FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

EVRİMSEL TABANLI KOLEKTİF ÖĞRENME SİSTEMLERİ KULLANARAK SALDIRI TESPİT SİSTEMLERİ TASARIMI

YÜKSEK LİSANS TEZİ

YAHYA BİLİR

İSTANBUL, 2022

FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

EVRİMSEL TABANLI KOLEKTİF ÖĞRENME SİSTEMLERİ KULLANARAK SALDIRI TESPİT SİSTEMLERİ TASARIMI

YÜKSEK LİSANS TEZİ

YAHYA BİLİR
(180221008)

Danışman
(Dr. Öğr. Üyesi Berna Kiraz)

DÜZELTİLMİŞ TEZ

İSTANBUL, 2022

FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
TEZ ONAY FORMU

14/10/2022

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Tezli Yüksek Lisans programı 180221008 numaralı Yahya BİLİR'in hazırladığı "Evrimsel Tabanlı Kolektif Öğrenme Sistemleri Kullanarak Saldırı Tespit Sistemleri Tasarımı" konulu Yüksek Lisans tezi ile ilgili Tez Savunma Sınavı, 14/10/2022 Cuma günü saat 10:00'da yapılmış, sorulara alınan cevaplar sonunda adayın tezinin **Kabulüne Oy Birliği** ile karar verilmiştir.

Tez adı değişikliği yapılması halinde: Tez adının .....
.....
şeklinde değiştirilmesi uygundur.

Jüri Üyesi	Karar
1. Dr. Öğr. Üyesi Berna KİRAZ (Danışman)	**Kabul**
2. Prof. Dr. Ayşe Şima UYAR	**Kabul**
3. Dr. Öğr. Üyesi Fatma CORUT ERGİN	**Kabul**
4.	.....
5.	.....
6. (İkinci Danışman)*.....	.....

**2. Danışman varsa doldurulması gerekmektedir.*

ETİK BİLDİRİM

Bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bağlı olduğum üniversite veya bir başka üniversitedeki başka bir çalışma olarak sunulmadığını beyan ederim.

Yahya Bilir

DÜZELTME METNİ

1. İçindekiler kısmı düzenlendi, şekil listesine eklemeler yapıldı.
2. Tablo eklemesi yapıldı ve tablo listesine yansıtıldı.
3. Önsöz, giriş ve sonuç bölümleri yeniden düzenlendi; eklemeler yapıldı.
4. Üçüncü, dördüncü ve beşinci bölümler yeniden düzenlendi, eklemeler yapıldı.
5. Sonuç bölümüne ekleme yapıldı.

TEŞEKKÜR

Bu tez çalışmasında desteklerini esirgemeyen hocam Dr. Öğr. Üyesi Berna KİRAZ'a, yüksek lisans konusunda teşviklerinden dolayı kurumum Takasbank'a ve çalışma arkadaşlarıma, bu süreçte en büyük destekçim olan aileme ve değerli hanımıma teşekkürü borç bilirim.

Yahya Bilir

EVRİMSEL TABANLI KOLEKTİF ÖĞRENME SİSTEMLERİ KULLANARAK SALDIRI TESPİT SİSTEMLERİ TASARIMI

Yahya Bilir

ÖZET

Son yıllardaki internet ve iletişim teknolojilerinde özellikle de nesnelerin interneti teknolojilerindeki gelişmeler ev otomasyonu, akıllı şehirler, gelişmiş sağlık ve üretim sistemlerinin oluşmasında temel etken olmuştur. İnternete bağlanan cihaz sayısı sürekli artmaktadır. Bu durum ağ cihazlarını ve hizmetlerini siber saldırıların hedefi haline ve ağ güvenliğinin hayati bir noktaya gelmesine neden olmuştur. Bu nedenle bu alandaki çalışmalar da çok kritik bir noktaya gelmiştir. İletişim ağlarının ve bu ağların arkasındaki tüm cihazların güvenliğini sağlamak için antivirüs yazılımları, güvenlik duvarı ve saldırı önleme sistemleri gibi güvenlik önlemlerinden faydalanılır. Bu önlemler birçok saldırıları engellese de bazı saldırıları engellemekte eksik kalırlar. Bu eksiklerin ortadan kaldırılması için saldırı önleme sistemleri ve saldırı tespit sistemleri devreye girer. Bu sistemlerin tasarımında yapay zeka, makine öğrenmesi derin öğrenme gibi algoritmalardan faydalanılmaktadır. Ağ trafiğini dinleyerek kötü niyetli ve şühpeli davranışların tespit edilmesi konusunda yardımcı olur. Saldırı önleme sistemleri saldırıların hem tespit edilmesi hem de önlemesi konusunda kullanılan bir mekanizma iken saldırı tespit sistemleri (STS) yalnızca güvenlik ihlallerinin tespit edilmesi ve analizi konusunda faydalanılan sistemlerdir. Bu çalışmada da NSL-KDD veri seti kullanılmış ve öznitelik seçiminde diferansiyel evrimsel algoritmasından faydalanarak Adaboost kolektif makine öğrenmesi kullanılarak ihlal tespit sistemi tasarımı gerçeklenmiştir.

Anahtar kelimeler: STS, NSL-KDD, makine öğrenmesi, adaboost, diferansiyel evrim algoritması.

INTRUSION DETECTION SYSTEMS DESIGN USING EVOLUTIONARY BASED ENSEMBLE LEARNING SYSTEMS

Yahya Bilir

ABSTRACT

In recent years, developments in internet and communication technologies, especially in internet of things, have been the main factor in the formation of home automation, smart cities, advanced health and production systems. The number of devices connected to the Internet is constantly increasing. This situation has made network devices and services the target of cyber attacks and network security has become a vital point. Therefore, studies in this field have reached a critical point. Security measures such as antivirus software, firewall and intrusion prevention systems are used to ensure the security of communication networks and all devices behind these networks. Although antivirus and firewalls block many attacks, they fail to prevent some attacks. Intrusion prevention systems (IPS) and intrusion detection systems (IDS) come into play to eliminate these security deficiencies. In the design of these systems, algorithms such as artificial intelligence, machine learning and deep learning are used. It helps detect malicious and suspicious behavior by listening to network traffic. While intrusion prevention systems are a mechanism used to both detect and prevent attacks, intrusion detection systems are systems that are only used for detecting and analyzing security breaches. In this study, NSL-KDD data set was used and a intrusion detection system was designed using Adaboost ensemble machine learning by using differential evolutionary algorithm for feature selection.

