MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK SYBIL BOTLARIN TESPİT EDİLMESİ

CANSU BETÜL ÖCEL

AĞUSTOS 2025

ÇANKAYA ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

VERİ ANALİTİĞİ ANABİLİM DALI
VERİ ANALİTİĞİ YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK SYBIL BOTLARIN TESPİT EDİLMESİ

CANSU BETÜL ÖCEL

AĞUSTOS 2025

ÖZET

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANILARAK SYBIL BOTLARIN TESPİT EDİLMESİ

ÖCEL, CANSU BETÜL
VERİ ANALİTİĞİ YÜKSEK LİSANS TEZİ

Danışman: Prof. Dr. Mehmet Reşit TOLUN
Ağustos 2025, 61 sayfa

Bu çalışma, NSL-KDD veri seti kullanılarak ağ tabanlı anomali tespiti amacıyla çeşitli makine öğrenmesi algoritmalarının performansını karşılaştırmalı olarak değerlendirmeyi amaçlamaktadır. NSL-KDD, saldırı türlerini dört ana başlıkta (DoS, Probe, R2L, U2R) toplayan, etiketli ve dengeli yapısıyla denetimli öğrenme yöntemleri için uygun bir veri seti olarak ele alınmıştır.

Çalışma kapsamında veri seti üzerinde öncelikle istatistiksel analizler ve veri keşif çalışmaları gerçekleştirilmiş, ardından veri ön işleme adımları uygulanmıştır. Bu süreçte kategorik değişkenler sayısal forma dönüştürülmüş, eksik veriler temizlenmiş ve azınlıkta kalan sınıflar SMOTE yöntemiyle dengelenmiştir. Özellik seçimi için Mutual Information (MI) yöntemi kullanılarak en bilgilendirici 15 değişken belirlenmiş ve model eğitimi bu özellikler kullanılarak gerçekleştirilmiştir. Sonrasında tüm değişkenler kullanılarak modeller tekrar eğitilmiş ve sonuçlar kıyaslanmıştır.

Modelleme aşamasında Lojistik Regresyon, Naive Bayes, Random Forest, K En Yakın Komşu (KNN), Destek Vektör Makineleri (SVM), AdaBoost ve Yapay Sinir Ağı (ANN) algoritmaları kullanılmıştır. Her model için hiper parametre optimizasyonu GridSearchCV veya RandomizedSearchCV yöntemleriyle yapılmıştır. Modellerin başarısı doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 skoru gibi değerlendirme metrikleri kullanılarak analiz edilmiştir.Elde edilen sonuçlar, NSL-KDD veri seti üzerinde bazı modellerin özellikle DoS gibi baskın sınıflarda yüksek doğruluk sağlarken, azınlıkta kalan R2L ve U2R saldırı türlerinde performans düşüşleri yaşandığını göstermektedir.

Bu durum, dengesiz veri setlerinde kullanılacak yöntemlerin dikkatli seçilmesinin gerekliliğine işaret etmektedir.

Anahtar Kelimeler: Anomali Tespiti, NSL-KDD, Özellik Seçimi, SMOTE, Makine Öğrenmesi, Bilgisayar Ağları

ABSTRACT

DETECTION OF SYBIL BOTS USING MACHINE LEARNING TECHNIQUES

OCEL, CANSU BETUL
DATA ANALYTICS MASTER'S THESIS

Supervisor: Prof. Dr. Mehmet Reşit TOLUN
August 2025, 61 pages

This study aims to comparatively evaluate the performance of various machine learning algorithms for network-based anomaly detection using the NSL-KDD dataset. NSL-KDD, which categorizes attack types into four main groups (DoS, Probe, R2L, U2R), has been considered a suitable dataset for supervised learning methods due to its labeled and balanced structure.

Within the scope of the study, initial statistical analyses and exploratory data analysis were conducted on the dataset, followed by data preprocessing steps. In this process, categorical variables were converted into numerical format, missing values were removed, and the minority classes were balanced using the SMOTE technique. For feature selection, the Mutual Information (MI) method was applied to determine the 15 most informative variables, and models were trained using these features. Subsequently, the models were retrained using all available features, and the results were compared.

During the modeling phase, Logistic Regression, Naive Bayes, Random Forest, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), AdaBoost, and Artificial Neural Network (ANN) algorithms were employed. Hyperparameter optimization was performed for each model using GridSearchCV or RandomizedSearchCV. Model performances were evaluated based on several metrics, including accuracy, precision, recall, and F1-score. The results indicate that some models achieved high accuracy particularly for dominant classes such as DoS, while performance dropped significantly for underrepresented classes like R2L and U2R.

These findings emphasize the importance of careful algorithm selection when dealing with imbalanced datasets.

Keywords: Anomaly Detection, NSL-KDD, Feture Selection, SMOTE, Machine Learning, Computer Networks

TEŞEKKÜR

Yüksek lisans öğrenimim süresince akademik çalışmalarımı yürütürken, aynı zamanda annelik sorumluluklarını da yerine getirmeye gayret ettiğim bu süreçte, maddi ve manevi desteklerini esirgemeyen tüm aile bireylerime, arkadaşlarıma ve tez danışmanıma teşekkür etmeyi bir borç bilirim.

Öncelikle, bilgi ve deneyimiyle tez sürecimin her aşamasında bana yol gösteren, yapıcı geri bildirimleri, anlayışlı ve sabırlı tutumu ve değerli katkılarıyla çalışmama yön veren saygıdeğer danışmanım Prof. Dr. Mehmet Reşit TOLUN 'a en içten teşekkürlerimi sunarım.

Bununla birlikte, her koşulda yanımda olan, özverisi ve desteğiyle bu sürecin her aşamasında bana güç veren kıymetli annem Şerife TÜRKEŞ 'e en derin şükranlarımı sunarım. Bu süreçte motivasyonumu kaybettiğim her an bana gösterdiği anlayış ve sağladığı destek, çalışmalarımı sürdürebilmem açısından son derece kıymetli olmuştur.

Ayrıca, her zaman yanımda olan, sabrı, desteği ve teşvik edici tutumuyla bu sürecin yükünü birlikte taşıyan değerli eşim Kubilay ÖCEL'e de gösterdiği anlayış ve sağladığı manevi destek için teşekkür ederim.

Bu süreçte desteklerini esirgemeyen, varlıklarıyla beni motive eden arkadaşlarım ve diğer aile üyelerime de teşekkür ederim. Akademik hedeflerime ulaşma yolunda gösterdikleri anlayış ve verdikleri desteği her zaman minnetle anacağım.

İÇİNDEKİLER

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

SİMGELER VE KISALTMALAR LİSTESİ

Simgeler

χ^2	Chi-squared
C	Regularizasyon parametresi
λ	Ceza terimi

Kısaltmalar

vd.	Ve diğerleri
IDS	:Intrusion Detection Systems - İzinsiz Giriş Tespit Sistemleri
DoS	:Denial of Service
DDoS	:Distributed Denial of Service
U2R	:User to Root
R2L	:Remote to Local
SYN	:Senkronizasyon
ICMP	:Internet Control Message Protocol - İnternet Kontrol Mesaj Protokolü
UDP	:User Datagram Protocol - Kullanıcı Datagram Protokolü
TCP	:Transmission Control Protocol - İletim Kontrol Protokolü
Bps	:Saniye başına bit
Pps	:Saniye başına paket
Rps	:Saniye başına istek
DNS	:Domain Name System - Alan Adı Sistemi
NTP	:Network Tima Protocol - Ağ Zamanlama Protokolü
IP	:İnternet Protokolü
URL	:Uniform Resource Loader - Tekdüze Kaynak Bulucu
SF	:Session Finished
REJ	:Reject
RSTR	:Reset
MI	:Mutual Information

sklearn	:Scikit-Learn
SMOTE	:Synthetic Minority Oversampling Technique
RFE	:Recursive Feature Elimination
KNN	:K Nearest Neighbor - K En Yakın Komşu
SVM	:Support Vector Machine - Destek Vektör Makineleri
AdaBoost	:Adaptive Boosting
SAMME	:Stagewise Additive Modeling using a Multiclass Exponential loss function
ANN	:Artificial Neural Network - Yapay Sinir Ağları
TP	:True Positive - Doğru Pozitif
TN	:True Negative - Doğru Negatif
FP	:False Positive - Yanlış Pozitif
FN	:False Negative - Yanlış Negatif

BÖLÜM I
AĞ SALDIRILARI VE KORUNMA YÖNTEMLERİ

Verilerin elektronik olarak aktarılması ihtiyacı nedeniyle son zamanlarda bilgisayar sistemlerinin ve internetin kullanımı bir dizi güvenlik, gizlilik ve mahremiyet sorunu ile ilişkilendirilmektedir. Bu nedenle, çalışmalar gizlilik ve güvenli bilgisayar sistemlerinin geliştirilmesine odaklanmıştır, ancak çabalara rağmen, bu sorunlar bilgisayar sistemlerinde hala devam etmektedir, öyle ki şu anda dünya üzerinde tamamen güvenli bir sistem bulunmamaktadır. Saldırılar farklı şekillerde ortaya çıkmaktadır (Amasarani 2022); bu saldırılar veri tabanında anormal davranış gösteren yeni yetkililerin varlığına tepki olarak ortaya çıkmaktadır. Bu nedenle, ağ sistemlerine yönelik farklı saldırı biçimlerine karşı koymak için çeşitli araçlar kullanılmıştır ve bu araçlardan biri de İzinsiz Giriş Tespit Sistemleridir (IDS). Bu araç her türlü izinsiz giriş için ağ sistemlerinin gerçek zamanlı izlenmesi için geliştirilmiştir. Bir sistemin güvenlik özelliklerini tehlikeye atmayı amaçlayan saldırıları tespit etmeyi hedeflerler (Aljanabi vd. 2021:4).

1.1 İZİNSİZ GİRİŞ TESPİT SİSTEMLERİ

İzinsiz giriş tespiti, bir bilgisayar sisteminde veya ağında meydana gelen olayları izleyerek ve bu girişlerin izinsiz olup olmadığını analiz eder. Asıl amaç bu izinsiz girişlerin önlenmesi değil, sisteme sızma girişimlerini tespit etmektir. İzinsiz giriş veya anormal faaliyetler tespit edildiğinde sistem operatörlerine veya güvenlik görevlilerine alarmlar oluşturabilir ve raporlayabilir. Tespit edilen izinsiz girişleri engellemek için harici yollar denenebilmektedir.

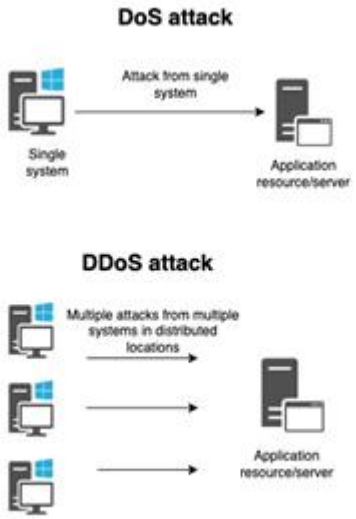
1.2 AĞ SALDIRILARI

Ağ saldırıları, bilgisayar ağlarına yönelik kötü niyetli girişimlerdir ve modern bilgi güvenliğinin önemli bir parçasını oluşturur. Bu saldırılar, veri çalma, hizmet kesintisi yaratma veya yetkisiz erişim sağlama gibi çeşitli amaçlarla gerçekleştirilir.

Bu tez kapsamında kullanılan veri seti içinde yer alan ağ saldırıları DoS, Probe, U2R ve R2L saldırılarıdır. Bu saldırıların türleri ve bu saldırılara karşı alınabilecek güvenlik önlemleri ile ilgili genel bilgiler aşağıda açıklanmaktadır.

1.2.1 Denial of Service (DoS) ve Distributed Denial of Service (DDoS) Saldırıları

DoS ve DDoS saldırılarını daha iyi algılayabilmek adına Şekil1'de bu saldırıların çalışma mekanizması gösterilmiştir.



Şekil 1: DoS ve DDoS saldırılarının çalışma mekanizmaları

1.2.1.1 DoS Saldırıları

DoS saldırıları, saldırılmak istenen sisteme aşırı miktarda istek göndererek sistemin kapasitesini aşar ve hizmetin kesintiye uğramasına sebebiyet verir. Bu tür saldırılar, belirli bir hizmetin veya kaynağın geçici veya kalıcı olarak kullanılamaz hale gelmesini amaçlar.

- **SYN Flooding**: Saldırgan, hedef sisteme çok sayıda yarım bağlantı isteği (SYN paketleri) gönderir. Sistem, bu istekleri işlemekle meşgul olur ve yeni bağlantı taleplerine yanıt veremez (Stallings 2017).
- **Ping of Death**: Saldırgan, hedef sisteme çok büyük boyutlu ICMP (Ping) paketleri gönderir, bu da sistemin çökmesine neden olabilir (Pfleeger P. C. ve Pfleeger S. L. 2007).
- **UDP Flood Attack**: Saldırgan, hedef sisteme çok sayıda UDP paketleri gönderir ve böylece sistemin kesintiye uğramasını veya çökmesini amaçlar (Log360 2024).

1.2.1.2 DDoS Saldırıları

DDoS saldırıları, DoS saldırılarının daha gelişmiş bir versiyonudur. Genel olarak birçok farklı kaynaktan eş zamanlı olarak gerçekleştirilmektedir. Bunun amacı ise, saldırının etkisini artırmak ve hedef sistemin çökmesine neden olmaktır.

- **Botnet Kullanımı**: Saldırganların amacı ilk etapta bir ağın içindeki cihazların kontrolünü ele geçirmektir. Ağ içindeki birçok bilgisayarı zararlı yazılımlarla kontrol altına alır ve bu bilgisayarları hedef sisteme karşı saldırı gerçekleştirmek için kullanır (Mirkovic ve Reiher 2004:39). Saldırganın kontrolü altında bulunan bu virüslü cihazlar topluluğuna botnet adı verilir (Log360 2024).

- **Volume Based Attacks**: UDP, ICMP ve diğer sahte paket akışlarını içerir. Saldırının amacı saldırıya uğrayan sitenin bant genişliğini satüre etmektir. Bu saldırıların büyüklüğü ise saniye başına bit (Bps) cinsinden ölçülür (Imperva 2024).

- **Protocol Attacks:** Bu saldırı türü gerçek sunucu kaynakları, güvenlik duvarları gibi ara iletişim ekipmanlarının kaynaklarını tüketmeyi amaçlar. Saldırının büyüklüğü ise saniye başına paket (Pps) cinsinden ölçülür (Imperva 2024).

