıkmasına rağmen, veri setindeki öznitelikler arasındaki bağımsızlık varsayımının her durumda sağlanamaması nedeniyle diğer modellere kıyasla daha düşük performans sergilemiştir.

Sonuç olarak, bu çalışma kapsamında elde edilen performans analizleri, farklı makine öğrenmesi algoritmalarının saldırı tespit probleminde birbirinden farklı güçlü ve zayıf yönlere sahip olduğunu göstermektedir. Bulgular, tek bir algoritmanın tüm senaryolar için ideal olmadığını; model seçiminin veri yapısı ve problem gereksinimlerine bağlı olarak yapılması gerektiğini ortaya koymaktadır.

4.6 CICEV2023 Veri Seti Üzerinde Elde Edilen Sonuçlar

Bu çalışmanın genişletilmiş değerlendirme aşamasında, 2024 yılı sonrası yayımlanan CICEV2023 DDoS saldırı veri seti kullanılarak ek deneyler gerçekleştirilmiştir. Amaç, UNSW-NB15 veri kümesi üzerinde elde edilen bulguların güncel ve farklı bir ağ trafiği ortamında ne ölçüde genellenebilir olduğunu incelemektir.

CICEV2023 veri seti üzerinde, UNSW-NB15 için kullanılan aynı veri ön işleme adımları, aynı özellik kümesi ve aynı makine öğrenmesi algoritmaları (Rastgele Orman, Karar Ağacı, Destek Vektör Makineleri, k-En Yakın Komşu ve Naive Bayes) uygulanmıştır.

Elde edilen sonuçlar, Rastgele Orman ve k-En Yakın Komşu algoritmalarının, CICEV2023 veri setinde de genel doğruluk, F1-skoru ve ROC-AUC metrikleri açısından öne çıktığını göstermiştir. Bu durum, söz konusu algoritmaların yalnızca belirli bir veri kümesine özgü değil, farklı ve güncel ağ saldırı senaryolarında da tutarlı performans sergilediğini ortaya koymaktadır.

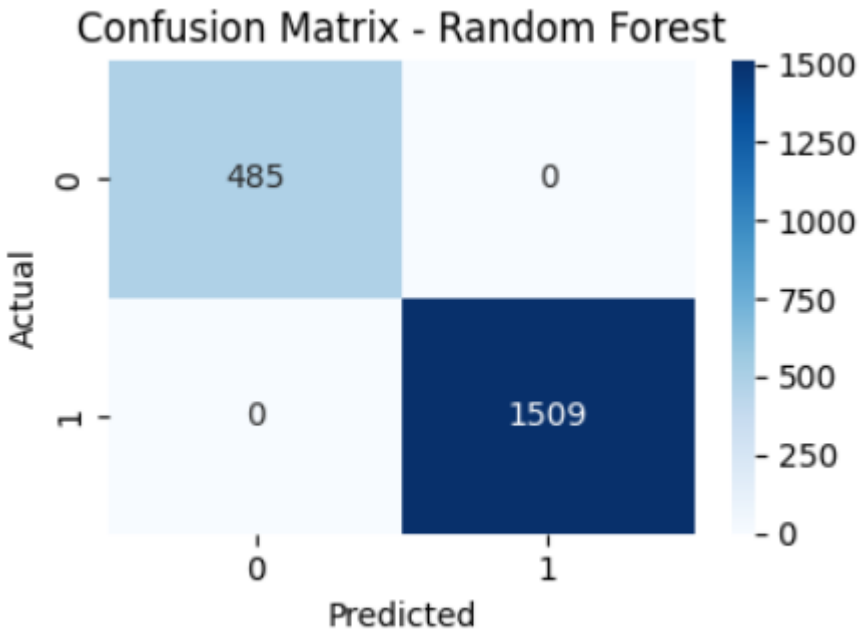
Naive Bayes algoritmasının ise, CICEV2023 veri setinde de karmaşık ve yüksek boyutlu trafik örüntülerini ayırt etmede sınırlı kaldığı gözlemlenmiştir. Bu sonuçlar, UNSW-NB15 üzerinde elde edilen bulgularla yüksek düzeyde tutarlılık göstermektedir.

