MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KOLEKTİF MAKİNE ÖĞRENMESİ TABANLI AĞ SALDIRI TESPİTİ

ŞURA EMANET

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı

DANIŞMAN
Doç. Dr. Önder Demir

EŞ-DANIŞMAN
Dr. Öğr. Üyesi Gözde Karataş Baydoğmuş

İSTANBUL, 2022

MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KOLEKTİF MAKİNE ÖĞRENMESİ TABANLI AĞ SALDIRI TESPİTİ

ŞURA EMANET

523618021

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı

DANIŞMAN
Doç. Dr. Önder Demir

EŞ-DANIŞMAN
Dr. Öğr. Üyesi Gözde Karataş Baydoğmuş

İSTANBUL, 2022

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesindeki tüm aşamalarda bilgi, birikim ve tecrübeleri ile bana rehberlik eden danışmanım sayın Doç. Dr. Önder Demir'e, çalışma süresince hiçbir zaman yardımlarını esirgemeyen kıymetli eş-danışmanım sayın Dr. Öğr. Üyesi Gözde Karataş Baydoğmuş'a, maddi ve manevi destekleriyle her daim yanımda olan çok değerli aileme sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

TEŞEKKÜR i

İÇİNDEKİLER ii

ÖZET iv

ABSTRACT vi

SEMBOLLER viii

KISALTMALAR ix

ŞEKİL LİSTESİ x

TABLO LİSTESİ xi

1. GİRİŞ 1

1.2. LİTERATÜR ÖZETİ 3

2. MATERYAL VE YÖNTEM 8

2.1. Bilgi Güvenliği, Siber Saldırılar ve Çeşitleri 8

2.1.1. DoS ve DDoS Saldırıları 10

2.1.2. Botnet Saldırıları 11

2.1.3. Kaba Kuvvet Saldırıları 12

2.1.4. SQL Enjeksiyon Saldırıları 13

2.1.5. Ağa İçeriden Sızma (Infilteration) Saldırıları 14

2.2. Saldırı Tespit Sistemleri 15

2.2.1. Veri İşleme Zamanı 17

2.2.2. Mimari Yapı 18

2.2.3. Bilgi Kaynağı 18

2.2.4. Saldırı Tespit Yöntemi 19

2.2.5. Korunan Sistem 20

2.3. Saldırı Tespit Sistemlerinde Kullanılan Yöntemler 22

2.4. Makine Öğrenmesi Türleri 24

2.4.1. Gözetimli Öğrenme 25

2.4.2. Gözetimsiz Öğrenme 25

2.4.3. Pekiştirmeli Öğrenme 26

2.5. Makine Öğrenmesi Teknikleri 26
2.5.1. Adaptif Yükseltme Algoritması 26
2.5.2. Karar Ağacı Algoritması 27
2.5.3. Ekstra Ağaçlar Algoritması 28
2.5.4. Lojistik Regresyon Algoritması 29
2.5.5. Gradyan Artırma Algoritması 31
2.5.6. Pasif-Agresif Algoritmalar 32
2.5.7. Çok Katmanlı Algılayıcı Yapay Sinir Ağı 33
2.6. Öznitelik Seçimi Yöntemleri 34
2.6.1. Filtreleme Yöntemleri 34
2.6.2. Sarmal Yöntemler 36
2.6.3. Gömülü Yöntemler 37
2.7. Kolektif Öğrenme Yöntemleri 37
2.8. CSE-CIC-IDS2018 Veri Kümesi 40
2.9. Performans Değerlendirme Metrikleri 44
2.10. Önerilen Sistem 46
3. BULGULAR VE TARTIŞMA 49
3.1. Veri Önişleme 49
3.2. Öznitelik Seçimi 51
3.3. Model Seçimi ve Kolektif Öğrenme 59
4. SONUÇ VE GELECEK ÇALIŞMALAR 62
KAYNAKLAR 65

ÖZET

KOLEKTİF MAKİNE ÖĞRENMESİ TABANLI AĞ SALDIRI TESPİTİ

İnternet kullanımının hızla yayılması ve buna paralel olarak çevrimiçi ortamlarda vakit geçiren kullanıcı sayısının gün geçtikçe fazlalaşması, siber risk ve tehditleride beraberinde getirmektedir. Kötü amaçlı kullanıcılar bilgi, fikir, para gibi birçok önemli unsurun paylaşıldığı bu ortamlarda bulunan sistem ve uygulamaları önemli ölçüde zarara uğratabilmektedir.

Saldırı Tespit Sistemleri (STS), İnternet ortamındaki sistem ve uygulama güvenliğinin sağlanmasında kritik bir role sahiptir. Bu sistemler yardımıyla internet ağında gerçekleşen aktiviteler ve trafik analiz edilerek olası atak, ihlal ve tehditler tespit edilir. Eğitimlerinde klasik yöntemlerin yanı sıra, çok sayıda makine öğrenmesi teknikleri kullanılabilmektedir. Son geliştirilen STS'ler, -dinamik bir güvenlik mekanizması oluşturulabilmesi için- makine öğrenmesi tekniklerinin tercih edildiği çalışmaların sayısının giderek arttığını göstermektedir. Bu çalışmada, öznitelik seçimi ve kolektif öğrenme yöntemlerinden faydalanılarak yüksek doğruluk oranına sahip performanslı bir STS elde etme üzerinde durulmuştur. Kullanılan veri kümesi kalitesinin de doğrudan STS verimliliği üzerinde etkisi olması sebebiyle, veri kümesi olarak saldırı çeşitliliği yüksek, bilinen güncel STS veri kümelerinden olan CIC-CSE-IDS2018 tercih edilmiştir.

İlk aşamada, -saldırı tespit sürecinin iyileşmesi ve süresinin azalması adına- öznitelikler Spearman'ın Korelasyon Analizi, Özyinelemeli Öznitelik Seçimi (RFE) ve Ki-Kare Test metotları uygulanarak belirlenmiştir. Belirlenen özniteliklerle oluşturulan yeni veri kümeleri ile orijinal boyuttaki veri kümelerinin karşılaştırılmasında Karar Ağacı, Gradyan Artırma, Adaptif Yükseltme, Lojistik Regresyon, Pasif-Agresif, Ekstra Ağaçlar ve Çok Katmanlı Algılayıcı sınıflandırıcılarından faydalanılmıştır. Yapılan performans denemelerinde Katmanlı 5-Katlamalı Çapraz Doğrulama tekniği kullanılmıştır. Bu tekniğin kullanılması nedeniyle oluşan hesaplama ve zaman maliyetini düşürmek için çok-çekirdekli paralelleştirme (multi-core parallellism) uygulanmıştır. Sonrasında, elde edilen performans sonuçlarının karşılaştırmalı bir analizi yapılmıştır. Sonuçlar, sistem başarımının Spearman'ın korelasyon analizi ve Ki-Kare test yöntemleri ile düştüğünü fakat RFE yöntemi ile arttığını göstermiştir. %98,76 doğruluk oranı ile en başarılı sınıflandırıcı Ekstra Ağaçlar olsa da çalışma süre kriteri göz önünde bulundurulduğunda sırayla %95,15 ve %98,65 doğruluk oranları ile Lojistik Regresyon ve Karar Ağacı sınıflandırıcıları da ön plana çıkmıştır.

Pek çok çalışma, topluluk modelini kullanan bir sistemin sınıflandırmada tek bir sınıflandırıcı kullanan sisteme göre daha iyi sonuçlar verebileceğini göstermiştir. Bu sebeple ikinci aşamada, kompleks fakat daha yüksek doğruluk oranı sağlayan bir topluluk modeli oluşturma fikri üzerinde durulmuştur. Sınıflandırma algoritmalarından her birinin faydasını birleştiren "oylama" isimli toplu öğrenme yaklaşımı uygulanarak, ilk aşamada yer alan performans sonuçları üzerinden seçilen sınıflandırıcılar ile kolektif bir model üretilmiştir. Kolektif model için Karar Ağacı, Ekstra Ağaç ve Lojistik Regresyon sınıflandırıcıları seçilmiştir. Sonuçlar, %98,82 doğruluk oranı ile kolektif modelin tek bir sınıflandırıcının bulunduğu bireysel yaklaşımlardan daha üst bir performans gösterdiğini ortaya koymuştur.

ABSTRACT

COLLECTIVE MACHINE LEARNING BASED NETWORK INTRUSION DETECTION

The fast-moving propagation of internet usage and the corresponding increase in the number of user spending time online bring cyber risks and threats along. Malicious computer users can cause momentous damage to the systems and applications in the internet environment where many important elements such as information, ideas and money are shared.

Intrusion Detection Systems (IDSs) have a critical role in ensuring system and application security in the Internet environment. With the help of these systems, activities and traffic on the Internet network are analyzed and possible attacks, violations and threats are detected. In addition to classical methods, many machine learning techniques can be used in their training. Recently developed IDSs show that the number of studies in which machine learning techniques are preferred in order to create a dynamic security mechanism, is increasing day by day. In this study, it is focused on obtaining a high-performance IDS that works with high accuracy by using feature selection and ensemble learning methods. Since the quality of the dataset used has a direct effect on IDS efficiency, CIC-CSE-IDS2018, which is one of an up-to-date IDS dataset known, with a high attack variety, was preferred.

In the first stage, the features were determined by applying Spearman's correlation analysis, Recursive Feature Elimination (RFE) and Chi-Square test methods in order to improve attack detection process and reduce its time. Decision Tree, Gradient Boosting, Adaptive Boosting, Logistic Regression, Passive-Aggressive, Extra Trees and Multilayer Perceptron classifiers were used to compare the original datasets with the new datasets consisting of the specified features. Stratified 5-Fold Cross Validation technique was used in performance tests. In order to reduce computational and time cost incurred due to the fact that all experiments were performed with using this technique, multi-core parallelism has been applied. Afterwards, a comparative analysis was made for the performance results obtained. The results showed that, the system performance decreased with Spearman’s correlation analysis and Chi-Square test methods, but increased with RFE method.

Although the model with the highest performance belongs to the Extra Trees classifier with an accuracy rate of 98.76%, considering the execution time metric, Logistic Regression and Decision Tree classifiers came to the fore with accuracy rates of 95.15% and 98.65%, respectively.

Many studies have shown that a system using the ensemble model can give better results in classification than a system using a single classifier. For this reason, in the second stage, the idea of creating a complex but higher accuracy ensemble model was discussed. By applying the ensemble learning approach called “voting”, which combines the benefits of each of the classification algorithms, a collective model was produced with the classifiers selected based on the performance results obtained in the first stage. Decision Tree, Extra Tree and Logistic Regression classifiers were chosen for the collective model. The results revealed that the collective model outperformed the individual approaches consisting of a single classifier, with an accuracy rate of 98.82%.

SEMBOLLER

$\boldsymbol{\beta}$: Regresyon katsayısı

$\boldsymbol{fi}$: İ. zayıf sınıfa ait bilgi

$\boldsymbol{\theta i}$: İ. zayıf sınıfa ait ağırlık bilgisi

S : Veri seti

η : Lojit dönüşüm

KISALTMALAR

ADA : Adaptif Yükseltme

ANN : Yapay Sinir Ağları

CSE : İletişim Güvenliği Kuruluşu

CIC : Kanada Siber Güvenlik Enstitüsü

CFS : Korelasyon Tabanlı Öznitelik Seçimi

CNN : Evrişimsel Sinir Ağları

DDoS : Dağıtık Hizmet Engelleme

DNN : Derin Sinir Ağları

DoS : Hizmet Engelleme

DT : Karar Ağacı

EFS : Topluluk Öznitelik Seçimi

ENN : Düzenlenmiş En Yakın Komşu

ET : Ekstra Ağaçlar

GB : Gradyan Artırma

GE : Dilbilgisel Gelişim

ICMP : İnternet Kontrol Mesaj Protokolü

IDS : Saldırı Tespit Sistemi

ID3 : Yinelemeli Dikotomizer 3

KNN : K-En Yakın Komşu

LAN : Yerel Alan Ağı

LGBM : Hafif Gradyan Artırma

LR : Lojistik Regresyon

MLP : Çok Katmanlı Algılayıcı

NB : Naive Bayes

OneR : Tek Sınıf Sınıflandırma

PA : Pasif-Agresif

PCA : Temel Bileşen Analizi

QDA : Karesel Ayrım Analizi

RF : Rastgele Orman

RFE : Özyinelemeli Öznitelik Seçimi

RNN : Tekrarlayan Sinir Ağları

SMOTE : Sentetik Azınlık Aşırı Örnekleme

SVM : Destek Vektör Makinesi

ŞEKİL LİSTESİ

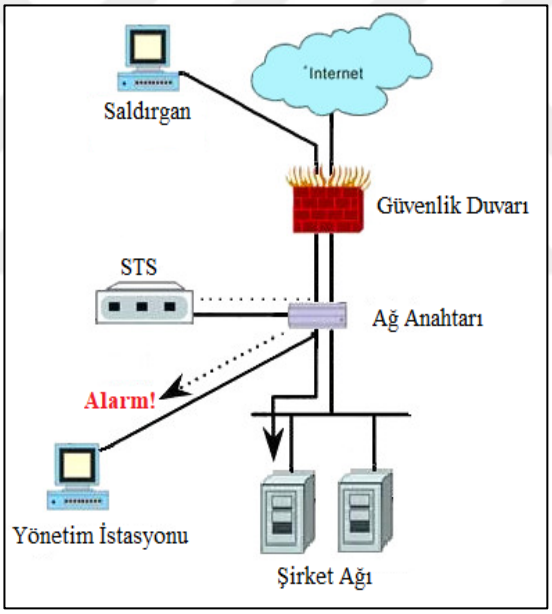
Şekil 1.1. STS içeren örnek topoloji 1
Şekil 2.1. Gizlilik, Bütünlük ve Erişilebilirlik Üçlüsü 9
Şekil 2.2. Botnet Yaşam Döngüsü (Kara ve ark., 2011) 12
Şekil 2.3. STS Temel İşlemleri 15
Şekil 2.4. STS Mimarisi 16
Şekil 2.5. STS'lerin Sınıflandırılması 17
Şekil 2.6. Makine Öğrenimi Türleri 25
Şekil 2.7. Torbalama (Bagging) Algoritması İşlem Adımları 38
Şekil 2.8. Yükseltme (Boosting) Algoritması İşlem Adımları 39
Şekil 2.9. Yığınlama (Stacking) Algoritması İşlem Adımları 39
Şekil 2.10. Oylama (Voting) Algoritması İşlem Adımları 40
Şekil 2.11. Veri Kümesindeki Etiket Sayısı 41
Şekil 2.12. Etiket Kategori Dağılımı 42
Şekil 2.13. Karmaşıklık Matrisi 45
Şekil 2.14. Geliştirilen Makine Öğrenmesi Modeli 48
Şekil 3.1. Küçültülmüş Veri Kümesi Etiket Kategori Dağılımı 51
Şekil 3.2. Tüm Öznitelikler Kullanılarak Elde Edilen Doğruluk ve Hata Oranları 52
Şekil 3.3. Ki-Kare Test ile Seçilen Öznitelikler Kullanılarak Elde Edilen Doğruluk ve Hata Oranları 54
Şekil 3.4. Spearman'ın Sıralama Korelasyon Katsayısı ile Seçilen Öznitelikler Kullanılarak Elde Edilen Doğruluk ve Hata Oranları 56
Şekil 3.5. RFE ile Seçilen Öznitelikler Kullanılarak Elde Edilen Doğruluk ve Hata Oranları 58
Şekil 3.6. Çoğunluk Oyu Sınıflandırıcı Tahminleri 60
Şekil 3.7. RFE ile Seçilen Öznitelikler Kullanılarak Elde Edilen Doğruluk, Hata, Kesinlik, Duyarlılık, F1-Skor ve Süre Değerleri 61
Şekil 3.8. Kolektif Yöntem ile Elde Edilen Doğruluk, Hata, Kesinlik, Duyarlılık, F1-Skor ve Süre Değerleri 62

TABLO LİSTESİ

Tablo 2.1. Adaptif Yükseltme Hiper-parametre ve Değerleri 27
Tablo 2.2. Karar Ağacı Hiper-parametre ve Değerleri 28
Tablo 2.3. Ekstra Ağaçlar Hiper-parametre ve Değerleri 29
Tablo 2.4. Lojistik Regresyon Hiper-parametre ve Değerleri 31
Tablo 2.5. Gradyan Artırma Hiper-parametre ve Değerleri 32
Tablo 2.6. Pasif-Agresif Hiper-parametre ve Değerleri 33
Tablo 2.7 Çok Katmanlı Algılayıcı Hiper-parametre ve Değerleri 33
Tablo 2.8. CICFlowMeter-V3 ile Elde Edilen Öznitelikler 42
Tablo 3.1. Etiketler ve Değerleri 50
Tablo 3.2. Lojistik Regresyon Modeli ile Karşılaştırma Sonuçları 51
Tablo 3.3. Karar Ağacı Modeli ile Karşılaştırma Sonuçları 51
Tablo 3.4. Saldırı ve Normal Ağ Trafik Yüzdesi 51
Tablo 3.5. Tüm Öznitelikler ile Doğruluk Başarımları 52
Tablo 3.6. Ki-Kare Test Sonuçları 53
Tablo 3.7. Ki-Kare Öznitelik Seçimi ile Doğruluk Başarımları 54
Tablo 3.8. Spearman Korelasyon Analiz Sonuçları 55
Tablo 3.9. Spearman'ın Sıralama Korelasyon Katsayısı Öznitelik Seçimi ile Doğruluk Başarımları 55
Tablo 3.10. Özyinelemeli Öznitelik Eliminasyon Sonuçları 57
Tablo 3.11. Özyinelemeli Öznitelik Eliminasyonu ile Doğruluk Başarımları 57
Tablo 3.12. Tüm Öznitelikler- Duyarlılık, Kesinlik, F1-Skor Başarımları 58
Tablo 3.13. RFE ile Seçilen Öznitelikler- Duyarlılık, Kesinlik, F1-Skor Başarımları 59
Tablo 3.14. RFE Yöntemi ile Tahmin Sonuçları 60
Tablo 3.15. Önerilen Kolektif Model ile Tahmin Sonuçları 62

1. GİRİŞ

Ağ güvenliği altyapısında yaygın olarak kullanılmakta olan Saldırı tespit sistemleri (STS), anomali ve imza tabanlı saldırıları tespit etmede başarılı bir şekilde kullanılmaktadır. Kötüye kullanım (imza) tespiti yaklaşımında, saldırıya ait karakteristik özellikler önceden kaydedilmiş olan saldırı modelleri ya da imzaları ile karşılaştırılır ve bir eşleşme durumu aranır. Karşılaştırma sonucunda eşleşme bulunursa aktivite saldırı olarak tanımlanır. Bu yaklaşım, yeni ya da bilinmeyen saldırıların tespiti konusunda zayıftır. Bu sebeple daha çok bilinen saldırıları tespit etmek amaçlı kullanılır. Anomali tespiti yaklaşımında ise saldırıları tespit sürecinin temeli, saldırı durumunda normal kullanıcı davranışında sapmalar meydana gelmesi ile farklı davranışların üretilmesine dayanmaktadır (Mishra ve ark., 2018).



Şekil 1.1. STS içeren örnek topoloji

STS'ler, politika ihlallerini ya da kötü amaçlı faaliyetleri saptayabilmek adına ağda gerçekleşen aktiviteleri izler. İzleme ile edindiği bilgilerin analizini yapar ve neticesinde aktivitenin bir saldırı vakası olup olmadığına karar verir. Önemli güvenlik bileşenlerinden biri olan bu sistemler, genellikle güvenlik duvarından sonra bir ağ Terminal Erişim Noktası (TAP)'na ya da anahtara bağlanır ve ağ trafiğinin TAP vasıtasıyla gönderildiği ya da STS'ye yayıldığı satır içi olmayan bir modda kullanılır. Verilen bilgiler doğrultusunda ağ tabanlı STS içeren örnek bir topoloji Şekil 1.1'de sunulmuştur.

Öğrenebilme kabiliyetleri dolayısıyla makine öğrenmesi teknikleri, STS'lerin saldırıları algılama noktasındaki yeteneklerini geliştirmek için sıklıkla başvurulan bir yöntem olmuştur. Bu sebeple araştırmacılar, en düşük yanlış alarm ve en yüksek doğruluk oranlarının elde edilmesi hedefiyle sistemlerin verim ve başarımını artırmak için bu yöntemlere yönelmiştir (Kim ve ark., 2018). Son yıllarda saldırı tespiti alanında yapılan çalışmalarda makine öğrenmesinin RF (Rastgele Orman), DT (Karar Ağacı), SVM (Destek Vektör Makineleri) ve ANN (Yapay Sinir Ağları) gibi popüler algoritmaları kullanılmaya başlanmış ve birçok iyileştirme sağlanmıştır.

Ancak, bilindiği üzere her algoritmanın kendine göre avantajları ve dezavantajları bulunmaktadır. Bu sebeple, bazı algoritmalar bir saldırı türünde iyi performans gösterebilirken, diğer türlerde düşük performans gösterebilmektedir (Gao ve ark., 2019). Örneğin, DT'nin anlaşılması ve görselleştirilmesi kolaydır. Çok az veri hazırlığı gerektirerek hem sayısal hem de kategorik verileri işleyebilir. Fakat, iyi genellenemeyen karmaşık ağaçlar da oluşturabilir veya verilerdeki küçük değişiklikler tamamen farklı bir ağacın üretilmesine neden olabileceğinden güvenilir sonuçlar sağlamayabilir.

Bu çalışmada, optimal saldırı tespiti elde etmek için her bir algoritmanın faydalarını birleştiren bir topluluk öğrenme modeli uygulanmıştır. Toplu öğrenmenin tek tahminciye göre avantajı, genelleştirilebilirliği (generalizability) ve sağlamlığı (robustness) artırmak için birkaç temel sınıflandırıcıdan gelen tahmin sonuçlarını birleştirme yeteniğidir (Gao ve ark., 2019).

STS'nin verimliliği, veri kümesinin kalitesi ve öğrenme modeli ile doğrudan ilişkilidir. Pek çok çalışma, anonimlik (gizlilik veya etik nedeniyle), trafik çeşitliliğinin olmaması, simüle edilmiş trafik (gerçek bir üretim ağından değil), güncel olmayan saldırı trafiği gibi bilinen eksiklikleri bulunan veri kümelerine dayandırılmaktadır (Aljawarneh ve ark., 2018). Bu eksiklikleri bertaraf etmek ve etkili STS'ler geliştirilmesine katkı sağlamak amaçlı Kanada Siber Güvenlik Enstitüsü, İletişim Güvenliği Kuruluşu iş birliği ile gerçek ağ trafiğine sahip, saldırı çeşitliliği yüksek CSE-CIC-IDS2018 isimli güncel bir STS veri kümesi sunmuştur (CyberSecurity, 2018).

Çalışmada, bu güncel veri kümesi üzerinde kolektif öğrenme ve öznitelik seçim yöntemleri kullanılarak saldırı tespit performansının iyileştirilmesi üzerinde durulmuştur. Kolektif modelleri iyileştirmek için çeşitli faktörlerin dikkate alınması gerekir. Bunlar; öznitelik seçimi, temel sınıflandırıcı seçimi ve kolektif öğrenme yöntemidir.

Tespit sürecini iyileştirmek ve saldırı tespit modelinin tespit süresini azaltmak için öznitelikler CSE-CIC-IDS2018 veri kümesinden Spearman'ın Korelasyon Analizi, Özyinelemeli Öznitelik Seçimi (RFE) ve Ki-Kare Test yöntemlerinin uygulanmasıyla elde edilen sonuçlara göre belirlenmiştir. Temel sınıflandırıcı seçimi için ADA (Adaptif Yükseltme), DT (Karar Ağacı), LR (Lojistik Regresyon), MLP (Çok Katmanlı Algılayıcı), ET (Ekstra Ağaçlar), PA (Pasif-Agresif) ve GB (Gradyan Artırma) makine öğrenmesi yöntemleri ile elde edilen performans sonuçları karşılaştırılmıştır. Sonrasında, sınıflandırma algoritmalarının her birinin faydasını birleştiren "oylama" adı verilen topluluk öğrenme yaklaşımı uygulanarak, en iyi sonuçların elde edildiği üç sınıflandırma yöntemi ile kolektif bir model oluşturulmuştur. Kolektif model için seçilen sınıflandıcılar; LR, DT ve ET'dir.

Sonuçlar, uygun ayar parametreleri kullanılarak RFE metotu uygulandığında, işlem süresinin kısalarak sistem başarımının arttığını ve tek bir sınıflandırıcı içeren bireysel yaklaşımlara kıyasla oluşturulan kolektif modelin %98,82 doğruluk oranı ile daha üstün bir performans sergilediğini göstermiştir.

1.2. LİTERATÜR ÖZETİ

Makine öğrenmesi algoritmalarının performansı büyük ölçüde eğitildiği veri kümesi ile ilişkilidir (Sommer ve ark., 2010). Mevcut çalışmaların çoğunda, geliştirilen makine öğrenmesi tabanlı saldırı tespit sistemleri NSL-KDD, KDD Cup 99 ve DARPA veri kümeleri kullanılarak eğitilmiştir. Ancak bazı araştırmacılar güncel olmayan bu veri kümelerinin yapılacak yeni çalışmalar için tercih edilmemesini tavsiye etmişlerdir (Aljawarneh ve ark., 2018; Sommer ve ark., 2010). Güncel olmayan ve bilinen eksiklikleri olan veri kümeleri kullanılarak geliştirilen STS'ler güncel ve farklı saldırıları algılama noktasında eksik kalacaktır. Dolayısıyla, güncel ve yüksek saldırı çeşitliliğine sahip veri kümeleri ile çalışmak daha etkili STS'ler geliştirilmesi adına önemlidir.

Sistematik bir şekilde siber güvenlik veri kümesi oluşturmak amacıyla profil kavramının kullanıldığı CSE-CIC-IDS2018, İletişim Güvenliği Kurumu (CSE) ve Kanada Siber Güvenlik Enstitüsü (CIC) arasındaki ortak bir projenin sonucudur. Bu veri kümesi uygulamalar, protokoller veya daha düşük seviyeli ağ varlıkları için saldırıların ayrıntılı tanımlarını içerir. CICIDS2017 veri kümesinin güncel bir versiyonu olan CSE-CIC-IDS2018 farklı saldırı çeşitlerine sahip, bilinen yeni STS veri kümelerindendir.