Keywords: IDS, NSL-KDD, machine learning, adaboost, differantial evolution.

ÖNSÖZ

Bu çalışmada kolektif öğrenme algoritmaları ve diferansiyel evrim algoritmasından faydalanılarak saldırı tespit sistemi tasarımı amaçlanmıştır. Birden fazla zayıf öğrenici bir araya getirilerek daha güçlü bir model geliştirilebileceği düşünülmüştür. Burada destek vektör makinesi, karar ağaçları, Naive Bayes, K-en yakın komşu gibi makine öğrenmesi algoritmalarından faydalanılması amaçlanmaktaydı. Ancak Adaboost ile kullanılması sırasında eğitim verisi ile modelin eğitilmesi aşamasında ilerleme kaydedilememiştir. Araştırmalarda Adaboost ile destek vektör makinesi'nin kullanılamadığı bilgisine ulaşılmıştır. Adaboost ile varsayılan temel öğrenici değeri olan karar ağaçları algoritmasından faydalanılmıştır. Karar ağaçlarının temel öğrenici olarak kullanılmasına karar verilmiştir. Tahmin edici sayısı, öğrenme oranı ve algoritma Adaboost hiper parametrelerinin belirlenen seçenekler arasında neler olacağı greedy yöntem kullanılarak, öznitelik seçiminde ise diferansiyel evrim algoritmasından faydalanılarak belirlenmiştir. Bu çalışmalar sırasında desteğini esirgemeyen Dr. Öğretim Üyesi Berna KİRAZ hocama teşekkürü borç bilirim.

Eylül, 2022 Yahya Bilir

İÇİNDEKİLER

SEMBOLLER

F : Mutasyon oranı

CR : Çaprazlama oranı

w : Uygunluk değeri hesaplama katsayısı

ŞEKİL LİSTESİ

Sayfa

TABLO LİSTESİ

Sayfa

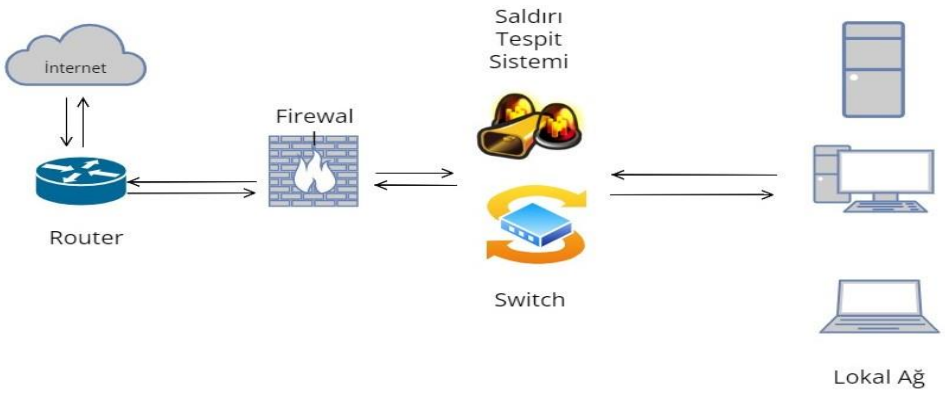
KISALTMALAR

AIDS	Anomali Tabanlı Saldırı Tespit Sistemi (Anomaly Based Intrusion Detection System)
ANN	Yapay Sinir Ağları (Articifial Neural Network)
CFS	Korelasyon Temelli Özellik Seçimi (Correlation Based Feature Selection)
CPU	İşlemci (Central Processing Unit)
DE	Diferansiyel Evrim Algoritması
DDoS	Dağıtık Servis Dışı Bırakma (Distributed Denial of Service)
DoS	Servis Dışı Bırakma (Denial of Service)
DT	Karar Ağaçları (Decision Tree)
DVM	Destek Vektör Makinesi
ELM	Aşırı Öğrenme Makineleri (Extreme Learning Machine)
FTP	Dosya Transfer Protokolü
GE	Genetik Algoritma
ICMP	Internet Kontrol Mesaj Protokolü (Internet Control Message Protocol)
IDS	Intrusion Detection System
IFS	Geliştirilmiş Özellik Seçimi (Improved Feature Selection)
IG	Bilgi Kazancı (Information Gain)
IoT	Nesnelerin İnterneti (Internet of Things)
KNN	K-En Yakın Komşu (K-Nearest Neighbors)
NB	Naive Bayes
OWASP	Open Web Application Security Project
PSO	Parçacıklı Sürü Optimizasyon Algoritması
RAM	Bellek (Random Access Memory)
RF	Rastgele Orman Algoritması (Random Forest)
SIDS	İmza Tabanlı Saldırı Tespit Sistemi (Signature Based Intrusion Detection System)
SSH	Güvenli Kabuk (Secure Shell)
STS	Saldırı Tespit Sistemi
SVM	Destek Vektör Makinesi (Support Vector Machine)

GİRİŞ

Teknolojik gelişmelere bağlı olarak, gerçek dünyadaki işlemlerin çoğu siber dünyada kullanılabilir hale getirilmiştir. Böylece bankacılık, alışveriş, çevrimiçi sınavlar, elektronik ticaret, iletişim gibi birçok işlem bu yeni ortamda yoğun olarak kullanılmaktadır. Akıllı telefonların yaygınlaşmasıyla birlikte insanlar bu küresel ağa bağlanabilmekte ve her an ve her yerden işlem gerçekleştirebilmektedir. Bu dijitalleşme, insanların günlük işlerini kolaylaştırsa da, sunucuların zayıflığı ve yeni ortaya çıkan izinsiz giriş teknikleri nedeniyle, ağlar, yalnızca bazı bilgileri veya parayı çalmak için değil, internetin anonim yapısından yararlanan davetsiz misafirler tarafından ağ hizmetlerinin çalışmasını yavaşlatmak için sıkça saldırıya uğramaktadır.