Çizelge 4.8 UNSW-NB15 ve CICEV2023 veri setleri karşılaştırılması

Veri Seti	Model	Accuracy	Precision (w)	Recall (w)	F1-Score (w)
UNSW-NB15	Random Forest	0.8775	0.8871	0.8775	0.8755
UNSW-NB15	Decision Tree	0.8752	0.8822	0.8752	0.8735
UNSW-NB15	SVM	0.7739	0.8157	0.7739	0.7598
UNSW-NB15	KNN	0.8869	0.8925	0.8869	0.8856
UNSW-NB15	Naive Bayes	0.6395	0.7181	0.6395	0.6216
CICEV2023	Random Forest	1.0000	1.0000	1.0000	1.0000
CICEV2023	Decision Tree	1.0000	1.0000	1.0000	1.0000
CICEV2023	SVM (Linear)	1.0000	1.0000	1.0000	1.0000
CICEV2023	KNN	0.8722	0.8689	0.8722	0.8641
CICEV2023	Naive Bayes	0.7143	0.8603	0.7143	0.7342

Yukarıdaki çizelge incelendiğinde, UNSW-NB15 veri seti üzerinde modellerin farklı performanslar sergilediği, özellikle Random Forest ve k-En Yakın Komşu algoritmalarının daha dengeli sonuçlar ürettiği görülmektedir. Buna karşılık CICEV2023 veri setinde Random Forest, Decision Tree ve SVM modellerinin %100 doğruluk elde etmesi, veri setindeki saldırı ve normal trafik örüntülerinin belirgin biçimde ayrıştığını göstermektedir. Bu durum, veri setinin saldırı tespiti açısından yüksek ayırt ediciliğe sahip olduğunu, ancak gerçek ağ ortamlarındaki karmaşıklığı sınırlı düzeyde yansıttığını göstermektedir. Bu nedenle, modellerin genellenebilirliği UNSW-NB15 gibi daha zorlayıcı veri kümeleri ile birlikte değerlendirilmiştir.