CSE-CIC-IDS2018, CICIDS2017 ve popüler diğer veri kümeleri kullanılarak elde edilen anomali temelli saldırı tespitine yönelik bulguların bildirildiği çalışmaların bazıları şöyledir: Sharafaldin ve diğerleri, saldırı tespiti için kullanılan mevcut veri kümelerinin saldırı çeşitliliği noktasında yeteriz kalması, kullanımlarının güvenli olmaması ve güncel saldırıları kapsamaması gibi nedenlerle CICIDS2017 isimli yeni bir veri kümesi üretmiştir. Bu veri kümesi kullanılarak karşılaştırması yapılan 6 farklı makine öğrenmesi yöntemi şunlardır; NB, KNN, RF, ID3, QDA ve ADA. Sonuçlar, ID3'ün %98 F1-Skoru ile en başarılı yöntem olduğunu göstermiştir (Sharafaldin ve ark., 2018).

Wankhede ve diğerleri, DoS saldırıları tespit etmek amacıyla CICIDS2017 veri kümesi üzerinde ANN ve RF makine öğrenmesi tekniklerini kullanmıştır. Çalışmada, veri kümesi bölümlenmesinin saldırı tespit başarımına etkisi de gösterilmek istenmiştir. Bu sebeple, eğitim veri kümesi %20-%80 arasında bölümlenmiş, MLP ve RF algoritmaları ile bir başarım karşılaştırması yapılmıştır. Sonuçlar, RF tekniğinin %99,95 doğruluk oranı ile en yüksek başarımı sunduğunu göstermiştir. Ayrıca, MLP tekniği için %30, RF tekniği için de %50 bölümlemenin ideal olduğu belirtilmiştir (Wankhede ve ark., 2018).

Zhou ve diğerleri, eğitimi CSE-CIC-IDS2018 veri kümesi ile yapılmış bir modelin daha önceden bilinmeyen (Zero-Day) saldırıları tespit etme noktasındaki başarısını incelemiştir. NB, DT, QDA, MLP, KNN ve RF olmak üzere 6 farklı makine öğrenmesi tekniği 10-Katlamalı Çapraz Doğrulama ile aynı veri kümesi üzerinde saldırı tespiti amaçlı denenmiştir. Yapılan tüm denemelerde her saldırı çeşidi için normal trafik ile ikili (binary) karşılaştırma yapılmıştır ve en başarılı modelin DT olduğu ortaya konmuştur (Zhou ve ark., 2019). Ardından, eğitim veri kümesi Normal ve Saldırı olmak üzere etiketlenmiş ve model bu veri kümesi üzerinde eğitilmiştir. Test veri kümesi için 1 haftalık normal trafik ve Ponmocup Trojan, ZeroAccess, Bitcoin Miner, DDoS Bot'a Darkness, Nuclear Ransomware, Drowor Worm, Google Doc Macadocs ve False Content Injection olmak üzere 8 farklı yeni saldırı trafiği oluşturulmuştur. Sonuçlar, DT modelinin saldırıları %96 doğruluk oranı ile saptadığını göstermiştir.

Kanimozhi ve diğerleri, MLP tekniği ile CSE-CIC-IDS2018 veri kümesi üzerinde Botnet saldırı tespiti yapmıştır. Oluşturulan model varsayılan hiperparametreler kullanıldığında aşırı uyum (overfitting) durumuna düşmüştür. Bu sebeple GridSearch yönteminden faydalanılarak hiper parametre optimizasyonuna gidilmiştir. Sonuçlar, saldırıların %99,97 doğruluk oranı ile tespit edilebildiğini göstermiştir.

Yulianto ve diğerleri, saldırı çeşitleri açısından dengesiz dağılıma sahip olan CICIDS2017 veri kümesi üzerinde SMOTE, PCA ve EFS metotlarını kullanarak ADA tabanlı STS performansını iyileştirmeye yönelik denemeler yapmıştır. Deneme sonuçlarına göre, en iyi performans iyileştirmesini %90,01 F1-Skoru ile EFS ve SMOTE metotlarının beraber kullanımı sağlamıştır (Yulianto ve ark., 2018).

Wani ve diğerleri, DDoS saldırılarını tespit etmek amacıyla Bulut Bilişim Ortamı üzerinde NB, RF ve SVM metotlarını uygulamıştır. Oluşturulan veri kümesinden kullanılan 9 öznitelik ile elde edilen sonuçlar, %99,80 F1-Skoruna sahip olan SVM'nin en başarılı metot olduğunu göstermiştir (Wani ve ark., 2019).

McKay ve diğerleri, OneR, KNN, J48, RF, NB ve MLP metotları ile CICIDS2017 veri kümesi üzerinde Botnet saldırıları tespitini amaçlamıştır. Çalışmada, normal ve dengeli olarak veri kümesi bölümlenmiş ve eğitimi dengeli veri kümesi kullanılarak yapılan modellerin tümünde daha iyi sonuçlar alındığı gözlenmiştir. J48, %98,73 doğruluk oranı ile saldırı tespitindeki en başarılı metot olmuştur (McKay ve ark., 2019).

Kanimozhi ve diğerleri, Botnet saldırılarını tespit etmek amacıyla CSE-CIC-IDS2018 veri kümesi üzerinde KNN, NB, SVM, RF, ADA ve MLP olmak üzere altı farklı makine öğrenimi sınıflandırıcısını karşılaştırmıştır. Çalışmada, sınıflandırıcıların performans değerlendirmeleri kalibrasyon eğrileri kullanılarak yapılmıştır. KNN, NB, SVM, RF, ADA ve MLP sınıflandırıcıları için sırasıyla %99,84, %99,53, %99,94, %99,92, %99,92 ve %99,98 F1-Skor değerleri elde edilmiştir. Sonuçlar, bu metrik ile doğru orantılı olarak, MLP'nin kalibrasyon eğrisinin mükemmel eğriye en yakın olduğunu göstermiştir (Kanimozhi ve ark., 2019).

Ferrag ve diğerleri, RNN, SVM, CNN ve RF metotları ile CSE-CIC-IDS2018 veri kümesi üzerinde DoS, DDoS, Web, Brute-Force, Infilteration ve Botnet saldırılarının tespitini hedeflemiştir. Sonuçlara göre, edinilen en iyi doğruluk değerleri sırasıyla %92,19, %96,12, %96,18, %98,55, %98,71 ve %96,23'tür (Ferrag ve ark., 2019).

Pehlivanoğlu ve diğerleri, CSE-CIC-IDS2018 veri kümesi üzerinde tek seviyeli ve iki seviyeli hibrit olarak ayırdıkları yöntemleri kullanarak saldırı tespit denemeleri yapmıştır. Çalışma için kullanılan makine öğrenmesi yöntemleri şunlardır; RF, LGBM, CNN, CNN + RF, RF+RF ve LGBM + RF. Sonuçlar, en iyi saldırı tespitinin %86,00 makro F1-Skoru ve %98,00 doğruluk oranı ile CNN+RF hibrit modeli tarafından yapıldığını göstermiştir (Pehlivanoğlu ve ark., 2019).

Filho ve diğerleri, DoS saldırılarını tespit etmek amacıyla CSE-CIC-IDS2018 CICIDS2017, CIC-DOS ve ürettikleri yeni veri kümesi üzerinde RF makine öğrenmesi metotunu uygulamıştır. Çalışma sonucunda sırasıyla %99,00, %99,00, %100,00 ve %99,00 F1-Skor değerleri elde edilmiştir (Filho ve ark., 2019).

Zhou ve diğerleri, öznitelik seçimi ve kolektif öğrenme metotlarından faydalanarak NSL-KDD, AWID ve CICIDS2017 veri kümeleri üzerinde doğruluk oranı ve performansı yüksek etkili bir STS geliştirmek için çalışmıştır. Çalışmanın ilk adımında boyut azaltmak için, öznitelikler arasındaki korelasyona dayalı olarak en uygun alt kümeyi seçen CFS-BA isimli sezgisel (heuristic) bir algoritma önerilmiştir.

Ardından, C4.5, RF ve ForestPA (Forest by Penalizing Attributes) algoritmalarını birleştiren kolektif bir yaklaşım sunulmuştur. Saldırı tespitinde, temel öğrenicilerin olasılık dağılımlarını birleştirmek için oylama tekniğinden faydalanılmıştır. NSL-KDD, AWID ve CIC-IDS2017 veri kümelerini kullanan deneysel sonuçlar, önerilen CFS-BA-Kolektif yöntemin çeşitli metrikler altında diğer ilgili ve son teknoloji yaklaşımlardan daha iyi performans gösterebildiğini ortaya koymuştur (Zhou ve ark., 2019).

Fitni ve diğerleri, -performanslı bir STS geliştirilebilmesi adına- CSE-CIC-IDS2018 veri kümesi üzerinde öznitelik seçimi ve kolektif öğrenme tekniklerine odaklanmıştır. Kolektif modelde yer alacak baz sınıflandırıcıları tespit etmek için yedi farklı tek sınıflandırıcı karşılaşılaştırılmıştır. Elde edilen sonuçlar doğrultusunda LR, DT ve GB sınıflandırıcıları ile kolektif bir model oluşturulmuştur. Temel sınıflandırıcıların olasılık dağılımlarını birleştirmek için oylama tekniğinden faydalanılmıştır. Bu çalışmada, boyut azaltmak için korelasyon tabanlı bir öznitelik seçim yöntemi uygulanmıştır. Değerlendirme sonuçları, öznitelik seçimi sonrasında 80 üzerinden 23 özniteliğin belirlendiğini ve kolektif modelin şu başarımlara sahip olduğunu göstermiştir; F1-Skor %97,90, Doğruluk %98,80; Duyarlılık %97,10; ve Kesinlik %98,80 (Fitni ve ark., 2020).

Aksu ve diğerleri, port tarama saldırılarını tespit etmek amacıyla CICIDS2017 veri kümesi üzerinde SVM ve MLP metotlarını uygulamıştır. Çalışmada, yapılan saldırı tespitine yönelik elde edilen F1-Skor değerlerinin sırasıyla %65,00 ve %95,00 olduğu ve 7 katmanlı MLP modelinin daha yüksek başarım ile saldırı tespiti sağladığı bildirilmiştir (Aksu ve ark., 2018).

Ullah ve diğerleri, UNSW-15 ve CICIDS2017 veri kümelerini kullanarak Nesnelerin İnterneti ağlarında anomali tespiti yapmak amacıyla iki seviyeden oluşan bir model üzerinde çalışmıştır. İlk seviyede DT metotu ile ikili (binary) sınıflandırma yapılmıştır. İlk seviyede saldırı olarak nitelendirilen paketler bir sonraki seviyeye iletilmiştir. İkinci seviyede SMOTE, ENN ve RFE metotlarından faydalanılarak önemli öznitelikler belirlenmiştir. Sonuçlar, CICIDS2017 veri kümesinde bir ve ikinci seviye için %100 F1-skoru ile UNSW-15 veri kümesinde ilk seviye için %99,00, ikinci seviye için %97,00 F1-skoru ile saldırı tespitinin sağlanabildiğini göstermiştir (Ullah ve ark.,2019).

Yılmaz ve diğerleri, CICIDS2017 ve ISOT veri kümeleri üzerinde GE algoritmasını kullanarak botnet saldırılarının erken tespit edilebilmesi üzerine çalışmıştır. GE modelin eğitimi veri kümelerindeki botnet ve normal trafik verileri kullanılarak yapılmıştır. Sonrasında, oluşturulan model ile gerçek zamanlı veriden edinilen trafik akışları sınıflandırılmıştır. Sonuçlar, ISOT veri kümesi üzerinde önerilen yöntemin literatürdeki diğer çalışmalara nispeten en iyi doğruluk oranını sunduğunu göstermiştir (Yılmaz ve ark., 2019).

Internet ve toplumsal yaşamın hızla bütünleşmesi ve internet üzerinden sunulan hizmetlerin sayısının gün geçtikçe artması ağ altyapısını tehdit eden birçok riski de beraberinde getirmektedir. Bu sebeple Cil ve diğerleri, ağ trafiği üzerinden elde edilen paketleri kullanarak DDoS saldırı tespiti yapan bir model önerisinde bulunmuştur. Çalışmada, popüler derin öğrenme metotlarından biri olan DNN kullanılmıştır ve veri kümesi olarak CIC-DDoS2019 tercih edilmiştir. DNN metotunun seçilmesinin sebebi öznitelik çıkarma, sınıflandırma işlemlerini içeren katmanlara sahip olması ve eğitildikçe kendini güncelliyor olmasıdır. Sonuçlar, %99,99 başarı oranı ile ağ trafiğine yapılan saldırıların tespit edildiğini ve %94,57 doğruluk oranı ile saldırı türlerinin sınıflandırıldığını göstermektedir (Cil ve ark., 2021).

Arslan, çalışmasında ağ trafik verilerinin işlenmesini kolaylaştıracak bir veri önişleme yöntemi önermiştir. Sonrasında, makine öğrenmesi tekniklerinlerinden faydalanarak ağı analiz etmiş ve saldırıları tespit denemeleri yapmıştır. Önerilen veri önişleme, trafiğin analiz sürecini önemli ölçüde hızlandırmış ve başarımı yükseltmiştir. Çalışmanın gerçeklenmesinde CSE-CIC-IDS2018 veri kümesi kullanılmıştır. Sonuçlar, ET algoritmasının ikili sınıflandırmada %98,5, RF algoritmasının da çoklu sınıflandırmada %99,0 doğruluk başarımı elde ettiğini göstermiştir (Arslan, 2021).

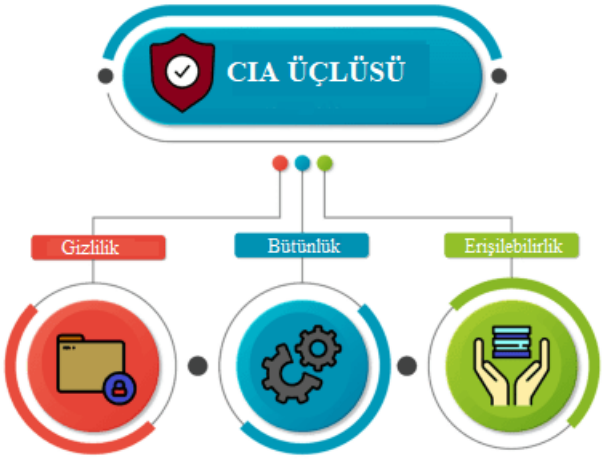
Emhan ve diğerleri, anomalileri tespit etmek için kullanılan makine öğrenmesi algoritmalarının verimini artırmayı amaçlayan bir çalışmada bulunmuştur. Çalışmanın gerçeklenmesinde NSL-KDD veri kümesi kullanılmıştır. İlk aşama olarak filtreleme yöntemlerinden Ki-Kare, Kazanç Oranı, One-R, Simetrik Belirsizlik Katsayısı, Bilgi Kazancı ve Korelasyon Tabanlı Öznitelik Seçimi ile boyut azaltma yapılmıştır. Sonrasında, her bir yöntem için belirlenen özniteliklerden ilk 8'i seçilmiştir. Seçilen öznitelikler ile nihai veri kümeleri oluşturularak sınıflandırıcılara gönderilmiştir. Saldırı tespiti için RF ve KNN metotlarından faydalanılmıştır. OneR, Ki-Kare, Bilgi Kazancı yöntemleri kullanılarak belirlenen öznitelikler RF metotu ile sınıflandırıldığında en optimum sonuçların elde edildiği bildirilmiştir (Emhan ve ark., 2019).

2. MATERYAL VE YÖNTEM

Bu bölümde bilgi güvenliği, siber saldırılar ve çeşitleri ile ilgili genel bir giriş yapılmıştır. Ardından saldırı tespit sistemleri, sınıfları ve saldırı tespit sistemlerinde kullanılan yöntemlerden bahsedilerek, çalışma kapsamında faydalanılan makine öğrenmesi tekniklerine değinilmiştir. Makine öğrenmesi teknikleri incelendikten sonra, öznitelik seçimi ve kolektif öğrenme yöntemleri ile ilgili bilgi verilmiştir. Son olarak, deneysel sonuçlara ulaşmak için kullanılan CSE-CIC-IDS2018 veri kümesi, performans ölçütleri ve önerilen sistemden bahsedilmiştir.

2.1. Bilgi Güvenliği, Siber Saldırılar ve Çeşitleri

Bilgilerin izinsiz veya yetkisiz bir şekilde erişimlerden, kullanımından, değiştirilmesinden, ifşa edilmesinden, ortadan kaldırılmasından veya hasar verilmesinden korunması işlemine bilgi güvenliği denir. Gizlilik (Confidentiality), Bütünlük (Integrity) ve Erişilebilirlik (Availability) dünya çapında bilgi güvenliğinin üç temel bileşeni olarak baz alınmaktadır. Bu bileşenler CIA üçlüsü olarak da bilinmektedir. CIA üçlüsü, güvenlik politikalarının oluşturulması noktasında rehberlik eden bir tasarım modelidir. Risk yaklaşımları, korunacak olan bilgiler, süreçler gibi bileşenler öncelikle bu model çerçevesinde değerlendirilerek koruma altına alınmaya çalışılır.



Şekil 2.1. Gizlilik, Bütünlük ve Erişilebilirlik Üçlüsü

Gizlilik: Verinin yetkisiz erişime karşı muhafaza edilmesidir.
Bütünlük: Verinin yetkisiz kişilerce değiştirilmemesidir.
Erişilebilirlik: Verinin yetkili kişilerce ihtiyaç duyulduğu anda kullanılabilir ve ulaşılabilir halde olmasıdır.

Bilişim sistemleri ve özellikle İnternet teknolojilerinin yaygınlaşması, bilgi güvenliği ihlallerine maruz kalma riskinin artmasını da beraberinde getirmiştir. Bu sebeple, İnternete bağlı cihazların çeşitli tehditlere karşı güvenliği, sistemin bütünlüğü ve sürekli erişilebilir halde olmasının sağlanması oldukça önemli konular haline gelmiştir (Ren ve ark., 2016). Sürekliliği sağlanması saldırılara karşı alınan tedbirlerin güncel olmasına bağlıdır. Tedbirlerin güncel olması için de yeni saldırı ve yöntemlerinin takip edilerek öğrenilmesi ve mevcut sistemlere eklenmesi gerekir.

Bir kaynağın gizliliğini, bütünlüğünü, güvenilirliğini ya da erişilebilirliğini engelleme amacıyla gerçekleştirilen tüm eylemler saldırının tanımını oluşturur. Bu bağlamda siber saldırı, bilgisayar tabanlı sistemlerin fonksiyonlarının durdurulması, etkinliklerinin zayıflatılması ya da yetkisiz bir şekilde ağın çevrimiçi hareketlerinin izlenmesi olarak tanımlanabilir. Bilgi sistemlerini, altyapıları veya ağları hedef alan pekçok siber saldırı çeşiti vardır. Bunlar;

- Aktif Saldırılar (Active Attaks)
- Pasif Saldırılar (Passive Attacks)
- Dolandırıcılık Saldırıları (Fraud Attacks)
- Düşük Oranlı TCP Saldırıları (Low Rate TCP Attacks)
- Bilgi Tarama Saldırıları (Probe Attacks)

- İstismar Saldırıları (Exploit Attacks)
- Dinleme Saldırıları (Sniffing Attacks)
- Gizli Kanal Saldırıları (Covert Channel Attacks)
- İçeriden Saldırılar (Insider Attacks)
- Yan Kanal Saldırıları (Side Channel Attacks)
- Yakın Saldırılar (Close-in Attacks)
- Siber Tacizler (Cyber Harrassment)
- Kötü Amaçlı Saldırılar (Malicious Attacks)
- Hizmet Engelleme Saldırıları (Denial of Service Attacks)
- Dağıtık Hizmet Engelleme Saldırıları (Distributed Denial of Service Attacks)
- Dağıtık Saldırılar (Distributed Attacks)
- SQL Enjeksiyon Saldırıları (SQL Injection Attacks)
- Şifre Saldırıları (Password Attacks)

Saldırı çeşitleri ve detayları için (Anwar ve ark., 2017) referansı incelenebilir. Yaygın olan ve bu tez çalışmasında kullanılan veri kümesinde yer alan saldırı çeşitleri; Hizmet Engelleme (DoS, DDoS), SQL Enjeksiyon (SQL Injection), Kaba kuvvet (Brute Force), Ağa içeriden sızma (Infilteration) ve Botnet saldırılarıdır. Devam eden kısımda bu saldırılar hakkında bilgi verilmiştir.

2.1.1. DoS ve DDoS Saldırıları

DDoS saldırıları, İnternete bağlı bir hostun hizmetlerini -kapasite sınırlarının üzerinde veriye maruz tutma yoluyla- geçici veya süresiz bir şekilde aksatarak, kullanıcıların makineye veya ağ kaynaklarına erişimlerini engellemeyi hedefler. Bu saldırıların ortaya çıkış versiyonları DoS'tur yani bir tek kaynak ile hedefe saldırıda bulunulmasıdır. Zaman geçtikçe bu versiyon geliştirilmiş ve fazla sayıda kaynak kullanılarak hedefe saldırıda bulunulması sağlanarak DDoS'a dönüşmüştür.

Sistemler oluşturulurken kullanıcı sayısı, anlık istek sayısı ve hat kapasiteleri gibi parametreler için değer tahminleri yapılır. Oluşturulan tasarımın bu tahminlerin biraz üstünde olan yükleri de kaldırabilir olması gerekir dolayısıyla tasarım oluşturulurken, yapılan değer tahminleri minimum olarak kabul edilip, biraz daha fazla yük için tolerans düşünülmelidir. DDoS saldırısı, kurban web ağının taşıyabileceği yükün çok üzerinde kullanıcı ve istek sayısı ile sistemi dondurarak erişilebilirliği engeller. DoS ve DDoS saldırılarının bazı belirtileri şöyledir (Gezgin ve ark., 2013):

- Beklenmedik bir şekilde ağdaki performansın düşmesi
- Web sitelerine ya da sunucu hizmetlerinde yaşanan yavaşlıklar, bağlantı kopmaları
- Spam e-posta sayısının artması
- Uzun süreli hizmet kesintileri
- Sunucu kaynak tüketimindeki anlık yığılmalar

Yaygın olarak bilinen DDoS saldırı türleri ve nitelikleri şöyledir:

- Protokol Odaklı Saldırılar: Sunucu kaynakları, güvenlik duvarları veya yük dengeleyiciler gibi ağ ekipman kaynaklarını aşırı tüketerek hizmetin engellenmesini hedefler.

  Bu saldırılar, erişimi engellemek için protokol yığınının 3. ve 4. katmanındaki zayıflıkları kullanır.
- Hacim Odaklı Saldırılar: Sunucularda kullanılan tüm bant genişliğini tüketerek tıkanıklık oluşturmak için büyük miktarda veri, botnet'ten gelen istekler gibi büyük bir trafik hedefe gönderilir.
- Uygulama Katmanlı Saldırılar: Bu saldırılarda, sunucu sistemlerindeki GET ve POST özelliği taşıyan formlar kullanılır. GET ve POST istekleriyle hedeflenen sisteme aşırı yüklenilerek sistem kaynaklarını tüketmek ve sunucuyu cevap veremez hale getirmek hedeflenir.
- SYN Seli: Bu saldırılarda, bağlantı istek paketlerinin (SYN) art arda gönderimi sağlanarak TCP el sıkışmasının tamamlanması engellenir ve cihazın trafiğe yanıt veremez hale gelmesi hedeflenir.
- UDP Seli: Sunucuda çalışan portları kilitlemek için kullanılan saldırılardır. Bu saldırılarda, çok sayıda UDP paketi hedeflenen sunucuya gönderilir ve portların kapanması ya da hizmet verememesi hedeflenir.
- Ping Seli: ICMP flood olarak da bilinen bu saldırılarda temel amaç ICMP talep paketlerini kullanarak hedeflenen sistemin giden ve gelen bant genişliğine aşırı yük bindirmek ve bu şekilde sistemi çalışamaz hale getirmektir.

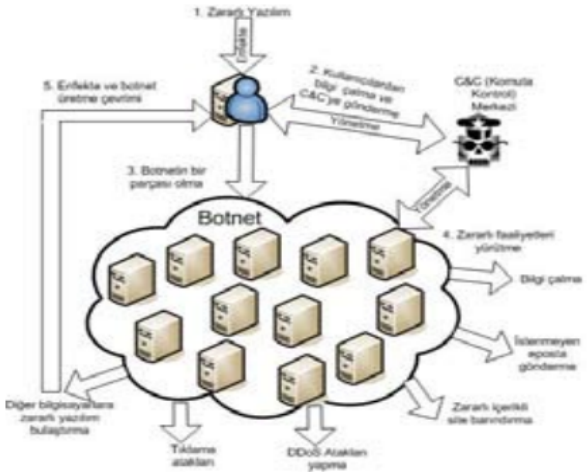
2.1.2. Botnet Saldırıları

Son yıllarda, geniş çaplı internet saldırıları için yaygın olarak Botnet zararlı yazılımları kullanılmaktadır. Bot, önceden planlanmış işleri yapan makine olarak tanımlanan "Robot" kelimesinden türetilmiştir.

Botnetler, botların tek bir merkezden yönetilen büyük gruplarıdır. Botnetler, botları belirli bir düzende belirli amaçlar doğrultusunda yönlendirirler. Botnetlerin yönettiği bilgisayarlar köle (zombie) ya da botnet üyesi olarak isimlendirilmektedir. Botnetler zararlı yazılımlardır. Solucan ve virüslerdeki gibi bu yazılımlar da zafiyet bulunan bilgisayarlara bulaşarak yayılır. Bu zararlı yazılımları diğerlerinden farklı kılan başlıca faktörler komuta kontrol (C&C- Command&Control) merkezi ile haberleşerek, kendilerini güncelleyebilmeleri ve yönetilebilmeleridir (Kara ve ark., 2011). Botnetler de günümüzdeki diğer zararlı yazılımlar gibi çoğunlukla aşağıdaki yollar ile yayılırlar (Microsoft Security Report, 2010):

- Bilişim Teknoloji ürünlerdeki güvenlik zaafiyetleri
- Güvenli olmayan ya da zayıf olan politikalardaki zaafiyetler,
- Sosyal mühendislik taktikleri

Botnetin yaşam döngüsü enfeksiyon, kullanıcıdan bilgi çalma ve C&C'ye iletme, bağlantıyı sürdürme, zararlı faaliyetleri yürütme, enfekte etme ve botnet oluşturma olmak üzere beş fazdan oluşmaktadır. Şekil 2.2 tipik bir botnet yaşam döngüsünü göstermektedir.