Güvenlik yöneticileri, ağı korumanın bir yolu olarak güvenlik duvarlarına ek olarak geleneksel olarak parola koruma mekanizmalarını, şifreleme tekniklerini ve erişim kontrollerini tercih ederler. Ancak bu teknikler sistemi korumak için yeterli değildir. Bu nedenle, birçok yönetici, Şekil 1'de gösterildiği gibi, ağ trafiğini izleyerek kötü niyetli saldırıları tespit etmek için Saldırı Tespit Sistemlerinin kullanımını tercih eder.



Şekil 1 Saldırı Tespit Sistemi ve Lokal Ağ Yapısı [37]

İzinsiz giriş, bir bilgi sistemi içindeki verilerin gizliliğine, kullanılabilirliğine veya bütünlüğüne zarar veren her türlü yetkisiz faaliyet olarak tanımlanabilir. Saldırı tespit sistemleri, bu tür aktiviteyi tespit etmek için oldukça tercih edilen bir araçtır. İmza Tabanlı Saldırı Tespit Sistemleri (SIDS), Anomali Tabanlı Saldırı Tespit Sistemleri (AIDS) ve bu ikisinin birleştirildiği hibrit sistemler olarak gruplanabilir.

İmza tabanlı tespit sistemleri, kötü niyetli faaliyetlerin imzalarını bir bilgi veri tabanında saklar ve örüntü eşleştirme tekniklerini kullanarak izinsiz girişleri tespit etmeye çalışır. Anomali tabanlı tespit sistemleri ise, faaliyetlerin normal davranışlarını öğrenmeye çalışır ve diğerlerini şüpheli olarak sınıflandırır. Bu tür bir sistemde imza tabanı kullanmaya gerek yoktur ve sistem daha önce karşılaşılmamış sıfır gün saldırılarını tanımlayabilir. Hibrit sistemler, sıfırıncı gün saldırılarının yanlış pozitif oranını azaltarak bilinen kötü niyetli faaliyetlerin tespit oranını artırmak için SIDS ve AIDS'in entegrasyonundan oluşur.

AIDS'in avantajları nedeniyle, mevcut saldırı tespit sistemlerinin çoğu ya doğrudan bir AIDS kullanır ya da hibrit bir yaklaşım içinde bundan yararlanır. Bu IDS'lerin, veri setini işleyerek makine öğrenme modeli aracılığıyla eğitilmesi gerekir. Bu konudaki çalışmaların çoğu, gereksiz bilgiler ve dengesiz hacimlerde veri türleri içeren eski veri setlerini benimsemiştir. Güncel veriler içeren bazı yeni veri setleriyle karşılaşabilinse de, veri türlerinin dengesiz boyutu araştırmacılar için hala bir zorluktur.

Bu çalışmada saldırı tespit sistemi tasarımı aşamasında en fazla kullanılan veri setlerinden olan NSL-KDD veri setinden faydalanılmıştır. Bu veri setinin tercih edilme nedenlerinden biri de KDD’99 veri setindeki gereksiz kayıtlardan temizlenmiş olmasıdır. NSL-KDD veri seti kullanılarak Adaboost kolektif öğrenme algoritmasından faydalanılarak saldırı tespit sistemi tasarlanması amaçlanmıştır. Adaboost parametrelerinin tercihinde greedy yöntem, veri setine ait öznitelik seçiminde evrimsel algoritma olan diferansiyel evrim algoritmasından faydalanılmıştır.

Bu çalışmada 1. Bölümde saldırı tespit sistemi tasarımında en fazla kullanılan makine öğrenmesi algoritmalarından ve öznitelik seçiminde sıkça faydalanılan

evrimsel tabanlı algoritmalardan bahsedilmiştir. 2. Bölümde saldırı tespit sistemleri geliştirilmesi sırasında yapılan akademik çalışmalar, veri setleri ve atak tiplerinden bahsedilmiştir. 3. Bölümde STS tasarımında diferansiyel evrim algoritmasının kullanımından bahsedilmiştir. 4. Bölümde STS tasarım aşamalarından ve kullanılan NSL-KDD veri seti detaylarından bahsedilmiştir. 5. Bölümde ise yapılan çalışmaların sonuçlarından bahsedilmiştir.

BİRİNCİ BÖLÜM

1. TEMEL KAVRAMLAR

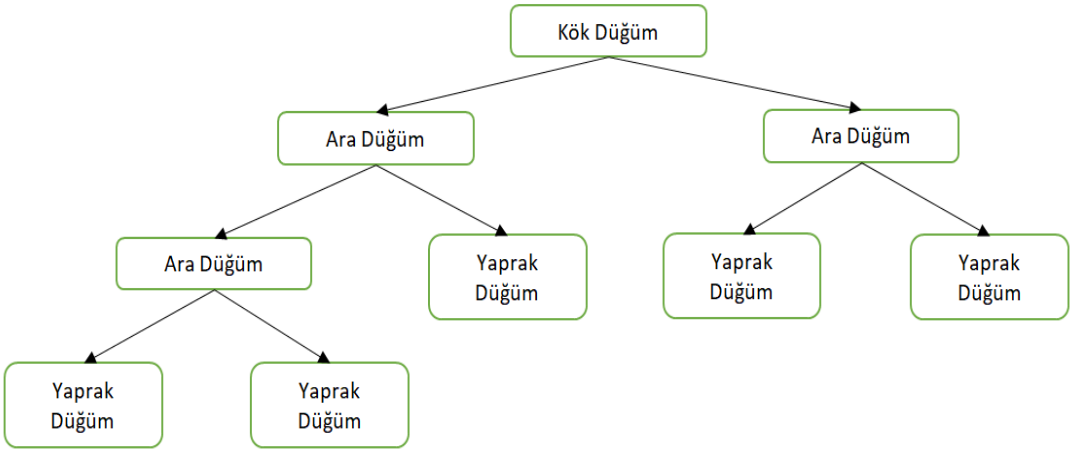
1.1. MAKİNE ÖĞRENMESİ

Makine öğrenmesi, büyük verilerden faydalı veriler çıkarmak amacıyla matematiksel modeller kullanarak makinelerin otomatik öğrenmesini sağlayan tüm yöntem ve algoritmaları içeren yapay zeka türüdür [1,2]. Saldırı tespit sistemi geliştirmeleri kapsamında Karar Ağacı, K-En Yakın Komşu (KNN), Yapay Sinir Ağı (YSA), Destek Vektör Makinesi (SVM), K-Ortalama Kümeleme, Hızlı Öğrenme Ağı ve Kolektif (Ensemble) Öğrenme Yöntemleri gibi makine öğrenmesi tekniklerine sıkça başvurulmaktadır [3].