4.6.1 Karışıklık matrisi CICEV2023

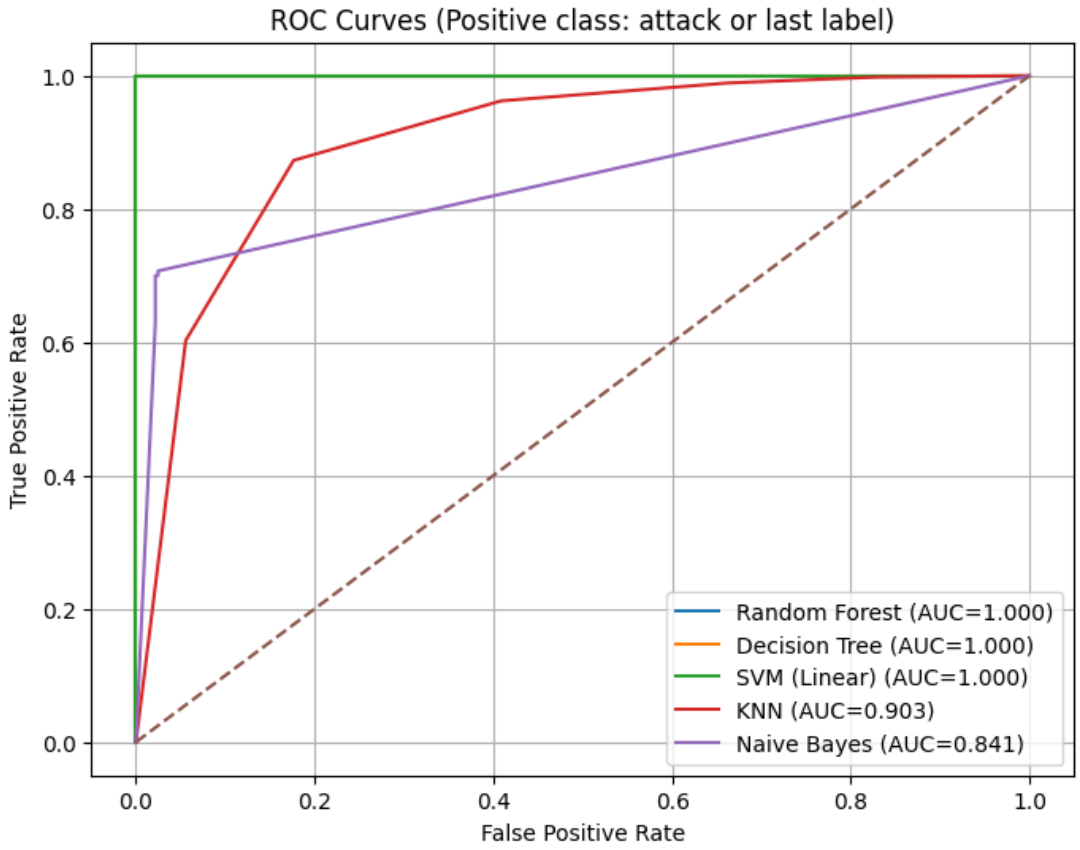


Şekil 4.7 Rastgele Orman karışıklık matrisi (CICEV2023)

CICEV2023 veri seti için oluşturulan karışıklık matrisi, saldırı ve normal trafik sınıflarının tamamen doğru şekilde ayrıştırıldığını göstermektedir. Bu durum, veri setindeki saldırı örüntülerinin belirgin yapısından kaynaklanmaktadır.

4.6.2 ROC–AUC analizi (CICEV2023)

ROC–AUC eğrisi, modelin farklı eşik değerlerinde de yüksek ayırt edicilik sağladığını göstermektedir.



Şekil 4.8 ROC curve (CICEV2023)

CICEV2023 veri seti üzerinde bazı modellerin %100 performans değerlerine ulaşması, veri setinin saldırı ve normal trafik arasında keskin ayrımlar içermesinden kaynaklanmaktadır. Bu nedenle, çalışmanın ana değerlendirmesi daha karmaşık ve gerçekçi trafik yapısına sahip UNSW-NB15 veri seti üzerinden yapılmıştır.

5. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, ağ trafiği verileri üzerinde makine öğrenmesi yaklaşımlarından yararlanılarak saldırı tespitine yönelik bir sınıflandırma yapısı oluşturulmuştur. Çalışma kapsamında yalnızca UNSW-NB15 veri kümesi kullanılmış, veri ön işleme sürecinin ardından beş farklı denetimli öğrenme yöntemi uygulanarak modellerin saldırı tespit başarımı karşılaştırmalı olarak incelenmiştir.

Kullanılan sınıflandırma modelleri; Karar Ağacı, Naive Bayes, Rastgele Orman, k-En Yakın Komşu ve Destek Vektör Makineleri algoritmalarından oluşmaktadır. Modellerin performansları, sınıflandırma başarımını daha kapsamlı biçimde değerlendirebilmek amacıyla doğruluk, kesinlik, duyarlılık ve F1 skoru ölçütleri kullanılarak analiz edilmiş ve elde edilen sonuçlar tablo hâlinde sunulmuştur.

Elde edilen bulgular, tek bir makine öğrenmesi algoritmasının bütün saldırı tipleri için ideal bir performans sunmadığını, her yöntemin belirli senaryolarda daha avantajlı olduğunu göstermiştir. Rastgele Orman modeli, karar ağaçlarının topluluk yaklaşımını kullanması sayesinde genel doğruluk ve F1 skorunda diğer modellerden daha başarılı bir sonuç üretmiştir. Buna karşılık Naive Bayes modeli, basit varsayımlarına rağmen bazı sınıflarda yüksek duyarlılık değerleri göstermiş, özellikle düşük boyutlu özellik uzaylarında verimli çalışmıştır. Destek Vektör Makineleri ise ayrım sınırlarını daha belirgin oluşturduğu durumlarda rekabetçi performans sergilemiştir.