Şekil 2.2. Botnet Yaşam Döngüsü (Kara ve ark., 2011)

2.1.3. Kaba Kuvvet Saldırıları

Siber güvenlik tehditleri arasında önemli bir yere sahip olan kaba kuvvet saldırıları, deneme yanılma yoluyla parola-kullanıcı adlarını elde etme, mesajın şifreleme anahtarını ya da gizli bir web sayfasını bulma teşebbüsüdür. Her bir kaba kuvvet saldırısında hassas verileri ele geçirmek için farklı yöntemler kullanılmaktadır.

Sözlük, basit kaba kuvvet, ters kaba kuvvet, hibrit kaba kuvvet saldırıları ve kimlik bilgisi doldurma popüler kaba kuvvet yöntemlerindendir. Kaba kuvvet saldırılarından korunma yollarından bazıları aşağıdaki gibidir:

- Kimlik hırsızlığı, veri kaybı, hesaplara yetkisiz erişim durumlarına karşı korunmak için güçlü parola kullanmak,
- CAPTCHA kullanmak,
- Çok faktörlü kimlik doğrulama sistemi kullanmak,
- Şifrenin kırılmasını zorlaştırmak ve süresini artırmak amacıyla parola karmaşıklığını artırmak,
- Belirli bir sayıda hatalı giriş yapıldığı durumda sonraki denemeyi engelleyici özellikler kullanmak.

2.1.4. SQL Enjeksiyon Saldırıları

SQLi olarak da bilinen SQL Enjeksiyonu, veriye dayalı uygulamalarda SQL sorgularına müdahale ederek -hassas verilere erişim amaçlı- saldırı gerçekleştirmek için yapılan bir kod enjeksiyon tekniğidir. SQL Enjeksiyonu, en çok tercih edilen saldırı yöntemlerinden biridir. Veri tabanını (SQL) yanıltarak sistem üzerindeki verilere ulaşmaya imkân sağlayan bu saldırı yöntemiyle bir web uygulamasının veri tabanı içerisinde yer alan veriler görüntülenebilir, değiştirilebilir veya silinebilir. SQL Enjeksiyonu, gelişen güvenlik önlemleriyle zararsız hale getirilmiş, önlenebilir bir Internet saldırısı olarak düşünülse de SQL Enjeksiyon açıklıkları dünyanın en güvenli sistemlerinde dahi bulunabilmektedir.

SQL Enjeksiyonu genel olarak üç ana kategoriye ayrılır. Bunlar;

- Bant-içi (In-band) SQL Enjeksiyonu
- Çıkarımsal (Blind) SQL Enjeksiyonu
- Bant-dışı (Out-of-band) SQL Enjeksiyonu

Bant-içi SQL Enjeksiyonu: Saldırgan, saldırılarını başlatmak ve sonuçlarını almak için aynı iletişim kanalını kullanır. Hata tabanlı ve Birliğe dayalı SQL Enjeksiyonu olmak üzere iki farklı çeşidi bulunmaktadır.

- Hata Tabanlı SQL Enjeksiyonu: Saldırganlar, veritabanından hata mesajları almak amaçlı SQL sorguları enjekte ederek, veri tabanı ve yapısı hakkında bilgi alır.

- Birliğe Dayalı SQL Enjeksiyonu: Saldırganlar, tek bir HTTP cevabı almak için iki veya daha fazla SELECT ifadesinin sonuçlarını birleştiren UNION SQL operatöründen yararlanır. Alınan yanıt, veri tabanı içeriği hakkında bilgi almak için değerlendirilir.

Çıkarımsal SQL Enjeksiyonu: Saldırganlar, sunucuya veri göndererek yapısı ile ilgili bilgi edinmek için sunucunun yanıtını ve davranışlarını gözlemler. Sunucu yanıt ve davranış kalıplarına dayanması sebebiyle bu tür saldırıların yürütülmesi yavaştır ancak diğer SQL Enjeksiyon türleriyle eşit derecede zararlı olabilirler. Boole tabanlı ve zamana dayalı olmak üzere iki çeşidi vardır.

- Boole Tabanlı SQL Enjeksiyonu: Bu saldırıda, Web uygulama sorgularını Doğru/Yanlış koşullarına çeviren işe yarar payloadlar oluşturulur. Sonrasında, uygulama davranışların farklı Doğru/Yanlış koşullarıyla nasıl değiştiğine bakarak sorguların sonuçları değerlendirilir.
- Zamana Dayalı SQL Enjeksiyonu: Bu saldırıda, veritabanı yanıt vermeden önce belirli bir süre beklemesini sağlayan SQL sorguları kullanılır. Yanıt süresi, saldırganın sorguyu doğru veya yanlış olarak belirlemesine yardımcı olur. Sonuca göre, hızlı ya da ertelemeli bir şekilde HTTP cevabı gönderilir.
  Bu durum, saldırganın veri tabanından hiçbir bilgi döndürülmemesine rağmen kullanılan yükün geçerli ya da sahte bilgi olup olmadığına karar vermesine imkân verir.

Bant-dışı SQL Enjeksiyonu: Bant-dışı SQL enjeksiyon saldırıları en az kullanılan SQL enjeksiyon tekniğidir. Bant-dışı SQL enjeksiyon saldırısı, uygulamanın verileri bir protokol (HTTP, DNS veya SMB) aracılığıyla iletmesini ister.

2.1.5. Ağa İçeriden Sızma (Infilteration) Saldırıları

Direkt olarak ağın ihlali ya da ağdaki ana bilgisayarlardan birinin enfekte edilmesiyle gerçekleşebilir.

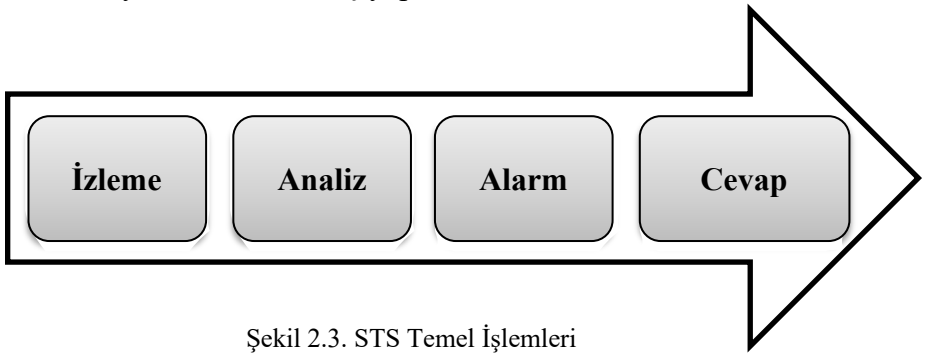
Saldırıyı yapan, hedefe zararlı dosyayı elektronik posta gönderimi gibi sıradan bir yolla ileterek uygulamada bulunan zaafiyetlerden faydalanabilir ya da sonra kullanmak üzere bir arka kapı bırakarak bilgisayardan ve ağdan yararlanabilir.

2.2.Saldırı Tespit Sistemleri

Internet başta olmak üzere diğer ağlara pek çok erişim noktası ile bağlı olan ağlar, günümüzde oldukça karmaşık bir yapıdadır. Siber saldırı ve tehditlerin gün geçtikçe çoğalması ve çeşitlenmesi, bu karmaşık yapının artık sadece şifreleme yöntemleri ya da güvenlik duvarı gibi basit yöntemlerle korunamayacağını açıkça gözler önüne sermektedir. Bu durum, ağ trafiğini izleyip analiz ederek, saldırı teşebbüslerinin anlık tespitini yapan sistemlerin geliştirilmesini sağlamıştır. Bu bağlamda, STS'ler internet ağında gerçekleşen aktiviteleri ve trafiği analiz ederek olası atak, ihlal ve tehditleri tespit etmede başarılı bir şekilde kullanılmaktadır. Saldırı Tespit ve Önleme Sistemleri Rehberi'nde yer alan tanıma göre saldırı tespiti bilgisayar ya da ağ sistemindeki aktiviteleri izleyerek ve analizini yaparak saldırı olması muhtemel vakaları saptamadır. Bahsi geçen vakalar, standart güvenlik uygulamalarının, kabul edilebilir kullanım politikalarının ya da bilgisayar güvenlik politikalarının ihlali ya da ihlal edilmesi tehdididir (Scarfone ve ark., 2007).

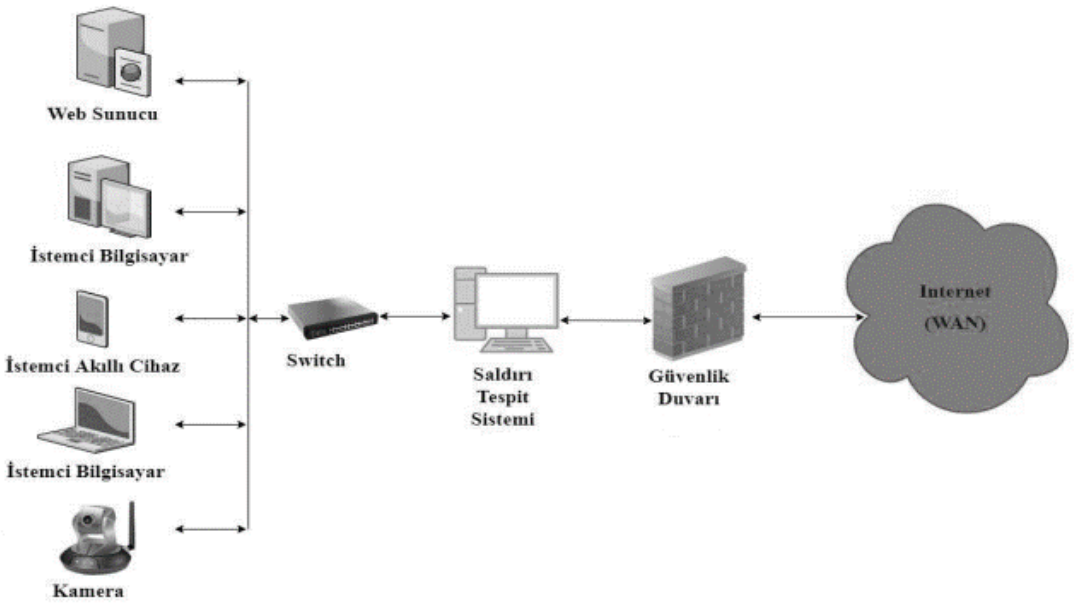
STS'ler önleyici bir sistem olarak değil, sisteme zarar gelmeden önce alarm verme görevini üstlenen bir siber güvenlik tedbiri olarak düşünülmelidir. STS'ler, kullanılan tekniklere ve korunan sisteme bağlı olarak Şekil 2.3'te belirtilen dört işlevi gerçekleştirmektedir (Chezian, 2017). Birbirini takip eden ve devamlı tekrarlanan bu işlemlerden ilki izlemedir.

Muhafaza edilecek sistemin sunucu olması durumunda gelen ve giden veri paketleri, bilgisayar ağı olması durumunda ise kullanıcı faaliyetleri ve paketler izlenebilir. İzlenme sonrasında elde edilen bilgiler ön işlemlerden geçirilir. Sonrasında kullanılan teknikler doğrultsunda analizi yapılır. Bu analizin neticesinde, izlenen paketin ya da faaliyetin saldırı vakası olma durumuna karar verilir. Saldırı olması durumunda uyarı verilerek, bir cevap ile sistem yöneticisine dönüş yapılır.



Şekil 2.3. STS Temel İşlemleri

STS'ler ağ faaliyetlerini izleyerek ağ güvenliğini kontrol eden yazılımlardan oluşur. Bu yazılımlar, ağ denetleyicisinin ağı var olabilecek güvenlik açıkları açısından incelemesine ve saldırganlar bu potansiyel açıklıklardan yaralanmadan önce sistemi güvence altına almasına olanak tanır. Bilgi güvenliği araştırmalarında sıkça yer alan STS'ler, bir uyarı sistemi olarak hizmet vererek, internet ağ yapısının muhafaza edildiği son uç olma vazifesini üstlenmektedir. Bu son kalkan aşıldığında saldırganlar tüm sistemi ele geçirebilme imkanına sahip olabilmektedir. Şekil 2.4'te görüldüğü üzere, STS'ler genel olarak ağın arka bağlantı noktasında yer almaktadır.

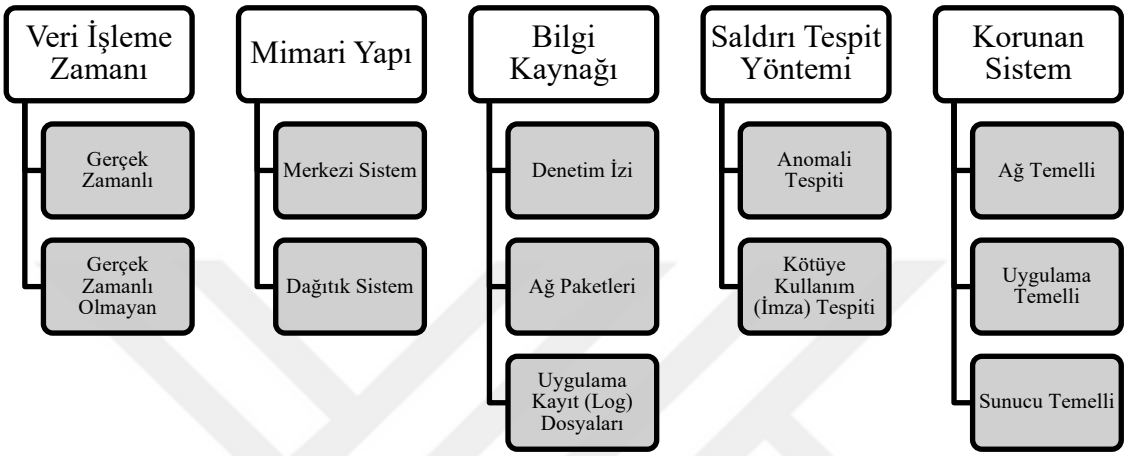


Şekil 2.4. STS Mimarisi

İdeal bir STS'nin sahip olması gereken özellikler şu şekilde sıralanabilir:

- Zamanlılık: Anormal davranışın öngörülen süre içinde veya hemen sonra tespit edilebilmesidir.
- Yüksek algılama oranı: Anormal davranışların çoğunun tespit edilebilmesidir.
- Düşük yanlış alarm oranı: Çok az sayıda yanlış alarma izin verilmesidir.
- Belirlilik: Saldırı tespiti yapıldığında daha iyi bir yanıt almak için yeterli ve detaylı bilginin mevcut olmasını gerektirir.
- Ölçeklenebilirlik: Büyük ve küçük ağlara için uygulanabilir olmasıdır.
- Düşük öncelik bilgisi: Potansiyel saldırganlar ve stratejileri hakkında en az öncelik bilgisine ihtiyaç duyulmasıdır. (Garg ve ark., 2016).

Saldırı tespit sistemlerinin sınıflandırmalarında pek çok ölçüt kullanılmıştır. STS stratejilerinin de belirlendiği, kullanılan genel ölçütler korunan sistem, saldırı tespit yöntemi, bilgi kaynağı, mimari yapı ve veri işleme zamanıdır. Saldırı tespit yöntemi en çok bilinen sınıflandırma türüdür. Bu türde STS'ler, Kötüye Kullanım (İmza) Tespiti ve Anomali Tespiti olmak üzere ikiye ayrılır. Şekil 2.5 STS'lerin genel ölçütler doğrultusunda sınıflandırmasını içermektedir.



Şekil 2.5. STS'lerin Sınıflandırılması

2.2.1. Veri İşleme Zamanı

Aktivite izleme ve analiz etme arasındaki süredir. Bu ölçüt için saldırı tespit sistemleri ikiye ayrılmaktadır. Bunlar gerçek zamanlı ve gerçek zamanlı olmayan zamanlı'dır.

2.2.1.1. Gerçek Zamanlı Saldırı Tespit Sistemleri

Gerçek zamanlı STS'lerde verinin anlık analizi yapılır. Analiz sonucunda aktivitenin saldırı olduğuna karar verilirse gerekli işlemler yine anlık gerçekleştirilir. Saldırılar ciddi kayıplara sebep olabileceğinden ticari amaçlı kullanılan sistemler çoğunlukla gerçek zamanlı olarak tasarlanmıştır.

2.2.1.2. Gerçek Zamanlı Olmayan Saldırı Tespit Sistemleri

Gerçek zamanlı olmayan STS'lerde, veri kaydedilir ve analizinin yapılması amaçlı uygun STS'ye yönlendirilir (Murali ve ark., 2005). Belirlenen tehditler ve olası saldırılar için sistem kullanıcısına uyarı verilir. Saldırının profili çıkartılır. Sonuçlar geçmişte kaydedilmiş veriler kullanılarak gerçekleştirilen analizler doğrultusunda elde edilir. Gerçek zamanlı olmayan saldırı tespit sistemleri çoğunlukla sistem üzerinde belirli, kaydedilmiş güvenlik zafiyetlerini kapatmak için kullanılır.

2.2.2. Mimari Yapı

İşlevsel parçaların karşılıklı olarak nasıl yerleştirildikleri üzerinde durulur. Ana işlevsel parçalar şunlardır; analizin yapıldığı sunucu ve çevresi, izlenen sistem ve problemler için izlenen hedef. Bu ölçüt için saldırı tespit sistemleri ikiye ayrılmaktadır. Bunlar merkezi sistem ve dağıtık sistem'dir.

2.2.2.1. Merkezi Sistem

İzleme, tespit ve raporlama işlemlerinin tümünün kontrolleri merkezde yapılmaktadır. Merkezi sistem kontrolleri için fiziksel yakınlık önemli bir etken değildir.

2.2.2.2. Dağıtık Sistem

İzleme, tespit ve cevap işlemleri merkezi bir sistemde olmayıp kurulan ajan (agent) tabanlı programlar yardımıyla kontrol edilir. Çoğunlukla çok geniş ağlarda kullanılırlar. Bir kuruma ait farklı bölgelerde bulunan tüm bilgisayar sistemlerini içeren saldırı tespit sistmeleri dağıtık sistem için örnek olarak gösterilebilir.

2.2.3. Bilgi Kaynağı

Bilgi kaynaklarının analiz ve karşılaştırmasında kullanılmaktadır. Bilgi kaynakları, bilgisayar veya ağ faaliyetlerinin dinlenmesi yoluyla edinilebildiği gibi, kullanıcı profillerinin davranış modelleri üzerinden de elde edilebilir.

Geliştirilecek olan STS'nin amacına göre bilginin nasıl edinileceği belirlenir. Bu ölçüt için saldırı tespit sistemleri üçe ayrılmaktadır. Bunlar denetim izi, ağ paketleri ve uygulama kayıt dosyaları'dır.

2.2.3.1. Denetim İzi

Bilgi güvenliğini sağlamak amaçlı sistem aktivitelerinin tarih bazında sıralanmış şeklidir. İzlenen aktivitelerdeki sıralama bozulduğunda veya herhangi bir değişiklik oluştuğunda denetim izleri sayesinde aktiviteler yeniden yapılandırılır ve test edilir.

STS'ler denetim izini, genellikle sistemde tanımlı kullanıcı ve grupların hareket profillerini çıkarmak için kullanır. Sistemde tanımlı yetkiler doğrultusunda kullanıcıların günlük olarak yaptıkları işler, o kullanıcıların profillerini oluşturur. Kullanıcı profili oluştuğunda, STS'ler oluşan profil dışı hareketleri saldırı olarak tanımlarlar. Doğru alarm yüzdesini artırmak amaçlı kullanıcı profillerinin güncel durumda kalması gerekir.

2.2.3.2. Ağ Paketleri

Ağ paketleri, koklayıcı programlar kullanılarak trafiğin dinlenmesi ile edinilir ve çoğunlukla hizmet engelleme saldırılarını tespit etme amaçlı kullanılır. Sunucu tabanlı STS'lerden farklı olarak, paketlerden edinilen bilgiler ile birlikte ağ katmanında gerçekleşen saldırılar da tespit edilebilir. Bu tespitlerin yapılabilmesi STS'nin ağ yapılandırılmasında ideal noktaya yerleştirilemesi ile sağlanabilir.

2.2.3.3. Uygulama Kayıt Dosyaları

Uygulama katmanında meydana gelen saldırıların tespitinde kullanılır. Uygulama kayıt dosyaları, diğer bilgi kaynaklarına nispeten kolayca elde edilebilir fakat sınırlı sayıda saldırıyı tespit etme imkânı sunar.

2.2.4. Saldırı Tespit Yöntemi

Bu ölçüt için saldırı tespit sistemleri ikiye ayrılmaktadır. Bunlar, anomali tespiti ve kötüye kullanım (imza) tespiti'dir. Anomali tespiti yaklaşımında, sistemdeki kullanıcı davranışları modellenirken, kötüye kullanım (imza) tespiti yaklaşımında saldırgan davranışları modellenir.

2.2.4.1. Anomali Tespiti

Anomali tespitinde, sistem üzerindeki normal olmayan davranışların analizi ve tespiti yapılır. STS'ler için anomali, normal davranıştan sapma anlamına gelir.

Bu sistemdeki çalışma mantığı, kullanıcıların normal çalışma sürecindeki hareketlerinin takip edilerek normal davranış profillerinin çıkartılmasıdır. Bu profiller oluştuğunda, sistem üzerindeki hareketler izlenerek anormal davranışlar tespit edilir ve saldırı olarak tanımlanır. Saldırıların başarılı bir şekilde tespit edilebilmesi, normal davranış profillerinin doğru belirlenmesine bağlıdır.

Anomali tabanlı STS'lerin imza tabanlı STS'lere göre avantajı hakkında çok fazla bilgi sahibi olunmayan saldırıların tespit edilebilir olmasıdır. Bu tür STS'lerdeki dezavantaj, normal davranışın saldırı olarak nitelendirilme yüzdesinin yüksekliğidir.

Anomali tespit etmek için yapay zekâ, veri madenciliği ve istatistiksel yöntemler gibi birçok farklı teknikten faydalanılabilir. Bu teknikler ve yaklaşımlar 2.3 Saldırı Tespit Sistemlerinde Kullanılan Yöntemler bölümünde açıklanmaktadır.

2.2.4.2. Kötüye Kullanım (İmza) Tespiti

Kötüye kullanım tespitinin temelinde saldırgan hareket modellerinin çıkarılması bulunmaktadır. Çıkarılan modeller Saldırı imzası olarak isimlendirilirler ve önceden karşılaşılmış saldırıların incelenip karakteristik özelliklerinin çıkarılması sonucu oluşurlar. Saldırılara ilişkin imza veri tabanlarının oluşmasının ardından kötüye kullanım kontrolü yapan dedektörler ağdaki hareketleri izler ve belirlenen imzalarla karşılaştırır. Karşılaştırma sonucunda eşleşen hareketler saldırı olarak tanımlanır.

İmza tabanlı STS'lerin anomali tabanlı STS'lere göre avantajı, imzası bilinen tüm saldırıların tespit edilebilmesidir. Bu tür STS'lerdeki dezavantaj, imzasının bilinmiyor olması durumunda saldırının normal olarak nitelendirilme yüzdesinin yüksek olabilmesidir. Bu yüzdenin azalması yeni saldırı imzalarının saldırılara ilişkin imza veri tabanına eklenmesine bağlıdır.

2.2.5. Korunan Sistem

Kordukları sisteme göre STS'ler ağ, sunucu ve uygulama temelli olmak üzere üçe ayrılırlar.

2.2.5.1. Ağ Temelli

Ağ temelli STS'ler dinlenen ağ trafiği üzerinden analiz yaparak, ağ sistemine yönelik saldırıları tespit etmeye çalışır. Ticari amaçlı kullanılan STS'ler çoğunlukla ağ temellidir. Ağ temelli STS'lerin bazı avantajları ve dezavantajları aşağıda verilmiştir (Aydın, 2005).

Avantajlar;

- Geniş bir ağ denetlenebilir.
- Ağdaki işleyiş etkilenmez.
- Oldukça gizli olmaları sebebiyle çoğu saldırgan tarafından tespit edilemezler.
- Genel olarak pasif aygıtlardır.

Dezavantajlar;

- Geniş ağdaki tüm paketlerin işlenmesi zordur.
- Yoğun trafik altında gerçekleşen bir saldırı tespit edilemeyebilir.
- Şifrelenmiş bilgi analiz edilemez.
- Başlatılan saldırılar tespit edilebilir.
- Parçalı ağ paketlerinden oluşan ağ tabanlı saldırılar tespit edilemeyebilir.

2.2.5.2. Sunucu Temelli

Sunucu temelli STS'ler işletim sistemine ait hesap izlerinin ve sistem kayıtlarını toplayarak bu veriler üzerinde çalışırlar. Böylece işletim sistemine yönelik saldırılara sebep olan sistem çağrıları ve kullacılar tespit edilebilir. Sunucu temelli STS'lerin bazı avantajları ve dezavantajları aşağıda verilmiştir (Aydın, 2005).

Avantajlar;

- Olayların gerçekleştiği lokal sunucuyu izleme kabiliyetleri sebebiyle ağ temelli STS'lerde yakalanamayan saldırıların tespitini yapabilirler.
- Ağ trafiğinin şifrelendiği bir ortamda çalışabilirler.
- Anahtarlamalı ağlar üzerinde iyi çalışırlar.
- Truva atları ve diğer yazılımsal saldırıları tespit edebilirler.

Dezavantajlar;

- Yönetimi oldukça zordur.
- Üzerinde çalıştığı sunucunun saldırıya altında olması devre dışı kalmasına sebep olabilir.
- İşletim sistemi izleme modunda veri boyutu çok büyük olabilir.
- Ağın taranmasından haberdar olunmaz.