1.1.1. Karar Ağaçları

Karar ağaçları, verilen verisetinin karar kuralları uygulanarak hem sınıflandırılması hem de regrasyonu için kullanılan makine öğrenmesi algoritmasıdır. Model, düğümler, dallar ve yapraktan oluşan geleneksel bir ağaç yapısına sahiptir [4]. Her düğüm bir özniteliği veya bir özelliği temsil eder. Dal bir kararı veya kuralı temsil ederken, her yaprak olası bir sonucu veya sınıf etiketini temsil eder [5]. Karar ağacı algoritması, bir ağaç oluşturmak için en iyi özellikleri otomatik olarak seçer ve daha sonra aşırı sığmayı önlemek için ağaçtan alakasız dalları kaldırmak için budama işlemini gerçekleştirir.

Karar ağaçlarının avantajlarından bazıları, görselleştirilebilmesi, tahmin sürecinin yorumlanmasının nispeten kolay olması, çok çıktılı görevleri çözme yeteneğine sahip olması ve kapsamlı veri temizliği gerektirmemesidir [6].
En yaygın karar ağacı modelleri CART, C4.5 ve ID3.58'dir. Random Forest (RF) ve XGBoost gibi birçok gelişmiş öğrenme algoritması, birden çok karar ağacı kullanılarak geliştirilir [3].



Şekil 1.1 Karar Ağacı [7]

1.1.2. K-En Yakın Komşu

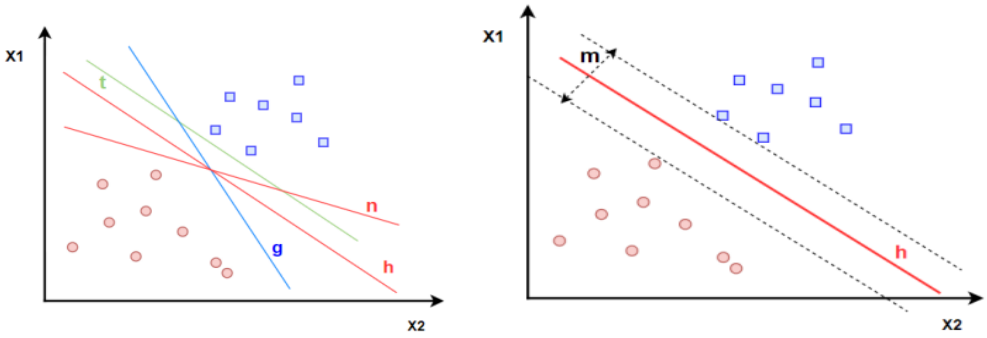
K-en yakın komşu algoritması, belirli bir veri örneğinin sınıfını tahmin etmek için özellik benzerliği fikrini kullanan en basit denetimli makine öğrenimi algoritmalarından biridir. Komşulara olan mesafesini hesaplayarak komşularına göre bir örnek tanımlar. KNN algoritmasında K parametresi modelin performansını etkiler. K değeri çok küçükse, model aşırı öğrenmeye yatkın olabilir. Çok büyük bir K değeri seçimi, yanlış sınıflandırılmasına neden olabilir [3].

1.1.3. Destek Vektör Makinesi

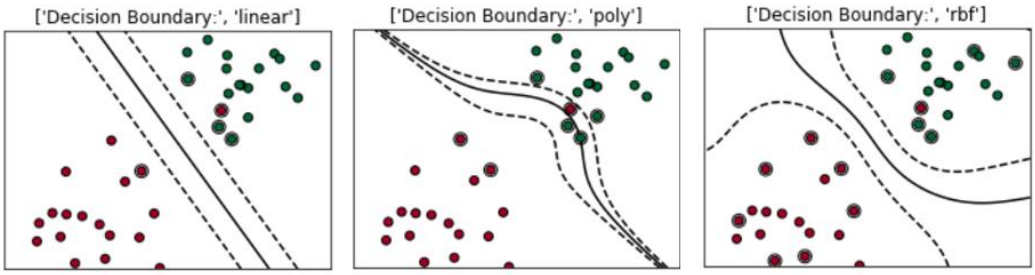
Destek Vektör Makinesi, n boyutlu özellik uzayında maksimum marj ayırma hiper düzlemi fikrine dayanan denetimli bir makine öğrenmesi algoritmasıdır. Hem doğrusal hem de doğrusal olmayan problemlerin çözümü için kullanılır. Doğrusal olmayan problemler için çekirdek fonksiyonları kullanılır. Buradaki fikir, önce düşük boyutlu bir girdi vektörünü, çekirdek işlevini kullanarak yüksek boyutlu bir özellik uzayına eşlemektir. Ardından, destek vektörlerini kullanarak bir karar sınırı olarak çalışan optimal bir maksimum marjinal hiper düzlem elde edilir [8,9]. NIDS için, normal ve kötü niyetli sınıfları doğru bir şekilde tahmin ederek verimliliğini ve doğruluğunu artırmak için destek vektör algoritması kullanılabilir [10,11].

Doğrusal olmayan destek vektör, veri setinin doğrusal bir fonksiyonla tam veya belirli bir hata ile ayrılamaması durumunda kullanılan algoritmalardır. Gerçek yaşam problemlerinde bir veri setinin hiper düzlem ile doğrusal olarak ayrılması

çoğunlukla mümkün değildir. Dolayısıyla sınıfları ayırma işlemi, ayırma eğrisinin tahmin edilmesiyle mümkün olmaktadır. Ancak uygulamada eğrinin tahmin edilmesi oldukça zordur [12].