- **Amplification Attacks**: DNS, NTP gibi hizmetler kullanılarak hedef sisteme çok sayıda büyük cevap paketleri gönderilir (Log360 2024). Saldırının büyüklüğü saniye başına istek (Rps) olarak ölçülür (Imperva 2024).

1.2.1.3 DoS ve DDoS Saldırılarının Farkları

- DoS saldırılarında tek bir sistem hedef sisteme saldırmayı amaçlarken, DDoS saldırılarında birden fazla sistem hedef sisteme saldırır. Dolayısıyla DoS saldırılarında hedef bilgisayara tek bir konumdan gönderilen veri paketi yüklenirken, DDoS saldırılarında birden fazla konumdan gönderilen veri paketi yüklenir.

- DoS saldırıları DDoS saldırılarına kıyasla daha yavaştır.

- DoS saldırıları tek bir sistem kullanıldığı için kolayca engellenebilir. Ancak DDoS saldırılarını engellemek zordur çünkü birden fazla cihaz paket göndermekte ve birden fazla konumdan saldırılmaktadır.

- DoS saldırılarında DoS saldırı araçları ile sadece tek bir cihaz kullanılırken DDoS saldırılarında aynı anda saldırmak için volume botlar kullanılır.

- DDoS saldırıları ile karşılaştırıldığında, DoS saldırılarının izini sürmek daha kolaydır.

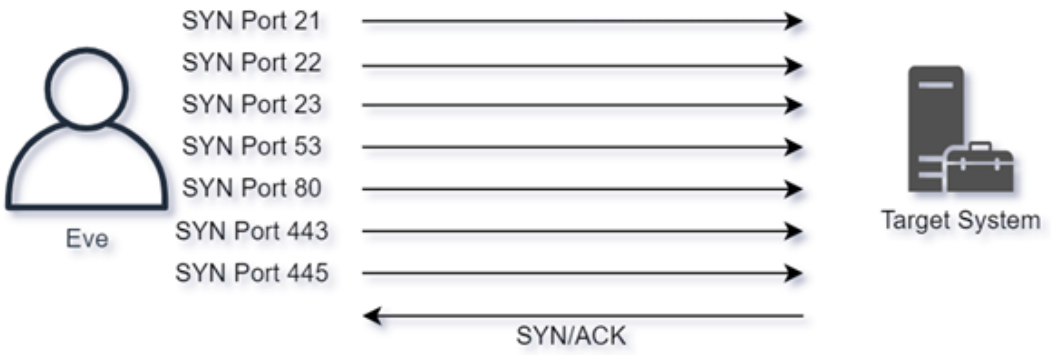
- DoS saldırılsrındaki trafik hacmi DDoS saldırılarına kıyasla daha azdır. DDoS saldırıları, saldırganın hedef ağında büyük hacimlerde trafik oluşturmasına olanak tanır (Amasarani 2022).

1.2.2 Probe Saldırıları

Probe saldırıları, bir ağ veya sistem hakkında bilgi toplamak amacıyla gerçekleştirilir. Saldırganlar, açık portları, güvenlik açıklarını ve zayıflıkları belirlemek için tarama yaparlar. Bu bilgiler, daha sonra gerçekleştirilecek olan daha karmaşık saldırılar için kullanılabilir. Diğer bir deyişle probe saldırıları, bir saldırganın hedefi taradığı saldırı sınıfıdır.

Hedeflenen bir ağ veya ana bilgisayar hakkında bilgi toplamak için yani keşif amacıyla kullanılır. Keşif saldırıları, bir ağa bağlı makinelerin türleri ve sayıları hakkında bilgi toplamanın oldukça yaygın bir yoludur. Yüklenmiş yazılım türlerini ve/veya kullanılan uygulamaları belirlemek için bir ana bilgisayara saldırılabilir. Probe saldırıları, bir ana bilgisayarı veya ağı tehlikeye atmaya yönelik gerçek bir saldırının ilk adımı olarak kabul edilir. Bu saldırılar belirli bir hasara neden olmasa da diğer saldırıları başlatmak için yararlı bilgiler elde edebilecekleri için kurum/kuruluşlar için ciddi tehditler olarak kabul edilirler (Mohiuddin Ahmed vd. 2015).

- **Port Tarama**: Port Tarama, bir uygulamaya paketler göndererek ve herhangi bir yanıt arayarak çalışır. Bu TCP için son derece kolaydır, çünkü bir TCP hizmeti mevcutsa her zaman bir SYN/ACK paketi ile yanıt verecektir. Ancak UDP için bu daha zordur. Hizmetin kullanılabilir olup olmadığını tespit etmek için, çoğu durumda saldırganın uygulamayı yanıt vermeye zorlayan belirli bir girdi göndermesi gerekir. UDP'de barındırılan çoğu uygulama, İstemciler iletişime geçmek için gereken girdiyi tam olarak göndermedikçe yanıt vermeyecektir (W3Schools 2024). Piyasada birçok tarayıcı bulunmakla birlikte örnek olarak nmap verilebilir. Şekil 2'de bir TCP protokolü için örnek bir port taraması görselleştirmesi verilmiştir.



Şekil 2: TCP protokolü için port tarama örneği (W3Schools 2024)

- **Network Mapping**: Ağda aktif olan ana bilgisayarları belirlemenin bir yolu, ağdaki tüm IP adreslerine bir ping göndermektir. Bu genellikle Ping Taraması (ping sweep) olarak adlandırılır (W3Schools 2024).

1.2.3 User to Root (U2R) Saldırıları

U2R saldırıları, bir sistem üzerinde sınırlı ayrıcalıklara sahip yetkisiz bir kullanıcının yönetici erişimi elde etmeye çalıştığı bir siber saldırı türüdür. Bu saldırıda, saldırgan normal bir kullanıcı hesabıyla erişim sağlar ve daha sonra sistem üzerinde tam kontrol sağlayan yönetici erişimini elde etmek için ayrıcalıklarını artırmaya çalışır. Bu saldırı türü bu yetki erişimini genellikle sistemdeki veya uygulamalardaki güvenlik açıklarından yararlanarak sağlar. Saldırganlar yönetici erişimine ulaşmak için arabellek taşmaları (buffer overflow) gibi çeşitli teknikler kullanabilir. U2R saldırılarının tespit edilmesi ve önlenmesi tipik olarak erişim kontrolleri, ayrıcalık ayrımı, düzenli güvenlik güncellemeleri gibi güvenlik önlemlerinin uygulanmasını ve devam eden bir saldırıya işaret edebilecek şüpheli faaliyetlerin izlenmesini gerektirmektedir.

- **Buffer Overflow**: Buffer (ara bellek), verileri bir konumdan diğerine aktarılırken geçici olarak tutan bellek depolama bölgeleridir. Veri hacmi bellek bufferının depolama kapasitesini aştığında bir buffer taşması (buffer overflow) meydana gelir. Sonuç olarak, verileri buffera yazmaya çalışan program bitişik bellek konumlarının üzerine yazar (Imperva 2024, Buffer Overflow Attack).

Buffer taşmaları her tür yazılımı etkileyebilir. Genellikle hatalı biçimlendirilmiş girdilerden veya buffer için yeterli alan ayrılmamasından kaynaklanır. İşlem çalıştırılabilir bir kodun üzerine yazılırsa, programın ön

görülemeyen şekilde davranmasına ve yanlış sonuçlar, bellek erişim hataları veya çökmelere neden olabilir (Imperva 2024, Buffer Overflow Attack).

Saldırganlar, bir uygulamanın belleğinin üzerine yazarak buffer taşması sorunlarından yararlanır. Bu, programın çalışma yolunu değiştirerek dosyalara zarar veren veya özel bilgileri açığa çıkaran bir yanıtı tetikler. Örneğin, bir saldırgan BT sistemlerine erişim elde etmek için uygulamaya yeni talimatlar göndererek fazladan kod ekleyebilir. Saldırganlar bir programın bellek düzenini biliyorlarsa, tamponun depolayamayacağı girdiyi kasıtlı olarak besleyebilir ve yürütülebilir kod içeren alanların üzerine yazarak kendi kodlarıyla da değiştirebilirler (Imperva 2024, Buffer Overflow Attack).

- **Rootkits**: Rootkit, saldırganın bir sistemde en yüksek ayrıcalıkları elde edebilmesini sağlayan araçtır.

Rootkitler oldukça tehlikelidir çünkü cihazınızdaki varlıklarını gizlemek üzere tasarlanmışlardır. Cihazınıza bir rootkit yerleştiren bir saldırgan cihazınıza uzaktan erişebilir ve kontrol edebilir. Kök seviyesinde erişim sağladıkları için rootkitler antivirüs yazılımınızı devre dışı bırakmak, faaliyetlerinizi gözetlemek, hassas verileri çalmak veya cihazda başka kötü amaçlı yazılımlar çalıştırmak gibi amaçlar için kullanılabilir (MalwareBytes, tarih yok). Burada cihaz yalnızca bir bilgisayar değil akıllı telefonlarınızdan tutun endüstriyel kontrol sistemlerine kadar tüm bilişim teknoloji sistemlerini kapsamaktadır. Bir rootkit, kötü amaçlı yazılım bulaşmasının bir parçası olarak hedef sisteme yüklenir. Kötü amaçlı yazılım için birçok saldırı yolu bulunsa da genellikle bir warez web sitesi veya bilinmeyen göndericiden gelen bir e-posta eki gibi güvenilmeyen bir kaynaktır (ENISA 2024)

Rootkitler yerleştirildikleri yere göre sınıflandırılmaktadır. Bir rootkit uygulamada, çekirdekte, hipervizörde veya donanımda bulunabilir. Aşağıdaki liste, enjekte edilmesi, tespit edilmesi ve kaldırılması en kolay olandan en sofistike ve tespit edilmesi ve kaldırılması çok daha zor olana doğru sıralanmıştır (ENISA 2024).

- Uygulamalar
- Çekirdek
- Hipervizör
- Donanım

1.2.4 Remote to Local (R2L) Saldırıları

R2L saldırıları, uzaktan bir saldırganın yerel kullanıcı haklarına erişim kazanarak sistem üzerinde yetkisiz işlemler gerçekleştirmesini hedefler. Bu saldırılar, genellikle phishing, trojanlar ve exploitler kullanılarak gerçekleştirilir.

- **Phishing (Kimlik Avı)**: Phisher olarak da bilinen bir saldırganın, güvenilir veya kamuya açık bir kuruluştan gelen elektronik iletişimleri taklit ederek, meşru kullanıcıların gizli bilgilerini hileli bir şekilde almaya çalıştığı bir saldırı türüdür. Bu tür iletişimler çoğunlukla e-postalar aracılığıyla yapılmaktadır. Bu saldırılar, kullanıcıları, söz konusu kimlik bilgilerini toplayan sahte web sitelerine yönlendirerek çalışmaktadır. Genellikle kimlik avcılarının ilgilendiği gizli bilgiler, şifreler, kredi kartı numaraları ve kimlik numaralarıdır (Myers ve Jakobsson 2006).
  - **Phishing Saldırı Türleri**:
    - **Aldatıcı Kimlik Avı:** Aldatıcı kimlik avı için en çok kullanılan yöntem e-postadır. Bu saldırı türünde saldırgan hedefe bir banka ya da başka bir işletmedeki hesabının risk altında olduğu gibi asılsız e-postalar göndererek hedefi bir linke yönlendirir.
    - **Donanım Tabanlı Kimlik Avı:** Genel olarak, kötü amaçlı yazılımlar ya sosyal mühendislik yoluyla ya da bir güvenlik açığından yararlanılarak yayılır. Tipik bir sosyal mühendislik saldırısı, bir kullanıcıyı bir e-posta ekini açmaya veya bir web sitesinden bir dosya indirmeye ikna etmektir ve genellikle ekin pornografi, müstehcen ünlü fotoğrafları veya dedikodu ile ilgili olduğunu iddia eder. Bazı indirilebilir yazılımlar kötü amaçlı yazılım da içerebilir. Donanım tabanlı kimlik avının farklı yöntemleri mevcuttur (Myers ve Jakobsson 2006). Bunlar;
    - **Keyloggers (Tuş Kaydediciler):** Tuş kaydediciler, bir web tarayıcısına veya bir aygıt sürücüsüne yüklenen bir kötü amaçlı yazılım türüdür. Tuş kaydediciler genellikle girilen verileri izler ve ilgili verileri kimlik avı sunucularına gönderir (Myers ve Jakobsson 2006).
    - **Oturum Ele Geçirme:** Oturum ele geçirme, kullanıcının yerel bilgisayarında kötü amaçlı yazılım tarafından gerçekleştirilebileceği gibi man in the middle saldırısının bir parçası olarak uzaktan gerçekleştirilebilir (Myers ve Jakobsson 2006).
    - **Web Trojanları:** Web trojanları, kimlik bilgilerini toplamak için oturum açma ekranları üzerinde açılan kötü amaçlı programlardır. Kullanıcı bir web

sitesine bilgi girdiğini sanırken aslında bilgiler yerel olarak girilmekte ve daha sonra kötüye kullanılmak üzere kimlik avcısına iletilmektedir (Myers ve Jakobsson 2006).

- **Ana Bilgisayar Dosya Zehirlenmeleri:** Bir kullanıcı URL çubuğuna www.xxx.com gibi bir adres yazarsa ya da bir yer imi kullanırsa, kullanıcının bilgisayarının siteyi ziyaret etmeden önce bu adresi sayısal bir adrese çevirmesi gerekir. Windows gibi birçok işletim sistemi, DNS araması yapılmadan önce ana bilgisayar adlarını aramak için bir kısayol "hosts" dosyasına sahiptir. Bu adres değiştirilirse, kullanıcı kötü amaçlı bir adrese yönlendirilebilir. Kullanıcı oraya gittiğinde, meşru görünen bir site görecek ve aslında amaçlanan meşru site yerine kimlik avcısına giden gizli bilgileri girecektir (Myers ve Jakobsson 2006).
- **Sistemin Yeniden Yapılandırılması Saldırıları:** Sistem yeniden yapılandırma saldırısının bir türü, bir kullanıcının DNS sunucularını değiştirmektir.Diğer sistem yeniden yapılandırma saldırısı türü de işletim sistemini yeniden yapılandırarak tüm web trafiğinin genellikle kullanıcının makinesinin dışındaki bir web proxy'sinden geçirilmesidir. Bu, bir sonraki sayfada ele alınan man in the middle saldırısının bir şeklidir. Kötü niyetli erişim noktası, gizli bilgiler için kullanıcının trafiğini izler veya kimlik bilgileri isteyen kendi sayfalarını meşru sayfaların yerine koyar (Myers ve Jakobsson 2006).
- **Veri Hırsızlığı:** Kötü amaçlı yazılım bir kullanıcının bilgisayarında çalışmaya başladığında, bilgisayarda depolanan gizli bilgileri doğrudan çalabilir. Bu bilgiler arasında parolalar, yazılım aktivasyon anahtarları, hassas e-postalar ve hedefin bilgisayarında depolanan diğer veriler yer alabilir. Kimlik numarası gibi kalıplara uyan bilgileri arayan verileri otomatik olarak filtreleyerek, çok sayıda hassas bilgi elde edilebilir. Veri hırsızlığı, kişisel bilgisayarların genellikle daha iyi korunan kurumsal bilgisayarlarda da depolanan aynı gizli bilgileri içerdiği gerçeğine dayanarak, kurumsal casusluğu amaçlayan kimlik avı saldırıları için de yaygın olarak kullanılmaktadır (Myers ve Jakobsson 2006).
    - **DNS Tabanlı Kimlik Avı:** İnternet iki temel isim alanından oluşur. Bunlar; İnternet Protokolü (IP) adresleme sistemi ve DNS'ler. Bir DNS sunucusu, alan adlarının (domain name) karşılık gelen IP adresleriyle eşleştirilmesinden başka bir şey olmayan DNS kayıtlarını saklar (Mockapetris 1987). Bu DNS sunucuları insan tarafından okunabilir alan adlarını İnternet üzerindeki bilgisayar sunucularına atanan IP adreslerine çözümler (Purkait 2015:333).