2.2.5.3. Uygulama Temelli

Sunucu temelli STS'lerin alt bir kümesi olan bu STS'lerde olayların yazılım uygulaması kullanılarak analizi yapılır. Bilgi kaynağı uygulamanın günlük dosyalarıdır. Analiz motoruna, belirlinen uygulamalara ait özel özellikler bildirilirse, yetkisiz kullanıcıların gerçekleştirdiği saldırılar tespit edilebilir (Axelsson, 2000). Uygulama temelli STS'lerin bazı avantajları ve dezavantajları aşağıda verilmiştir (Aydın, 2005).

Avantajlar;

- Kullanıcı ve uygulama etkileşimi incelenerek bireysel kullanıcıların yetkilerini aşan davranışları tespit edilebilir.
- Şifrelenmiş veri işlenebilir.

Dezavantajlar;

- Saldırılara daha fazla açıktırlar.
- Truva atları ve diğer yazılımsal saldırıları tespit edemezler.

2.3.Saldırı Tespit Sistemlerinde Kullanılan Yöntemler

Anomali ve imza tabanlı yaklaşımların modellenmesi için günümüze kadar pek çok yöntem kullanılmıştır (Patcha ve ark, 2007). Bu yöntemler, verinin modellenmesi, kural tablolarının oluşturulması ya da sınıflandırılması gibi amaçlar için geliştirilmiştir. Kullanılan yöntemler doğrultusunda edinilen veriler, saldırı tespit süreçlerinin uygulanması konusunda gereken alt yapıyı sağlamaktadır. STS'lerde kullanılan yöntemlerden bazıları aşağıda verilmiştir.

- Kural Tabanlı Sistemler: Sisteme ait trafik üzerinden kurallar oluşturulur. Saldırı tespiti yapılırken oluşturulan bu kurallar doğrultusunda sınıflandırma yapılır (Ilgun, 1995).
- Uzman Sistemler: Spesifik bir konu ile alakalı donanımlı ve problemlere sadece o konunun uzmanı olan biri tarafından sunulabilecek çözümler getirebilen bilgisayar programları olarak tanımlanabilir. Sızmayı tespit eden sistemlerin ilkleri kural-tabanlı uzman sistemlerdir (Axelsson, 2000).
- Veri Madenciliği: Büyük bir veri yığını arasından fayda sağlayan bilgilerin çıkarılmasıdır. Örüntü (pattern) ve veri arasında olan ilişkilerden kural çıkarımı yapmak amacıyla uygulanır. Böylece, hesap izlerini kullanarak normal kullanıcı faaliyetleri tanımlanır (Lee ve ark., 2001).
- Örüntü Eşleme: Sistem üzerinde tanımı yapılmış ve karşılaşılması beklenmeyen bazı kelimelerin tespit edilmesi için kullanılır. Örnek olarak, "parolayı kopyala" komutunun bir atak olduğu örüntü eşleme ile kolayca belirlenebilir (Axelsson, 2000).
- Açıklayıcı İstatistikler: Kullanıcı veya sistem hareketleri, belli bir zaman periyodunda erişilen dosya sayısı, kullanılan disk alanı/hafıza, kullanıcı oturum girişi ve oturum kapatma gibi farklı değişkenler bazında ölçülerek istatistiksel bir model oluşturulur. Basit istatistikler ile kullanıcı profilini oluşturup, bu profil üzerinden uzaklık vektörleri yardımıyla karara varan sistemlerdir. Hareket profilinin oluşturulma aşamasında, bir zaman periyodundaki ağ bağlantı sayısı ve kullanılan işlemci zamanı gibi metrikler de kullanılabilir. İstatistiklerin saldırgan tarafından öğrenilmesi ve saldırganın öğrendiklerine göre bir davranış göstermesi, bu yöntemin önemli dezavantajlarından biridir (Sundaram, 1996).

- Yapay Sinir Ağları: STS'lerde 1990'ların başında kullanılmaya başlanmış akıllı tekniklerdendir. Anomalileri tespit eden STS'ler için istatistiksel yöntemlerin bir alternatifidir (Cannady, 1998). Bu ağlar giriş/çıkış vektörleri arasında bağıntı kurarak kendi algoritmalarını uygularlar. Sonrasında genelleştirme yaparak yeni giriş/çıkış bağıntıları oluştururlar. Bu teknik, kullanıcı hareketlerinin öğrenilmesiyle gerçekleşmektedir. Spesifik olarak, bir kullanıcının önceki hareketlerinden yola çıkarak yeni hareketini tahmin ederler. YSA'nın normal sistem hareket izleriyle eğitilmesi, YSA'ların STS'lerdeki kullanımının ilk aşamasıdır. Olay akışları anormal veya normal olarak sınıflandırıldıktan sonra YSA'ya iletilir. Toplanan veriler ile sistem hareketleri doğrultusunda öğrenme değişimi gerçekleştirilebilir. Yani eğitim izin veriliyorsa sürekli hale getirilebilir. Burada yaklaşım, kullanıcının n tane komut ya da hareketinden, sonraki komut ya da hareketinin tahmin edilebilmesidir (Sundaram, 1996). Tahmin edebilme için gerekli ilk adım eğitim veri kümesinin oluşturulması ardından da eğitimin tamamlanmasıdır. Anomali tespit etme de kullanıldıkları gibi kötüye kullanım (imza) tespiti için de kullanılabilirler. Ağ saldırılarının sıkça değişiyor olması, ağ trafiğinin geniş çapta analizini yapan ve kural tabanlı sistemlerden daha esnek olan bir savunma sistemi ihtiyacı oluşturmuştur. YSA tabanlı kötüye kullanım (imza) tespiti yapan sistemlerle, kural tabanlı sistemlerde var olan bu tür problemlere çözüm getirilebilmektedir (Cannady, 1998).
- Destek Vektör Makineleri: Eğitim verilerini yüksek boyutlu bir özellik uzayında çizerek her vektörü kendi sınıfı ile etiketler. Genellikle özellik vektörünün seçilmesinde kullanılır. Hızları sebebiyle oldukça kullanışlı bir yöntemdirler. Geniş bir örnek kümesini öğrenebilir ve iyi ölçeklendirirler (Mukkamala ve ark., 2003).
- Yapay Bağışıklık Sistemi: İnsandaki bağışıklık sistemi düşünülerek geliştirilmiştir. Literatürde insan bağışıklık sistemi ile ilgili pek çok çalışma yer almaktadır. Bu çalışmalardan bazıları doğal bağışıklık sisteminin özelliklerini temel alarak anomali tespiti için biçimsel çatı ileri sürmüşlerdir ve doğal bağışıklık sisteminin bağışıklığı tanımladığını düşündükleri dört mühim özelliği, kullanmışlardır. Bu özellikler hata toleransı (error tolerance), farklılık (diversity), doğal dinamik yapı (dynamic nature) ve doğal dağıtık yapı (distributed nature)'dır (Esponda ve ark, 2004).

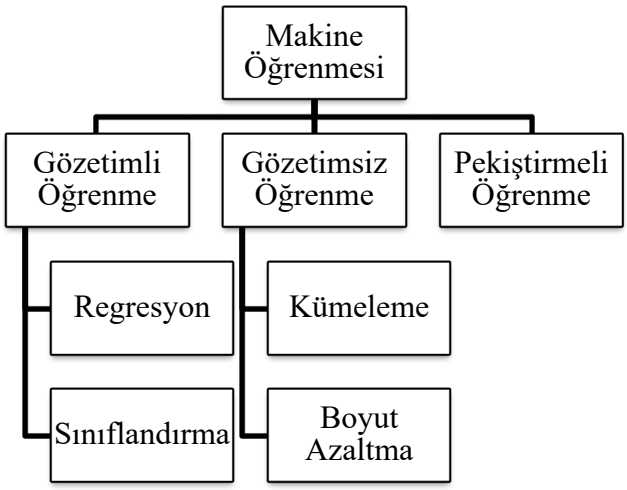
- Genetik Algoritmalar: Canlılardaki doğal gelişim prensibine dayanmaktadır (Stein ve ark., 2005). Bu algoritmalar, çözümlerin başlangıç popülasyonunu oluşturması için bir yöntem, genetik yapıyı değiştirecek genetik operatörler popülasyonun büyüklüğü ve çözümlerin uygunluğunun değerlendiriliceği bir değerlendirme fonksiyonu gibi kontrol parametrelerine ihtiyaç duyar.
  Bu algoritmalar STS'lerde, ağ trafik verilerindeki anormal trafik verilerini normal olanlardan ayırmak amaçlı basit kurallar uygulamak için kullanılır.

2.4.Makine Öğrenmesi Türleri

Temeli istatistik ve yapay zekâya dayanan, büyük veri yığınlarından faydalı bilgiyi edinmemize olanak tanıyan tekniklere sahip bir araştırma alanıdır. Bu bilgiler, verilerin makine öğrenmesi algoritmaları kullanılarak öğrenilmesi sonucunda varsayımlarda bulunmamızı sağlayan modellerden edinilir. Öğrenme fonksiyonu için farklı yaklaşımlara sahip pek çok algoritma geliştirilmiştir. Bir öğrenme algoritmasının girdileri deneyimlerden elde edilmiş eğitim verilerinden oluşmaktadır. Çıktılar ise farklı diğer sistemler tarafından da kullanılabilecek üzerinde fikir yürütülebilecek yeni bilgilerdir (Balta, 2018). Temel anlamda öğrenme fonksiyonu, belirli bir görevin işleyişini daha iyi hale getirmek için yeni bilgilerin kullanılmasıdır (Fielding, 1999). Öğrenme işlevi için genel olarak verilerden edinilen örüntüler (pattern) kullanılır.

Bu açıdan makine öğrenmesi ve veri madenciliği verilerden örüntüleri çıkarma konusunda benzerlik gösterirler. Veri madenciliği veriler üzerinden daha önce bilinmeyen yararlı bilgiler elde edilmesi konusuna yoğunlaşırken makine öğrenmesi tanımlama ve kestirim üzerine yoğunlaşır. Tanımlama ve kestirim, deneysel eğitim verilerinin bilinen özelliklerinin bilgisayar algoritmaları kullanılarak öğrenilmesi ile gerçekleştirilir (Balta, 2018). Eğitim verileri kullanılarak yapılan öğrenme işlemine tümevarım denir.

Tümevarımda amaç, eğitim verilerinden hedef değerlerinin tahminine yönelik bir model geliştirmek olduğunda bu öğrenme yaklaşımına, gözetimli öğrenme denir (Debeljak, 2012). Makine Öğreniminin üç farklı alt kategorisi bulunmaktadır. Bunlar Şekil 2.6'da belirtildiği üzere Gözetimli (Supervised), Gözetimsiz (Unsupervised) ve Pekiştirmeli (Reinforcement) Öğrenme'dir.



Şekil 2.6. Makine Öğrenimi Türleri

2.4.1. Gözetimli Öğrenme

Denetimli makine öğreniminde, modellerin eğitimi için kullanılan veri kümesi girdi verilerini ve girdi verilerine karşılık gelen, hedef değerlerinin bulunduğu çıktı verilerini içerir (Vieira ve ark., 2019; Subasi, 2020). Temel amaç girdi ve ilgili çıktı verilerinin (etiketlenmiş veri) arasındaki ilişkinin öğrenilmesi ve modelin eğitilmesidir. Model eğitildikten sonra, daha önce karşılaşılmamış olan test verilerinin çıktı hedef değerlerini başka bir deyişle etiketlerini tahmin etmek için kullanılır. Model performansı, tahmin edilen ve gerçek çıktı hedef değerlerinin karşılaştırılması sonucu edinilir. Bu öğrenme türünün gözetimli olarak isimlendirilmesinin sebebi, eğitilen modelin olması gereken çıktı hedef değerleri ile ilgili önceden bilgi sahibi olmasıdır. Denetimli öğrenme yöntemleri tahmin edilecek çıktı verilerinin sürekli ya da kategorik değişken olmalarına göre ikiye ayrılır.

Bunlar; sırasıyla sınıflandırma ve regresyondur (Vieira ve ark., 2019; Subasi, 2020). Rastgele Orman, Karar Ağaçları, K-En Yakın Komşu, Lojistik Regresyon ve Destek Vektör Makineleri popüler denetimli öğrenme algoritmalarından bazılarıdır. Denetimli öğrenme yöntemleri, yüz tanıma, anomali tespiti ve hava durum tahmini vb. alanlarda kullanılmaktadır.

2.4.2. Gözetimsiz Öğrenme

Gözetimsiz makine öğreniminde, model eğitimi için kullanılan veri kümesi yalnızca girdi verilerini içerir (Vieira ve ark., 2019; Subasi, 2020). Gözetimli öğrenme yönteminin aksine eğitim aşamasında kategorize edilmiş veya etiketlenmiş herhangi bir veri bulunmaz.

Bu yöntemdeki temel amaç girdi verileri arasındaki ilişkilerin, düzenlerin, benzerliklerin bulunması ve yoğunluk tahmini yapılmasıdır. Gözetimsiz öğrenme yöntemleri boyut azaltma ve kümeleme olmak üzere ikiye ayrılır (Vieira ve ark., 2019; Subasi, 2020). Temel Bileşen Analizi ve K-Means popüler gözetimsiz öğrenme algoritmalarından bazılarıdır. Gözetimsiz öğrenme yöntemleri, görüntü sıkıştırma, müşteri ilişkileri yönetimi vb. alanlarda kullanılmaktadır.

2.4.3. Pekiştirmeli Öğrenme

Pekiştirmeli makine öğreniminde, modellerin eğitimi için kullanılan veri kümesi gözetimsiz öğrenmedeki gibi yalnızca girdi verilerini içerir (Yılmaz, 2019). Bu yöntemde, girdilerin beklenen çıktısı ile modelin tahmin ettiği çıktı arasında karşılaştırma yaparak, üretilen sonucunun doğru olup olmadığını modele bildiren bir danışman bulunur.

Danışman üretilen sonucu modele tanıtmaz, yalnızca model tarafından üretilen sonucun doğruluğu ya da yanlışlığı ile ilgili geri dönüş sağlar (Hamzaçebi, 2011; Yılmaz, 2019). Model tarafından üretilen sonuç doğru ise model ödüllendirilir, aksi halde cezalandırılır. Bu şekilde modelin danışman tarafından gelen geri dönüşler doğrultusunda en uygun adımları atarak öğrenmeye devam etmesi pekiştirmeli öğrenme olarak isimlendirilir (Shobha ve Rangaswamy, 2018; Bilgin, 2018; Yılmaz, 2019). Q-Learning algoritması popüler pekiştirmeli öğrenme algoritmalarından biridir. Pekiştirmeli öğrenme yöntemleri, satranç, tavla oyunları, çeşitli optimizasyon problemleri, konuşma tanıma vb. alanlarda kullanılmaktadır.

2.5.Makine Öğrenmesi Teknikleri

Çalışmada kullanılan makine öğrenmesi algoritmaları Karar Ağacı, Gradyan Artırma, Adaptif Yükseltme, Lojistik Regresyon, Pasif-Agresif, Ekstra Ağaçlar ve Çok Katmanlı Algılayıcı'dır. Literatürde var olan diğer çalışmalarla kıyaslayabilmek adına algoritma hiperparametreleri için sklearn kütüphanesinde tanımlı varsayılan değerler kullanılmıştır. Kullanılan algoritmalarla ilgili detaylar devam eden kısımda yer almaktadır.

2.5.1. Adaptif Yükseltme Algoritması

1996 yılında Yoav Freund ve Robert Schapire tarafından tanıtılan ADA, Gödel ödülüne layık görülmüş ilk başarılı boosting algoritmasıdır. Bu algoritmada amaç, zayıf sınıflandırıcıları bir araya getirerek etkili bir sınıflandırıcı oluşturmaktır.

Çalışma mantığı şöyledir: Her bir tekrarda temel sınıflandırıcı adı verilen basit bir öğrenme algoritması çağırılır ve sınıflandırıcıya hatası ile ters orantılı olacak şekilde bir ağırlık katsayısı verilir. Ağırlık katsayılarına bağlı ağırlıklı oylama sonucunda çözüm elde edilir. ADA algoritması sınıflandırma problemlerinin çözümü için ileri sürülmüştür. Daha sonra, Drucker (Drucker, 1997) tarafından regresyon problemlerine uyarlanmış ve uyarlanan bu sürüm literatüre AdaBoost.R2 olarak geçmiştir (Acılar, 2020). Tablo 2.1'de algoritmanın sklearn kütüphanesi üzerinde tanımlı varsayılan hiper-parametre değerleri yer almaktadır.

Tablo 2.1. Adaptif Yükseltme Hiper-parametre ve Değerleri

Hiper-Parametre	Değer
Tahmin Edici Sayısı	50
Öğrenme Oranı	1.0
Algoritma	Samme.R

Veri kümesinin sınıflandırılması için kullanılan son denklem (2.1)'de gösterilmiştir;

$$\mathrm{F(x)} = \operatorname{sign}\left(\sum_{i=0}^{n} \theta_i f_i(\mathrm{x})\right) \qquad (2.1)$$

Buna göre fi, i'inci zayıf sınıf bilgisi ve θi buna karşılık gelen ağırlık bilgisidir.

2.5.2. Karar Ağacı Algoritması

Karar Ağaçları, sınıflandırma işlemlerinde yaygın olarak kullanılan gözetimli öğrenme yöntemlerinden biridir. Hızlı adaptasyonu, sadeliği ve yüksek algılama hassasiyeti sebebiyle sıklıkla tercih edilmektedir.

Karar ağaçları üç bölümden oluşur bunlar; karar düğümleri, dallar ve yapraklardır (Han ve ark., 2000). Karar düğümü gerçekleştirilecek olan testtir. Test sonucu, ağacın veri eksilmeksizin dallara bölünmesini sağlar. Her düğümde test ve dallara bölünme işlemleri sıralı olarak gerçekleşir. Bu bölünme işlemi üst seviyedeki ayrımlara bağlıdır. Ağaç dallarından her biri sınıflama işlemini yapmaya adaydır.

Eğer bir dalın ucunda sınıflama işlemi yapılamıyorsa, o dalın sonunda bir karar düğümü üretilir. Ancak dalın sonunda belirli bir sınıf oluşuyorsa, o dalın sonunda bir yaprak vardır ve bu yaprak veri üzerinde belirlenmek istenen sınıflardan birine aittir.

Karar ağaçlarında işlem, kök düğümünden başlar ve yukarıdan aşağıya doğru yaprağa ulaşılana kadar ardışık düğümleri takip ederek devam eder. Karar ağacı tekniğini kullanarak verinin sınıflandırılması iki basamaklı bir işlemdir (Han ve ark., 2000). Birinci basamak olan öğrenme basamağında, önceden bilinen bir eğitim verisi, model geliştirmek amaçlı sınıflandırma algoritması tarafından analiz edilir. Öğrenilen model, sınıflandırma kuralları veya karar ağacı olarak gösterilir. Sonraki basamak sınıflandırmadır. Bu basamakta test verisi sınıflandırma kurallarının veya karar ağacının doğruluk değerini belirlemek amaçlı kullanılır. Eğer doğruluk değeri geçerli bir oranda ise, kurallar yeni verilerin sınıflandırmasında kullanılır (Özekes, 2003).

Yaygın olarak bilinen ve kullanılan karar ağacı algoritmaları şunlardır; C4.5, ID3 ve C5. Tablo 2.2'de algoritmanın sklearn kütüphanesinde tanımlı varsayılan hiper-parametre değerleri verilmiştir.

Tablo 2.2. Karar Ağacı Hiper-parametre ve Değerleri

Hiper-Parametre	Değer
Kriter	Gini
Ayrıştırıcı	Best
Minimum Yaprak	1
Minimum Ayrılma	2

Karar ağaçları için Entropi ve Bilgi Kazancı hesaplanmalıdır. Denklem (2.2) Entropinin nasıl hesaplandığını göstermektedir.

$$\mathrm{E(S)} = \sum_{\mathrm{x}\in X} -\mathrm{p(x)E(x)} \tag{2.2}$$

Burada S veri setidir, x, S'deki sınıfların kümesi ve p, x sınıfındaki öğelerin sayısının oranıdır. Denklem (2.3), bilgi kazancının nasıl hesaplandığını göstermektedir.

$$Gain\ (A,S) = E(S) - \sum_{\mathrm{a}\in \mathrm{T}} p(x)E(x) \tag{2.3}$$

2.5.3. Ekstra Ağaçlar Algoritması

Rastgele Orman makine öğrenmesi yönteminin farklı bir uyarlamasıdır. Bu yöntemde modelin eğitimi, Rastgele Orman yöntemindeki gibi veri kümesinin kopyaları kullanılarak yapılır.

Rastgele Orman yönteminden ayrıldığı nokta, düğümler dallara ayrılırken karar kriteri kullanılarak optimum ayrılma yerine rastgele bir kesme noktası seçmesidir. Buradaki kesme noktası, özelliğin ampirik aralığı içindeki (ağacın eğitim setinde) eşit dağılımdan belirlenir. Sonrasında, bölme için rastgele oluşturulanlardan en yüksek puanlı bölünme düğümü seçilir. Rastgele Ormanlara benzer şekilde, her bir düğümde dikkate alınacak rastgele seçilen özelliklerin sayısı belirlenebilir. (Geurts ve ark., 2006). Tablo 2.3'te algoritmanın sklearn kütüphanesi üzerinde tanımlı varsayılan hiper-parametre değerleri yer almaktadır.

Tablo 2.3. Ekstra Ağaçlar Hiper-parametre ve Değerleri

Hiper-Parametre	Değer
Kriter	Gini
Tahminci Sayısı	100
Minimum Ayrılma	2
Minimum Yaprak	1

2.5.4. Lojistik Regresyon Algoritması

Lojistik regresyon, bağımlı değişken ikili ya da ikiden daha fazla kategori içeren kesikli değişken olduğunda normallik varsayımı olmaksızın sınıflandırma ve atama işlemlerini yapmayı sağlayan bir regresyon yöntemidir. Bu sebep, Lojistik regresyonu sık tercih edilen bir yöntem kılmıştır (Tatlıdil, 2002; Hosmer ve ark., 2000).

Denklem (2.4) i. gözlem için genel doğrusal regresyon modelini göstermektedir.

$$y_i = \sum_{k=0}^{p} \beta_k x_{ik} + u_i \quad \text{i} = 1,2,3,\ldots,\text{n} \tag{2.4}$$

Genel doğrusal regresyon modelinde bağımsız değişkenler üzerinde bir kısıt yok iken, y_i bağımlı değişkenin sürekli olma koşulu vardır. Bu sebeple y_i bağımlı değişkeni -∞ ile +∞ arasında herhangi bir değer alabilmektedir.

Bağımlı değişken y_i 'nin 0-1 gibi kategorik değerler alması durumunda dağılım bozulmaktadır. Bu durumda, i'inci gözlemin 1 değerini alma olasılığı P (y_i = 1) olmak üzere, beklenen değer, aşağıdaki şekilde elde edilir.

$$E\ (y_i) = 1 \times \text{P}(y_i = 1) + 0 \times \text{P}(y_i = 0) = \text{P}(y_i = 1)$$

Denklem (2.5), elde edilen beklenen değerin regresyon denklemi olarak yazılışını göstermektedir.

$$E(y_i) = \mathrm{P}(y_i = 1) = \sum_{k=0}^{p} \beta_k x_{ik} \quad \mathrm{i} = 1,2,\dots,\mathrm{n} \tag{2.5}$$

Olasılık değerleri 0-1 arasında olan Denklem (2.5) doğrusal olasılık modeli olarak adlandırılmaktadır. Doğrusal olasılık modeli bağımsız değişkenlerin sınırsız değerler alması sebebiyle her zaman sağlanamamaktadır (Tatlıdil, 2002). Bu gibi durumlardan sakınmak için çeşitli dönüşümler yapılabilmektedir. En yaygın kullanılan iki dönüşüm birbirine yakın sonuçlar veren probit (probability unit) ve lojit (logit) dönüşümleridir (Güneri ve ark., 2004). Lojit dönüşümde, doğrusal olasılık modelinin olasılık değerleri üzerinde P/(1-P) dönüşümü yapıldıktan sonra bu oranın doğal logaritması alınarak sonuç değişkeninin sınırları -∞, +∞ yapılmaktadır. Denklem (2.5)'te gösterilen doğrusal olasılık modeline lojit dönüşüm yapıldıktan sonra Denklem (2.6)'da gösterilen lojistik model ya da lojit olarak isimlendirilen model oluşur.

$$E(y_i) = L_i = \log \frac{P_i}{(1 - P_i)} = \sum_{k=0}^{p} \beta_k x_{ik} \quad \mathrm{i} = 1,2,\dots,\mathrm{n} \tag{2.6}$$

Bu modelde Pi olasılık değeri aşağıdak şekilde elde edilir.

$$P_i = \frac{\exp\left(\sum_{k=0}^{p} \beta_k x_{ik}\right)}{\left(1 + \exp\left(\sum_{k=0}^{p} \beta_k x_{ik}\right)\right)} \qquad \mathrm{i} = 1,2,\dots,\mathrm{n} \tag{2.7}$$

Denklem (2.7)'de gösterilen fonksiyon 'lojistik fonksiyon' olarak adlandırılmaktadır. Lojit dönüşüm yapılarak, lojistik fonksiyon doğrusallaştırılabilir (Denklem (2.8)).

$$\eta = \ln \frac{P_i}{(1 - P_i)} \qquad \mathrm{i} = 1,2,\dots,\mathrm{n} \tag{2.8}$$

Denklem (2.9) dönüşüm uygulandıktan sonra elde edilen lojistik regresyon modelini göstermektedir. Bu fonksiyonda, $E\,(y_i)$ –∞, + ∞ arasında değer almaktadır (Agresti, 2002).

$$E\,(y_i) = \eta = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} \quad i = 1,2,\dots,n \tag{2.9}$$

n birimlik bir örneklem için lojistik regresyon tahmin denklemi aşağıdaki gibidir.

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \cdots + \hat{\beta}_k x_{ki} \quad i = 1,2,\dots,n$$

Lojistik regresyon denklemi kullanılarak Denklem (2.10)'da hesaplanan Pi, 0,5'ten büyük ise $\hat{y}i$=1, aksi takdirde $\hat{y}i$=0 biçiminde sınıflandırılır (Agresti, 2002; Tatlıdil ve ark., 2014).

$$P_i = \frac{e^{\hat{y}_i}}{1 + e^{\hat{y}_i}} \qquad \text{i} = 1,2,\ldots,\text{n} \tag{2.10}$$

Bu yöntem, bağımlı değişkenin ölçek türüne göre üçe ayrılır. Bağımlı değişken iki kategorili ise ikili (binary) lojistik regresyon, ikiden çok sayıda sırasız ve kategorili ise çok kategorili (multinominal) lojistik regresyon ve ikiden çok sayıda sıralı ve kategorili ise sıralı (ordinal) lojistik regresyon olarak isimlendirme yapılır (Alpar, 2017). Tablo 2.4'te algoritmanın sklearn kütüphanesi üzerinde tanımlı varsayılan hiper-parametre değerleri yer almaktadır.