Şekil 1.2 Çoklu ve Optimal Hiper Düzlem [12]



Şekil 1.3 Çekirdek Türleri: Doğrusal, Polinom Çekirdek, RBF(Radyal Temelli Fonksiyon) [13]

1.1.4. Kolektif Öğrenme Algoritmaları

Topluluk yöntemlerinin arkasındaki ana fikir, topluluk yoluyla öğrenerek farklı sınıflandırıcılardan faydalanmaktır. Çünkü her sınıflandırıcının bazı güçlü ve zayıf yönleri vardır. Bazıları belirli bir saldırı türünü tespit etmede iyi performans gösterebilir ve diğer saldırı türlerinde düşük performans gösterir. Topluluk yaklaşımı, birden çok sınıflandırıcıyı eğiterek zayıf sınıflandırıcıları birleştirmek ve ardından bir oylama algoritması kullanarak seçerek daha güçlü bir sınıflandırıcı oluşturmaktır [3].

Shen ve arkadaşları [14], temel sınıflandırıcı olarak ELM'yi seçerek bir topluluk yöntemi kullanan bir IDS önerdi. Önerilen metodolojiyi optimize etmek için, topluluk budama aşaması sırasında bir BAT optimizasyon algoritması kullanılır. Model, KDD Cup'99, NSL-KDD ve Kyoto veri setleri kullanılarak test edilmiştir.

Deneysel sonuçlar, topluluk tarzında birleştirilen birçok ELM'nin performansta bireysel ELM'den daha iyi performans gösterdiğini gösterdi.

Gao ve arkadaşları [15], DT, RF, KNN, Derin Sinir Ağı (DNN) gibi birkaç temel sınıflandırıcı kullanarak ve adaptif oylama algoritmasını kullanarak en iyisini seçerek bir uyarlamalı topluluk modeli önerdi. Önerilen metodoloji, NSL-KDD veri seti kullanılarak deneyler yapılarak doğrulandı. Deneysel sonuçlar, diğer modellerle karşılaştırılarak performansın verimliliği gösterildi.

Toplu öğrenmenin en popüler üç kombinasyon şeması, torbalama (bagging), yükseltme (boosting), regresyon için ağırlıklı ortalama şeklinde kümeleme (stacking) ve sınıflandırma problemleri için çoğunluk oylamasıdır [16]. Aşağıda bunlardan kısaca bahsedilecektir.

1.1.4.1. Torbalama Yöntemi (Bagging)

Torbalama, en basit ancak etkili ve sağlam topluluk tekniklerinden biridir [17]. Torbalama prosedürü, önyükleme ve toplama kavramı üzerinde çalışır. Seçilen zayıf öğrenenler için iyi dengelenmiş bir eğitim veri setini garanti etmek için orijinal N eğitim veri setinden değiştirilen N örneğinin birden çok sürümünü oluşturmak için önyükleme örnekleme yöntemini kullanır. Benzer şekilde, temel öğrenicileri bir araya getirerek ve çeşitli zayıf öğrenicilerin sınıflandırma sonuçları arasında en çok meydana gelen olanı nihai sınıflandırma olarak seçmek için çoğunluk oylamasını kullanarak optimize edilmiş, verimli ve etkili bir nihai model elde edilir ve böylece geleneksel ile karşılaştırıldığında performansın artması sağlanır. Ayrıca, torbalama tekniği, yüksek boyutlu bir yapıya sahip IDS veri kümelerindeki çok sayıda örnekle ilişkili zorlukları önemli ölçüde azaltmıştır [18]. Referans [19], algoritma ve diğer ilgili bilgiler gibi torbalama yaklaşımının kapsamlı ayrıntılarını içerir.

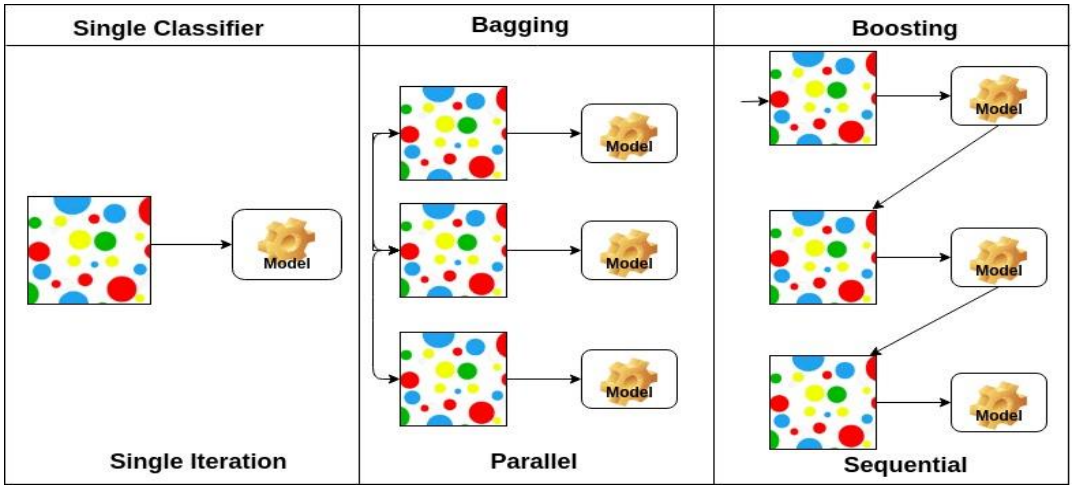
1.1.4.2. Yığın Yöntemi (Stacking)

Bagging ve bossting'in aksine yığınlama, rastgele orman, sinir ağları, lojistik regresyon ve destek vektör makineleri gibi çeşitli heterojen zayıf öğrenicileri iki katmanlı bir prosedürde toplamaya yönelik güvenilir ancak basit bir topluluk yöntemidir. Yığın yöntemine dayalı öğrenici seviyesi, çeşitli zayıf öğrenicilerin bir

k-katlı eğitim veri setinden öğrenmesini ve sınıflandırmalar için N sayıda sınıflandırıcı oluşturmasını sağlar [20].

1.1.4.3. Yükseltme Yöntemi (Boosting)

Boosting, eğitim veri setinin tamamından oluşturulan ilk zayıf öğrenicinin daha iyi sonuçlar için sonraki zayıf öğrenicileri tamamladığı sıralı bir grup yöntemidir. Temel fikir, doğru sınıflandırılmış örneklere küçük ağırlıklar ve yanlış sınıflandırılmış örneklere büyük ağırlıklar vermek gibi sınıflandırma sonuçlarına dayalı olarak eğitim setlerindeki örneklere ağırlıklar atamaktır [17]. Ardından, aşağıdaki zayıf sınıflandırıcılar, geliştirilmiş bir nihai model için önceki zayıf öğrencinin çıktısına dayalı olarak eğitilir. Bu ardışık prosedür, nihai geliştirilmiş model elde edilene kadar devam eder. Son olarak, ilgili zayıf öğrenenlerin parametreleri, önceki model tarafından eğitim veri setine dayatılan kayıp fonksiyonu azaltılarak belirlenir [18]. İki adet boosting algoritmasından bahsedilebilir: adaptif boosting(Adaboost) ve gradyan(gradient) boosting.