- **İçerik enjeksiyonlu Kimlik Avı:** İçerik enjeksiyonlu kimlik avı, meşru bir siteye kötü amaçlı içerik eklemek suretiyle çalışır. Bu kötü amaçlı içerik hedef kullanıcıyı başka sitelere yönlendirebilir, kullanıcının bilgisayarına kötü amaçlı yazılım yükleyebilir veya verileri kimlik avı sunucusuna yönlendirebilir.
- **Man in the Middle Kimlik Avı:** Kimlik avcısının kendisini kullanıcı ile meşru site arasında konumlandırdığı bir kimlik avı biçimidir. Meşru siteye gönderilmesi amaçlanan mesajlar, bunun yerine değerli bilgileri kaydeden, mesajları meşru siteye ileten ve yanıtları kullanıcıya geri ileten kimlik avcısına iletilir. Man in the middle saldırılarının bir kullanıcı tarafından tespit edilmesi zordur, çünkü site düzgün çalışacaktır ve yanlış herhangi bir şeyin olduğuna dair harici bir gösterge olmayabilir (Myers ve Jakobsson 2006).

BÖLÜM II
METOT VE YÖNTEM

2.1 VERİ SETİ HAKKINDA BİLGİ

Bu çalışma kapsamında kullanılmak üzere, siber güvenlik alanında referans veri seti olarak değerlendirilen, ağ tabanlı saldırı tespiti amacıyla kullanılan NSL-KDD veri seti tercih edilmiştir. NSL-KDD, 1999 yılında DARPA tarafından oluşturulan KDD Cup 99 veri setinin çeşitli eksikliklerinin giderilmesiyle geliştirilmiş bir versiyonudur. KDD 99 veri setinin en büyük dezavantajlarından biri, yüksek miktarda tekrar eden kayıtlar içermesiydi. Bu durum, makine öğrenmesi modellerinin öğrenme sürecinde ciddi dengesizliklere neden olmakta ve modelin genelleme yeteneğini olumsuz yönde etkilemekteydi (Mahbod Tavallaee vd. 2009:1). NSL-KDD veri seti, bu problemi ortadan kaldırmak amacıyla tekrarlayan kayıtları filtrelemiş ve daha dengeli bir dağılım sunmuştur.

NSL-KDD veri seti, her biri 41 özellikten oluşan ağ bağlantı kayıtlarını ve her kayda karşılık gelen saldırı tipini (veya normal bağlantıyı) içermektedir. Saldırılar dört ana kategoriye ayrılmaktadır: DoS (Denial of Service), Probe, R2L (Remote to Local) ve U2R (User to Root). Bu yapısıyla, denetimli öğrenme algoritmalarının eğitilmesi ve test edilmesi için uygun, etiketli bir veri yapısı sunmaktadır.

Veri seti hem eğitim hem de test aşamalarında kullanılmak üzere farklı örneklem grupları içermekte ve test kümesi, eğitim kümesinde bulunmayan saldırı türlerini de barındırarak modelin genel geçerliliğini test etmeye olanak tanımaktadır. Bu da siber güvenlik alanında geliştirilen modellerin bilinmeyen tehditlere karşı ne kadar başarılı olabileceğini değerlendirmek açısından büyük önem taşımaktadır (Revathi ve Malathi 2013:1848).

Sonuç olarak, NSL-KDD veri seti hem akademik hem de pratik uygulamalarda yaygın şekilde kullanılan, ağ tabanlı saldırı tespit sistemlerinin performansını ölçmek için uygun ve daha dengeli bir test ortamı sunmaktadır.

NSL-KDD veri setinde bulunan 41 adet özelliğin 32 tanesi nümerik değişkenler olmakla birlikte 9 tanesi kategorik değişkenlerdir. Veri setinde bulunan özelliklere ilişkin detaylı bilgi aşağıda verilen Tablo 1 ‘de görülmektedir.

Tablo 1: Veri Setine Ait Özellikler

No	Özellik Adı	Özelliğe ilişkin tanım	Tür	Değişken Türü
1	Duration	Bağlantı süresinin uzunluğu	Nümerik	Integer
2	Protocol Type	Bağlantıda kullanılan protokol	Kategorik	String
3	Service	Kullanılan hedef ağ servisi	Kategorik	String
4	Flag	Bağlantının durumu – Normal ya da Hatalı	Kategorik	String
5	Src Bytes	Tek bağlantıda kaynaktan hedefe aktarılan veri baytı sayısı	Nümerik	Integer
6	Dst Bytes	Tek bağlantıda hedeften kaynağa aktarılan veri baytı sayısı	Nümerik	Integer
7	Land	Kaynak ve hedef IP adresleri ve port numaraları eşitse bu değişken 1, değilse 0	Nümerik	Integer
8	Wrong Fragment	Bağlantıdaki toplam yanlış parça sayısı	Nümerik	Integer
9	Urgent	Bağlantıdaki acil paketlerin sayısı. Acil paketler, acil biti etkinleştirilmiş paketlerdir.	Nümerik	Integer
10	Hot	İçerikteki “hot” göstergelerin sayısı (ör. sistem çağrıları, kilit dosya erişimi)	Nümerik	Integer
11	Num Failed Logins	Başarısız oturum açma girişimlerinin sayısı	Nümerik	Integer
12	Logged In	Giriş Durumu: Başarıyla giriş yapılmışsa 1; aksi takdirde 0	Nümerik	Integer
13	Num Compromised	“Tehlikeli” durumların sayısı	Nümerik	Integer
14	Root Shell	Root shell erişimi varsa 1, yoksa 0	Nümerik	Integer
15	Su Attempted	“su root” komutu kullanılmışsa 1, kullanılmamışsa 0	Nümerik	Integer
16	Num Root	Root erişimi sayısı veya root seviyesinde işlem sayısı	Nümerik	Integer
17	Num File Creations	Bağlantıdaki dosya oluşturma işlemlerinin sayısı	Nümerik	Integer
18	Num Shells	Shell istemlerinin sayısı	Nümerik	Integer
19	Num Access Files	Erişim kontrol dosyaları üzerindeki işlem sayısı	Nümerik	Integer
20	Num Outbound Cmds	FTP oturumundaki çıkış komutu sayısı	Nümerik	Integer
21	Is Hot Logins	Oturum açma işlemi, “hot” listeye, yani root veya admin'e aitse 1; aksi takdirde 0	Nümerik	Integer
22	Is Guest Login	Oturum açma işlemi, “misafir” oturum açma ise 1; aksi takdirde 0	Nümerik	Integer

Tablo 1'in devamı

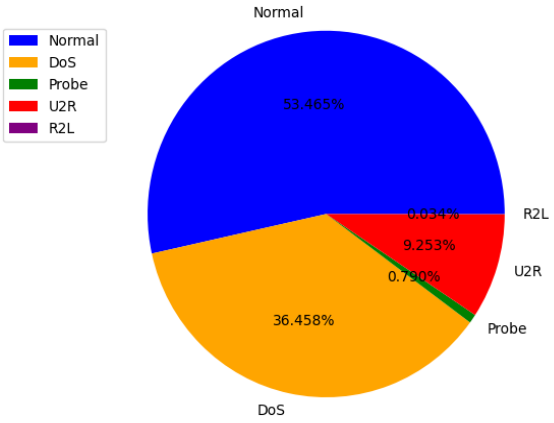
23	Count	Son iki saniye içinde geçerli bağlantıyla aynı hedef ana bilgisayara yapılan bağlantı sayısı	Nümerik	Integer
24	Srv Count	Son iki saniye içinde geçerli bağlantıyla aynı hizmete (bağlantı noktası numarası) yapılan bağlantı sayısı	Nümerik	Integer
25	Serror Rate	Count (23) içinde toplanan bağlantılar arasında s0, s1, s2 veya s3 bayrağını (4) etkinleştiren bağlantıların yüzdesi	Nümerik	Float
26	Srv Serror Rate	srv_count (24) içinde toplanan bağlantılar arasında s0, s1, s2 veya s3 bayrağını (4) etkinleştiren bağlantıların yüzdesi	Nümerik	Float
27	Rerror Rate	Count (23) olarak toplanan bağlantılar arasında REJ bayrağını (4) etkinleştiren bağlantıların yüzdesi	Nümerik	Float
28	Srv Rerror Rate	srv_count'ta (24) toplanan bağlantılar arasında REJ bayrağını (4) etkinleştiren bağlantıların yüzdesi	Nümerik	Float
29	Same Srv Rate	Count'ta (23) toplanan bağlantılar arasında aynı hizmete olan bağlantıların yüzdesi	Nümerik	Float
30	Diff Srv Rate	Count (23) olarak toplanan bağlantılar arasında farklı hizmetlere yapılan bağlantıların yüzdesi	Nümerik	Float
31	Srv Diff Host Rate	srv_count (24) içinde toplanan bağlantılar arasında farklı hedef makinelere giden bağlantıların yüzdesi	Nümerik	Float
32	Dst Host Count	Aynı hedef ana bilgisayar IP adresine sahip bağlantı sayısı	Nümerik	Integer
33	Dst Host Srv Count	Aynı bağlantı noktası numarasına sahip bağlantı sayısı	Nümerik	Integer
34	Dst Host Same Srv Rate	dst_host_count (32) içinde toplanan bağlantılar arasında farklı hizmetlere olan bağlantıların yüzdesi	Nümerik	Float
35	Dst Host Diff Srv Rate	dst_host_count (32) içinde toplanan bağlantılar arasında farklı hizmetlere olan bağlantıların yüzdesi	Nümerik	Float
36	Dst Host Same Src Port Rate	dst_host_srv_count (33) içinde toplanan bağlantılar arasında aynı kaynak bağlantı noktasına olan bağlantıların yüzdesi	Nümerik	Float
37	Dst Host Srv Diff Host Rate	dst_host_srv_count (33) içinde toplanan bağlantılar arasında farklı hedef makinelere olan bağlantıların yüzdesi	Nümerik	Float
38	Dst Host Serror Rate	dst_host_srv_count (32) içinde toplanan bağlantılar arasında s0, s1, s2 veya s3 bayrağını (4) etkinleştiren bağlantıların yüzdesi	Nümerik	Float

Tablo 1'in devamı

39	Dst Host Srv Serror Rate	dst_host_srv_count (33) içinde toplanan bağlantılar arasında s0, s1, s2 veya s3 bayrağını (4) etkinleştiren bağlantıların yüzdesi	Nümerik	Float
40	Dst Host Rerror Rate	dst_host_count(32) içinde toplanan bağlantılar arasında REJ bayrağını (4) etkinleştiren bağlantıların yüzdesi	Nümerik	Float
41	Dst Host Srv Rerror Rate	dst_host_srv_count (33) içinde toplanan bağlantılar arasında REJ bayrağını (4) etkinleştiren bağlantıların yüzdesi	Nümerik	Float
42	Class	Trafik girişinin sınıflandırılması	Kategorik	String
43	Difficulty Level	Zorluk seviyesi	Nümerik	Integer

2.2 VERİ SETİNİ TANIMAYA YÖNELİK YAPILAN ANALİZLER

Çalışma sırasında ilk olarak veri seti hakkında öngörü elde etmek amacıyla bazı analizler yapılmıştır. Öncelikli olarak veri setinde mevcut olan saldırıların ve normal kayıtların yüzdeleri incelenmiş olup elde edilen bilgiler pasta grafiği şeklinde Şekil 3'de verilmiştir.



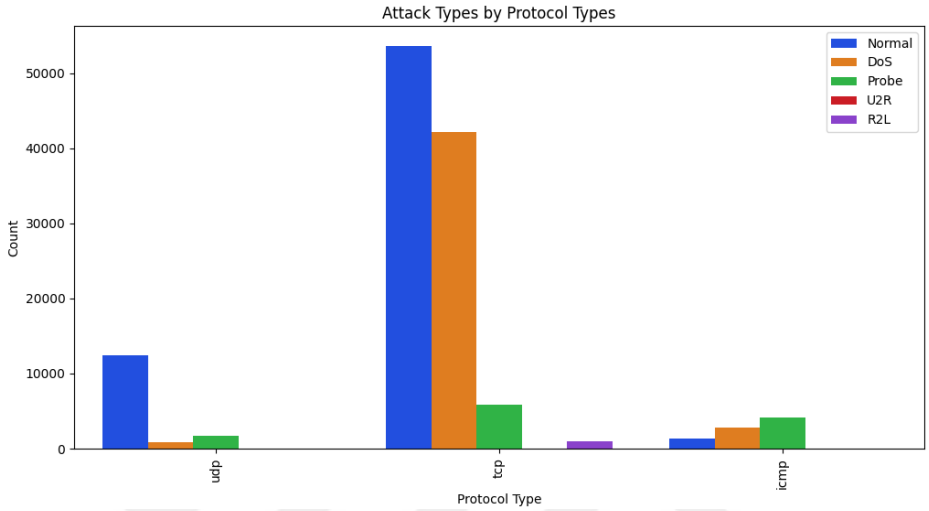
Şekil 3: Saldırıların oransal dağılımı

Pasta grafiğine bakıldığında, "Normal" olarak işaretlenen trafik kayıtları %53,46 ile veri setinin çoğunluğunu oluşturmaktadır. DoS saldırıları %36,46 oranında büyük bir yer tutmaktadır ve ağ tabanlı saldırı tespit sistemleri için en önemli kategorilerden biridir. U2R (User to Root) ve R2L (Remote to Local) saldırıları ise sırasıyla %9,25 ve %0,03 oranlarında daha nadir gözlemlenmektedir. Bu saldırılar düşük oranda temsil edilse de sistem üzerinde ciddi güvenlik açıklarına neden olabilecek türlerdir. Probe saldırıları sadece %0,79 oranında yer almaktadır.