Bu sonuçlar, saldırı tespit probleminin tek yönlü bir öğrenme modeli ile çözülemeyeceğini, çok sınıflı yapının ve veri kümesindeki dengesizliklerin model performansını doğrudan etkilediğini ortaya koymaktadır. Dolayısıyla gerçek zamanlı bir IDS mimarisinde, birden fazla sınıflandırma modelinin birlikte kullanıldığı hibrit yaklaşımın daha verimli olabileceği değerlendirilmektedir. Çalışmanın bulguları ayrıca, denetimli öğrenme modellerinin başarısının doğrudan öznitelik mühendisliği, anomali örneklerinin dağılımı ve eğitim için kullanılan veri miktarı ile ilişkili olduğunu göstermektedir.

Sonuç olarak, bu tez kapsamında uygulanan yöntemler, UNSW-NB15 veri kümesi kullanılarak makine öğrenmesi algoritmaları ile ağ saldırılarının otomatik olarak sınıflandırılabileceğini ve IDS sistemlerinin performansının istatistiksel metriklerle iyileştirilebileceğini kanıtlamıştır. Yapılan deneyler aynı zamanda veri bilimi temelli yaklaşımların geleneksel imza tabanlı saldırı tespit yöntemlerine göre daha esnek ve güncellenebilir olduğunu göstermektedir.

Güncel Veri Seti ile Doğrulama Sonuçlarının Değerlendirilmesi: Ek olarak kullanılan CICEV2023 veri seti üzerinde elde edilen sonuçlar, çalışmada önerilen yaklaşımın güncel veri kümelerinde de yüksek başarı sağlayabildiğini göstermiştir. Bununla birlikte, veri setlerinin yapısal farklılıklarının model performanslarını doğrudan etkilediği gözlemlenmiş; bu durum, saldırı tespit sistemlerinin farklı senaryolar altında değerlendirilmesi gerektiğini ortaya koymuştur.

Öneriler ve Gelecek Çalışmalar: Bu tezde ortaya konan bulgular, ağ saldırılarının sınıflandırılmasında denetimli makine öğrenmesi yöntemlerinin etkili şekilde kullanılabileceğini göstermiştir. Ancak çalışma, yalnızca UNSW-NB15 veri kümesi ve beş temel sınıflandırma algoritması ile sınırlandırılmıştır. Bu nedenle gelecek araştırmalarda, yöntemsel kapsamın genişletilmesi ve farklı veri kaynaklarının değerlendirilmesi önerilmektedir.

İlk olarak, **derin öğrenme tabanlı mimarilerin** kullanılması, özellikle yüksek boyutlu özellik uzaylarında daha başarılı sonuçlar üretebilir. LSTM, CNN veya melez ağ yapıları, akış tabanlı trafik analizlerinde zaman bağımlı örüntüleri yakalama açısından avantaj sağlayabilir. İkinci olarak, veri kümesindeki sınıf dengesizliğini azaltmaya yönelik **SMOTE, ADASYN veya GAN tabanlı sentetik veri üretimi** gibi yaklaşımlar, tespit oranlarını artırma potansiyeline sahiptir.

Ayrıca, çalışmanın statik veri seti üzerinden yürütülmesi, gerçek zamanlı trafiğin dinamik özelliklerini yansıtmamaktadır. Bu nedenle ileride yapılacak çalışmalarda, **canlı ağ ortamından toplanan akışların çevrimiçi işlenmesi ve anomali tespiti** üzerine yoğunlaşılması, IDS sistemlerinin operasyonel verimliliğini artıracaktır. Bunun yanında,

denetimli öğrenme yöntemleri yerine **yarı denetimli ve denetimsiz modellerin** incelenmesi, etiketlenmemiş trafik örneklerinin saldırı kategorilerine otomatik olarak atanmasına imkân sağlayabilir.