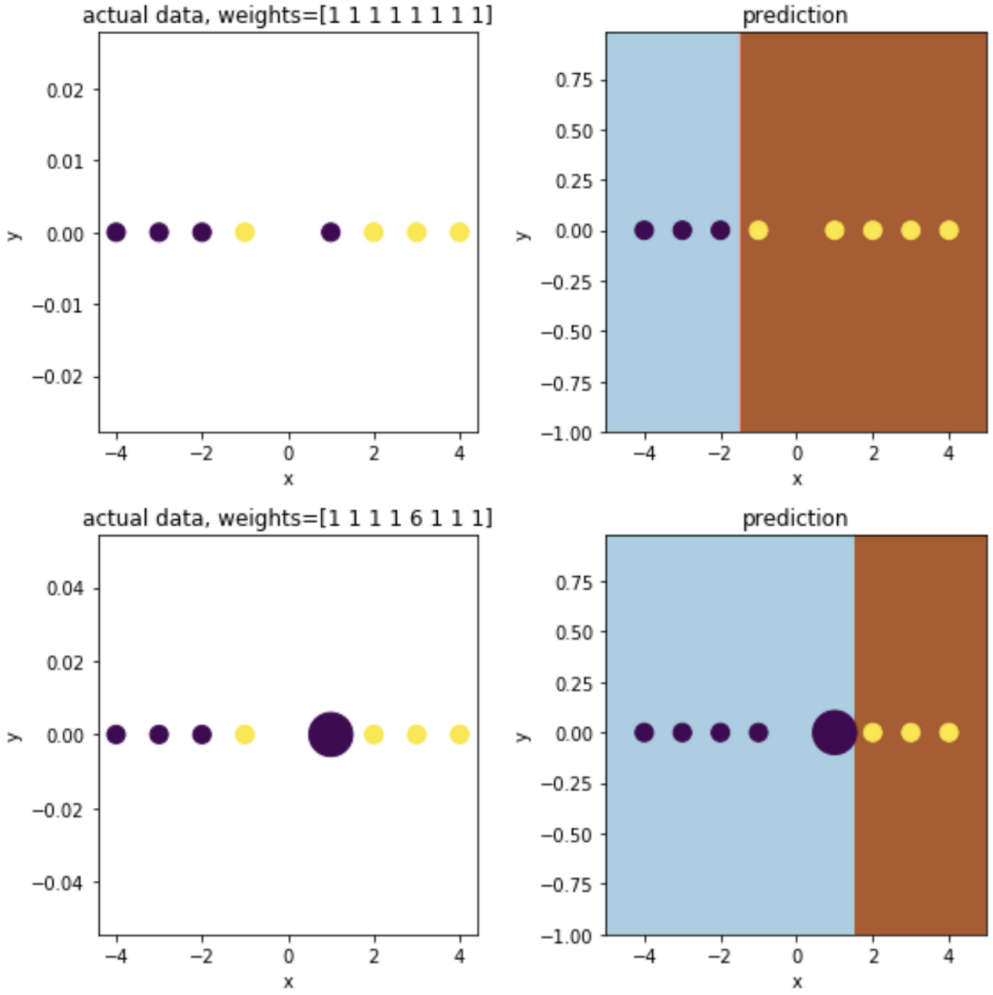
Şekil 1.4 Tek Sınıflandırıcı, Torbalama, Yükseltme Yöntemler [21]

1.1.4.3.1. Adaboost

Adaboost, sınıflandırma problemleri için geliştirilmiş özel bir boosting algoritmasıdır. Zayıflık, zayıf tahmincinin hata oranı ile tanımlanır:

$$for\ a\ weak\ classifier\ C$$
$$X: n \times d, Y: n \times k\ with\ sample\ weights\ W: n \times 1$$
$$Error = \frac{\sum_{j=1}^{n} W_j I(C(X_j) \neq Y_j)}{\sum_{j=1}^{n} w_j}, I(x) = \begin{cases} 1, & if\ x\ is\ True \\ 0, & if\ x\ is\ False \end{cases}$$

Her yinelemede Adaboost, yanlış sınıflandırılmış veri noktalarını tanımlar, yanlış olanların ağırlıklarını arttırır ve doğru noktaların ağırlıklarını azaltır, böylece bir sonraki sınıflandırıcı onları doğru yapmak için dikkat eder. Aşağıdaki şekil, ağırlıkların veriyi nasıl etkilediğini gösterir.



Şekil 1.5 Numune ağırlıkların karar sınırına etkisi [22]

Adaboost, artırılmış örnek ağırlıkları ile bir dizi modeli eğiterek, hatalara dayalı bireysel sınıflandırıcılar için alfa "güven" katsayıları üretir. Düşük hatalar, oylamada daha yüksek önem anlamına gelen büyük alfa'ya yol açar.

Adaboost için aşağıdaki parametrelerden bahsedilebilir.

base_estimator : obje, isteğe bağlı (varsayılan=None)

Güçlendirilmiş topluluğun oluşturulduğu temel tahmin edici. Eğer başka bir değer verilmediyse, temel tahminci DecisionTreeClassifier(max_depth=1) şeklindedir.

n_estimators : tamsayı, isteğe bağlı (varsayılan=50)

Yükseltmenin sonlandırıldığı maksimum tahmin edici sayısı. Mükemmel uyum durumunda, öğrenme prosedürü erken durdurulur.

learning_rate : float, isteğe bağlı (varsayılan=1.)

Öğrenme oranı, öğrenme hızı ile her sınıflandırıcının katkısını küçültür.

random_state : int, RandomState değeri veya None, isteğe bağlı (varsayılan=None)

1.1.4.3.2. Gradyan Boosting

Gradyan artırma, soruna biraz farklı yaklaşır. gradyan güçlendirme, veri noktalarının ağırlıklarını ayarlamak yerine, tahmin ile temel gerçek arasındaki farka odaklanır.