Dağılımdaki bu dengesizliğin, çalışma süresince makine öğrenmesi tabanlı modellerin eğitimi sırasında dikkat edilmesi gereken önemli bir nokta olacağı

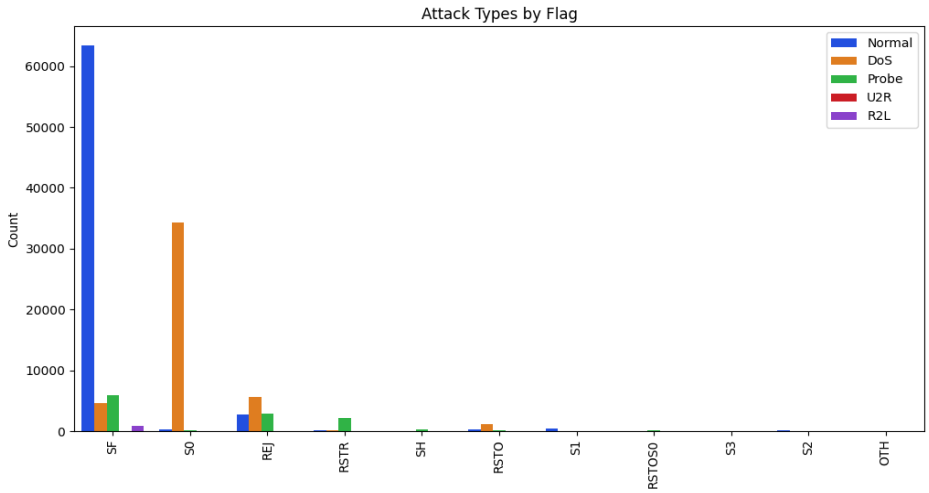
düşünülmüştür. Azınlıkta olan sınıfların öğrenilmesinin zor olacağı ve bunun da model performansını olumsuz etkileyebileceği öngörülmüştür.

Veri seti hakkında daha fazla fikir edinebilmek amacıyla kritik olduğu düşünülen özelliklerden protokol türlerine ve TCP/IP protokolündeki bayrak alanlarına göre dağılımları incelenmiştir.



Şekil 4: Protokol Tiplerine göre Saldırı Türleri

Şekil 4'de görüldüğü gibi, TCP protokolü hem normal hem saldırı verilerinde en yoğun kullanılan protokoldür. Bu özellikle DoS ve Probe türlerinin büyük çoğunlukla TCP üzerinden gerçekleştiğini göstermektedir. UDP protokolü, daha çok normal trafik için kullanılsa da bazı Probe saldırılarında da gözlemlenmektedir. ICMP protokolü ise, özellikle Probe ve kısmen DoS saldırılarında öne çıkmaktadır. Bu bulgular, ağ trafiğinin protokol tipine göre ayrıştırılmasının saldırıların tespiti açısından ayırt edici bir özellik olabileceğini göstermektedir.

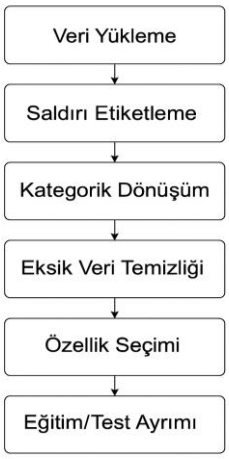


Şekil 5: Saldırı Türlerinin TCP/IP Bayraklarına Göre Dağılımı

Şekil 5’de saldırı türlerinin TCP/IP protokolündeki bayrak (flag) alanlarına göre dağılımı gösterilmiştir. Bu dağılıma göre, SF (Session Finished) bayrağı en çok "Normal" trafik ile ilişkilidir ve 60.000’den fazla kayıtla en baskın kategoridir. Bu durum, başarılı ve düzgün bir şekilde tamamlanan bağlantıların çoğunlukla normal trafik örnekleri olduğunu göstermektedir. S0 bayrağı ise büyük oranda DoS saldırılarıyla ilişkilendirilmiştir. REJ (Reject) ve RSTR (Reset) gibi bayraklar ise genellikle Probe ve daha az sıklıkla R2L saldırı türleriyle ilişkilidir. Bu durum, bu tür saldırıların hedef sistemin yanıt davranışını test etmeye çalıştığını ve bağlantının çoğu zaman reddedildiğini gösterir. Bu analiz, saldırı türlerinin TCP/IP bayraklarıyla doğrudan ilişkilendirilebileceğini ve bu bilgilerin anomali tespit sistemlerinde önemli bir özellik olarak kullanılabileceğini göstermektedir.

2.3 VERİ ÖN İŞLEME AŞAMASI

Bu çalışmada NSL-KDD veri seti üzerinde gerçekleştirilen makine öğrenmesi tabanlı anomali tespiti uygulaması için çeşitli ön işleme adımları uygulanmıştır. Veri ön işleme, modelin başarısını doğrudan etkileyen önemli bir aşama olup, veri setinin kalite ve uygunluk açısından hazırlanmasını amaçlamaktadır. Ön işleme sürecine ilişkin akış diyagramı aşağıda verilmiştir.



İlk olarak, veri setindeki “attack” sütunu analiz edilerek saldırı türleri beş ana kategoriye ayrılmıştır. Bunlar; Normal, Denial of Service (DoS), Probe, User to Root (U2R) ve Remote to Local (R2L) kategorileridir. Bu sınıflandırma, çalışmanın hedef değişkeni olan “category_attack” sütunu oluşturularak gerçekleştirilmiş ve her saldırı

tipi 0 ile 4 arasında bir sayısal değer ile temsil edilmiştir. Temsil edilen değerler Tablo 2'de gösterilmiştir.

Tablo 2: "category_attack" sütununa ilişkin etiket numaraları

Saldırı	***Etiket***
Normal	0
Denial of Service (DoS)	1
Probe	2
User to Root (U2R)	3
Remote to Local (R2L)	4

Veri setinde yer alan kategorik değişkenler (protocol_type, service, flag), makine öğrenmesi algoritmaları ile uyumlu hale getirilmek amacıyla "Label Encoding" yöntemiyle sayısal forma dönüştürülmüştür. Bu yöntem çalışmanın devamında detaylıca açıklanmıştır. Aşağıdaki tablo, LabelEncoder uygulanmadan önceki kategorik değerler ile dönüşüm sonrası karşılık gelen sayısal etiketleri göstermektedir. Bu dönüşüm sayesinde veriler makine öğrenmesi modelleri tarafından doğrudan işlenebilir hale gelmiştir. "Label encoding" uygulamasından önceki ve sonraki durum Tablo 3'de verilmiştir.

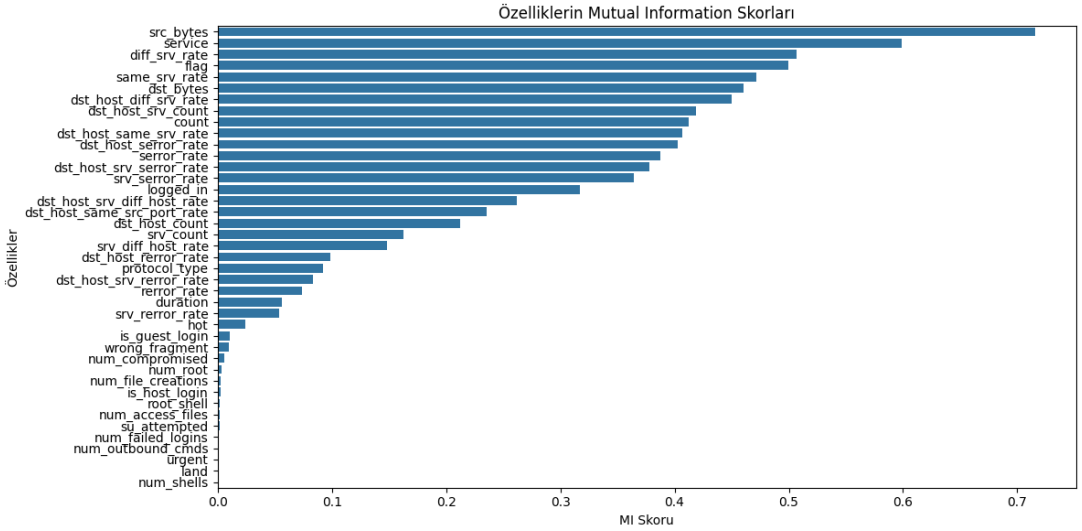
Tablo 3: Kategorik değişkenlere Label Encoder uygulanmadan önceki ve sonraki durum

Örnek No	**protocol_type (Öncesi)**	**protocol_type (Sonrası)**	**Service (Öncesi)**	**Service (Sonrası)**	**flag (Öncesi)**	**flag (Sonrası)**
1	tcp	0	http	3	SF	1
2	udp	1	domain_u	7	S0	0
3	icmp	2	eco_i	4	REJ	2

Kategorik değişkenlerin dönüştürülmesinin ardından, eksik verilerin analiz edilmesi ve çıkarılması işlemi gerçekleştirilmiştir. Gerekli kontroller sonucunda eksik (null) veri içeren örnekler veri setinden kaldırılmış ve modelin hatalı öğrenme riskinin önüne geçilmiştir.

Modelin başarımını artırmak amacıyla, özellik seçimi (feature selection) adımı uygulanmıştır. Bu kapsamda, her bir özelliğin hedef değişkenle olan ilişkisini ölçmek üzere "Mutual Information (MI)" yöntemi kullanılmıştır. MI, değişkenler arasındaki karşılıklı bilgi miktarını ölçerek hangi özelliklerin daha bilgilendirici olduğunu belirler. mutual_info_classif fonksiyonu yardımıyla yapılan bu işlem sonucunda, en yüksek bilgi kazancına sahip ilk 15 özellik seçilerek modelin bu sınırlı ama etkili değişkenler ile eğitilmesi sağlanmıştır. Sonrasında elde edilen sonuçlarda nasıl bir iyileşme sağlanacağının görülmesi amacıyla tüm değişkenler kullanılarak modeller

tekrarlanmıştır. Etkilerine ilişkin detaylı açıklamalar “Sonuçlar” kısmında verilmiştir. MI yöntemi ile elde edilen sıralamaya ilişkin bir örnek Şekil 6’da verilmiştir.



Şekil 6: Özelliklerin Mutual Information (MI) Skorları

Son olarak, nihai modelleme aşamasında kullanılmak üzere, category_attack dışındaki tüm sütunlar bağımsız değişkenler (X), category_attack ise bağımlı değişken (y) olarak ayrılmıştır. Bu yapılandırma sayesinde, modelin öğrenme süreci için uygun veri yapısı elde edilmiştir.

Label Encoding (Etiket Kodlaması): Makine öğrenmesi modelleri kategorik değişkenlere uygulanamamaktadır. Bu sebeple veri setinde yer alan kategorik değişkenler, modele beslenebilmesi amacıyla nümerik hale getirilmelidir. Bu çalışmada kategorik veriler “Label Encoding” yöntemi kullanılarak nümerik hale getirilmiştir. Scikit-Learn (sklearn) kütüphanesinin LabelEncoder fonksiyonu ile uygulanmıştır.

Veri Setini Dengelendirme: Bu çalışma kapsamında asıl odaklanılan nokta NSL-KDD veri setinde oldukça azınlıkta bulunan R2L ağ saldırıların tespiti idi. Bu durum elde edilen doğruluk değerlerinin istenilen seviyelere çıkmasına engel teşkil etmekteydi. Bu sebeple veri setinin SMOTE yöntemi kullanılarak dengeli hale getirilmesi amaçlandı.

SMOTE (Synthetic Minority Oversampling Technique) Yöntemi: Bu çalışmada kullanılan NSL-KDD veri setinde olduğu gibi bazı veriler sayıca azınlıkta (minority) olabilir. Bu tip durumlarda yapılacak çalışmada etkinliği artırmak için veride mevcut dengesizliğin azaltılması hedeflenir. Bu noktada çoğunlukla tercih

edilen yöntem SMOTE uygulamasıdır. Bu çalışmada da SMOTE yöntemi kullanılmıştır. SMOTE yöntemi, azınlık sınıfı örnekleri ve bunların k en yakın komşuları arasında doğrusal enterpolasyon gerçekleştirerek yeni sentetik veriler üretir. (Dina Elreedy vd. 2023:4903) SMOTE tekniğinin enterpolasyon mekanizması Şekil 7'de verilmiştir.

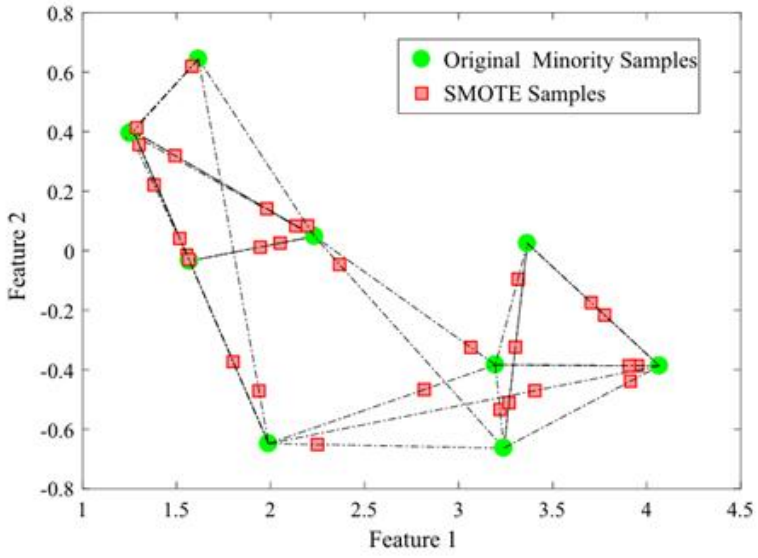


Fig. 1 The SMOTE interpolation mechanism displaying the original minority samples and the SMOTE generated patterns

Şekil 7: SMOTE enterpolasyon mekanizmasının gösterimi (Dina Elreedy vd. 2023:4903)

Özellik Seçimi: Özellik seçimi, makine öğrenmesinde kullanılan bir ön işleme tekniğidir ve veri kümesindeki en etkili özniteliklerin belirlenerek, modelin performansını artırmak ve öğrenme sürecini iyileştirmek amacıyla gerçekleştirilir. Veri kümesinde yer alan tüm özellikler model için her zaman faydalı olmayabilir. Bazı özellikler hedef değişkenle çok az ilişkilidir, bazıları ise birbirini tekrar edebilir. Bu tür gereksiz veya zararlı özelliklerin çıkarılması modelin karışıklığını ve aşırı öğrenme ihtimalini azaltırken, modelin eğitim süresini kısaltır.