Tablo 2.4. Lojistik Regresyon Hiper-parametre ve Değerleri

Hiper-Parametre	Değer
Ceza	L2
Tolerans	0.0001
Çözücü	Lbfgs
Maksimum İterasyon	100

2.5.5. Gradyan Artırma Algoritması

Gradyan artırma, sınıflandırma ve regresyon problemlerinde kullanılan bir makine öğrenmesi tekniğidir. Bu algoritmada temel amaç, yüksek doğruluk oranına sahip "güçlü bir öğrenici" oluşturmak için "zayıf öğrenici" olarak isimlendirilen birkaç basit modelin tekrarlamalı olarak biraraya getirilmesidir. Artırma algoritmasını kayıp fonksiyon kavramına bağlayan Friedman, artırmaya ilişkin istatistiksel bir bakış açısı ileri sürmüştür.

Bu bağlamda gradyan artırma, kayıp fonksiyonunun en aza indirildiği ek bir model keşfetmeyi hedefleyen sayısal bir optimizasyon algoritması olarak da değerlendirilebilir. Bu algoritma, tekrarlamalı bir şekilde her adım için kayıp fonksiyonunu en iyi azaltan yeni bir karar ağacı başka bir deyişle "zayıf öğrenici" eklemesi yapar. Regresyonda algoritma modeli bir tahminle başlar, bu genellikle kayıp fonksiyonunu en üst düzeyde azaltan bir karar ağacıdır (regresyon için ortalama hata karesidir). Ardından her adımda yeni bir karar ağacı mevcut artığa takılır ve artığı güncellemek için önceki modele eklenir.

Algoritma, kullanıcının belirlediği maksimum tekrarlama sayısına ulaşılana kadar yinelenir. Bu süreç, sözde aşamalı olarak isimlendirilmektedir. Adımların her biri için, önceki adımda modele eklenen karar ağaçlarında bir değişiklik yapılmaz. Model, karar ağaçlarının kalıntılara uygun bir hale getirilmesiyle, performansının iyi olmadığı bölgelerde iyileştirilir. Her yinelemede eklenen karar ağacı, öğrenme hızı olarak isimlendirilen küçülme parametresi α kullanılarak küçültülürse, gradyan artırma algoritması daha iyi sonuçlar alır. Gradyan artırma bağlamında küçülme prosedürünün ardındaki düşünce, daha yüksek sayıda küçük adımın, daha az sayıda büyük adımdan daha yüksek bir doğruluk sağlamasıdır. Öğrenme parametresi α, 0 ile 1 arasında bir değer alabilir ve ne kadar küçük olursa, model o kadar doğru olur. Bununla birlikte, daha güçlü bir küçülme (daha küçük α) seçmek, yakınsama elde etmek için daha yüksek sayıda yineleme anlamına gelir, çünkü α'nın değeri yineleme sayısı ile ters orantılıdır (Friedman, 2000; Kabak ve ark., 2021). Tablo 2.5 algoritmanın sklearn kütüphanesi tanımlı varsayılan hiper-parametre değerleri yer almaktadır.

Tablo 2.5. Gradyan Artırma Hiper-parametre ve Değerleri

Hiper-Parametre	Değer
Kayıp	Deviance
Öğrenme Oranı	1.0
Doğrulama Oranı	0.1
Maksimum Derinlik	3
Tahmin Edici Sayısı	100

2.5.6. Pasif-Agresif Algoritmalar

Pasif Agresif Algoritmalar Crammer tarafından geliştirilen, sınıflandırma ve regresyon problemleri için kullanılabilen çevrimiçi öğrenme algoritmalarıdır. Bu algoritmalar, doğru bir sınıflandırma sonucu için pasif kalırken yanlış bir sonuç için agresif hale gelir. Temel amaç, kaybı düzelten ve ağırlık vektörü normunda çok az değişikliğe neden olan güncellemeler yapmaktır. Uygulaması kolay ve hızlı olan bu algoritmalar, büyük veri akışlarını sınıflandırmada tercih edilen yöntemlerdendir. Tablo 2.6'da algoritmanın sklearn kütüphanesinde tanımlı varsayılan hiper-parametre değerleri yer almaktadır.

Tablo 2.6. Pasif-Agresif Hiper-parametre ve Değerleri

Hiper-Parametre	Değer
Kayıp	Hinge
Tolerans	0.001
Maksimum İterasyon	1000
İyileştirmesiz Yineleme Sayısı	5
Doğrulama Oranı	0.1

2.5.7. Çok Katmanlı Algılayıcı Yapay Sinir Ağı

İnsan beyninin çalışma sisteminden esinlenerek geliştirilen Yapay Sinir Ağları (YSA), esnek ve etkili bir makine öğrenme metotudur. YSA'nın çokça tercih edilen bir metot olmasının başlıca sebeplerinden biri klasik metotlarla çözüm getirilemeyen problemler için etkin bir alternatif olmasıdır.

Geleneksel ileri beslemeli YSA'lar içerisinde popüler bir model olan MLP basit bir yapıya sahiptir. MLP'ler, tek katmanlı algılayıcıların doğrusal olmayan problemlerin çözümünde başarısız olması üzerine geliştirilmişlerdir. MLP'ler üç katmandan oluşmaktadır, bunlar; bilgi girişinin olduğu girdi katmanı, bir veya daha çok gizli (ara) katman ve bir çıktı katmanı.

MLP'de katmanlar arası ileri ve geri yayılım olarak adlandırılan geçişler bulunmaktadır. İleri yayılım aşamasında, ağın çıktısı ve hata değeri hesaplanır. Geri yayılım aşamasında ise hesaplanan hata değerinin minimize edilmesi için katmanlar arası bağlantı ağırlık değerlerinin bir güncellemesi yapılır. MLP, doğrusal perceptrondaki en küçük kareler algoritmasının bir genelleştirilmesi olan geri yayılım (backpropagation) öğrenme algoritmasını kullanır (Arı ve ark., 2017). Tablo 2.7'de algoritmanın sklearn kütüphanesi tanımlı varsayılan hiper-parametre değerleri yer almaktadır.

Tablo 2.7 Çok Katmanlı Algılayıcı Hiper-parametre ve Değerleri

Hiper-Parametre	Değer
Çözücü	Adam
Gizli Katman Boyutu	100
Aktivasyon Fonksiyonu	Relu
Öğrenme Oranı	0.001
Maksimum İterasyon	200
Tolerans	1e-4

2.6.Öznitelik Seçimi Yöntemleri

Makine öğrenmesi ve veri madenciliği teknikleri büyük ölçekli bir veri içerisinden bilgi işleme ve çıkarma yapmak amacıyla yaygın bir şekilde kullanılmaktadır. Bu yöntemlerin alakasız ve gereksiz öznitelikler içeren büyük miktarda veri üzerinde uygulanıyor olması bilginin doğruluğunu etkilemesinin yanı sıra zaman açısından da maliyetlidir. STS'ler genellikle uzun işlem süresine ve düşük saldırı tespit oranına sahip, çeşitli gereksiz ve ilgisiz öznitelikler içeren büyük miktarda veriler üzerine çalışır. Bu sebeple makine öğrenimi tabanlı STS'lerde, öznitelik seçimi konusu sınıflandırma doğruluğunun artması, sınıflandırma süresinin azalması, desteklenmeyen bellek gereksiniminin önlenmesi adına önemlidir. Öznitelik seçimi, beklenen çıktı değişkenlerine ya da tahmine fayda sağlayabilecek önemli olan özniteliklerin belirlenmesidir. Bu yöntemlerinin sağladığı avantajlar ve dezavantajlar aşağıdaki gibidir:

Avantajlar (Siddique ve ark., 2017; Haq ve ark., 2015; Ahmad ve ark., 2019)

- Modelin hesaplama süresi azalır.
- Veri kümesinde bulunan gereksiz ve ilgisiz veriler çıkarılır.
- Gereksiz ve ilgisiz verilerin çıkarılması ile hesaplama karmaşıklığı azalır.
- Modelin sınıflandırma performansının iyileşmesine katkı sağlanır.
- Aşırı uyum gösterme (overfitting) durumunun engellenmesine katkı sağlanır.
- Boyutsal Bela (Curse of dimentionality) etkileri azalır.

Dezavantajlar (Bilgin, 2018)

- Ek bir işlem maaliyeti oluşur.
- Veri kümesindeki en önemli özelliklerin seçilmesi zaman alır.

Öznitelik seçiminde kullanılan yöntemler genel olarak üç gruba ayrılır, bunlar; en iyi bölen ölçütünü bulmaya dayalı olan gömülü yöntemler, öznitelikler üzerinde arama işlemleri gerçekleştiren sarmal yöntemler ve istatistiksel bilgiye dayalı olan filtreleme yöntemleridir (Saeys ve ark., 2007).

2.6.1. Filtreleme Yöntemleri

İstatistik alanında yaygın bir şekilde kullanılmakta olan filtreleme yöntemlerinde, öznitelik seçimi tercih edilen makine öğrenmesi algoritmasından tamamen bağımsız bir şekilde gerçekleştirilir. Bu yöntemler modelin eğitiminde kullanılacak eğitim veri kümesine uygulanmaktadır. Filtreleme yöntemleri, öznitelikleri değerlendirmede uyguladıkları yönteme göre ikiye ayrılır. Bunlar, bireysel öznitelik ölçütleri ve grup öznitelik ölçütleri'dir. Bireysel ölçütler, özniteliklerin uygunluk durumlarını bireysel olarak değerlendirir.

Son öznitelik alt kümesi oluşturmak için tüm öznitelikler belirli bir kritere göre sıralanır. Ardından, belirli bir eşik değerini aşan öznitelikler seçilir. Grup ölçütlerinde ise, öznitelikler bireysel olarak değerlendirilmez. Bunun yerine aday öznitelik alt kümelerinin bir değerlendirmesi yapılır. Bu şekilde, öznitelikler arası ilişkilerinde hesaba katılması hedeflenir (Diao, 2014; Onan ve ark., 2017).

2.6.1.1.Ki-Kare Testi

Gözlenen ve beklenen frekanslar arasındaki farkın anlamlı olup olmadığı temeline dayanmaktadır. Bu yöntemde, öznitelikler (X) ile Y arasında bir ilişki olup olmadığı test edilir. Test sonucuna bağlı olarak, Y ile bir ilişkisinin bulunmadığı saptanan öznitelikler veri kümesinden kaldırılır. Ki-Kare testi için kullanılan denklemler şöyledir (Inc, 2007; Ünver ve ark., 2006; Budak, 2018):

$$X^2 = \sum_{i=1}^{i}\sum_{j=1}^{j}(N_{ij} - N'_{ij})^2 \,/\, N'_{ij} \tag{2.11}$$

$$N'_{ij} = N_{i.}N_{.j}/\mathrm{N} \tag{2.12}$$

$$d = (I-1)(J-1) \tag{2.13}$$

Denklemlerdeki N_{ij} Y'nin i. ve X'in j. düzeyindeki gözlenen birim sayısı, N'_{ij} iki özellik bağımsız iken Y'nin i. ve X'in j. düzeyindeki beklenen birim sayısı, d test istatistiği için kullanılacak Ki-Kare dağılımının serbestlik derecesini göstermektedir.

Belirli bir sayıda öznitelik seçimi yapılmak istendiğinde, öznitelikler hesaplanan X^2 değerine göre büyükten küçüğe doğru sıralanır. Sonrasında, en üstten başlanarak istenen sayıda öznitelik seçilir (Budak, 2018).

2.6.1.2.Korelasyon Tabanlı Öznitelik Seçimi

CFS, öznitelik alt kümelerinin bilgi değerlerini ölçen fonksiyona ek olarak bir arama algoritması da kullanmaktadır. CFS'nin öznitelik altkümelerinin değerlerini ölçerken kullandığı yaklaşımda, her bir özniteliğin sınıf etiketini tahmin etmekteki başarısı ve aralarındaki iç korelasyon değerleri dikkate alınmaktadır. Bu yaklaşım, iyi öznitelik altkümelerinin ilgili sınıf ile yüksek birbirleri ile düşük korelasyona sahip özniteliklerden oluştuğu hipotezine dayanmaktadır (Hall, 1999; Budak, 2018).

$$M_s = \frac{kr'_{ci}}{\sqrt{k + k(k-1)\, r'_{ii'}}} \qquad (2.14)$$

Denklem 2.14'teki k altkümedeki özellik sayısı, kr'_{ci} Y ile öznitelik arasındaki ortalama korelasyonu, $r'_{ii'}$ özniteliklerin birbirleri arasındaki ortalama iç korelasyonunu göstermektedir.

2.6.2. Sarmal Yöntemler

Veri kümesindeki öznitelik alt küme kombinasyonlarının hangisinin makine öğenmesi modelinde daha iyi bir performans sağlayacağını bulmak amacıyla tüm kombinasyonları tarayarak çalışan bir yöntemdir. En iyi öznitelik alt kümesinin belirlenmesinde filtreleme yöntemlerine nispeten daha başarılıdır. Fakat tüm kombinasyonların denenmesi sebebiyle yüksek hesaplama zamanı maliyetleri bulunur (Guyon ve ark., 2003, Budak, 2018). Kombinasyonların performansları değerlendirilirken çeşitli sınıflandırıcı ve kriterlerden faydalanılabilir. Seçilecek öznitelik sayısı, makine öğrenmesi probleminin türüne göre değerlendirme metrikleri bunlara birer örnektir.

2.6.2.1.Yinelemeli Öznitelik Eliminasyonu

RFE, sarmalayıcı öznitelik seçiminde yaygın bir yöntemdir. RFE yöntemi, yinelemeli yöntem temelinde düşük katkı puanlı öznitelikleri sürekli olarak ortadan kaldırır ve ardından en düşük puana sahip n özniteliği silmek için her bir döngüde her özelliği sıralar.

RFE algoritmasına ait sözde kod aşağıdaki gibidir:

Girdi:

Eğitim kümesi T_s

p öznitelik kümesi $F_e = \{f_{1\ldots,}\, f_n\}$

Sıralama yöntemi $M_e\ (T_s,\ F_e\ )$

Çıktı:

Son sıralama R

Adımlar:

1. $\{1{:}\, n\}$ içinde i için tekrarla
2. $M_e\ (T_s,\ F_e\ )$ kullanarak F_e kümesini sırala
3. $f \leftarrow F_e$'de son sıradaki özellik
4. $R(n-i+1) \leftarrow f$
5. $F_e \leftarrow F_e - f$

2.6.3. Gömülü Yöntemler

Yapısında hem öznitelik seçimi hem de sınıflandırma algoritması içerir. Bu sebeple öznitelik seçme ve sınıflandırma işlemlerini eşzamanlı olarak yapabilir. Sarmal yöntemlerdeki gibi gömülü yöntemlerin de hesaplama maliyeti filtreleme yöntemlerine kıyasla daha yüksektir. (Guyon ve ark., 2003, Budak, 2018).

2.7.Kolektif Öğrenme Yöntemleri

Bu yöntemler gözetimli öğrenme çatısında bulunan sınıflandırıcılar için Nilsson tarafından ileri sürülmüştür (Yang, 2017; Nilsson, 1965).

Çoğunlukla tek bir sınıflandırıcı kullanılarak oluşturulmuş modellerin daha üstünde bir performans gösteren bu yöntemler, makine öğrenmesinde kullanılan etkili yaklaşımlar arasındadır (Subasi, 2020; Bilgin, 2018). Kolektif metotlarda, temel sınıflandırıcı olarak isimlendirilen bireysel sınıflandırıcıların sahip oldukları yetenekler biraraya getirilir ve sınıflandırma doğruluğunun geliştirilmesi hedeflenir.

Sınıflandırma problemlerinin yanı sıra regresyon problemlerine de uygulanabilirler (Sugiyama, 2016a). Torbalama (Bagging), Güçlendirme (Boosting), Yığınlama (Stacking), ve Oylama (Voting) en tanınmış kolektif öğrenme metotları arasındadır.

2.7.1.1.Torbalama (Bagging)

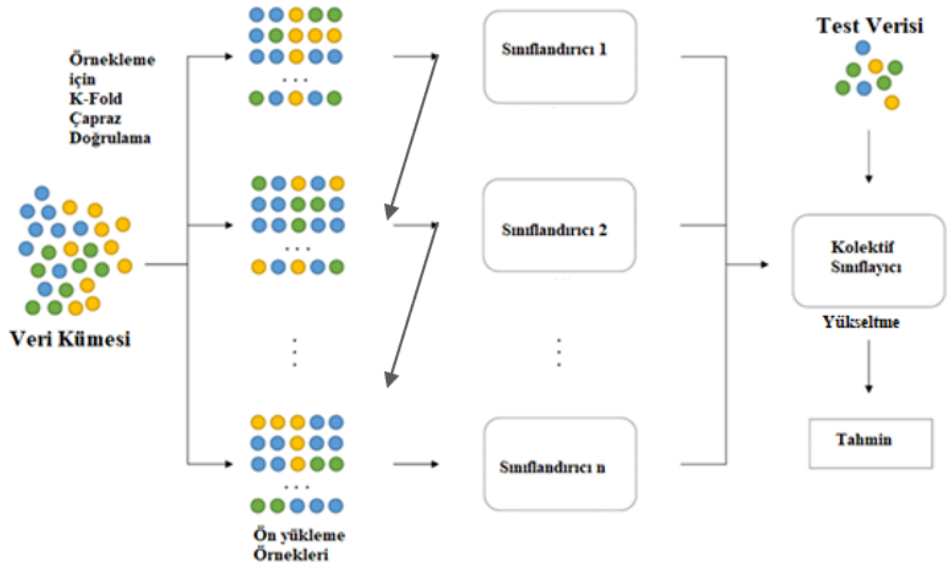
Bagging, “bootstrap aggregation” ifadesinin kısaltılmış versiyonudur. Bu teknik, sınıflandırma ve regresyon algoritmalarının tahmin doğruluğunu arttırmak amaçlı ileri sürülmüştür (Breiman, 1996; Sutton, 2005; Bilgin, 2018). Yeniden örneklem yöntemi ile edinilen sınıflandırıcıların birleştirilmesiyle oluşturulan bu algoritmada veri kümesinden, içerisine koyulacak şekilde tekrar tekrar örnekler alınarak yeni sınıflandırıcılar elde edilir. Elde edilen tüm sınıflandırıcılar ile kolektif bir model geliştirilir. Sınıflandırıcıların her birine bir oylama yapılır. Bireysel sınıflandırıcılar çoğunluk oyu için bir araya getirilmeden evvel küçük veri kümeleri ile eğitilir (Breiman, 2001). Torbalama tekniğinin farklı bir oluşumu olan Rastgele Orman torbalama kullanılarak eğitilmiş karar ağaçları topluluğudur (Zhang ve ark., 2012). Şekil 2.7 Torbalama kolektif öğrenme algoritması işlem adımlarını göstermektedir.



Şekil 2.7. Torbalama (Bagging) Algoritması İşlem Adımları

2.7.1.2.Yükseltme (Boosting)

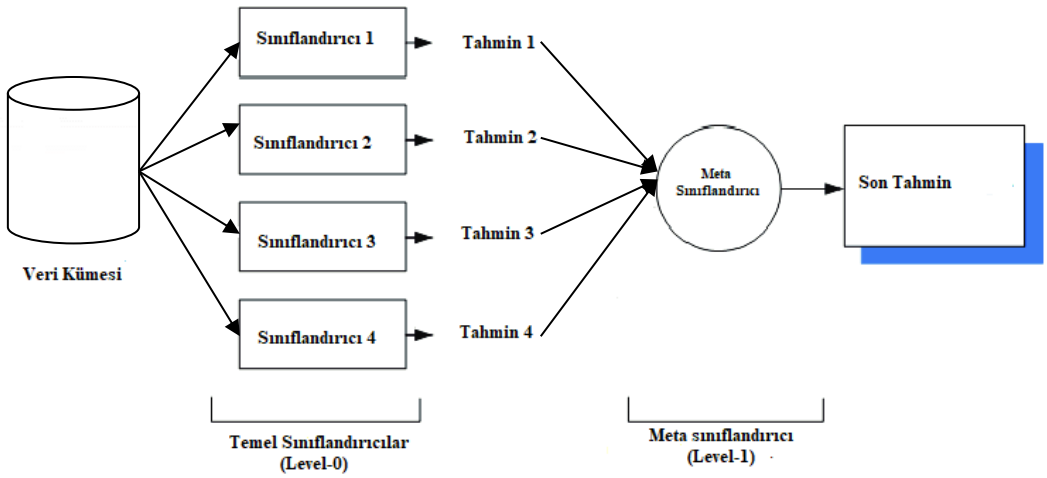
Zayıf olan sınıflandırıcıların bir araya getirilerek güçlü bir sınıflandırıcının oluşturulmasının hedeflendiği bir kolektif öğrenme yaklaşımıdır. Temel sınıflandırıcılar sıra ile eğitilir. Tahmin işlemleri için sonuçlarda sıra ile alınarak birleştirilir. Burada, önce gelen sınıflandırıcının hatalarına dikkat edilir ve bu hataların düzeltilmesine odaklanır (Zhou, 2012). Yükseltme, ikili (binary) sınıflandırmalar için tasarlanmış bir algoritmadır. Bu algoritmadaki prosedür, sıralı olan her bir sınıflandırıcının daha önce doğru bir şekilde sınıflandırılmamış örneklere odaklanmasıdır. Örnek olarak, herhangi bir D dağılımı üzerinde çalışan zayıf sınıflandırıcı için doğruluk fonksiyonunun f olduğunu ve eğitim setinin X uzayından seçildiğini varsayalım. Sonrasında, X uzayıdan tek seferde X_1, $X_2 ve$ X_3 olarak üç zayıf sınıflandırıcı kümesinin oluştuğunu düşünelim. Buradaki X_1 sınıflandırıcısı, h_1 eğitim kümesinin rastgele bir alt kümesinde torbalama algoritmasına benzer bir şekilde eğitilir. X_2 sınıflandırıcısı, h_2 orijinal veri kümesinin farklı bir altkümesi ile eğitilir ve yarısı h_1 tarafından doğru, diğer yarısı ise yanlış olarak sınıflandırılır. X_3 sınıflandırıcısı ise $h_1$ve h_2 tarafından eğitilmeyen veriler ile eğitilir. Sonrasında bu üç sınıflandırıcı çoğunluk oyu (majority voting) ile bir araya getirilir (Zhou, 2012). Şekil 2.8 Yükseltme kolektif öğrenme algoritması işlem adımlarını göstermektedir.



Şekil 2.8. Yükseltme (Boosting) Algoritması İşlem Adımları

2.7.1.3. Yığınlama (Stacking)

Yığınlama, torbalama ve yükseltme kolektif öğrenme yöntemlerinden farklı olarak, temel modellerin sonuçlarını birleştirmek için ayrı bir model bir meta öğrenici kullanır. Yüksek bir doğruluk performansı elde edilmesi amacıyla farklı türden sınıflandırıcıların tahminleri meta sınıflandırıcıya girdi olarak verilir. Bu yaklaşımda, eğitim veri kümesi kullanılarak oluşturulan farklı türde sınıflandırıcılardan tahminler alındıktan sonra, bu tahminler meta sınıflandırıcısında birleştirilir ve kolektif model oluşturularak nihai sonuçlar edinilir (Wolpert, 1992). Şekil 2.9 Yığınlama kolektif öğrenme algoritması işlem adımlarını göstermektedir.

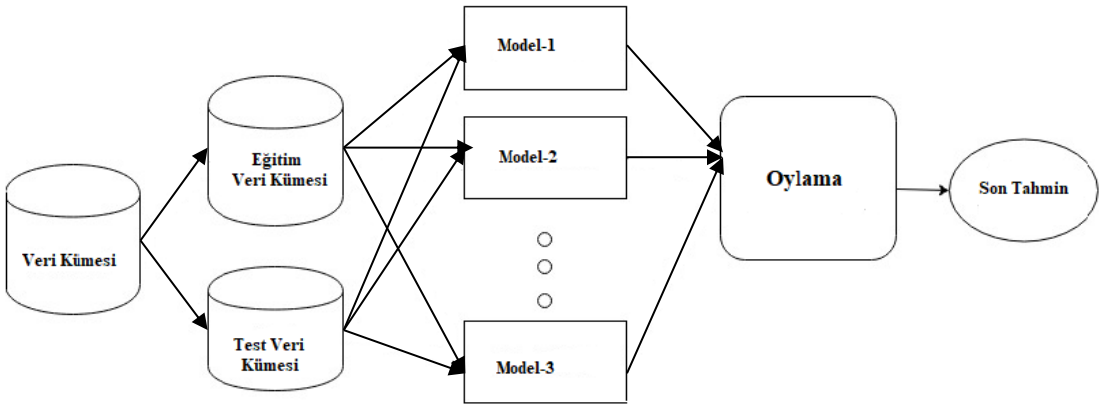


Şekil 2.9. Yığınlama (Stacking) Algoritması İşlem Adımları

2.7.1.4.Oylama (Voting)

Oylama, birçok sınıflandırıcıdan gelmiş olan tahminleri birleştirmenin en kolay metotlarından biridir. Bu metotta, sınıflandırıcılardan her birinin karakteristik özelliklerinden faydalanmak amacıyla paralel bir şekilde eğitilmiş olan farklı tür sınıflandırıcı grupları birleştirilir (Akman, 2011). Farklı tür sınıflandırıcıların sınıflandırma işlemi yapması doğruluk oranının yükselmesinde olumlu yönde bir etki oluşturmaktadır. Bu sebeple, sınıflandırıcı çeşitliliğinin sağlanması doğruluk oranını artıracaktır (Polikar, 2012). Torbalama ve yükseltme yaklaşımlarında benzer sınıflandırıcılardan çok sayıda sınıflandırıcı, veri kümesinin farklı alt kümeleri ile eğitilmektedir. Burada ise aynı veri kümesi üzerinde eğitilen farklı türden sınıflandırıcıların sınıflandırma yapması durumu söz konusudur (Rokach, 2010).