1.2. EVRİMSEL ALGORİTMALAR

1.2.1. Genetik Algoritma (GE)

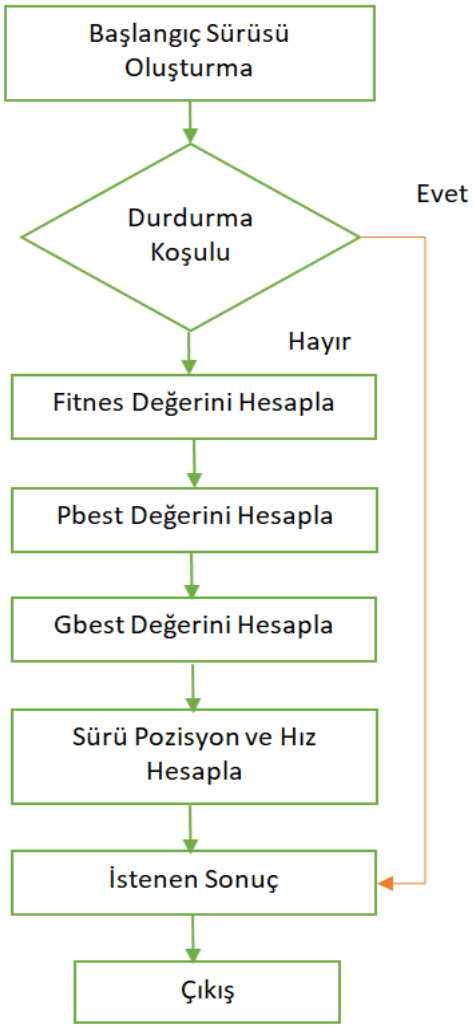
Genetik Algoritmalar, doğal seçilim ve genetiğin evrimsel fikirlerine dayanan uyarlanabilir buluşsal arama algoritmalarıdır [23]. Evrimsel tabanlı algoritmaların bir parçasını oluşturur. Ayrıca, genetik algoritmalar sezgisel optimizasyon teknikleridir. Başka bir deyişle, rastgele arama yöntemleridir. Bu nedenle, optimizasyon problemlerini çözmek için kullanılan rastgele bir aramanın akıllı bir şekilde kullanılmasını temsil ederler. Biyolojik doğal seçilim sürecine benzer şekilde, GA'lar popülasyon olarak adlandırılan bir başlangıç kümesinden rastgele bireyleri seçer. Bu

bireyler, yeni nesle ait yeni bir dizi bireyi çoğaltacak ebeveynler olarak kabul edilir [r22]. Genetik algoritmaların temel adımları aşağıdaki gibidir.

1. Rastgele çözümlerden oluşan başlangıç popülasyonu oluşturulması.
2. Herbir popülasyon bireyi için performans değerlendirilmesi yapılır.
3. Aşağıdaki adımlar tekrarlanarak yeni popülasyon oluşturulur.
    a. Seçim: Uygunluk (fitnes) değerlerine göre iki tane ebeveyn seçilir.
    b. Çaprazlama: Çaprazlama oranı göz önünde bulundurularak iki ebeveyn kullanılarak yavru oluşturulur.
    c. Mutasyon: Mutasyon oranına göre kromozom mutasyona uğratılır.
4. Yeni popülasyona yavrular yerleştirilir.
5. Algoritmanın bir sonraki iterasyonda çalışmasında yeni popülasyon kullanılır.
6. Koşullar sağlanırsa işlem durdurulur ve en iyi çözüm olarak belirlenir.
   Tekrar 2. Adımdan itibaren adımlar çalıştırılır.

1.2.2. Parçacıklı Sürü Optimizasyonu (PSO)

1995 yılında R Eberhart ve J Kennedy tarafından önerilmiştir [24]. Çeşitli mühendislik akışlarında uygulanan PSO algoritması kuş sürüsünün davranışından türetilmiştir. Balıklar ve kuşlar gibi sosyal hayvanların hızla değişen etkileşimlerinin ve hareketlerinin uyarlanmasını içerir. PSO algoritmasının bir akış diyagramı şekil 1.6'da sunulmaktadır. PSO algoritması, grup olarak birlikte çalışırken öğrenilen deneyimleri ve kişisel deneyimleri birleştirir. Optimize edilmiş çözümler, kuşların sürü davranışı ile elde edilir. Kuşlar, hedeflenen besin kaynakları için önceden belirlenmiş bir yol izlerler. En kısa yol olarak kabul edilen bu yol, parçacığın kişisel en iyi çözümü (pbest) olarak da adlandırılır. Her parçacık, kendi uçuş deneyimlerini ve gruptaki diğerlerinin deneyimlerini gözlemleyerek arama uzayında en iyi çözümü arar. Bir başka en iyi uygunluk değeri, gruptaki parçacıklardan herhangi birinin o parçacığın aralığına yakın gözlenmesiyle elde edilir. Buna gbest denir. Her parçacığın pbest ve gbest'e ulaşmaya yönelik ivme için ilişkili hızı vardır. PSO'nun temel konsepti, global optimal çözüme ulaşmak, böylece her bir parçacığı her adımda keyfi ağırlıkla pbest ve gbest'e doğru hareket ettirmektir. Bu algoritma, gelişmiş keşif ve kullanım sağlar.



Şekil 1.6 PSO Akış Diyagramı

1.2.3. Diferansiyel Evrim Algoritması

Diferansiyel evrim algoritması (DE) ilk olarak Price ve Storn [25] tarafından önerilmiştir. DE, karmaşık doğrusal olmayan problemler için yeni bir optimizasyon tekniği olarak potansiyeli konusunda önemli bir ilgi gördü ve çeşitli alanlarda başarıyla kullanıldı [26,27]. DE üç önemli işlemi kullanır: mutasyon, çaprazlama ve seçim. Rastgele oluşturulmuş ilk popülasyondan nihai bireysel çözüme evrimleşmek için bu üç operatörü kullanır [26]. Deneme vektörlerini oluşturmak için mutasyon ve çaprazlama kullanılır ve daha sonra yeni oluşturulan vektörlerin bir sonraki nesilde

hayatta kalıp kalamayacağını belirlemek için seçim kullanılır [28]. DE'nin prosedürü aşağıdaki gibi çalışır:

1. Adım: İlk algoritma parametreleri

Bunlar: ölçek faktörü F, çaprazlama oranı CR, popülasyon büyüklüğü M ve maksimum yineleme sayısı K.