Özellik seçimi yapmak amacıyla kullanılabilen çok sayıda yöntem vardır. Bunlar;

- Filtre Yöntemleri (Mutual Information (MI), Chi-squared (χ^2) testi, ANOVA F-test, Pearson korelasyon katsayısı
- Wrapper Yöntemler (Recursive Feature Elimination (RFE), Sequential Feature Selection)
- Embedded Yöntemler (Lasso Regresyonu) (Chandrashekar ve Sahin 2013:16), (Scikit-Learn Documentation 2025a)

Bu çalışma sırasında MI yöntemiyle ilişki durumuna göre değişkenler sıralandı. İlk 15 değişken ve tüm değişkenler kullanılarak sonuçlar elde edildi ve karşılaştırıldı. Bu sebeple bu tez çalışmasında yalnızca MI yöntemi ile ilgili teori yer almaktadır.

MI (Mutual Information): MI, bir özelliğin hedef değişkenle taşıdığı ortak bilgiyi ölçerek, hangi özelliklerin daha bilgilendirici olduğunu belirler. Bu yöntem, modelden bağımsız olup filtre yöntemleri kategorisine girer. Scikit-Learn (sklearn) kütüphanesinin mutual_info_classif fonksiyonu ile uygulanmıştır.

Mutual Information, rastgele değişkenler X ve Y arasındaki bilgi paylaşımını şu şekilde tanımlar:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2.1}$$

Burada p(x,y), X ve Y'nin ortak olasılık dağılımını, p(x) ve p(y) ise marjinal olasılıkları ifade eder. Bu formül, iki değişkenin birlikte gözlemlenme sıklığı ile bağımsız olasılıkları arasındaki farkı ölçer. MI değeri sıfıra yaklaştıkça değişkenler bağımsızdır; değer büyüdükçe aralarındaki bilgi paylaşımı artar. (Cover ve Thomas 2006)

2.4 KULLANILAN MAKİNE ÖĞRENMESİ MODELLERİ

Bu çalışmada kullanılan tüm makine öğrenmesi modellerine ilişkin teorik bilgi aşağıda sırasıyla açıklanmıştır.

2.4.1 Lojistik Regresyon

Lojistik regresyon, bağımlı değişkenin kategorik olduğu durumlarda kullanılan bir denetimli makine öğrenme algoritmasıdır. Bu yöntem bir olayın gerçekleşme olasılığını bağımsız değişkenler aracılığıyla tahmin etmeye çalışan bir istatistiksel yöntemdir. Lineer regresyonun aksine, lojistik regresyon çıktıları genellikle 0 ile 1 arasında olasılık değerleri üretir. (Hosmer vd. 2013)

Lojistik regresyon model, lineer regresyon modeline benzer bir yapıya sahiptir. Ancak modelde elde edilen doğrusal kombinasyon sigmoid fonksiyonu aracılığıyla 0-1 değer aralığına dönüştürülür. Bu dönüşüm aşağıdaki gibi ifade edilir:

$$p(x) = \frac{1}{1 + e^{-\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n}} \quad (2.2)$$

Burada,

$p(x)$: Bağımlı değişkenin 1 olma olasılığını,

β_i: x_i değişkenindeki 1 birimlik artışın olasılık oranına etkisini,

β_0: Sabit terimi ifade eder. (Kleinbaum ve Klein 2010)

Bu çalışmada Scikit-Learn kütüphanesinde LogisticRegression modeli kullanılmıştır. Aşağıda kullanılan model ve modele ilişkin değişkenler açıklanmıştır.

```
from sklearn.linear_model import LogisticRegression

Logistic_Regression_Classifier=LogisticRegression(C=10,
penalty='l1', solver='liblinear')

Logistic_Regression_Classifier.fit(train_df_selected,
y_train)
```

C parametresi, modelde uygulanan düzenlileştirme (regularization) miktarını kontrol eder. Daha teknik olarak,

$C = \frac{1}{\lambda}$ şeklinde ifade edilir ve burada λ, ceza terimini temsil eder.

C değeri büyüdükçe düzenlileştirmenin etkisi azalır. Bu çalışmada seçilen C=10 değeri, modelin katsayılarını çok fazla cezalandırmadan öğrenmesine olanak tanıyan, orta düzeyde zayıf bir düzenlileştirme anlamına gelmektedir. (Ng 2004)

Penalty parametresi, modelde hangi tür düzenlileştirmenin uygulanacağını belirtir. 'l1' değeri, Lasso düzenlileştirmesi anlamına gelir. L1 düzenlileştirme, bazı model katsayılarını sıfıra indirerek özellik seçimi sağlar ve modelin daha sade ve yorumlanabilir olmasına katkıda bulunur. (Ng 2004)

Solver parametresi, modelin optimizasyon sürecinde hangi algoritmayı kullanacağını belirler. Liblinear, özellikle küçük ve orta ölçekli veri kümelerinde etkin çalışan bir çözücüdür. Ayrıca liblinear, L1 düzenlileştirmesini destekleyen az sayıdaki çözücüden biridir ve bu sebeple L1 cezası tercih edildiğinde önerilen çözücüdür. (Pedregosa vd. 2011:2825)

Seçilen parametre değerleri kodlama aşamasında "GridSearchCV" metodu ile optimize edilerek seçilmiştir. "GridSearchCV" metodu, modelin etkinliğini artırmak için hiperparametre optimizasyonu yapmayı sağlayan bir metottur. "GridSearchCV", belirli bir parametre ızgarası (grid) üzerinde denemeler yaparak, modelin en iyi

performansı gösterdiği parametre kombinasyonunu belirlemeye çalışır. (Pedregosa vd. 2011:2825)

2.4.2 Naive Bayes

Bu algoritma, olasılıksal temellere dayanan ve yaygın olarak kullanılan bir denetimli öğrenme algoritmasıdır. Bayes teoremine dayabır ve öz nitelikler arasında koşulsal bağımsızlık varsayımı yapar. (Rish 2001:41)

Naive Bayes algoritması, bir sınıfa ait olma olasılığını aşağıdaki Bayes Teoremi ile hesaplar:

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \tag{2.3}$$

Burada,

P(C|X): Veri kümesinde X özelliklerine sahip bir örneğin C sınıfına ait olma olasılığı,

P(X|C) C sınıfı altında gözlemlenen X'in olasılığı,

P(C): Sınıfın genel olasılığı,

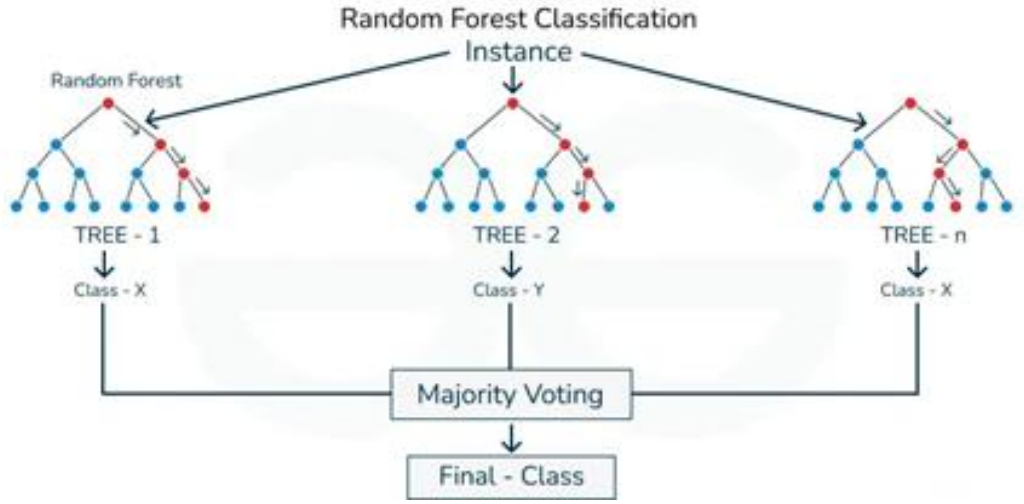
P(X): Özelliklerin gözlemlenme olasılığıdır. (Harry 2004:562)

Naive Bayes algoritması, özelliklerin dağılımına göre farklı şekillerde uygulanabilir. Bunlar; Gaussian, Multinomial ve Bernoulli Naive Bayes'dir. Bu çalışmada Scikit-Learn kütüphanesinden özelliklerin normal dağıldığını varsayan "GaussianNB" modeli kullanılmıştır.

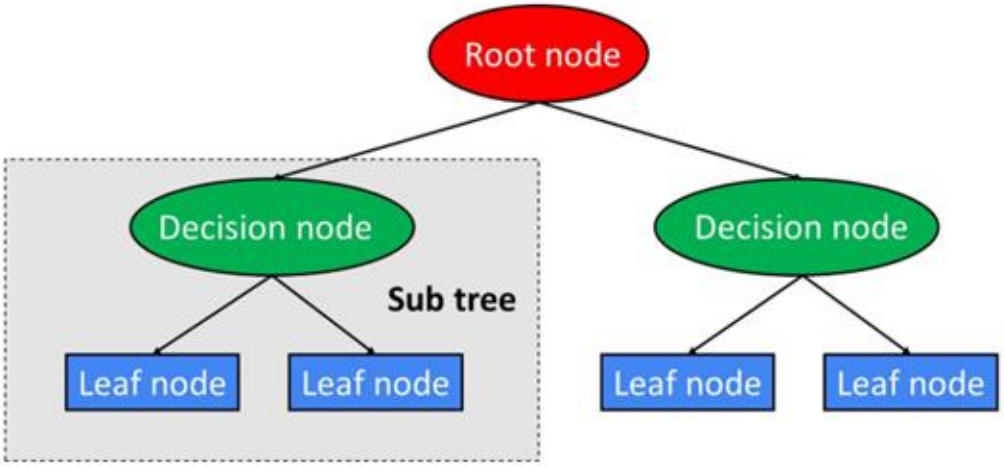
```
from sklearn.naive_bayes import GaussianNB

NB_Classifier=GaussianNB()
NB_Classifier.fit(train_df_selected,y_train
```

2.4.3 Rastgele Orman (Random Forest)

Bu algoritma sınıflandırma ve regresyon problemlerinde sıklıkla kullanılan bir çeşit denetimli öğrenme algoritmasıdır. Rastgele orman algoritması birden fazla karar ağacı oluşturarak bu ağaçlardan elde edilen sonuçlara göre nihai tahmin yapar. Random Forest Sınıflandırıcısının genel işleyişi Şekil 8'de, düğüm yapısı ise Şekil 9'da verilmiştir.



Şekil 8: Random Forest Sınıflandırıcısı (Geeks 2025)



Şekil 9: Random Forest Düğümleri (DataHackers 2025)

Bu çalışmada Scikit-Learn kütüphanesinde "RandomForestClassifier" modeli kullanılmıştır. Aşağıda kullanılan model ve modele ilişkin değişkenler açıklanmıştır. Modele ilişkin parametreler açıklanırken Rastgele Orman algoritmasında mevcut düğüm yapılarının net anlaşılabilmesi için Şekil 9'da verilen görselden faydalanabilirsiniz.

```
from sklearn.ensemble import RandomForestClassifier

Random_Forest_Classifier=RandomForestClassifier(n_estimators
=150,min_samples_split=2,min_samples_leaf=1,max_depth=20)
Random_Forest_Classifier.fit(train_df_selected,y_train)
```

Model içinde belirtilen "n_estimator" parametresi modelde kullanılacak ağaç sayısını belirtmektedir. "min_samples_split" değeri ise bir iç düğümün dallanabilmesi için gerekli olan minimum örnek sayısını vermektedir. Normal şartlarda bu değerin

küçük olması modelin aşırı öğrenme riskini artırabilir ancak bu çalışmada kullanılan veri seti için böyle bir sorun yaratmamıştır. “min_samples_leaf” parametresi ise leaf düğümlerinde bulunması gereken minimum örnek sayısını ifade etmektedir. “max_depth” parametresi ise her bir karar ağacının sahip olabileceği maksimum derinlik katmanını belirtir.

Bu çalışmada Rastgele Orman algoritması için belirlenen parametreler, “GridSearchCV” metodu kullanılarak optimize edilmiştir.

2.4.4 K En Yakın Komşu (K Nearest Neighbor-KNN)

Bu algoritma bir verinin gruplandırılması ile ilgili sınıflandırma tahmini yapmak için kullanılan bir çeşit denetimli öğrenme algoritmasıdır. Genel olarak KNN algoritmasının amacı sorgulanmak istenen noktaya en yakın “k” adet komşuyu uygun metotla belirleyerek sınıflandırma tahmini yapmaktır. KNN algoritmasını açıklayan ve yaygın olarak kullanılan diyagram Şekil 10’da verilmiştir.

Şekil 10: KNN Diyagramı (IBM, tarih yok)

Bu çalışmada Scikit-Learn kütüphanesinde KNeigborsClassifier modeli kullanılmıştır. Aşağıda kullanılan model ve modele ilişkin değişkenler açıklanmıştır.

```
from sklearn.neighbors import KNeighborsClassifier

KNN_Classifier_resampled=KNeighborsClassifier(n_neighbors=3
,weights='distance',algorithm='brute')
KNN_Classifier_resampled.fit(X_train_resampled,
y_train_resampled)
```

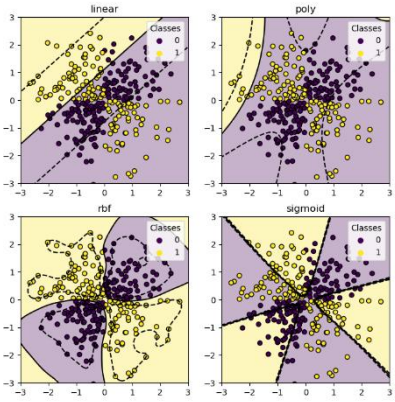
“n_neighbors” parametresi test edilen örneğin en yakın kaç komşusuna bakılacağını belirtir. “weights” parametresi komşuların tahmin üzerindeki etkilerinin nasıl ağırlıklandırılacağını gösterir. Distance seçildiğinde bu ağırlıklarındırma uzaklığa göre yapılır. Yani yakın komşulara daha fazla, uzak komşulara daha az ağırlık verilir. “algorithm” parametresi en yakın komşuları araken nasıl bir algoritma

izleneceğini belirtir. Brute algoritması tam kapsamlı bir arama algoritmasıdır, yani tüm veri noktalarının tek tek test noktasına olan uzaklığını öklid mesafesi cinsinden hesaplayarak karşılaştırır. Bu çalışmada kullanılan veri setinde brute algoritması uygulanabilirdir. Ancak daha büyük veri seti setlerinde modelin çok yavaş olmasına sebep olabilir. Ağırlık değerinin uzaklık olarak seçilmesi, yakın olan komşunun algoritma sırasında daha yüksek ağırlıkla ele alınmasını sağlayarak daha doğru tahmin yapılmasını sağlamaktır.