Oylama yaklaşımında en yaygın görülen birleştirme kuralı, Çoğunluk Oyu'dur (Polikar, 2012). Sürekli sonuçlar üreten sınıflandırıcılar için ortalama ve ağırlıklı ortalama alma kuralları da kullanılmaktadır (Zhou, 2012). Şekil 2.10 Oylama kolektif öğrenme algoritması işlem adımlarını göstermektedir.

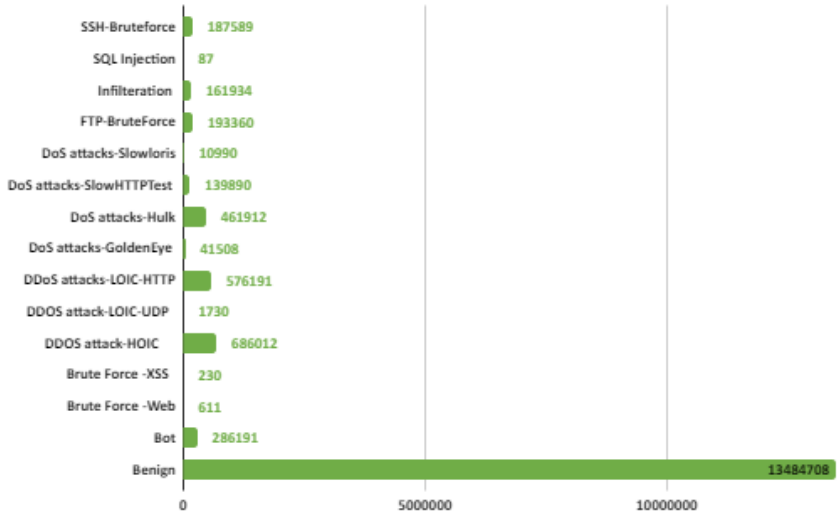


Şekil 2.10. Oylama (Voting) Algoritması İşlem Adımları

2.8.CSE-CIC-IDS2018 Veri Kümesi

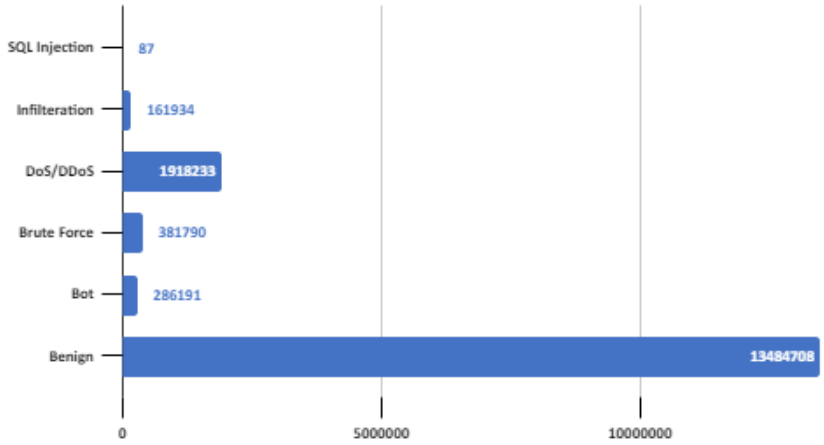
Anomali tespiti, yeni saldırıları tespit etme noktasındaki potansiyeli sebebiyle birçok araştırmacının ana odak noktası olmuştur. Fakat yapılan çalışmaların çoğunda geçerli ve güncel bir veri kümesi bulma konusunda zorluk yaşanması sebebiyle bilinen eksikleri olan veri kümeleri kullanılmıştır. UMASS, ISCX, DARPA, DEFCON, KYOTO, CDX, TWENTE, KDD'99 (Sharafaldin ve ark., 2018) bu veri kümelerinden bazılarıdır.

Kanada Siber Güvenlik Enstitüsü ve İletişim Güvenliği Kuruluşunun ortak bir çalışması sonucu üretilen ve herkesin kullanımına açık olan CSE-CIC-IDS2018 (CyberSecurity, 2018) veri kümesi, bilinen eksikleri bulunan önceki veri kümelerinde yer alan eksiklikler göz önünde bulundurularak oluşturulmuştur. CICIDS2017 veri kümesinin güncel versiyonu olan CSE-CIC-IDS2018, gerçek ağ trafiğine sahip, saldırı çeşitliliği oldukça fazla olan yeni STS veri kümelerinden biridir. Bu veri kümesi içerisinde, 10 gün boyunca yapılan farklı saldırı tiplerini içeren saldırı verileri yer almaktadır. Veri kümesi, her makine için ağ trafiği (PCAPs) ve sistem günlüklerini (Windows ve Ubuntu sistem günlükleri) içeren ham veriler kaydedilerek günlük olarak düzenlenmiştir. Ham verilerden öznitelik çıkarma işleminde ağ trafiği akış üreticisi ve analizörü olan CICFlowMeter-V3 uygulaması kullanılmış ve elde edilen 80 ağ trafik özniteliği CSV dosyası olarak kaydedilmiştir. Siber güvenlik konuları üzerine çalışma yapmak isteyen araştırmacılar, bu veri kümesinin CSV ve PCAP formatlarına kolaylıkla ulaşabilmektedir (CyberSecurity, 2018). Veri kümesi Amazon Web Servisleri üzerinden lokal bilgisayara alınıp incelendiğinde, içerisindeki etiketlemeler ve bu etiketlere ait örnek sayısının Şekil 2.11'de gösterildiği şekilde olduğu görülecektir.



Şekil 2.11. Veri Kümesindeki Etiket Sayısı

Veri kümesinde Botnet, DoS (Hulk, SlowHTTPTest, GoldenEye, Slowloris), Ağa içeriden sızma (Infilteration), BruteForce (Web, XSS, FTP, SSH), Web saldırıları (SQL Injection) ve DDoS (HOIC, LOIC-UDP, LOIC-HTTP) olmak üzere 6 tipte 14 farklı saldırı türü (2,748,235 saldırı) yer almaktadır. Şekil 2.12 saldırı tiplerinin veri kümesindeki dağılımlarını göstermektedir.



Şekil 2.12. Etiket Kategori Dağılımı

Tablo 2.8 CSE-CIC-IDS2018 veri kümesinde yer alan CICFlowMeter-V3 uygulaması kullanılarak elde edilen 80 ağ trafik özniteliğini ve kısa açıklamalarını içermektedir.

Tablo 2.8. CICFlowMeter-V3 ile Elde Edilen Öznitelikler

Öznitelik	**Açıklama**
ack_cnt	ACK'li paket sayısı
atv_avg	Bir akışın boşta kalmadan önce aktif olduğu ortalama süre
atv_max	Bir akış boşta olmadan önce aktif olan standart sapma süresi
atv_min	Boşta kalmadan önce bir akışın etkin olduğu minimum süre
atv_std	Bir akışın kullanılmadan önce etkin olduğu maksimum süre
bw_blk_rate_avg	Geri yönde ortalama yığın oranı
bw_byt_blk_avg	Geri yönde ortalama bayt yığın oranı
bw_hdr_len	Geri yönde başlıklar için kullanılan toplam bayt sayısı
bw_iat_avg	Geri yönde gönderilen iki paket arasındaki ortalama süre
bw_iat_max	Geri yönde gönderilen iki paket arasındaki maksimum süre
bw_iat_min	Geri yönde gönderilen iki paket arasındaki minimum süre
bw_iat_std	Geri yönde gönderilen iki paket arasındaki standart sapma süresi
bw_iat_tot	Geri yönde gönderilen iki paket arasındaki toplam süre
bw_pkt_blk_avg	Geri yönde ortalama paket yığın oranı
Bw_pkt_l_avg	Paketin geriye doğru ortalama boyutu
Bw_pkt_l_max	Geri yönde maksimum paket boyutu
Bw_pkt_l_min	Geri yönde minimum paket boyutu
Bw_pkt_l_std	Paketin geriye doğru standart sapma boyutu

bw_pkt_s	Saniyedeki geriye doğru paket sayısı
bw_psh_flag	PSH bayrağının geriye doğru giden paketlerde ayarlanma sayısı (UDP için 0)
bw_seg_avg	Geri yönde gözlemlenen ortalama boyut
bw_urg_flag	URG bayrağının geriye doğru giden paketlerde ayarlanma sayısı (UDP için 0)
bw_win_byt	İlk pencerede geriye doğru gönderilen bayt sayısı
cwe_cnt	CWE'li paket sayısı
down_up_ratio	İndirme ve yükleme oranı
ece_cnt	ECE'li paket sayısı
fin_cnt	FIN'li paket sayısı
fl_byt_s	Saniyede aktarılan paket sayısına olan akış bayt oranı
fl_dur	Akış Süresi
fl_iat_avg	İki akış arasındaki ortalama süre
fl_iat_max	İki akış arasındaki maksimum süre
fl_iat_min	İki akış arasındaki minimum süre
fl_iat_std	İki akışın standart sapma süresi
fl_pkt_s	Saniyede aktarılan paket sayısına olan akış paket hızı
Fw_act_pkt	İleri yönde en az 1 bayt TCP veri yükü olan paket sayısı
fw_blk_rate_avg	İleri yönde ortalama yığın oranı
fw_byt_blk_avg	İleri yönde ortalama bayt yığın oranı
fw_hdr_len	İleri yönde başlıklar için kullanılan toplam bayt sayısı
fw_iat_avg	İleri yönde gönderilen iki paket arasında geçen ortalama süre
fw_iat_max	İleri yönde gönderilen iki paket arasındaki maksimum süre
fw_iat_min	İleri yönde gönderilen iki paket arasındaki minimum süre
fw_iat_std	İleri yönde gönderilen iki paket arasındaki standart sapma süresi
fw_iat_tot	İleri yönde gönderilen iki paket arasında geçen toplam süre
fw_pkt_blk_avg	İleri yönde ortalama paket toplu hızı
fw_pkt_l_avg	Paketin ileri yönde ortalama boyutu
fw_pkt_l_max	İleri yönde maksimum paket boyutu
fw_pkt_l_min	İleri yönde minimum paket boyutu
fw_pkt_l_std	Paketin ileri yönde standart sapma boyutu
fw_pkt_s	Saniyede iletilen paket sayısı
fw_psh_flag	İleri yönde seyahat eden paketlerde PSH bayrağının ayarlanma

	sayısı (UDP için 0)
fw_seg_avg	İleri yönde gözlenen ortalama boyut
fw_seg_min	İleri yönde gözlemlenen min. segment boyutu
fw_urg_flag	URG bayrağının ileri yönde giden paketlerde ayarlanma sayısı (UDP için 0)
fw_win_byt	İlk pencerede ileri yönde gönderilen bayt sayısı
idl_avg	Bir akışın aktif hale gelmeden önce boşta kaldığı ortalama süre
idl_max	Bir akışın etkin olmadan önce kullanılmadığı maksimum süre
idl_min	Bir akışın etkin olmadan önce minimum boşta kaldığı süre
idl_std	Bir akış aktif olmadan önce boşta kalan standart sapma süresi
pkt_len_avg	Ortalama akış uzunluğu
pkt_len_max	Maksimum akış uzunluğu
pkt_len_min	Minimum akış uzunluğu
pkt_len_std	Bir akışın standart sapma uzunluğu
pkt_len_va	Minimum paket varış zamanı
pkt_size_avg	Ortalama paket boyutu
pst_cnt	PUSH'lu paket sayısı
rst_cnt	RST'li paket sayısı
subfl_bw_byt	Geri yönde bir alt akıştaki ortalama bayt sayısı
subfl_bw_pkt	Geri yönde bir alt akıştaki ortalama paket sayısı
subfl_fw_byt	İleri yönde bir alt akıştaki ortalama bayt sayısı
subfl_fw_pk	İleri yönde bir alt akıştaki ortalama paket sayısı
syn_cnt	SYN'li paket sayısı
tot_bw_pk	Geri yönde toplam paket
tot_fw_pk	İleri yönde toplam paket
tot_l_fw_pkt	Paketin ileri yönde toplam boyutu
urg_cnt	URG'li paket sayısı

2.9.Performans Değerlendirme Metrikleri

Makine öğrenmesi yöntemlerinde kullanılan performans değerlendirme metrikleri karmaşıklık matrisinden yola çıkılarak hesaplanmaktadır. Karmaşıklık matrisi, bir sınıflandırma modelinin veriler üzerindeki gerçek sonuçlara kıyasla yaptığı doğru ve yanlış tahmin sayısını gösterir.

Bir algoritmanın performansının görüntülenmesini ve sınıflar arasındaki karmaşıklığın kolayca tanımlanmasını sağlar. Şekil 2.13'teki örnek karmaşıklık matrisinde yer alan, DN doğru negatifi, YN yanlış negatifi, DP doğru pozitifi ve YP yanlış pozitifi temsil etmektedir.

- Doğru Negatif (DN): Gözlem negatif iken tahmin de negatiftir.
- Yanlış Negatif (YN): Gözlem pozitif iken tahmin negatiftir.
- Doğru Pozitif (DP): Gözlem ve tahminin her ikisi de pozitiftir.
- Yanlış Pozitif (YP): Gözlem negatiftir, ancak tahmin pozitiftir.

		Gerçek Kategori	
Tahmin Edilen Kategori		Kategori 1	Kategori 2
	Kategori 1	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
	Kategori 2	Yanlış Negatif (YN)	Doğru Negatif (DN)

Şekil 2.13. Karmaşıklık Matrisi

Duyarlılık (Recall); doğru bir şekilde sınıflandırması yapılan pozitif örnek sayısının, toplam pozitif örnek sayısına oranlanması ile elde edilir.

$$\text{Duyarlılık} = \frac{\text{DP}}{\text{DP} + \text{YN}}$$

Doğruluk (Accuracy); doğru bir şekilde sınıflandırması yapılan örneklerin toplam örnek sayısına oranlanması ile elde edilir.

$$\text{Doğruluk} = \frac{\text{DP} + \text{DF}}{\text{DP} + \text{DF} + \text{YP} + \text{YN}}$$

Kesinlik (Precision); doğru bir şekilde sınıflandırması yapılan pozitif örnek sayısının, toplam pozitif tahminlenmiş örnek sayısına oranlanması ile elde edilir.

$$\text{Kesinlik} = \frac{\text{DP}}{\text{DP} + \text{YP}}$$

Duyarlılık ve kesinlik bireysel olarak tam manasıyla anlamlı bir karşılaştırma sonucu elde etmemiz için kâfi değildir. Duyarlılık Tip 1 Hata (Yanlış Pozitif) değerini, kesinlik ise Tip 2 Hata (Yanlış Negatif) değerini göz önünde bulundurmaz. Bu nedenle her iki kriterin birlikte değerlendirilmesi ile oluşan, tüm hata maliyetlerini içeren F1-Skor tanımlanmıştır. Bu metrik, duyarlılık ve kesinlik kriterlerinin harmonik ortalamasından elde edilmektedir.

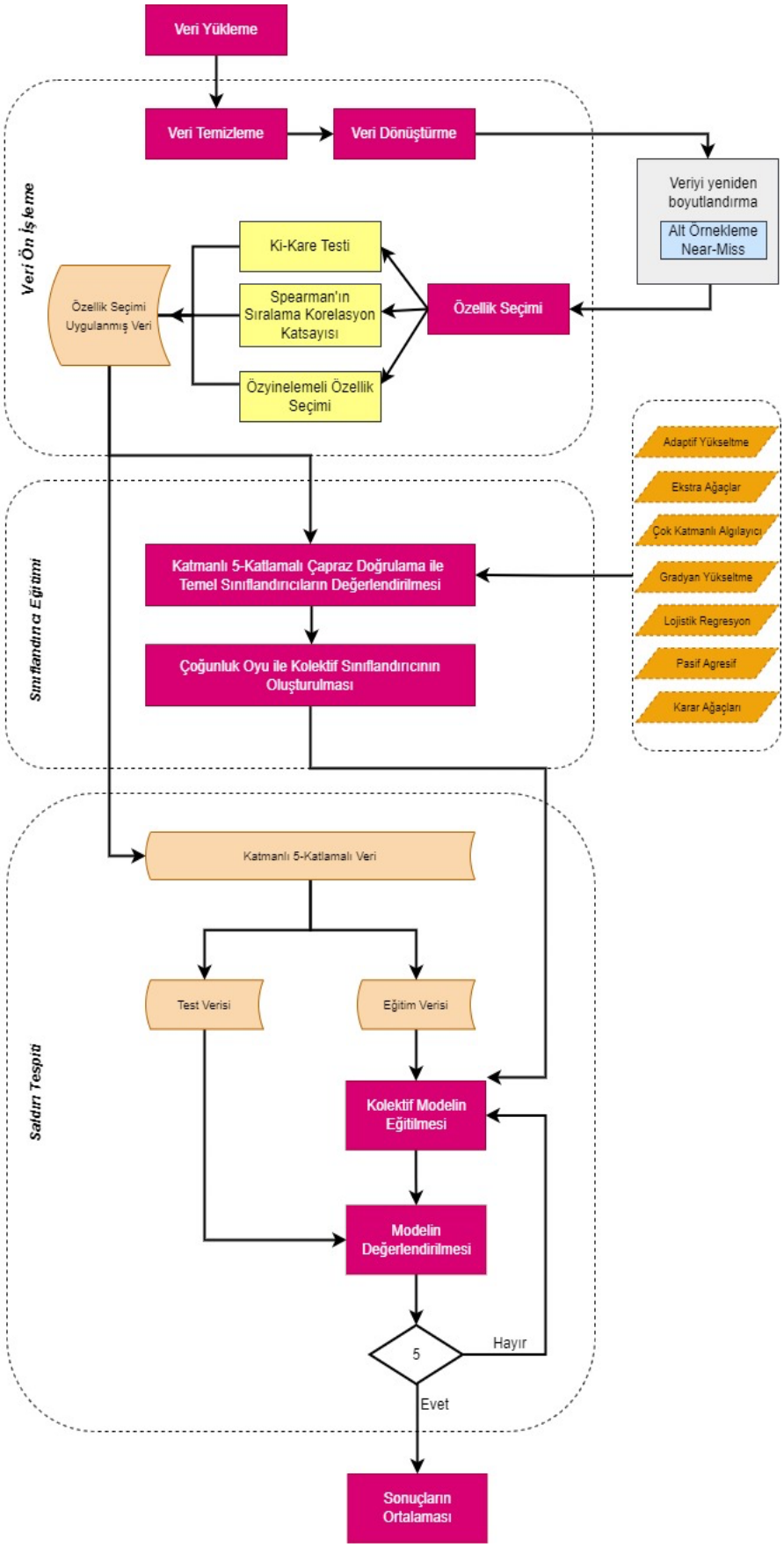
$$\mathrm{F1-Skor} = \frac{2 * \mathrm{Duyarlılık} * \mathrm{Kesinlik}}{\mathrm{Duyarlılık} + \mathrm{Kesinlik}}$$

2.10. Önerilen Sistem

STS'lerin gelişen teknoloji ile artan ihtiyaçlara yönelik istekleri karşılayabilir durumda olması gerekmektedir (Thomas ve ark., 2018). Pek çok araştırmada yer alan ve hala gelişmekte olan Makine Öğrenmesi bu anlamda başvurulan yöntemlerdendir (Athmaja ve ark., 2017). STS'lerde bu yöntemlerin kullanılmasındaki amaç sistemin, önceden bilmediği bir veriyi yüksek doğruluk oranı ile hızlı bir şekilde sınıflandırabiliyor olmasıdır (Sahingöz ve ark., 2019). STS'ler genellikle, düşük tespit oranı ve uzun işlem süresi ile neticelenen çeşitli gereksiz ve alakasız öznitelikler içeren büyük ölçekli veriler üzerinde işlem yapar (Amrita, 2013). Bu durum da öznitelik seçimini, -makine öğrenimi tabanlı STS'lerde performans iyileştirmeleri sağlamak adına- önemli bir konu haline getirmektedir. Öznitelik seçimi, sınıflandırma eğitim süresini azaltmak ve doğruluk tespitini geliştirmek amacıyla veri kümesinden en önemli özniteliklerinin seçilmesidir (Zhou ve ark., 2019).

Bu çalışmada, yüksek doğruluk ve verime sahip etkili bir STS oluşturmak amacıyla literatürde önerilen çeşitli öznitelik seçimi ve kolektif öğrenme tekniklerine odaklanılmıştır. Şekil 2.14 STS için geliştirilen makine öğrenmesi modeli yaklaşımını göstermektedir. Önerilen model veri ön işleme, sınıflandırıcı eğitimi ve saldırı tespiti olmak üzere üç aşamadan oluşmaktadır. Veri ön işlemede, CSE-CIC-IDS2018 veri kümesi üzerinde ilk olarak şu işlemler yapılmıştır: Tutarsızlıkların saptanması/temizlenmesi, hatalı verinin düzeltilmesi, eksik öznitelik değerlerinin tamamlanması, ölçeklendirme ve normalizasyon. Veri kümesindeki örneklem sayısının oldukça fazla olması ve sadelik gereksinimi sebebiyle, öznitelik seçimi yapılmadan önce veri kümesi saldırı dağılımları aynı kalacak şekilde %50 oranında küçültülmüştür.

Ardından, öznitelik seçim tekniklerinden faydalanılarak veri kümesindeki en önemliöznitelikler seçilmiştir. Sınıflandırıcı eğitimi aşamasında, öznitelik seçim yöntemlerinin her biri için 7 temel sınıflandırıcının karşılaştırılması yapılarak verim artışı sağlayan öznitelik seçim yöntemi için en yüksek performans gösteren üç temel sınıflandırıcı çoğunluk oyu (majority voting) ile birleştirilerek kolektif bir model oluşturulmuştur. Bu aşamada Karar Ağacı, Gradyan Artırma, Adaptif Yükseltme, Lojistik Regresyon, Pasif-Agresif, Ekstra Ağaçlar ve Çok Katmanlı Algılayıcı sınıflandırıcılarının performans değerlendirmeleri için Katmanlı 5-Katlamalı Çapraz Doğrulama (Stratified 5-Fold Cross Validation) tekniğinden faydalanılmıştır. Elde edilen sonuçlar doğruluk (accuracy), duyarlılık (recall), kesinlik (precision), F1-Skoru (F1-Score) ve hesaplama zamanı kriterlerine göre değerlendirilmiştir. Son aşama olan saldırı tespitinde ise çapraz doğrulama tekniği ile başarımı incelenen kolektif modelin bir değerlendirmesi yapılmıştır.



Şekil 2.14. Geliştirilen Makine Öğrenmesi Modeli

3. BULGULAR VE TARTIŞMA

STS için önerilen modeli gerçeklenmesinde Python dili kullanılmış, makine öğrenmesi ve veri işleme için Numpy (Numpy, 2022), Pandas (Pandas, 2022) ve Sklearn (Scikit-learn, 2022) kütüphanelerinden faydalanılmıştır. Değerlendirmelerde 64-bit Microsoft Windows işletim sistemli, Intel Core i7-7700K 4.2 GHz, 32 RAM teknik özelliklerine sahip bir bilgisayar kullanılmıştır.

3.1.Veri Önişleme

2.8 bölümünde detayları verilen CSE-CIC-IDS2018 veri kümesi kullanılarak uygulama yapılmadan önce, 10 farklı .csv dosyasında bulunan veriler üzerinde -makine öğrenme modelleri ile uyumlu ve verimli çalışılabilmesi adına- bazı ön işlemler gerçekleştirilmiştir. Bu ön işlemler aşağıdaki gibidir;

1) CICFlowMeter-V3 kullanılarak veri kümesinden öznitelikler çıkarılmıştır. Sonrasında Source IP, Flow ID, Destination IP, Destination Port ve Source Port öznitelikleri silinmiştir.
2) Timestamp özniteliği saldırının gerçekleştiği zaman bilgisini sunmaktadır. Bu öznitelik saldırının türü ya da durumu ile ilgili herhangi bir bilgi sağlamadığından veri kümesinden silinmiştir.
3) NaN değerinin yerine hataların önüne geçmek amacıyla 0 değeri koyulmuştur.
4) Flow Bytes/s ve Flow Pkts/s sütunlarındaki sonsuz değerlerinin yerine, bulunulan sutündaki maksimum değerin bir fazlası koyulmuştur.
5) InitBwd Win Bytes ve InitFwd Win Byts sütunlarında -1 değeri bulunabilmektedir. Bu yüzden, InitBwd Win BytsNeg ve InitFwd Win BytsNeg isimli sütunlar oluşturulmuş ve bu sütunlara orijinal öznitelikler dikkate alınarak; -1 değerini içeren veri ile karşılaştığında 1, aksi durumda 0 değerleri atanmıştır.
6) 'Label' verinin saldırı tipi bilgisini verir. Bu sütundaki değerler Tablo 3.1'de gösterilen şekilde değiştirilmiştir.

Tablo 3.1. Etiketler ve Değerleri

Etiket	Değer
Benign	0
Bot	1
Brute Force -XSS	2
Brute Force -Web	2
FTP-BruteForce	2
SSH-Bruteforce	2
DDOS attack-GoldenEye	3
DDOS attack-LOIC- HTTP	3
DDoS attacks-LOIC-UDP	3
DoS attacks- HOIC	3
DoS attacks- Slowloris	3
DoS attacks-SlowHTTPTest	3
DoS attacks-Hulk	3
Infilteration	4
SQL Injection	5

7) Min-Max ölçekleme kullanılarak veri 0-1 aralığında ölçeklendirilmiştir.
8) Ölçeklendirilen veri üzerinde normalizasyon uygulanmıştır.