Bu tez çalışması ile, makine öğrenmesi yöntemlerinin ağ güvenliği alanındaki potansiyeli ortaya konmuş ve ağ saldırılarına karşı daha etkili çözümler geliştirilmesi yönünde önemli bir adım atılmıştır. Yapılacak ek çalışmalar ile bu alandaki gelişmelerin daha ileriye taşınması mümkündür.

Son olarak, mevcut çalışma yalnızca performans metrikleri üzerinden değerlendirilmiştir. Gelecek araştırmalarda **hesaplama maliyeti, tepki süresi, model karmaşıklığı ve bellek gereksinimleri** gibi pratik faktörlerin de karşılaştırılması, saldırı tespit sistemlerinin gerçek sistemlere entegrasyonunda kritik bir tasarım rehberi sunacaktır.

5.1 Tezin Katkıları

Bu çalışma, ağ saldırılarının tespitine yönelik makine öğrenmesi tabanlı çözümlere odaklanmış olup literatüre üç temel katkı sunmaktadır. İlk olarak, araştırma kapsamında **UNSW-NB15 veri kümesi sistematik biçimde işlenmiş**, özellik seçimi, veri temizleme, normalizasyon ve train/test ayrımı gibi adımlar açık bir yöntem çerçevesinde tanımlanmıştır. Bu süreç, veri kümesinin saldırı tespit sistemlerinde kullanılabilirliğini artırmakta ve sonraki çalışmalar için yeniden üretilebilir bir deney ortamı oluşturmaktadır.

İkinci olarak, bu çalışmada beş farklı denetimli sınıflandırma yaklaşımı (Karar Ağacı, Naive Bayes, Rastgele Orman, k-En Yakın Komşu ve Destek Vektör Makineleri) aynı veri kümesi üzerinde uygulanarak karşılaştırmalı bir değerlendirme yapılmıştır. Bu sayede, farklı algoritmaların çok sınıflı saldırı tespit problemindeki davranışları incelenmiş; her bir yöntemin güçlü yönleri, sınırlılıkları ve performans farklılıkları ortaya konmuştur. Elde edilen bulgular, tek bir modelin tüm saldırı türleri için en yüksek başarıyı

garanti etmediğini ve sınıflandırma performansının veri dağılımı ile sınıf dengesine bağlı olarak değiştiğini göstermiştir.

Üçüncü olarak, saldırı tespit sistemlerinin değerlendirilmesinde kullanılan performans ölçütlerinin birlikte ele alınmasının önemi vurgulanmıştır. Doğruluk, kesinlik, duyarlılık ve F1 skoru gibi metriklerin ortak yorumlanması sayesinde, modellerin yalnızca genel başarı düzeyleri değil, aynı zamanda yanlış alarm ve saldırıyı kaçırma eğilimleri de analiz edilmiştir. Bu yaklaşım, makine öğrenmesi tabanlı IDS tasarımlarında daha bilinçli ve dengeli kararlar alınmasına katkı sağlamaktadır.

Genel çerçevede çalışma, **akademik araştırmalar ile pratik IDS gereksinimleri arasındaki boşluğun azaltılmasına katkı sağlayarak**, ileride gerçek zamanlı saldırı analizi çalışmalarına temel oluşturabilecek bir deneysel değerlendirme modeli ortaya koymuştur.

Genel olarak bu tez, ağ trafik analizinde makine öğrenmesi yöntemlerinin uygulanabilirliğini nicel verilerle göstermiş; gelecekte gerçek zamanlı saldırı analizine yönelik daha kapsamlı çalışmaların yapılabilmesi için deneysel bir temel sunmuştur.

Bu Tezden Üretilen Yayın Çalışmaları: Bu tez çalışması kapsamında elde edilen deneysel bulgular doğrultusunda, derin öğrenme ve üretici yapay sinir ağları (GAN) tabanlı sentetik veri artırma yöntemlerinin DDoS saldırı tespitine etkisini inceleyen bir bilimsel makale hazırlanmıştır.