Bu çalışmada KNN algoritması için belirlenen parametreler, "RandomizedSearchCV" metodu kullanılarak optimize edilmiştir. RandomizedSearchCV, bir makine öğrenmesi modelinin hiperparametrelerini rastgele örnekleme yoluyla test eden bir arama yöntemidir. GridSearchCV'nin aksine tüm kombinasyonları denemek yerine belirli sayıda rastgele seçim yaparak daha hızlı hesaplama sağlar. Belirlenen hiperparametre dağılımı üzerinden rastgele seçilen sayıda model eğitilir ve çapraz doğrulama (cross-validation) ile test edilir. Böylece daha kısa sürede, genel olarak iyi bir parametre seti bulunabilir. (Bergstra ve Bengio 2012:281)

2.4.5 Destek Vektör Makineleri (Support Vector Machine-SVM)

Bu algoritma sınıflandırma ve regresyon problemleri için kullanılan bir denetimli öğrenme algoritmasıdır. Amacı, iki veya daha fazla sınıfı en geniş aralığı sağlayacak şekilde bir hiper düzeleme ayırmaktır. Bu hiper düzlem verilerin ait olduğu sınıfları en iyi şekilde ayıran düzlemdir. Karar aşamasında yalnızca bu düzleme en yakın olan veriler dikkate alınır. (Corinna ve Vapnik 1995:273) SVM algoritmasının en güçlü yönlerinden biri çekirdek (Kernel) fonksiyonlarıdır. Çekirdek fonksiyonları sayesinde doğrusal olarak ayrılmayan veriler de ayrılabilir hale gelebilmektedir. Bunlar, lineer, polinom, gaussian ve sigmoid çekirdek fonksiyonlarıdır. (Scholkopf ve Smola 2001). Bu fonksiyonlara ait sınıflandırma grafikleri Şekil 11'de verilmiştir. Bu çalışmada lineer çekirdek fonksiyonu kullanılmıştır.



Şekil 11: Destek Vektör Makineleri Sınıflandırma (Scikit-Learn Documentation 2025b)

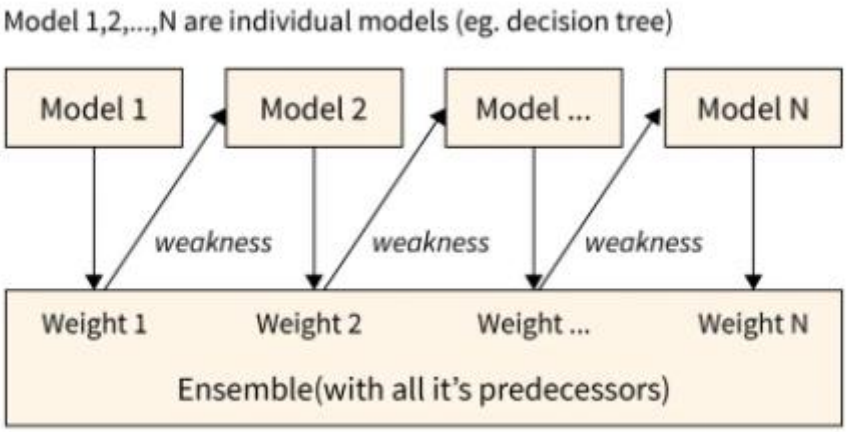
Bu çalışmada Scikit-Learn kütüphanesinde "LinearSVC" modeli default olarak kullanılmıştır. Aşağıda kullanılan model ve modele ilişkin değişkenler açıklanmıştır.

```
from sklearn.svm import LinearSVC

SVM_Classifier=LinearSVC()
SVM_Classifier.fit(train_df_selected,y_train)
```

Bu çalışmada SVM algoritması için belirlenen parametreler, "RandomizedSearchCV" metodu kullanılarak optimize edilmiştir.

2.4.6 AdaBoost (Adaptive Boosting)

Bu modeli açıklarken öncelikle modelin isminde yer "Boosting" teriminin anlamını açıklamanın anlaşılırlık açısından önem arz ettiğini düşünüyorum. Bir model eğitilirken, genellikle bu noktada karar ağaçları kullanılır, tüm veri noktaları ilk etapta eşit ağırlığa sahiptir. Model eğitimi tamamlandıktan sonra model tarafından yapılan hatalar tespit edilerek, hatanın ait olduğu veri noktaları modelin eğitiminde daha yüksek ağırlıklar verilecek şekilde tekrarlanır. Tahmin yaparken hata yapılan veri noktalarına daha fazla ağırlık verme eylemi "Boosting" olarak adlandırılır. (Majumdar 2023:244) Bu işlem iteratif olarak tekrarlanır. Sonunda minimum hata elde edilene kadar işlem devam eder. Tahmin yaparken minimum hatayı veren model seçilir. AdaBoost algoritması da bu mantığa dayanmaktadır. Bu algoritmanın çalışma prensibini açıklayan görsel Şekil 12'de verilmiştir. Ayrıca AdaBoost algoritmasının avantajları ve dezavantajları Tablo 4'de açıklanmıştır.



Şekil 12: AdaBoost Algoritmasının Çalışma Prensibi (Uniyal 2024)

Tablo 4: AdaBoost Algoritmasının Avantajları ve Dezavantajları (Uniyal 2024)

Avantajları	Dezavantajları
AdaBoost, birden fazla zayıf öğreniciyi daha güçlü bir sınıflandırıcıda birleştirerek genellikle bireysel modellerden daha iyi doğruluk sağlar.	AdaBoost aykırı değerlerden veya gürültülü veri noktalarından aşırı derecede etkilenebilir, bu da veriler bu tür birçok nokta içeriyorsa düşük performansa yol açabilir.
Algoritma, yanlış sınıflandırılmış noktalara odaklanarak hatalara uyum sağlar ve bu da karmaşık sınıflandırma görevlerini yerine getirme yeteneğini artırır.	Birden fazla öğreniciyi sırayla eğitmek, özellikle büyük veri kümeleri için veya karmaşık modeller kullanıldığında zaman alıcı olabilir.
AdaBoost'un scikit-learn gibi standart kütüphaneler kullanılarak uygulanması kolaydır ve karar ağaçları gibi çeşitli zayıf öğrenicilerle birleştirilebilir.	AdaBoost, uygun şekilde ayarlanmadığı takdirde, özellikle çok fazla zayıf öğrenici kullanıldığında veya model karmaşıklığı çok yüksekse aşırı uyum sağlayabilir.
Hem ikili hem de çok sınıflı sınıflandırma problemleriyle iyi çalışır ve regresyon görevlerini ele almak için de genişletilebilir.	AdaBoost büyük ölçüde zayıf öğrenicilerin performansına dayanır. Bu modeller çok basit veya etkisizse, nihai sınıflandırıcı iyi performans göstermeyebilir.
AdaBoost, zorlu durumları vurgulayarak, bir sınıfın yetersiz temsil edildiği dengesiz veri kümelerini etkili bir şekilde ele alabilir.	

Bu çalışmada Scikit-Learn kütüphanesinde AdaBoostClassifier modeli kullanılmıştır. Aşağıda kullanılan model ve modele ilişkin değişkenler açıklanmıştır.

```
from sklearn.ensemble import AdaBoostClassifier

AdaBoost_Classifier=AdaBoostClassifier(n_estimators=100,
algorithm='SAMME',learning_rate=1.0,random_state=42)
AdaBoost_Classifier.fit(train_df_selected,y_train)
```

“n_estimators”, modelin oluşturacağı zayıf sınıflayıcı (örneğin, karar ağaçları) sayısını belirtir. Daha fazla sınıflayıcı, modelin daha karmaşık yapıları öğrenmesine imkân tanır, ancak aşırı öğrenmeye de neden olabilir. Bu çalışmada 100 adet sınıflayıcı ile yeterli performans elde edilmiştir. “algorithm” temel sınıflayıcıların sınıf tahminlerine dayalı olarak ağırlıkları nasıl güncelleyeceğini belirler. (Freund ve Schapire 1996:119) AdaBoost algoritması Scikit-learn içinde iki farklı modda çalışabilir: SAMME ve SAMME.R. SAMME (Stagewise Additive Modeling using a Multiclass Exponential loss function), çok sınıflı sınıflandırma için kullanılan bir versiyondur. SAMME.R, sınıf olasılıklarını kullanarak genellikle daha hızlı ve daha doğru sonuçlar verebilir, ancak bu çalışmada SAMME tercih edilmiştir. “learning_rate”, her bir zayıf sınıflayıcının katkısını belirler. Daha düşük değerler modelin öğrenmesini yavaşlatabilirken, daha yüksek değerler modelin daha agresif öğrenmesini sağlar.Bu çalışmada varsayılan değer olan 1.0 kullanılmıştır. Literatürde bu değerin genellikle başlangıç için yeterli olduğu belirtilmektedir. (Fahresi vd. 2024:175) “random_state” için sabit bir başlangıç değeri belirlenmiştir. Bu sayede modelin her çalıştırıldığında aynı sonuçları vermesi sağlanır.

Bu çalışmada AdaBoost algoritması için belirlenen parametreler, “GridSearchCV” metodu kullanılarak optimize edilmiştir.

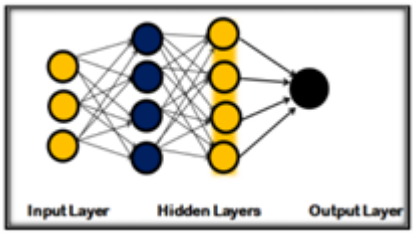
```
grid = GridSearchCV(estimator=model,
param_grid=param_grid,verbose=3,scoring='accuracy',n_jobs=-
1)
grid.fit(train_df_selected, y_train)
```

2.4.7 Yapay Sinir Ağları (Artificial Neural Network-ANN)

Yapay sinir ağları, insan beynindeki sinir hücrelerinden esinlenerek geliştirilmiş bir algoritmadır. Girdiler ve çıktılar arasındaki karmaşık ilişkilerde modellerin veya örüntülerin kurulduğu doğrusal olmayan bir veri modelleme sistemidir. Sinir ağları üstün öğrenme yeteneklerine sahiptir. Bunlar genellikle el yazısı ve yüz tanıma gibi daha karmaşık görevler için kullanılır. Sinir ağı aynı zamanda “perceptron” olarak da adlandırılır. (Qamar ve Zadari 2023:124) Sinir ağı şunlardan oluşur;

- "girdi" birimleri katmanı
- "kaplanmış" birimler katmanı
- "çıktı" birimleri katmanı

Veri, giriş katmanına gelir ve çıkışa ulaşana kadar katman ağ boyunca ilerler. Basit haliyle yapay sinir ağlarını açıklayan görsel Şekil 13'de verilmiştir.



Şekil 13: Yapay Sinir Ağları (Qamar ve Zadari, 2023)

Bu çalışmada Scikit-Learn kütüphanesinde MLPClassifier modeli kullanılmıştır. MLPClassifier methodu "multilayer perceptron" yani çok katmanlı perceptron mimarisini kullanan bir yapay sinir ağı modelidir. Aşağıda kullanılan model ve modele ilişkin değişkenler açıklanmıştır.

```
from sklearn.neural_network import MLPClassifier

ANN_Classifier = MLPClassifier(solver= 'adam', max_iter=
200, hidden_layer_sizes= (100,), activation= 'tanh')
ANN_Classifier.fit(train_df_selected, y_train)
```

"solver" parametresi ağırlıkları güncellemek için kullanılan optimizasyon algoritmasını belirtir. Bu çalışmada 'adam' algoritması kullanılmıştır. 'Adam' Algoritması uyarlanabilir öğrenme oranına sahip bir gradyan iniş yöntemidir ve genellikle hızlı ve etkili sonuçlar verir (Kingma ve Ba 2015). "max_iter" parametresi modelin belirlenen maksimum iterasyon boyunca eğitileceğini belirtir. Bu çalışmada 200 maksimum iterasyon kullanılmıştır. "hidden_layer_sizes" parametresi gizli katmanların yapısını belirtir. Bu çalışmada model, tek bir gizli katman içerir ve bu katmanda 100 adet yapay nöron bulunur. Gizli katman sayısı ve boyutu modelin öğrenme kapasitesini doğrudan etkiler. "activation" parametresi, gizli katmanlarda kullanılacak aktivasyon fonksiyonunu belirtir. Bu çalışmada 'tanh' fonksiyonu kullanılmıştır. Bu fonksiyon her nöronun çıktısını -1 ile 1 arasında sıkıştırır ve özellikle sıfır merkezli verilerde öğrenmeyi kolaylaştırabilir.

Bu çalışmada MLPClassifier algoritması için belirlenen parametreler, "RandomizedSearchCV" metodu kullanılarak optimize edilmiştir.

2.4.8 Model Değerlendirme Metrikleri

Çalışmada kullanılan makine öğrenmesi modellerinin başarısını değerlendirmek için çeşitli ölçütler kullanılır. Bu ölçütler, modelin ne kadar doğru tahmin yaptığını ve hangi tür hatalara daha yatkın olduğunu anlamaya yardımcı olur. Veri setinde mevcut olan dengesizlik sebebiyle birden fazla metrik ayrı ayrı ele alınarak incelenmiştir.