Veri kümesi 16.232.943 örneklem içermektedir. Bu büyük veri kümesini sadeleştirmek ve hesaplama zamanını düşürmek için Near-Miss alt örnekleme algoritmasından faydalanılarak veri kümesi %50 oranında küçültülmüştür. Veri kümesi %50 oranında küçültüldükten sonra örneklem sayısı 8.116.473 olmuştur. En yakın üç azınlık sınıfı örneğine minimum ortalama mesafeye sahip çoğunluk sınıf örnekleri üzerinden eleme yapılarak küçültme sağlanmıştır. Bu işlem yalnızca örneklem sayılarını azaltmak üzere yapılmıştır. Dolayısıyla, öznitelik sayısında da saldırı dağılımlarında da herhangi bir değişiklik olmamıştır. Küçültmede, orijinal veri kümesi ile edinilen F1-Skor ve doğruluk değerlerinin küçültülmüş veri kümesi ile edinilenlere yaklaşık eşit olmasına dikkat edilmiştir. Önerilen sisteminin pratikte hangi doğrulukta çalışacağını kestirebilmek adına denemelerin tümünde Katmanlı 5-Katlamalı Çapraz Doğrulama kullanılmıştır. Çalışmanın devamında küçültülmüş veri kümesi kullanılarak işlem yapılmıştır. Orijinal ve küçültülmüş veri kümeleri için Süre, F1-Skor, Doğruluk değer karşılaştırmaları Tablo 3.2 ve Tablo 3.3'te verilmiştir.

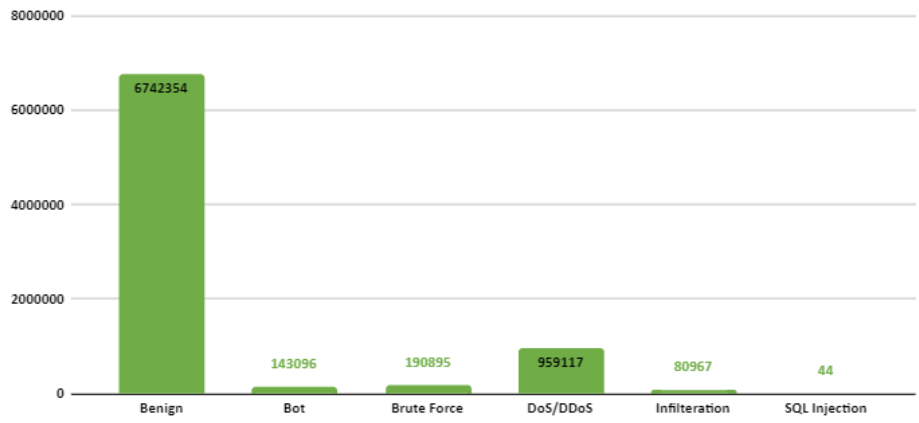
Tablo 3.2. Lojistik Regresyon Modeli ile Karşılaştırma Sonuçları

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

Tablo 3.3. Karar Ağacı Modeli ile Karşılaştırma Sonuçları

Veri kümesi	Doğruluk (%)	F1-Skor (%)	Süre (dk.)
Orijinal	98,75	97,33	78,29
Küçültülmüş	98,65	97,34	49,30

Küçültülmüş veri kümesindeki etiketlerin her biri için atak dağılımları Şekil 3.1'de gösterilmektedir.



Şekil 3.1. Küçültülmüş Veri Kümesi Etiket Kategori Dağılımı

9) Label sütunu ikili (binary) sınıflandırmaya uygun şekilde sayısallaştırılmıştır. Tablo 3.4 veri kümesinde yer alan kötü amaçlı ve normal ağ trafik yüzdesini göstermektedir.

Tablo 3.4. Saldırı ve Normal Ağ Trafik Yüzdesi

Tür	Sayı	Yüzde
Saldırı	1.374.119	16,93
Normal	6.742.354	83,07

3.2.Öznitelik Seçimi

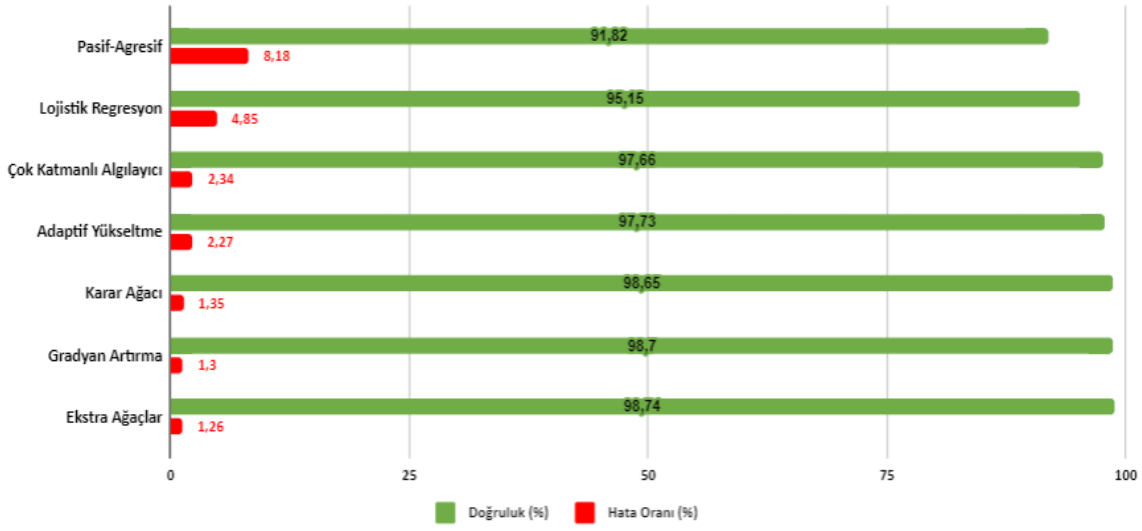
Çapraz doğrulama tekniği, modellerin görülmeyen veriler üzerindeki performaslarının olabildiğince objektif ve doğru değerlendirilmesini sağlar fakat beraberinde hesaplama ve zaman maliyetleri getirir. Çalışmada, bu maliyetlerin düşürülmesi amacıyla *joblib* kütüphanesinden faydalanılarak parallelleştirme uygulanmıştır.

Tablo 3.5 orijinal veri kümesi üzerinde makine öğrenmesi modelleri ile elde edilen doğruluk oranları ve süre değerlerini göstermektedir. Sonuçlar, ağaç tabanlı olan algoritmaların oldukça başarılı olduğunu ortaya koymaktadır. %98,74 doğruluk oranı ile ET'nin başarımı en yüksek model olduğu görülmektedir. Doğruluk oranları ve süre değerleri birlikte değerlendirildiğinde LR ve DT modelleri de öne çıkmaktadır.

Tablo 3.5. Tüm Öznitelikler ile Doğruluk Başarımları

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

Şekil 3.2 sınıflandırıcıların orijinal veri kümesindeki tüm öznitelikleri kullanarak elde ettikleri saldırı tespit başarımlarını doğruluk ve hata metrikleri üzerinden göstermektedir. Değerler, optimum sonucun ET sınıflandırıcısına ait olduğunu göstermektedir.



Şekil 3.2. Tüm Öznitelikler Kullanılarak Elde Edilen Doğruluk ve Hata Oranları

Alakasız özniteliklerin elenerek veri kümesinin kaliteli bir hale getirilmesi STS performansı için önemlidir.

Bu sebeple, CSE-CIC-IDS2018 veri kümesinden STS için en uygun olacak alt kümeyi belirlemek amacıyla aşağıdaki üç öznitelik seçim yaklaşımı ile denemeler yapılmıştır.

1. Ki-Kare testi ile özniteliklerden her birinin puan hesaplaması yapılmış, düşük puan alan öznitelikler veri kümesinden çıkarılmıştır.
2. Spearman korelasyon analizi ile yüksek korelasyonlu öznitelikler tespit edilerek veri kümesinden çıkarılmıştır.
3. RFE ile tüm özniteliklerin bir sıralaması yapılmış, en alakasız öznitelikler ayar parametrelerinde belirtilmiş olan öznitelik sayısına ulaşılıncaya kadar veri kümesinden çıkarılmıştır.

Tablo 3.6 Ki-Kare test sonuçlarını göstermektedir. Tablo içerisinde modelde kullanılacak olan yüksek puana sahip öznitelikler yer almaktadır.

Tablo 3.6. Ki-Kare Test Sonuçları

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

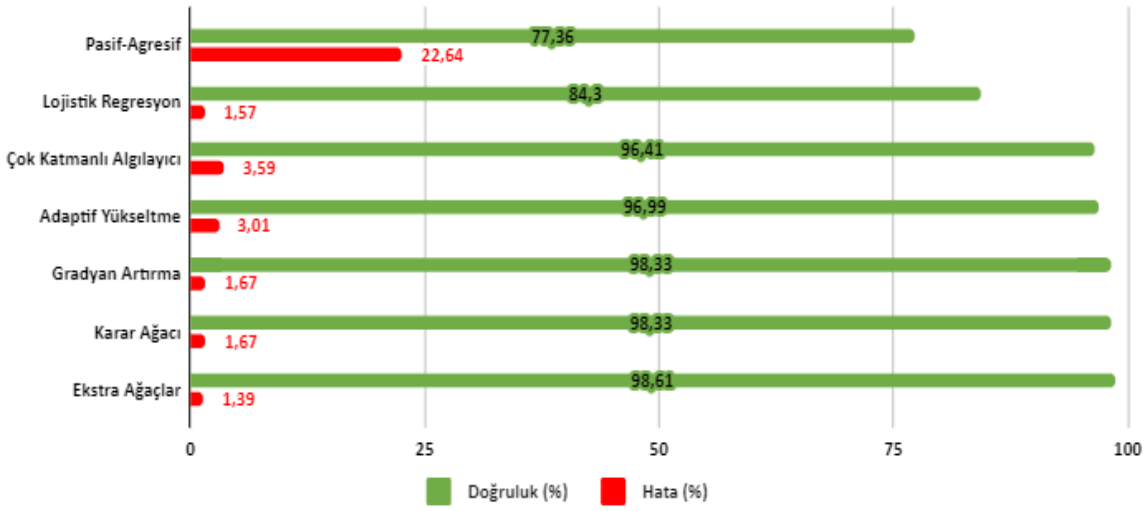
Tablo 3.7, seçilen öznitelikler ile oluşturulmuş veri kümesi üzerinde modellerin saldırı tespitine yönelik doğruluk başarımlarını içermektedir. Sonuçlara göre, ağaç tabanlı algoritmaların saldırı tespit başarımları yüksektir.

Fakat tüm öznitelikleri içeren orijinal veri kümesi ile kıyaslandığında doğruluk başarımlarının az miktarda düştüğü görülmektedir. Ek olarak, veri boyutunun indirgenmesi işlem yükünde azalma meydana getirdiği için işlem süresi kısalmıştır.

Tablo 3.7. Ki-Kare Öznitelik Seçimi ile Doğruluk Başarımları

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

Şekil 3.3 sınıflandırıcıların Ki-Kare testi ile seçilmiş öznitelikleri kullanarak elde ettikleri saldırı tespit başarımlarını doğruluk ve hata metrikleri üzerinden göstermektedir. Değerler, Şekil 3.2'deki gibi ET'nin en başarılı sınıflandırıcı olduğunu göstermektedir.



Şekil 3.3. Ki-Kare Test ile Seçilen Öznitelikler Kullanılarak Elde Edilen Doğruluk ve Hata Oranları

Tablo 3.8 Spearman'ın korelasyon analizine ait sonuçları göstermektedir. Bu çalışma kapsamında, 0,8 eşik değerini geçen öznitelikler yüksek korelasyonlu kabul edilmiştir. Korelasyon analizi sonrası 77 üzerinden 25 öznitelik seçilmiştir.

Tablo 3.8. Spearman Korelasyon Analiz Sonuçları

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

Tablo 3.9, seçilen öznitelikler ile oluşturulmuş veri kümesi üzerinde modellerin saldırı tespitine yönelik doğruluk başarımlarını içermektedir. Sonuçlar, ağaç tabanlı algoritmaların başarımlarının Ki-Kare testi ile elde edilenlere nispeten daha yüksek olduğunu göstermektedir. Denemeleri yapılan her iki öznitelik seçim yöntemi için elde edilen sonuçlar tüm öznitelikleri içeren orijinal veri kümesi kullanılarak elde edilen sonuçlarla karşılaştırıldığında, işlem süresine yönelik olumlu bir etkinin oluştuğu fakat doğruluk oranlarında artış sağlanmadığı görülmüştür. Ek olarak, lineer modellerin de başarımlarında düşüş olmuştur.

Tablo 3.9. Spearman'ın Sıralama Korelasyon Katsayısı Öznitelik Seçimi ile Doğruluk Başarımları

Model	Doğruluk (%)	Hata (%)	Süre (dk.)
ADA	97,2	2,8	7,53
DT	98,42	1,58	1,32
ET	98,69	1,31	12,1
GB	98,35	1,65	28,23
LR	81,27	18,7	0,53
MLP	95,94	4,06	194,53
PA	85,07	14,93	0,18

Şekil 3.4 sınıflandırıcıların Spearman'ın Sıralama Korelasyon Katsayısı ile seçilmiş öznitelikleri kullanarak elde ettikleri saldırı tespit başarımlarını doğruluk ve hata metrikleri üzerinden göstermektedir. Değerler, 25 öznitelik içeren veri kümesi üzerinde çalışan sınıflandırıcılara ait hata oranlarının tüm öznitelikleri içeren veri kümesi kullanılarak elde edilenlere göre daha yüksek olduğunu göstermektedir.



Şekil 3.4. Spearman'ın Sıralama Korelasyon Katsayısı ile Seçilen Öznitelikler Kullanılarak Elde Edilen Doğruluk ve Hata Oranları

RFE metotu tüm öznitelikleri içeren veri kümesi ile bir model oluşturur ve özniteliklerin her biri için önem puanı hesaplar. Sonra, daha az öneme sahip olan öznitelikleri veri kümesinden çıkarır. Oluşan yeni veri kümesi ile tekrar bir model oluşturur ve önem puan hesaplaması yapar. Bahsi geçen bu işlemler, RFE metotununun ayar parametrelerinde belirtilen öznitelik sayısına ulaşılıncaya kadar yinelenir. Bu metotta seçilmesi istenen öznitelik miktarı bir ayar parametresidir. Bir diğer ayar parametresi ise, önem puanlarını belirleyecek makine öğrenmesi tekniğidir. Tüm öznitelikleri içeren orijinal veri kümesi kullanılarak elde edilen sonuçlarda Karar Ağacı modeli doğruluk ve süre metrikleri açısından diğer modellerden daha başarılı olduğu için önem puanı belirleyicisi olarak bu model tercih edilmiştir. Sırasıyla 60, 50, 40 ve 30 öznitelik sayısı için sınıflandırma başarımları değerlendirilmiştir. 40 öznitelik sayısının işlem süresini kısaltmanın yanında başarımı da artırdığı görülmüştür. RFE metotuna ait sonuçlar Tablo 3.10'da verilmiştir.

Tablo 3.10. Özyinelemeli Öznitelik Eliminasyon Sonuçları

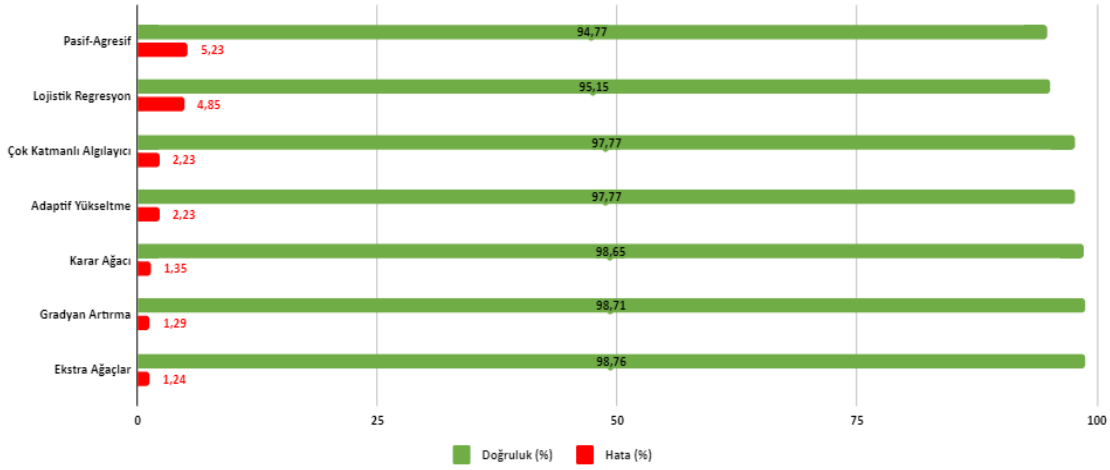
	Öznitelik		Öznitelik
1	Bwd IAT Max	21	ACK FlagCnt
2	Bwd IAT Mean	22	Active Max
3	Bwd IAT Std	23	Active Min
4	Bwd IAT Tot	24	Bwd IAT Min
5	BwdPktLenMax	25	BwdHeaderLen
6	BwdPktLenStd	26	BwdPkts/s
7	Flow IAT Max	27	BwdSeg Size Avg
8	Flow IAT Mean	28	ECE FlagCnt
9	Flow IAT Min	29	FwdAct Data Pkts
10	FlowByts/s	30	FwdHeaderLen
11	FlowDuration	31	FwdPkts/s
12	FlowPkts/s	32	FwdSeg Size Min
13	Fwd IAT Max	33	IdleMin
14	Fwd IAT Mean	34	InitBwd Win BytsNeg
15	Fwd IAT Min	35	Pkt Size Avg
16	Fwd IAT Tot	36	PktLenMax
17	FwdPktLenMax	37	PktLenStd
18	FwdPktLenMean	38	PSH FlagCnt
19	FwdPktLenStd	39	RST FlagCnt
20	Protocol	40	URG FlagCnt

Tablo 3.11, seçilen öznitelikler ile oluşturulmuş veri kümesi üzerinde modellerin saldırı tespitine yönelik doğruluk başarımlarını içermektedir.

Tablo 3.11. Özyinelemeli Öznitelik Eliminasyonu ile Doğruluk Başarımları

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

Şekil 3.5 sınıflandırıcıların RFE ile seçilmiş öznitelikleri kullanarak elde ettikleri saldırı tespit başarımlarını doğruluk ve hata metrikleri üzerinden göstermektedir. Değerler, RFE uygulanmış veri kümesi üzerinde çalışan sınıflandırıcıların hata oranlarının azaldığını veya sabit kaldığını göstermektedir.



Şekil 3.5. RFE ile Seçilen Öznitelikler Kullanılarak Elde Edilen Doğruluk ve Hata Oranları

Tablo 3.11'deki sonuçlar ve tüm öznitelikleri içeren veri kümesi ile elde edilen sonuçlar doğruluk oranı ve süre değerleri açısından kıyaslandığında RFE metotunun sistem başarımında artış sağladığı görülmüştür. Fakat bu evrede, CSE-CIC-IDS2018 gibi dengesiz bir veri kümesinde sadece doğruluk değerini göz önünde bulundurmak yanıltıcı olabileceğinden kesinlik, duyarlılık, F1-Skor değerleri de inceleme alınmıştır. Tüm öznitelikleri içeren orijinal veri kümsi kullanılarak elde edilen başarımlar Tablo 3.12'de verilmiştir.

Tablo 3.12. Tüm Öznitelikler- Duyarlılık, Kesinlik, F1-Skor Başarımları

Model	Duyarlılık (%)	Kesinlik (%)	F1-Skor (%)
ADA	94,47	96,44	95,42
DT	97,28	97,4	97,34
ET	97,19	97,82	97,5
GB	96,32	98,51	97,38
LR	92,42	89,35	90,79
MLP	96,04	95,36	95,61
PA	87,46	90,07	90,47

Tablo 3.13, seçilen öznitelikler ile oluşturulmuş veri kümesi üzerinde modellerin saldırı tespitine yönelik duyarlılık, kesinlik ve F1-Skor başarımlarını içermektedir.

Tablo 3.13. RFE ile Seçilen Öznitelikler- Duyarlılık, Kesinlik, F1-Skor Başarımları

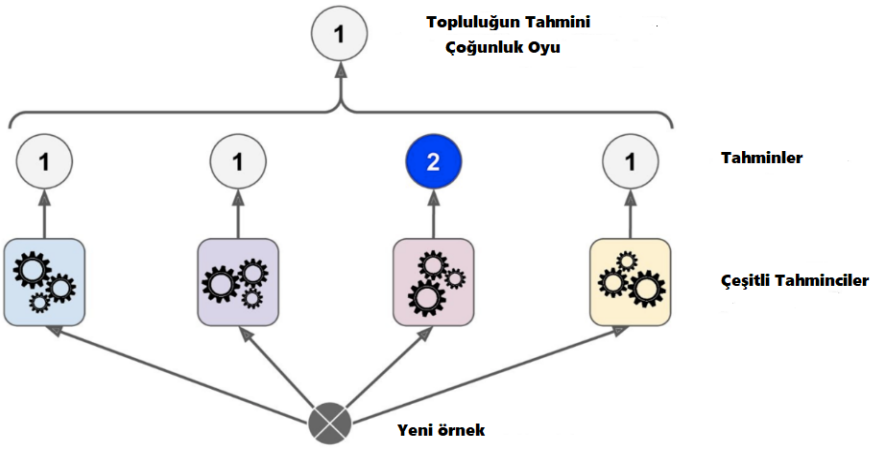
Model	Duyarlılık (%)	Kesinlik (%)	F1-Skor (%)
ADA	94,49	96,58	95,5
DT	97,28	97,4	97,34
ET	97,21	97,86	97,53
GB	96,34	98,52	97,39
LR	92,41	89,35	90,79
MLP	95,8	95,51	95,68
PA	86,19	87,03	79,62

RFE metotu PA modelinin doğruluk değerinde artış sağlamasına rağmen saldırı tespit başarısını düşürmüştür. Duyarlılık ve kesinlik değerleri diğer modeller için genel olarak iyileşmiştir. Doğruluk değerleri ile birlikte bu değerlerinde artıyor olması modellerin saldırı tespit başarısının arttığını göstermektedir. Duyarlılık ve kesinlik bireysel olarak model başarımını belirlemek için yeterli ölçütler değildir. Duyarlılık ve kesinlik ölçütleri sırası ile False Positive ve False Negative değerlerini dikkate almaz. Buradaki False Positive saldırı olarak yanlış nitelendirme, False Negative ise normal trafik olarak yanlış nitelendirmedir. Bu sebeple, F1-Skor ölçütü model başarımını belirleme noktasında en anlamlı ölçüttür. Sonuçlara göre, en başarılı model ET olmuştur fakat süre ölçütü de dikkate alındığında DT'nin %98,65 oranı ile daha başarılı olduğu söylenebilir. Tüm bu bilgiler ışığında, makine öğrenmesi tabanlı STS'lerde uygun ayar parametreleri ile kullanılan RFE'nin başarım ve performansı artırabileceği sonucuna varılmıştır. Bu sebeple, ikinci aşamada oluşturulacak kolektif modelin değerlendirmeleri RFE metotu ile seçilen özniteliklerden oluşan veri kümesi ile yapılmıştır.

3.3.Model Seçimi ve Kolektif Öğrenme

Kolektif öğrenme, doğruluğu artırabilecek tahminler üretmek için birkaç öğrenme algoritmasının birleştirildiği bir makine öğrenme yöntemidir (Polikar, 2012). Kolektif öğrenmenin torbalama, yükseltme, yığınlama ve oylama olmak üzere 4 farklı türü bulunmaktadır. Bunlardan oylamada, birkaç tek sınıflandırıcının gücü, kararlar için bir kombinasyon kuralının uygulanmasını kolaylaştırır (More ve ark., 2016). Bu çalışma ikili sınıflandırmaya odaklandığından, çözüm olarak çoğunluk oyu tercih edilmiştir.

Çoğunluk oyu demokratik kurallara uyar, yani en fazla oyu alan sınıf sonuçtur (More ve ark., 2016). Her bir sınıflandırıcı kendi tahmini üretir ve model her sınıflandırıcıda üretilen en yüksek sayıda tahmin sonucunu seçer. Şekil 3.6 çoğunluk oyu sınıflandırıcısının bir gösterimini sağlamaktadır.



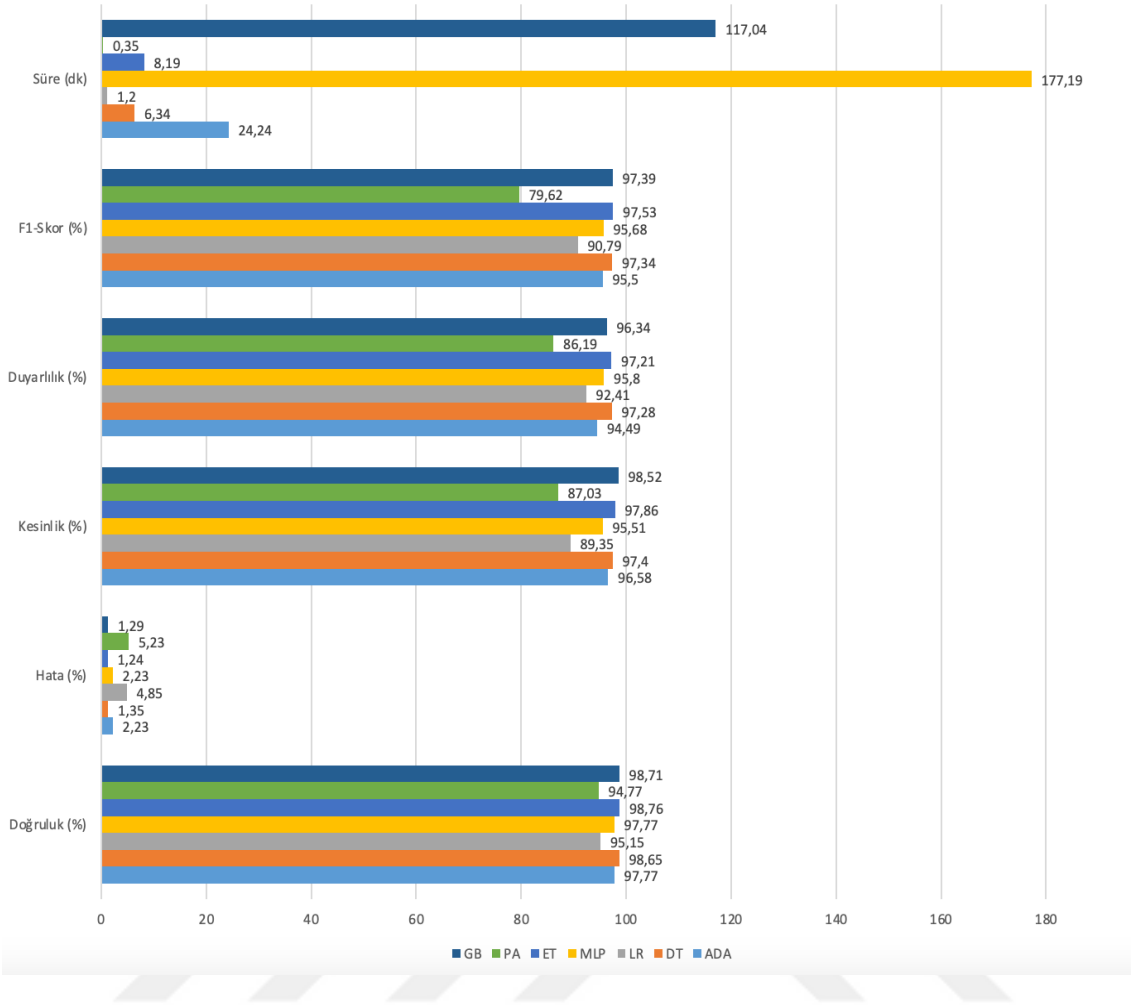
Şekil 3.6. Çoğunluk Oyu Sınıflandırıcı Tahminleri

Çalışmada kolektif model için, özyinelemeli öznitelik eliminasyonu temelinde 40 öznitelikli bir veri kümesi kullanılmıştır. Tüm sınıflandırıcılar, Python scikit-learn kitaplığındaki varsayılan değer parametreleri ile geliştirilmiştir (Scikit-learn, 2022). Tablo 3.14 Her bir öğrenme algoritması için tahmin sonuçlarını göstermektedir.