Synthetic Data Augmentation for DDoS Attack Detection: A GAN-Based Deep Learning Approach başlıklı bu çalışma henüz herhangi bir dergi veya konferansta yayımlanmamış olup, yayın sürecine sunulması planlanmaktadır

5.2 Çalışmanın Sınırlılıkları ve Tartışma

Bu çalışma belirli varsayımlar ve teknik kısıtlar altında yürütüldüğü için bazı sınırlılıklara sahiptir. Öncelikle, araştırma yalnızca **UNSW-NB15 veri kümesine** odaklanmıştır. Bu veri kümesi geniş saldırı kategorileri sunmasına rağmen, gerçek ağ ortamlarındaki tüm tehdit çeşitliliğini temsil etmemektedir. Dolayısıyla modellerin performansı, farklı veri setlerine veya kurumsal ağlardan elde edilen gerçek zamanlı trafikl e karşılaştırıldığında değişiklik gösterebilir.

İkinci olarak çalışma, **denetimli öğrenme algoritmalarına** yönelmiştir. Bu yaklaşım etiketlenmiş örnekler üzerinden eğitim gerektirdiği için, yeni ortaya çıkan saldırı türlerini veya etiketlenmemiş anomalileri tespit etme kapasitesi sınırlıdır. Denetimsiz veya yarı denetimli öğrenme yöntemleri bu açıdan daha esnek olabilir fakat bu tez kapsamında değerlendirilmemiştir.

Üçüncü olarak, veri kümesindeki sınıf dağılımı dengesizdir ve bazı saldırı kategorileri oldukça düşük örnek sayısına sahiptir. Modellemede kullanılan klasik sınıflandırma algoritmaları, **küçük sınıflara karşı duyarlılık kaybı** yaşayabileceği için bazı saldırı türleri yeterince temsil edilememiştir. Veri çoğaltma, yeniden örnekleme veya sentetik veri üretimi gibi teknikler bu çalışma kapsamında uygulanmamıştır.

Son olarak, araştırma yalnızca **sınıflandırma performans ölçütlerine** (doğruluk, kesinlik, duyarlılık, F1) odaklanmış, hesaplama maliyeti, çalışma süresi, hafıza kullanımı, model karmaşıklığı ve ölçeklenebilirlik gibi pratik kriterler değerlendirilmemiştir. Bu nedenle sonuçlar, gerçek zamanlı ağ izleme sistemlerine doğrudan aktarılmadan önce ek mühendislik analizleri gerektirebilir.

KAYNAKLAR

Ahmed M, Mahmood AN, Hu J (2016) A survey of network anomaly detection techniques. *Journal of Network and Computer Applications* 60, 19–31

Alaiz-Rodríguez R, Japkowicz N (2008) Assessing the impact of changing environments on classifier performance. *Proceedings of the 8th IEEE International Conference on Data Mining*

Aljawarneh S, Yassein MB (2017) Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *International Journal of Computer Applications* 167(10), 1–7

Almseidin M, Alzubi A, Kovacs S, Alkasassbeh M (2017) Evaluation of machine learning algorithms for intrusion detection system. *Procedia Computer Science* 127, 124–129

Bhuyan MH, Bhattacharyya DK, Kalita JK (2014) Network anomaly detection: a machine learning perspective. *Computer Networks* 55(15), 3448–3470

Biermann E, Cloete E, Venter L (2001) A comparison of intrusion detection systems. *Computers & Security* 20(8), 676–683

Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials* 18(2), 1153–1176

Calzarossa MC, Massari L, Tessera D (2023) Machine Learning Techniques for Cybersecurity Applications: A Survey. *ACM Computing Surveys* 55(12)

Cannady J (1998) Artificial neural networks for misuse detection. *Proceedings of the 21st National Information Systems Security Conference*

Debar H, Dacier M, Wespi A (2000) A revised taxonomy for intrusion-detection systems. *Annales des Télécommunications* 55(7–8), 361–378

Denning DE (1987) An intrusion-detection model. *IEEE Transactions on Software Engineering* SE-13(2), 222–232

Depren O, Topallar M, Anarim E, Ciliz MK (2005) An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications* 29(4), 713–722