2.4.8.1 Doğruluk (Accuracy)

Doğruluk, modelin doğru tahmin ettiği örneklerin toplam örnek sayısına oranıdır. Ancak bu çalışmada olduğu gibi dengesiz veri setlerinde bu metrik yanıltıcı olabilir.

$$Doğruluk = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.4}$$

Burada;

TP: True Positive – Doğru Pozitif

TN: True Negative – Doğru Negatif

FP: False Positive – Yanlış Pozitif

FN: False Negative – Yanlış Negatif (Raschka ve Mirjialili 2017)

2.4.8.2 Kesinlik (Precision)

Kesinlik, yanlış pozitif sonuçların önemli olduğu durumlarda kullanılır. Modelin pozitif olarak tahmin ettiği örneklerin ne kadarının doğru pozitif olduğunu ölçer. Anomali tespit sistemlerinde yanlış pozitifler önem arz ettiği için bu çalışmada kullanılan modellerin değerlendirilmesinde göz önünde bulundurulmuştur. (Géron 2023)

$$Kesinlik = \frac{TP}{TP + FP} \tag{2.5}$$

2.4.8.3 Duyarlılık (Recall)

Duyarlılık, yanlış negatiflerin önemli olduğu durumlarda kullanılır. Gerçek pozitif örneklerin ne kadarının doğru bir şekilde tahmin edildiğini gösterir. Bu çalışmada olduğu gibi anomali tespit sistemlerinde duyarlılık önemli bir metriktir, bu sebeple modellerin değerlendirilmesinde ele alınmıştır. (Aggarwal 2015)

$$Duyarlılık = \frac{TP}{TP + FN} \tag{2.6}$$

2.4.8.4 F1 Skoru

F1 skoru, kesinlik ve duyarlılığın harmonik ortalamasıdır. Bu metrik, iki ölçütün dengesini sağlar ve dengesiz veri setlerinde sıklıkla kullanılmaktadır. (Sarker 2021:1)

$$F1 = 2 \,.\frac{Kesinlik \,.Duyarlılık}{Kesinlik + Duyarlılık} \tag{2.7}$$

BÖLÜM III
SONUÇLAR

Bir önceki bölümde detaylı şekilde açıklanmış olan veri ön işleme teknikleri kullanılarak modellere beslenmek için hazırlanan veri, yine önceki bölümde açıklanmış olan farklı makine öğrenmesi modellerine beslenmiştir. Gerçekleştirilen denemelere ilişkin sonuçlar aşağıda detaylı şekilde verilmiştir.

3.1 KULLANILAN MODELLERİN SONUÇLARI VE KARŞILAŞTIRMASI

Yapılan çalışmalara ait sonuçlar net bir şekilde irdeleyebilme açısından 2 başlık altında düzenlenmiştir.

3.1.1 Veri Seti Dengelendirme Öncesi

Tablo 5:SMOTE tekniği kullanılmadan elde edilen sonuçlar

Model	**Doğruluk (15 öznitelik)**	**Doğruluk (41 öznitelik)**
Random Forest	%56	%57
Logistic Regresyon	%52	%58
SVM	%48	%57
AdaBoost	%60	%61
Naive Bayes	%53	%49
KNN	%65	%66
ANN	%59	%59

Tablo 6: Random Forest modeli 15 öznitelik ile

Doğruluk: 0.74			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.64	0.97	0.77
1 (DoS)	0.96	0.76	0.85
2 (Probe)	0.83	0.58	0.68
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.60	0.00	0.00

Tablo 7: Logistic Regresyon modeli 15 öznitelik ile

Doğruluk: 0.69			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.63	0.93	0.75
1 (DoS)	0.90	0.69	0.78
2 (Probe)	0.59	0.55	0.57
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.53	0.00	0.01

Tablo 8: SVM modeli 15 öznitelik ile

Doğruluk: 0.73			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.77	0.77	0.77
1 (DoS)	0.70	0.97	0.81
2 (Probe)	0.67	0.65	0.66
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.00	0.00	0.00

Tablo 9: AdaBoost modeli 15 öznitelik ile

Doğruluk: 0.76			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.68	0.97	0.80
1 (DoS)	0.94	0.78	0.86
2 (Probe)	0.79	0.71	0.75
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.00	0.00	0.00

Tablo 10: Naive Bayes modeli 15 öznitelik ile

Doğruluk: 0.37			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.63	0.18	0.28
1 (DoS)	0.34	0.89	0.49
2 (Probe)	0.00	0.00	0.00
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.12	0.00	0.00

Tablo 11: KNN modeli 15 öznitelik ile

Doğruluk: 0.65			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.64	0.90	0.75
1 (DoS)	0.74	0.62	0.68
2 (Probe)	0.53	0.50	0.51
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.00	0.00	0.00

Tablo 12: ANN modeli 15 öznitelik ile

Doğruluk: 0.74			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.64	0.97	0.77
1 (DoS)	0.96	0.75	0.85
2 (Probe)	0.84	0.58	0.69
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.60	0.00	0.00

Tablo 13: Random Forest modeli 41 öznitelik ile

Doğruluk: 0.75			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.66	0.97	0.78
1 (DoS)	0.96	0.77	0.86
2 (Probe)	0.86	0.64	0.73
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.95	0.01	0.03

Tablo 14: Logistic Regresyon modeli 41 öznitelik ile

Doğruluk: 0.76			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.67	0.96	0.79
1 (DoS)	0.96	0.78	0.86
2 (Probe)	0.78	0.70	0.74
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.29	0.00	0.00

Tablo 15: SVM modeli 41 öznitelik ile

Doğruluk: 0.70			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.67	0.91	0.77
1 (DoS)	0.94	0.68	0.79
2 (Probe)	0.57	0.59	0.58
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.21	0.08	0.12

Tablo 16: AdaBoost modeli 41 öznitelik ile

Doğruluk: 0.77			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.68	0.97	0.80
1 (DoS)	0.96	0.82	0.88
2 (Probe)	0.80	0.67	0.73
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.80	0.00	0.01

Tablo 17: Naive Bayes modeli 41 öznitelik ile

Doğruluk: 0.29			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.24	0.03	0.06
1 (DoS)	0.32	0.82	0.46
2 (Probe)	0.00	0.00	0.00
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.00	0.00	0.00

Tablo 18: KNN modeli 41 öznitelik ile

Doğruluk: 0.66			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.66	0.93	0.77
1 (DoS)	0.69	0.67	0.68
2 (Probe)	0.52	0.27	0.36
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.00	0.00	0.00

Tablo 19: ANN modeli 41 öznitelik ile

Doğruluk: 0.76			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.66	0.97	0.79
1 (DoS)	0.96	0.78	0.86
2 (Probe)	0.86	0.65	0.74
3 (U2R)	0.00	0.00	0.00
4 (R2L)	1.00	0.01	0.03

3.1.2 Veri Seti Dengelendirme Sonrası

Tablo 20: SMOTE tekniği kullanıldıktan sonra sonuçlar

Model	**Doğruluk (15 öznitelik)**	**Doğruluk (41 öznitelik)**
Random Forest	0.75	0.75
Logistic Regression	0.69	0.78
SVM	0.69	0.78
AdaBoost	0.68	0.71
Naive Bayes	0.67	0.77
KNN	0.77	0.76
ANN	0.74	0.76

Tablo 21: Random Forest (SMOTE) modeli 15 öznitelik ile

Doğruluk: 0.75			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.66	0.97	0.79
1 (DoS)	0.94	0.76	0.84
2 (Probe)	0.77	0.64	0.70
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.50	0.00	0.01

Tablo 22: Logistic Regresyon (SMOTE) modeli 15 öznitelik ile

Doğruluk: 0.69			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.71	0.81	0.76
1 (DoS)	0.90	0.69	0.78
2 (Probe)	0.50	0.80	0.61
3 (U2R)	0.00	0.20	0.01
4 (R2L)	0.36	0.16	0.22

Tablo 23: SVM (SMOTE) modeli 15 öznitelik ile

Doğruluk: 0.69			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.70	0.81	0.75
1 (DoS)	0.90	0.68	0.78
2 (Probe)	0.49	0.80	0.64
3 (U2R)	0.00	0.20	0.01
4 (R2L)	0.38	0.15	0.22

Tablo 24: AdaBoost (SMOTE) modeli 15 öznitelik ile

Doğruluk: 0.68			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.67	0.76	0.71
1 (DoS)	0.94	0.74	0.83
2 (Probe)	0.60	0.86	0.71
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.15	0.10	0.12

Tablo 25: Naive Bayes (SMOTE) modeli 15 öznitelik ile

Doğruluk: 0.67			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.74	0.76	0.75
1 (DoS)	0.96	0.71	0.82
2 (Probe)	0.62	0.67	0.64
3 (U2R)	0.00	0.40	0.00
4 (R2L)	0.35	0.24	0.28

Tablo 26: KNN (SMOTE) modeli 15 öznitelik ile

Doğruluk: 0.77			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.71	0.93	0.81
1 (DoS)	0.94	0.81	0.87
2 (Probe)	0.76	0.77	0.76
3 (U2R)	0	0	0
4 (R2L)	0.47	0.12	0.19

Tablo 27: ANN (SMOTE) modeli 15 öznitelik ile

Doğruluk: 0.74			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.69	0.89	0.78
1 (DoS)	0.88	0.81	0.84

Tablo 27'nin devamı

2 (Probe)	0.64	0.64	0.64
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.62	0.08	0.14

Tablo 28: Random Forest (SMOTE) modeli 41 öznitelik ile

Doğruluk: 0.76			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.66	0.97	0.79
1 (DoS)	0.96	0.77	0.85
2 (Probe)	0.85	0.62	0.72
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.97	0.07	0.13

Tablo 29: Logistic Regresyon (SMOTE) modeli 41 öznitelik ile

Doğruluk: 0.78			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.79	0.90	0.84
1 (DoS)	0.91	0.75	0.82
2 (Probe)	0.81	0.87	0.84
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.72	0.33	0.46

Tablo 30: SVM (SMOTE) modeli 41 öznitelik ile

Doğruluk: 0.78			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.73	0.90	0.80
1 (DoS)	0.89	0.75	0.81
2 (Probe)	0.82	0.84	0.83
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.73	0.36	0.48

Tablo 31: AdaBoost (SMOTE) modeli 41 öznitelik ile

Doğruluk: 0.74			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.75	0.95	0.83
1 (DoS)	0.97	0.59	0.74
2 (Probe)	0.44	0.80	0.57
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.87	0.30	0.45

Tablo 32: Naive Bayes (SMOTE) modeli 41 öznitelik ile

Doğruluk: 0.76			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.87	0.83	0.85
1 (DoS)	0.89	0.78	0.83
2 (Probe)	0.63	0.84	0.72
3 (U2R)	0.00	0.20	0.00
4 (R2L)	0.55	0.36	0.43

Tablo 33: KNN (SMOTE) modeli 41 öznitelik ile

Doğruluk: 0.76			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.67	0.96	0.79
1 (DoS)	0.96	0.76	0.85
2 (Probe)	0.80	0.73	0.76
3 (U2R)	0	0	0
4 (R2L)	0.84	0.04	0.08

Tablo 34: ANN (SMOTE) modeli 41 öznitelik ile

Doğruluk: 0.78			
Sınıf	Kesinlik	Duyarlılık	F1 Skoru
0 (Normal)	0.72	0.92	0.81
1 (DoS)	0.97	0.81	0.88
2 (Probe)	0.82	0.71	0.76
3 (U2R)	0.00	0.00	0.00
4 (R2L)	0.57	0.21	0.31

3.2 SONUCA DAYALI YORUM

Bir önceki bölümde Tablo 5 ile Tablo 34 arasında yer alan tüm sayısal değerlere ilişkin yorumlama bu bölümde yapılmıştır. Tez çalışması süresinde modellerin doğruluk değerlerinin yanıltıcı olmaması amacıyla kesinlik, duyarlılık ve f1 skorları da bu sebeple dikkate alınmış ve ayrı ayrı değerlendirilmiştir. Yorumlamalar kullanılan model bazlı olarak tek tek ele alınacaktır.

- Random Forest

Tablo 5, Tablo 6, Tablo 13, Tablo 20, Tablo 21 ve Tablo 28'de verilen sonuçlara dayalı olarak Random Forest modeli incelendiğinde ilk olarak 15 öz nitelik kullanılması ya da tüm öz niteliklerin kullanılması durumları arasında büyük bir farklılık görülmemiştir. Aynı zamanda veriyi dengeli hale getirmek adına uygulanan SMOTE tekniği de sonuçlarda ciddi bir iyileşmeye sebep olmamıştır. Bu çalışmada geliştirilen Random Forest modellerinin duyarlılığı genel olarak "Normal" ve "DoS" sınıflarında yüksek olup "Probe" sınıfında ortalama bir seviyededir. Ancak "U2R" ve "R2L" sınıflarına duyarlı değillerdir.

- Lojistik Regresyon

Tablo 5, Tablo 7, Tablo 14, Tablo 20, Tablo 22 ve Tablo 29'da verilen sonuçlara dayalı olarak Lojistik Regresyon modeli incelendiğinde ise ilk olarak 15 öz nitelik kullanılması durumuna kıyasla tüm öz nitelikler kullanıldığında modelin doğruluk değerinde artış gözlemlenmiştir. Aynı zamanda veriyi dengeli hale getirmek adına uygulanan SMOTE tekniği de sonuçlarda ciddi bir iyileşmeye sebep olmasa bile özellikle "R2L" sınıfı için kesinlik, duyarlılık ve f1 skor değerlerinde iyileşmeye sebep

olmuştur. Bu çalışmada geliştirilen Lojistik Regresyon modellerinin duyarlılığı genel olarak “Normal”, “DoS” ve “Probe” sınıflarında yüksek iken “R2L” sınıfı için düşüktür. Ancak “U2R” sınıfına duyarlı değillerdir.

- SVM

Tablo 5, Tablo 8, Tablo 15, Tablo 20, Tablo 23 ve Tablo 30’da verilen sonuçlara dayalı olarak SVM modeli incelendiğinde ilk olarak 15 öz nitelik kullanılması durumuna kıyasla tüm öz nitelikler değerlendirmeye alındığında doğruluk değerinin biraz azaldığı gözlemlenmiştir. Aynı zamanda veriyi dengeli hale getirmek adına uygulanan SMOTE tekniği de sonuçların iyileşmesine katkı sağlamamıştır. Her durumda DoS saldırıları ve Probe saldırılarına modelin iyi seviyede duyarlı, U2R ve R2L saldırıları için duyarlı olmadığı gözlemlenmiştir.

- AdaBoost

Tablo 5, Tablo 9, Tablo 16, Tablo 20, Tablo 24 ve Tablo 31’de verilen sonuçlara dayalı olarak AdaBoost modeli incelendiğinde ilk olarak 15 öz nitelik kullanılması durumuna kıyasla tüm öz nitelikler değerlendirmeye alındığında doğruluk değerinin etkilenmediği gözlemlenmiştir. Aynı zamanda veriyi dengeli hale getirmek adına uygulanan SMOTE tekniği de sonuçların iyileşmesine katkı sağlamamıştır. Modelin DoS saldırıları ve Probe saldırıları için oldukça duyarlı iken U2R ve R2L saldırıları için duyarlı olmadığı gözlemlenmiştir.