Tablo 3.14. RFE Yöntemi ile Tahmin Sonuçları

Model	Doğruluk (%)	Hata (%)	Duyarlılık (%)	Kesinlik (%)	F1-Skor (%)	Süre (dk.)
ADA	97,77	2,23	94,49	96,58	95,50	24,24
DT	98,65	1,35	97,28	97,40	97,34	6,34
LR	95,15	4,85	92,41	89,35	90,79	1,20
MLP	97,77	2,23	95,80	95,51	95,68	177,19
ET	98,76	1,24	97,21	97,86	97,53	8,19
PA	94,77	5,23	86,19	87,03	79,62	0,35
GB	98,71	1,29	96,34	98,52	97,39	117,04

Şekil 3.7 RFE yöntemi kullanılarak doğruluk, kesinlik, duyarlılık, F1-skor ve süre metrikleri bazında elde edilen model başarımlarını göstermektedir.



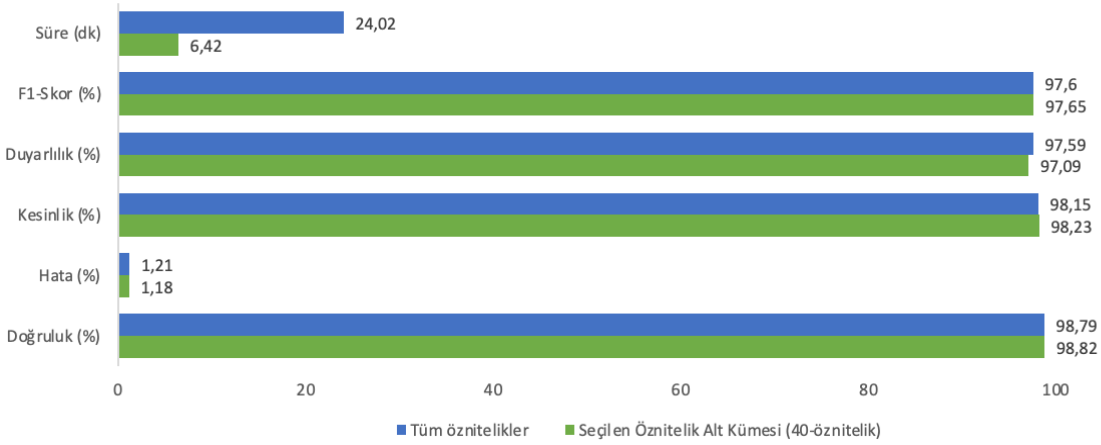
Şekil 3.7. RFE ile Seçilen Öznitelikler Kullanılarak Elde Edilen Doğruluk, Hata, Kesinlik, Duyarlılık, F1-Skor ve Süre Değerleri

Gradyan, Adaptif Yükseltme ve Çok Katmanlı Algılayıcı modelleri yüksek doğruluk oranlarına sahiptir. Fakat bu modellerde tahmin süreçleri diğer modellere göre oldukça uzundur. Pasif Agresif modelinde tahmin süreci oldukça kısa fakat doğruluk oranı düşüktür. Tahmin süreci ve doğruluk oranı birlikte değerlendirildiğinde Karar Ağacı, Lojistik Regresyon ve Ekstra Ağaçlar modellerinin hızlı tahmin süreleri ile yüksek doğruluk oranına sahip sonuçlar ürettiği görülmüştür. Bu sebeple, kolektif modeli oluşturmak için bu üç model tercih edilmiştir. Tablo 3.15 kolektif model için tahmin sonuçlarını göstermektedir.

Tablo 3.15. Önerilen Kolektif Model ile Tahmin Sonuçları

Kolektif Model	Doğruluk (%)	Hata (%)	Kesinlik (%)	Duyarlılık (%)	F1-Skor (%)	Süre (dk.)
Seçilen Öznitelik Alt Kümesi (40-öznitelik)	98,82	1,18	98,23	97,09	97,65	6,42
Tüm öznitelikler	98,79	1.21	98,15	97,59	97,60	24,02

Şekil 3.8 Kolektif model ile elde edilen başarımları göstermektedir.



Şekil 3.8. Kolektif Yöntem ile Elde Edilen Doğruluk, Hata, Kesinlik, Duyarlılık, F1-Skor ve Süre Değerleri

Önerilen kolektif modelin doğruluğu, tüm özniteliklerin kullanımındaki sonuçlara nispeten daha düşük bir tahmin süresi ile %98,82'dir. Spesifik olarak, önerilen model tespit süresini 24 dk. 2 sn'den 6 dk. 42 sn.'yeye kadar düşürmüştür.

4. SONUÇ VE GELECEK ÇALIŞMALAR

Bu çalışmada, öznitelik seçimi ve kolektif öğrenme metotlarını uygulamanın makine öğrenmesi tabanlı STS'lerde performansı artırabileceği sonucuna ulaşılmıştır. Çalışmanın gerçeklenmesinde CSE-CIC-IDS2018 veri kümesi kullanılmıştır. İlk aşamada, Spearman 'ın Korelasyon Analizi, Özyinelemeli Öznitelik Eliminasyonu (RFE) ve Ki-Kare Test metotlarından faydalanarak boyut azaltmaya gidilmiştir. Gereksiz ve önemsiz özniteliklerin ayıklanması ile saldırı tespit sürecinin ve performansının iyileştirilmesi hedeflenmiştir.

İncelenen öznitelik seçim metotlarının saldırı tespit başarımına etkisini görmek için her bir metot tarafından belirlenen özniteliklerle oluşturulan yeni veri kümeleri üzerinde denemeler yapılmıştır. Yeni veri kümelerinin orijinal boyuttaki veri kümesi ile karıştırılmasında Karar Ağacı, Gradyan Artırma, Adaptif Yükseltme, Lojistik Regresyon, Pasif-Agresif, Ekstra Ağaçlar ve Çok Katmanlı Algılayıcı sınıflandırıcıları kullanılmıştır. Yapılan tüm denemelerde Katmanlı 5-Katlamalı Çapraz Doğruluma Tekniği kullanılmıştır. Bu tekniğin beraberinde getirdiği zaman ve hesaplama maliyetlerini düşürmek için çok-çekirdekli paralelleştirme uygulanmıştır. Spearman'ın korelasyon analizi ve Ki-Kare test metotları ile elde edilen sonuçlar, ağaç tabanlı modellerin saldırıları %97 üzerinde bir başarım ile tespit edebildiğini göstermiştir. Ağaç tabanlı model başarımları işlem süresi ve doğruluk değer kriterlerine göre değerlendirildiğinde Spearman'ın korelasyon analizinin, Ki-Kare testinden daha üstün bir başarım sağladığı görülmüştür. Spearman'ın korelasyon analizi ve Ki-Kare test metotlarına ait sonuçlar orijinal veri kümesi kullanılarak elde edilmiş sonuçlar ile kıyaslandığında her iki yöntemin işlem süresi açısından olumlu bir etki oluşturduğu fakat, doğruluk oranlarında herhangi bir artış sağlamadığı sonucuna ulaşılmıştır. Spesifik olarak, lineer modellerin başarısında azalma meydana gelmiştir. RFE metotu işlem süresi ve doğruluk metriklerinin her ikisi için tüm modellerde sistem başarımını artırmıştır. CSE-CIC-IDS2018 gibi dengeli dağılıma sahip olmayan bir veri kümesinde sadece doğruluk oranı üzerinden bir değerlendirme yapmak yetersiz kalabilceğinden bu aşamada duyarlılık, kesinlik ve F1-Skor metrikleri de incelemeye alınmıştır. Deneysel sonuçlar, Spearman'ın Sıralama Korelasyon Katsayısı ve Ki-Kare test metotlarının veri boyutunda azalma sağlaması nedeniyle işlem süresini %45 oranında kısalttığını fakat hata oranını sırasıyla %10,52 ve %14,46 oranında artırdığını ortaya koymuştur. Ayrıca, RFE metotu -uygun ayar parametrelerinin kullanıldığı durumda- işlem süresinde %38 oranında bir kısalma sağlamış ve sistemin hata oranını da %2,95'e kadar azaltmıştır. Dolayısıyla, makine öğrenmesi tabanlı STS'lerde başarım ve performans iyileştirmesi RFE metotunun doğru ayar parametreler ile kullanılması yoluyla sağlanabilir.

Çalışmanın ikinci aşamasında, -doğruluk başarımını yükseltebilmek adına- bireysel sınıflandırıcıların faydalarının birleştirildiği kolektif bir model üretilmiştir. Bu model, ilk aşamada elde edilen performans sonuçları üzerinden seçilen sınıflandırıcılar ile oluşturulmuştur. Kolektif modelin performans değerlendirmeleri için RFE temelinde 40 öznitelikli bir veri kümesi kullanılmıştır.

Tahmin süreci ve doğruluk oranı birlikte değerlendirildiğinde Karar Ağacı, Lojistik Regresyon ve Ekstra Ağaçlar modelleri hızlı tahmin süreleri ile yüksek doğruluk oranına sahip sonuçlar ürettiğinden, kolektif modeli oluşturmak için bu üç model kullanılmıştır. Önerilen kolektif model, tüm özniteliklerin kullanımındaki sonuçlara nispeten daha düşük bir tahmin süresi ve %98,82'lik yüksek bir doğruluk oranına sahiptir. Spesifik olarak, önerilen model tespit süresinde yaklaşık %73'lük bir azalma ve doğruluk oranında da %3'lük bir artış sağlamıştır.

İleriki çalışmalarda, hiper parametre ayarlama teknikleri (hyperparameter-tuning techniques) varsayılan ve optimal parametrelerle sınıflandırıcıların karşılaştırılmasında kullanılabilir. Önerilen modelin DNN (Evrişimsel Sinir Ağı), RNN (Yinelemeli Sinir Ağı) ve ANN (Yapay Sinir Ağı) gibi sinir ağı algoritmaları ile karşılaştırmalı bir analizi yapılarak, modelin performansını değerlendirmek için farklı STS veri kümelerinden faydalanılabilir.

KAYNAKLAR

Kim, K., Aminanto, M.E., Tanuwidjaja, H.C. (2018). Network Intrusion Detection Using Deep Learning: A Feature Learning Approach. Singapore: Springer, pp. 35-45.

Mishra, P., Varadharajan, V., Tupakula, U., Pilli, E.S. (2018). A Detailed Investigation and Analysis of Using Machine Learning Technique for Intrusion Detection. IEEE Communications Surveys and Tutorials, PP (99): 1-1.

Gao, X., Shan, C., Hu, C., Niu, Z., Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. IEEE Access, 7, 82512-82521.

Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), pp. 1-8.

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

Sommer, R., Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. IEEE Symposium on Security & Privacy, Oakland, 05-316.

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

Wankhede, S., Kshirsagar, D. (2018). DoS Attack Detection Using Machine Learning and Neural Network. Proc. 4th Int. Conf. Comput. Commun. Control Autom., pp. 1-5.

Zhou, Q., Pezaros, D. (2019). Evaluation of Machine Learning Classifier for Zero-Day Intrusion detection-An Analysis on CIC AWS 2018 Dataset, CoRR, vol. abs/1905.03685.

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

Yulianto, A., Sukarno, P., Suwastika, N.A. (2018). Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset. Journal of Physics: Conference Series, 1192.

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

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

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

Ferrag, M.A., Maglaras, L. (2019). DeliveryCoin: An IDS and Blockchain-Based Delivery Framework for Drone-Delivered Services. Computers, vol. 8, no. 3, pp. 58.

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

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

Fitni, Q.R.S., Ramli, K. (2020). Implementation of Ensemble Learning and Feature Selection for Performance Improvements in Anomaly-Based Intrusion Detection Systems. Proc. IEEE Int. Conf. Ind. 4.0 Artif. Intell. Commun. Technol. (IAICT), pp. 118-124.

Aksu, D., Aydin, M.A. (2018). Detecting Port Scan Attempts with Comparative Analysis of Deep Learning and Support Vector Machine Algorithms. International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), pp. 77-80.

Ullah, I., Mahmoud, Q.H. (2019). A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks. 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, pp. 1-6

Yilmaz, S., Sen, S. (2019). Early Detection of Botnet Activities Using Grammatical Evolution. Theory and Applications of Models of Computation, pp. 395-404

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

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

Emhan, Ö., Akın, M. (2019). Filtreleme Tabanlı Öznitelik Seçme Yöntemlerinin Anomali Tabanlı Ağ Saldırısı Tespit Sistemlerine Etkisi. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(2), 549-559.

Ren, S.Q., Tan, B.H.M., Sundaram, S., Wang, T., Ng, Y., Chang, V., Aung, K.M.M. (2016). Secure Searching on Cloud Storage Enhanced by Homomorphic Indexing. Future Gener. Comput. Syst. Vol.65, pp. 102–110.

Anwar, S., Mohamad Zain, J., Zolkipli, M. F., Inayat, Z., Khan, S., Anthony, B., Chang, V. (2017). From Intrusion Detection to an Intrusion Response System: Fundamentals, Requirements, and Future Directions. Algorithms, 10(2), 39.

Gezgin, M. D., Buluş, E. (2013). Kablosuz Ağlar için Bir DoS Saldırısı Tasarımı. Bilişim Teknolojileri Dergisi, 6(3), 17-23.

Kara, M., Şişeci, N.E. (2011). Botnetlerle Mücadelede Dünyadaki ve Türkiye'deki Durum, Akademik Bilişim'11- XIII. Akademik Bilişim Konferansı Bildirileri

Microsoft Security Intelligence Report v9, http://www.microsoft.com/security/sir, (2010).

Scarfone, K., Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (idps). NIST Special Publication, vol. 800, p. 94.

Chezian, D. R. (2017). Intrusion Detection System Techniques and Tools: A Survey. Scholars Journal Engineering Technoogy, 5(3), 122-130

Garg, A., & Maheshwari, P. (2016). A Hybrid Intrusion Detection System: A Review. In 2016 10th International Conference on Intelligent Systems and Control (ISCO) (pp. 1-5). IEEE.

Murali, A., Rao, M. (2005). A Survey on Intrusion Detection Approaches. First International Conference on Information and Communication Technologies, IEEE Communications Society Press, 233-240.

Aydın, M. A. (2005). Bilgisayar Ağlarında Saldırı Tespiti İçin İstatistiksel Yöntem Kullanılması. İstanbul, İstanbul Teknik Üniversitesi.

Axelsson, S. (2000). Intrusion Detection Systems: A Survey and Taxonomy. Technical Report 99-15, Dept. Of Computer Eng., Chalmers University of Technology, Göteborg, Sweden, 1-23.

Patcha, A., Park, J.M. (2007). An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. Computer Networks, 51(12): 3448-3470.

Lee, W., Stolfo, S.J., Chan, P.K., Eskin, E., Fan, W., Miller, M., Hershkop, S., Zhang, J. (2001). Real Time Data Mining-based Intrusion Detection. Second {DARPA} Information Survivability Conference and Exposition (DISCEX II), Anaheim, CA, 89-100.

Ilgun, K., Kemmerer, R., Porras, P. (1995). State Transition Analysis: A Rule Based Intrusion Detection System. Software Engineering, 21(3): 181-199

Sundaram, A. (1996). An Introduction to Intrusion Detection. Crossroads: The ACM Student Magazine, New York, USA, 2(4), 3-7.

Cannady J. (1998). Artificial Neural Networks for Misuse Detection. Proceedings of the 1998 National Information Systems Security Conference (NISSC'98), Arlington, VA, 443-456.

Esponda, F., Forrest, S., Helman, P. (2004). A Formal Framework For Positive and Negative Detection Schemes. IEEE Transactions on Systems, Man, and Cybernetics, Part B, Cybernetics, 34(1): 357-373.

Mukkamala, S., Sung, A.H., Abraham, A. (2003). Intrusion Detection Using Ensemble of Soft Computing Paradigms. Third International Conference On Intelligent Systems Design and Applications, Germany: Springer, 239-248.

Stein, G., Chen, B., Wu, A.S., Hua, K.A. (2005). Decision Tree Classifier for Network Intrusion Detection with GA-based Feature Selection. ACM Southeast Regional Conference (2), Kennesaw, Georgia, 136-141.

Mitchell, T. M. (1997). Machine Learning, New York.

Balta, A. (2018). Makine Öğrenmesi Teknikleri ile Ekolojik Verilerin Değerlendirilmesi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Elâzığ, 84s.

Fielding, A.H. (1999). Machine Learning Methods for Ecological Applications. Springer Science & Business Media.

Debeljak, M. (2012). Inductive Machine Learning in Ecological Modeling: Invited Talk. University of Tennessee, Oak Ridge National Laboratory.

Vieira, S., Lopez Pinaya, W. H. ve Mechelli, A. (2019). Machine Learning: Methods and Applications to Brain Disorders. Elsevier, ss. 1–20, doi: 10.1016/B978-0-12-815739-8.00001-8.

Subasi, A. (2020). Practical Machine Learning for Data Analysis Using Python. Elsevier, ss. 91–202, doi: 10.1016/B978-0-12- 821379-7.00003-5.

Yılmaz, A. (2019). Yapay Zekâ, 6. Baskı, KODLAB Yayınevi, ss. 39–57.

Hamzaçebi, Ç. (2011). Yapay Sinir Ağları: Tahmin Amaçlı Kullanımı MATLAB ve Neurosolutions Uygulamalı, Ekin Basım Yayın Dağıtım.

Bilgin, M. (2018). Makine Öğrenmesi. 2. Basım, İstanbul, Papatya Yayıncılık.

Shobha, G. ve Rangaswamy, S. (2018). Handbook of Statistics, 1. Baskı, Elsevier B.V., ss. 197–228, doi: 10.1016/bs.host.2018.07.004.

Drucker, H. (1997). Improving Regressors Using Boosting Techniques. 14th International Conference on Machine Learning, 107–115.

Acılar, A., M. (2020). AdaBoost.R2 Regresyon Algoritması ile Konutların Isıtma ve Soğutma Yüklerinin Tahmin Edilmesi, Ejons International Journal on Mathematics, Engineering and Natural Sciences, ISSN No. 2602-4136.

Özekes, S. (2003). Veri Madenciliği Modelleri ve Uygulama Alanları. İstanbul Ticaret Üniversitesi Dergisi, Cilt 2, Sayı 3, 65-82.

Han, J., Kamber, M. (2000). Data Mining Concepts and Techniques, Morgan Kaufmann Publishers 1st Ed., San Francisco, USA.

Tatlıdil, H. (2002). Uygulamalı Çok Değişkenli İstatistiksel Analiz, Ankara: Engin Yayınları.

Hosmer, D.W., Lemeshow, S. (2000). Applied Logistic Regression (2nd ed.), New York: Wiley and Sons, Inc.

Kuyucu, Y. E. (2012). Lojistik Regresyon Analizi (LRA), Yapay Sinir Ağları (YSA) ve Sınıflandırma ve Regresyon Ağaçları (C&RT) Yöntemlerinin Karşılaştırılması ve Tıp

Alanında Bir Uygulama, Yüksek Lisans Tezi, Gaziosmanpaşa Üniversitesi Sağlık Bilimleri Enstitüsü, Tokat.

Özdamar, K. (2002). Paket Programlar ile İstatistiksel Veri Analizi (4.baskı). Eskişehir: Kaan Kitapevi.

Güneri, N., Apaydın, A. (2004) Öğrenci Başarılarının Sınıflandırılmasında Lojistik Regresyon Analizi ve Sinir Ağları Yaklaşımı, Ticaret ve Turizm Eğitim Fakültesi Dergisi, 1.

Agresti, A. (2002). Categorical Data Analysis (2nd ed.). Florida: Wiley and Sons, Inc.

Tatlıdil, H., Demirağ, İ. (2014) Türkiye'de Yoksulluğun Sosyo-Ekonomik ve Demografik Değişkenlerle İlişkilerinin Lojistik Regresyon ve Mars Yöntemleri Kullanılarak İncelenmesi, TISK Academy/TISK Akademi, 9(17).

Alpar, R. (2017) Çok Değişkenli İstatistiksel Yöntemler, 5. baskı, Ankara: Detay Yayıncılık.

Geurts, P., Ernst, D., Wehenkel, L. (2006). Extremely Randomized Trees. Machine Learning. 63: 3–42

Friedman, J. (2000). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 11-28.

Kabak, M., Yüce, T. (2021). Makine Öğrenmesi Algoritmaları ile Detay Üretim Alanları İçin İş Merkezi Kırılımında Üretim Süresi Tahminleme. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, Cilt 37, Sayı 1.

Arı, A., Berberler, B. E. (2017). Yapay Sinir Ağları ile Tahmin ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı. ACTA INFOLOGICA, vol 1 no, 2, pp. 55-73.

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

Diao, R. (2014). Feature Selection with Harmony Search and Its Applications. Ph.D. Thesis, Aberystwyth University, 213p (Unpublished).

Onan, A., Korukoğlu, S. (2017). Metin Sınıflandırmada Öznitelik Seçim Yöntemlerinin Değerlendirilmesi.

Inc, S. (2007). SPSS Clementine 12.0 Algorithms Guide. SPSS Inc, Chicago.

Ünver, Ö., Gamgam, H. (2006). Uygulamalı Temel İstatistik Yöntemler. Seçkin Yayıncılık, Ankara

Budak H., (2018). Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi Cilt 22, Özel Sayı, 21-31

Hall, M. (1999). Correlation-based Feature Selection for Machine Learning, The University of Waikato, PhD Thesis, Hamilton.

Guyon, I. Elisseeff, A. (2003). An Introduction to Variable and Feature Selection, Journal of Machine Learning Research. 3, 1157-1182.

Haq, N. F., Onik, A. R. ve Shah, F. M. (2015). An Ensemble Framework of Anomaly Detection Using Hybridized Feature Selection Approach (HFSA). İçinde: 2015 SAI Intelligent Systems Conference (IntelliSys). IEEE, ss. 989–995. doi: 10.1109/IntelliSys.2015.7361264.

Siddique, K., Akhtar, Z., Lee, H., Kim, W. ve Kim, Y. (2017). Toward Bulk Synchronous Parallel-Based Machine Learning Techniques for Anomaly Detection in High-Speed Big Data Networks. Symmetry, 9(9), s. 197. doi: 10.3390/sym9090197.

Ahmad, T. ve Aziz, M. N. (2019). Data Preprocessing and Feature Selection for Machine Learning Intrusion Detection Systems", ICIC Express Letters, 13(2), ss. 93–101. doi: https//doi.org/10.24507/icicel.13.02.93.

Nilsson, N. J. (1965). Learning Machines: Foundations of Trainable Pattern-classifying Systems. New York, McGraw Hill.

Yang, Y. (2017). Ensemble Learning. İçinde: Temporal Data Mining Via Unsupervised Ensemble Learning. Elsevier, ss. 35–56. doi: 10.1016/B978-0-12- 811654-8.00004-X.

Sugiyama, M. (2016a). Ensemble Learning. İçinde: Introduction to Statistical Machine Learning. Elsevier, ss. 343–354. doi: 10.1016/B978-0-12-802121- 7.00041-8.

Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), ss. 123–140. doi: 10.1007/BF00058655.

Sutton, C. D. (2005). Classification and Regression Trees, Bagging, and Boosting. İçinde: Handbook of Statistics. Elsevier Masson SAS, ss. 303–329. doi: 10.1016/S0169-7161(04)24011-1.

Breiman L. (2001) Random Forests. 1th ed. Boston, Springer.

Zhang C, Ma Y. (2012). Ensemble Machine Learning: Methods and Applications. 5th ed. Boston, Springer.

Zhou, Z. H. (2012). Ensemble Methods: Foundations and Algorithms. Chapman and Hall / CRC, p.70.

Wolpert D.H. (1992). Stacked Generalization. Neural Networks, 5(2): 241-59.

Polikar R. (2012). Ensemble Learning. Ensemble Machine Learning, 10th ed. Boston, Springer 2012:1-34.

Akman M, Genç Y, Ankarali H. (2011). Random Forests Yöntemi ve Sağlık Alanında Bir Uygulama. Türkiye Klinikleri Biyoistatistik, 3(1): 36-48

Rokach L. (2010). Pattern Classification Using Ensemble Methods, World Scientific 2010: vol.75.22-90.

CyberSecurity, Erişim Tarihi: 02.06.2022, https://www.unb.ca/cic/datasets/ids-2018.html

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

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

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

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

Scikit-learn.org. Scikit-learn: Machine Learning in Python - scikit-learn 1.0.1 Documentation. [online] https://scikit-learn.org/stable/, 2022.

NumPy.org. Numpy Documentation. [online] https://numpy.org/, 2022.

Pandas.org. Pandas Python Data Analysis Library Documentation. [Online] https://pandas.pydata.org/, 2022.

More, S., Gaikwad, P. (2016). Trust-based Voting Method for Efficient Malware Detection. Procedia Comp Science 79 657-667.ö