García-Teodoro P, Díaz-Verdejo J, Maciá-Fernández G, Vázquez E (2009) Anomaly-based network intrusion detection: techniques, systems and challenges. *Computers & Security* 28(1–2), 18–28

Gupta BB, Tewari A, Jain AK, Agrawal DP (2021) Fighting against phishing attacks: state of the art and future challenges. *Neural Computing and Applications* 28(12), 3629–3654

Hodo E, Bellekens X, Hamilton A, Dubouchaud J, Iorkyase E (2016) Threat analysis of IoT networks using artificial neural network intrusion detection system. *Proceedings of the International Symposium on Networks, Computers and Communications (ISNCC)*

Kayacik HG, Zincir-Heywood AN, Heywood MI (2005) Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets. *Proceedings of the Third Annual Conference on Privacy, Security and Trust*

Kim G, Lee S, Kim S (2014) A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications* 41(4), 1690–1700

Koziol J (2003) *Intrusion Detection with Snort*. Sams Publishing

Le Q, Ou Y (2010) Application of machine learning to network intrusion detection. *Proceedings of the International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*

Li Y, Xia J, Zhang S, Yan J, Ai X, Dai K (2012) An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert Systems with Applications* 39(1), 424–430

Lunt TF (1993) A survey of intrusion detection techniques. *Computers & Security* 12(4), 405–418

Moustafa N, Slay J (2015) UNSW-NB15: A comprehensive data set for network intrusion detection systems. *2015 Military Communications and Information Systems Conference (MilCIS)*

Mukkamala S, Sung AH, Abraham A (2005) Intrusion detection using an ensemble of intelligent paradigms. *Journal of Network and Computer Applications* 28(2), 167–182

Panda M, Patra MR (2007) Network intrusion detection using Naive Bayes. *International Journal of Computer Science and Network Security (IJCSNS)* 7(12), 258–263

Patcha A, Park JM (2007) An overview of anomaly detection techniques: existing solutions and latest technological trends. *Computer Networks* 51(12), 3448–3470

Peddabachigari S, Abraham A, Grosan C, Thomas J (2007) Modeling intrusion detection system using hybrid intelligent systems. *Journal of Network and Computer Applications* 30(1), 114–132

Revathi S, Malathi A (2013) A detailed analysis on NSL-KDD dataset using various machine learning techniques. *International Journal of Engineering Research & Technology (IJERT)* 2(12)

Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A (2019) A survey of network-based intrusion detection data sets. *Computers & Security* 86, 147–167

Sahu BR, Dash SK, Behera HS (2018) A novel feature selection method for intrusion detection system based on new entropy and improved rough set. *Expert Systems with Applications* 117, 61–75

Shone N, Ngoc TN, Phai VD, Shi Q (2018) A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence* 2(1), 41–50

Sommer R, Paxson V (2010) Outside the closed world: on using machine learning for network intrusion detection. *Proceedings of the IEEE Symposium on Security and Privacy*

Subba B, Biswas S, Karmakar S (2018) A neural network based system for intrusion detection and attack classification. *International Journal of Computer Applications* 182(47), 1–8

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD Cup 99 data set. *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*

Tsai CF, Hsu YF, Lin CY, Lin WY (2009) Intrusion detection by machine learning: a review. *Expert Systems with Applications* 36(10), 11994–12000

Tupsamudre H, Radhakrishnan B, Tripathy S (2019) URL phishing detection using lexical features with machine learning. *2019 International Conference on Communication and Signal Processing (ICCSP)*

Wang W, Sheng Y, Wang J, Zeng X, Ye X, Huang Y (2017) HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access* 6, 1792–1806

Xia Y, Wang X, Sun X, Wang X (2018) A new unsupervised anomaly detection approach for intrusion detection system. *Information Sciences* 480, 63–72

Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* 5, 21954–21961

Zhang J, Zulkernine M (2006) Anomaly-based network intrusion detection with unsupervised outlier detection. *Proceedings of the IEEE International Conference on Communications*