- Naive Bayes

Tablo 5, Tablo 10, Tablo 17, Tablo 20, Tablo 25 ve Tablo 32’de verilen sonuçlara dayalı olarak Naive Bayes modeli incelendiğinde ilk olarak 15 öz nitelik kullanılması durumuna kıyasla tüm öz nitelikler değerlendirmeye alındığında doğruluk değerinin olumlu etkilenmediği gözlemlenmiştir. Naive Bayes modeli SMOTE işlemi yapılmadan önce oldukça kötü sonuçlar vermiştir. Ancak SMOTE işlemi sonrasında doğruluk, duyarlılık, kesinlik ve f1 skor metriklerinde büyük bir iyileşme gözlemlenmiştir. Buradan yola çıkarak Naive Bayes modelinin veri setinin denge durumundan en çok etkilenen model olduğu farkedilmiştir. Ancak diğer modellerde olduğu gibi, bu model de R2L ve U2R saldırılarını tespit etmekte yetersiz kalırken SMOTE işlemi sonrası Normal, DoS ve Probe sınıflarını iyi seviyede tespit edebilmektedir.

- KNN

Tablo 5, Tablo 11, Tablo 18, Tablo 20, Tablo 26 ve Tablo 33'de verilen sonuçlara dayalı olarak KNN modeli incelendiğinde ilk olarak 15 öz nitelik kullanılması durumuna kıyasla tüm öz nitelikler değerlendirmeye alındığında doğruluk değerinin etkilenmediği gözlemlenmiştir. Tez çalışması sırasında denenen tüm modeller içerisinde en yüksek doğruluk değeri SMOTE uygulamasından önce, KNN modeli kullanılarak elde edilmiştir. SMOTE uygulamasından sonraki durumda KNN modeli ile elde edilen doğruluk değeri diğer modellere yakın olmakla birlikte R2L sınıfına ilişkin en başarılı veriler KNN modelinde elde edilmiştir. KNN modeli Normal, DoS ve Probe saldırı sınıflarına iyi seviyede duyarlıyken, R2L saldırıları için yetersiz kalmış ve U2R saldırıları içinse tamamen başarısız olmuştur. Ancak elde edilen verilere göre veri dengeli hale getirme işlemi sonrası R2L saldırısı için mevcut veri seti ile elde edilen en iyi sonuç da bu model ile elde edilmiştir.

- ANN

Tablo 5, Tablo 12, Tablo 19, Tablo 20, Tablo 27 ve Tablo 34'de verilen sonuçlara dayalı olarak ANN modeli incelendiğinde ilk olarak 15 öz nitelik kullanılması durumuna kıyasla tüm öz nitelikler değerlendirmeye alındığında doğruluk değerinde büyük bir değişiklik olmadığı gözlemlenmiştir. KNN modeli de Normal, DoS ve Probe saldırı sınıflarına iyi seviyede duyarlıyken U2R ve R2L saldırıları için yetersiz kalmıştır. Veri dengeli hale getirme aşamasının bu model için de etkili olmadığı gözlemlenmiştir.

Genel olarak sonuçlar incelendiğinde görünen şu ki; veri setinde mevcut olan dengesizlik sebebiyle geliştirilen modeller her saldırı türü için iyi sonuçlar vermemektedir. Özellikle U2R ve R2L sayılarının veri seti içerisindeki yüzdesinin düşük olması sebebiyle bu saldırı türleri için kullanılan modeller uygun sonuçlar vermemektedir. Veri setini dengeli hale getirme işlemi bu saldırı türleri için sonuçları biraz iyileştirse de yine de sonuçlar yetersiz kalmıştır. Ancak Normal, DoS ve Probe sınıflarının veri setindeki yoğunluğunun da etkisiyle bu sınıflara mahsus olaylar çok daha iyi şekilde tespit edilebilmektedir.

Bu çalışma süresince yapılan çalışmalar sonucunda U2R ve R2L saldırıları için veri setinde yeterli örnek olmaması sebebiyle bu saldırılar tespit edilememiştir. Veri setini dengeli hale getirme işlemi bu saldırı türleri için sonuçları biraz iyileştirse de yine de sonuçlar yetersiz kalmıştır. Ancak Normal, DoS ve Probe sınıflarının veri

setindeki yoğunluğu sayesinde bu sınıflara mahsus olaylar çok daha iyi şekilde tespit edilebilmektedir.

Bunlara ek olarak modeller ayrı ayrı incelendiğinde Naive Bayes modelinin veri setinin denge durumundan en çok etkilenen model olduğu gözlemlenmiştir.

Gelecek çalışmalara ilişkin yorum yapmak gerekirse, son yıllarda tabular veriler üzerinde doğrudan çalışan yeni derin öğrenme mimarileri geliştirilmiştir. Özellikle SAINT (Self-Attention and Intersample Attention Transformer) ve TabPFN (Tabular Prior-Data Fitted Networks) modelleri, klasik yöntemlere kıyasla daha güçlü öğrenebilme kabiliyetleri ile öne çıkmaktadır.

SAINT, Transformer mimarisine dayalı olarak hem öz nitelikler arasındaki ilişkileri hem de örnekler arası bağımlılıkları aynı anda modelleyebilen bir yapıya sahiptir. Bu sayede karmaşık etkileşimlerin bulunduğu tabular veri setlerinde başarılı sonuçlar elde edebilmektedir. (Somepalli vd. 2021)

TabPFN, önceden büyük miktarda sentetik veri üzerinde eğitilmiş bir "prior-fitted" ağdır. Küçük veri setlerinde bile birkaç adımda hızlı ve yüksek doğrulukla sonuçlar üretebilmektedir. Bu özellik dengesiz dağılıma sahip veriler ile yapılan çalışmalarda önemli avantajlar sağlayabilmektedir. (Hollmann vd. 2023)

Dolayısıyla ilerleyen çalışmalarda, SAINT ve TabPFN gibi daha modern derin öğrenme tabanlı yöntemlerin NSL-KDD ve benzeri siber güvenlik veri setleri üzerinde uygulanması, daha yüksek doğruluk ve genelleme başarısı elde edilmesine katkı sağlayabilir.

KAYNAKÇA

AGGARWAL Charu C. (2015), *Data Mining: The Textbook*, Springer, New York.

AHMED Mohiuddin, MAHMOOD Abdun Naser ve HU Jiankun (2015), "A Survey of Network Anomaly Detection Techniques", *Journal of Network and Computer Applications,* Sayı 60, ss.19-31.

ALJANABI Mohammad, ISMAIL Mohd Arfian, HASAN Raed Abdulkareem ve SULAIMAN Junaida (2021), "Intrusion Detection: A Review", *Mesopotamian Journal of Cybersecurity*, Cilt 2021, ss. 1–4.

AMASARANI Isuri (2022), "DoS vs. DDoS Attacks", https://medium.com/@isuriamasarani87/dos-vs-ddos-attacks-bde2ef270b40, ET. 24.05.2024.

BERGSTRA James ve BENGIO Yoshua (2012), "Random Search for Hyper-Parameter Optimization", *Journal of Machine Learning Research*, Cilt 13, ss. 281–305.

CHANDRASHEKAR Girish ve SAHIN Ferat (2013), "A Survey on Feature Selection Methods", *Computers and Electrical Engineering*, Cilt 40, Sayı 1, ss. 16–28.

CORTES Corinna ve VAPNIK Vladimir (1995), "Support-Vector Networks", *Machine Learning*, Cilt 20, Sayı 3, ss. 273–297.

COVER Thomas M. ve THOMAS Joy A. (2006), *Elements of Information Theory*, Wiley-Interscience Publication, New Jersey.

DATA HACKERS (2025), "#012 Machine Learning – Introduction to Random Forest", https://datahacker.rs/012-machine-learning-introduction-to-random-forest/, ET. 22.04.2025.

ELREEDY Dina, ATIYA Amir F. ve KAMALOV Firuz (2023), "A Theoretical Distribution Analysis of Synthetic Minority Oversampling Technique (SMOTE) for Imbalanced Learning", *Machine Learning*, Sayı 113, ss. 4903–4923.

ENISA (2024), “Rootkits”, https://www.enisa.europa.eu/topics/incident-response/glossary/rootkits, E.T: 24.05.2024.

FAHRESI Sahrul Yudha, NUGRAHA Adhitya, LUTHFIARTA Ardytha ve PRIMADYA Nauval Dwi (2024), “Optimizing Performance of AdaBoost Algorithm Through Undersampling and Hyperparameter Tuning on CICIoT 2023 Dataset”, *Techné Jurnal Ilmiah Elektroteknika*, Cilt 23, Sayı 2, ss. 175–184.

FREUND Yoav ve SCHAPIRE Robert E. (1996), “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”, *Journal of Computer and System Sciences*, Cilt 55, Sayı 1, ss. 119–139.

GÉRON Aurélien (2023), *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3. Baskı, O'Reilly Media, California.

GEKS FOR GEEKS (2025), “Random Forest Classifier Using Scikit-learn”, https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/, ET. 22.04.2025.

HOLLMANN Noah, MÜLLER Samuel, EGGENSPERGER Katharina ve HUTTER Frank (2023), “TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second”, *International Conference on Learning Representations (ICLR) 2023,*Kigali,Rwanda.

IBM (2025), “KNN Algorithm”, https://www.ibm.com/think/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20(KNN)%20algorithm..., ET. 24.03.2025.

IMPERVA (2024), “Buffer Overflow Attack”, https://www.imperva.com/learn/application-security/buffer-overflow/, ET. 24.05.2024.

IMPERVA (2024), “DDoS Attacks”, https://www.imperva.com/learn/ddos/ddos-attacks, ET. 21.05.2024.

IQBAL H. SARKER (2021), “Machine Learning: Algorithms, Real-World Applications and Research Directions”, *SN Computer Science*, Cilt 2, Sayı 3, ss. 1–21.

JAKOBSSON Markus ve MYERS Steven (2006), *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*, Wiley, New Jersey.

KLEINBAUM D. G. ve KLEIN M. (2010), *Logistic Regression: A Self-Learning Text*, 3. Baskı, Springer, Atlanta.

KINGMA Diederik P. ve LEI BA Jimmy (2015), "Adam: A Method for Stochastic Optimization", *International Conference on Learning Representations (ICLR) 2015,* San Diego, CA, ABD.

LOG360 (2024), "DoS and DDoS Attacks", https://www.manageengine.com/log-management/cyber-security-attacks/what-is-denial-of-service-attack.html, ET. 21.05.2024.

MAJUMDAR Partha (2023), *Mastering Classification Algorithms for Machine Learning*, BPB Online, Londra, ss. 244.

MALWARE BYTES (2024), "Rootkit", https://www.malwarebytes.com/rootkit, ET. 24.05.2024.

MOCKAPETRIS P. (1987), "Domain Implementation and Specification", www.ietf.org/rfc/rfc1035.txt, ET. 29.05.2024.

MIRKOVIC Jelena ve REIHER Peter (2004), "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms", *ACM SIGCOMM Computer Communication Review*, Cilt 34, Sayı 2, ss. 39–53.

MOHIT UNIAL (2024), "AdaBoost Algorithm in Machine Learning", https://www.appliedaicourse.com/blog/adaboost-algorithm-in-machine-learning/, ET. 28.05.2025.

NG Andrew Y. (2004), "Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance", *Proceedings of the 21st International Conference on Machine Learning (ICML-04),* ss 78, ACM, New York.

PEDREGOSA Fabian, Varoquaux Gaël, Gramfort Alexandre, Michel Vincent, Thirion Bertrand, Grisel Olivier, Blondel Mathieu, Prettenhofer Peter, Weiss Ron, Dubourg Vincent, Vanderplas Jake, Passos Alexandre, Cournapeau David, Brucher Matthieu, Perrot Matthieu, ve Duchesnay Édouard (2011), "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, Cilt 12, ss. 2825–2830.

PFLEEGER P. Charles ve PFLEEGER Shari Lawrence (2007), *Security in Computing*, 4. Baskı, Prentice Hall, New Jersey.

PURKAIT Swapan (2015), “Examining the Effectiveness of Phishing Filters Against DNS Based Phishing Attacks”, *Information & Computer Security*, Cilt 23, Sayı 3, ss. 333–346.

SOMEPALLI Gowthami, GOLDBLUM Micah, SCHWARZSCHILD Avi, BRUSS C. Bayan ve GOLDSTEIN Tom (2021), “SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training”, https://www.researchgate.net/publication/352081653_SAINT_Improved_Neural_Networks_for_Tabular_Data_via_Row_Attention_and_Contrastive_Pre-Training, ET. 16.09.2025

QAMAR Roheen ve ZADARI Baqar Ali (2023), “Artificial Neural Networks: An Overview”, *Mesopotamian Journal of Computer Science*, Cilt 2023, ss. 124–133.

RASCHKA Sebastian ve MIRJIALILI Vahid (2017), *Python Machine Learning – Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*, 3. Baskı, Packt Publishing, Birmingham.

REVATHI S. ve MALATHI A. (2013), “A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection”, *International Journal of Engineering Research & Technology (IJERT)*, Cilt 2, Sayı 12, ss. 1848–1853.

RISH Irina (2001), “An Empirical Study of the Naive Bayes Classifier”, *IJCAI Workshop on Empirical Methods in Artificial Intelligence*, Cilt 3, Sayı 22, ss. 41–46.

SARKER Iqbal H. (2021), “Machine Learning: Algorithms, Real-World Applications and Research Directions”, *SN Computer Science*, Cilt 2, Sayı 3, ss. 1–21.

SCHOLKOPF B. ve SMOLA A. J. (2001), *Learning with kernels: Support vector machines, regularization, optimization, and beyond*, MIT Press, Londra.

SCIKIT-LEARN DOCUMENTATION (2025a), “Feature Selection”, https://scikit-learn.org/stable/modules/feature_selection.html, ET. 30.05.2025.

SCIKIT-LEARN DOCUMENTATION (2025b), “Plot Classification Boundaries with Different SVM Kernels”, https://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html?utm_source=, ET. 24.06.2025.

STALLINGS William (2017), *Network Security Essentials: Applications and Standards*, 6. Baskı, Pearson, Essex.

TAVALLAEE Mahbod, BAGHERI Ebrahim, LU Wei ve GHORBANI Ali A. (2009), "A Detailed Analysis of the KDD CUP 99 Data Set", *IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)*, ss. 1–6, Ottawa.

UNIAL Mohit (2024), "AdaBoost Algorithm in Machine Learning", https://www.appliedaicourse.com/blog/adaboost-algorithm-in-machine-learning/, ET. 28.05.2025.

W3SCHOOLS (2024), "Cyber Security Network Mapping & Port Scanning", https://www.w3schools.com/cybersecurity/cybersecurity_mapping_port_scanning.php, ET. 24.05.2024.

ZHANG Harry (2004), "The Optimality of Naive Bayes", *AAAI*, Cilt 1, Sayı 2, ss. 562–567.