2. Adım: Rastgele M aday çözümler üretilir

İlk aday çözümler, bir üniformadan üretilir. $[x^j_L, x^j_U]$ (j = 1, 2,..., N) aralığındaki dağılım, burada N, değişkenlerin sayısıdır.

3. Adım: Mutasyon

Mutasyon operatörü yalnızca çözüm vektörlerinin çeşitliliğini artırmakla kalmaz, aynı zamanda DE algoritması için çözüm uzayının keşif kabiliyetini de geliştirir. Her ebeveyn için, $\boldsymbol{x_i^k}$ (i =1, 2,. . ., M) neslinin k (k = 1, 2,. ., K), bir deneme vektörü, $\boldsymbol{x_i^{k+1}}$, bir hedef vektörün mutasyona uğratılmasıyla oluşturulur. Mutasyon operatörüne göre, deneme vektörü aşağıdaki denklem kullanılarak hesaplanır:

$$v^{k+1}_i = x^k_{is} + F(x^k_{i1} - x^k_{i2})$$

4. Adım: Çaprazlama

DE, öğelerin bulunduğu ayrık bir yeniden birleştirme yaklaşımını izler. $\boldsymbol{x_i^k}$ ebeveyn vektöründen gelen elemanlar, yavruları üretmek için $\boldsymbol{v_i^k}$ deneme vektörünün elemanları ile birleştirilir ve yavru $\boldsymbol{u_i^k}$ oluşturulur.

$$\boldsymbol{u_{i,j}^{k+1}} = \begin{cases} \boldsymbol{v_{i,j}^{k+1}}, & \boldsymbol{eğer\ r\ ve < Cr\ yada\ j = r_{1\sim D}} \\ \boldsymbol{x_{i,j}^k}, & \boldsymbol{aksi\ halde} \end{cases}$$

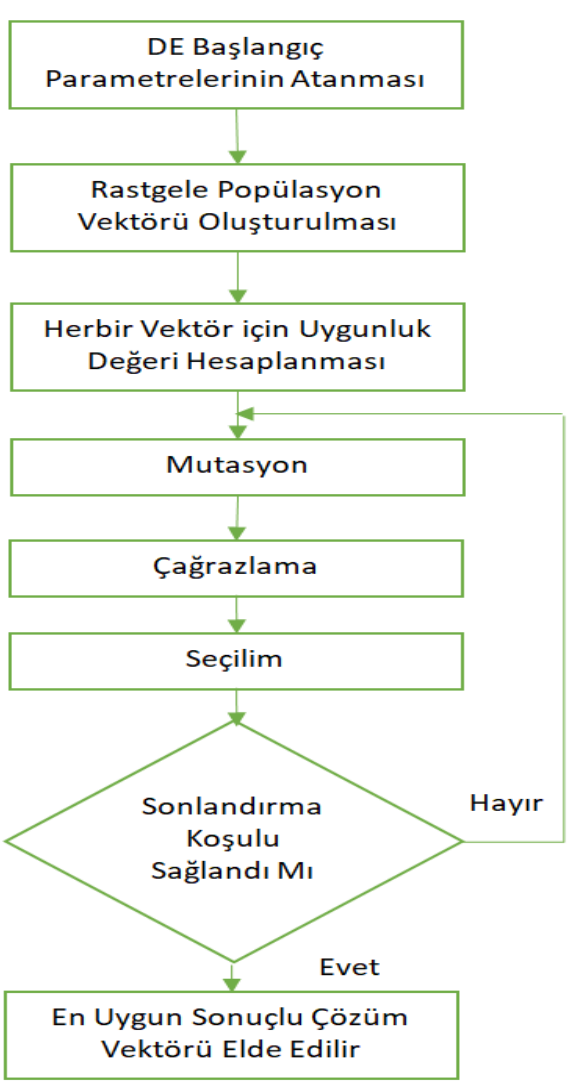
burada, $r_{1\sim D}$, [1, D] içinde rastgele bir tamsayıdır. Cr, geçiş oranını temsil eder ve $Cr \in [0,1]$.

5. Adım: Seçim

Üretilen yavru $\boldsymbol{u_i^{k+1}}$, yalnızca yavrunun uygunluğu ebeveyninkinden daha iyiyse, $\boldsymbol{x_i^k}$ ebeveyninin yerini alır.

6. Adım: Durdurma kriterini kontrol edin.

Durdurma kriteri (maksimum yineleme sayısı K) karşılanırsa, hesaplama sonlandırılır. Aksi takdirde, Adım 3-5 tekrarlanır. Durum çalışması için özellikle F = 0.5, CR = 0.7 olmak üzere ortak bir parametre değerleri kümesinin kullanılmasına karar verilmiştir.



Şekil 1.7 DE Akış Diyagramı

İKİNCİ BÖLÜM

2. SALDIRI TESPİT SİSTEMLERİ (STS/IDS)

Saldırı tespit sistemleri, saldırı, ihlal ve algılama sistemi kelimelerinin bir araya gelmesinden oluşan bir tanımdır. Bilgisayar ve ağ sistemlerinde bütünlüğü, gizliliği, erişilebilirliği tehlikeye atan davranışlara ihlal denir [3]. Saldırı tespit sistemleri de bu tür sistem güvenliğini tehlikeye atan veya şüpheli tüm davranışları tespit etmek üzere bilgisayar ve ağ sistemlerini sürekli izlemek üzere geliştirilen yapılardır. Bilgisayar ve ağ sistemindeki davranışlara göre uyarılar üretilmesini sağlayacaktır.

2.1. İLGİLİ ÇALIŞMALAR

Saldırı tespit sistemi, sistemin normal veya anormal davranışını öğrenmek ve yeni trafiği sınıflandırmaya yardımcı olan modeller oluşturmak için makine öğrenimi algoritmaları veya sınıflandırıcılar kullanır. Optimal bir makine öğrenimi tabanlı algılama sistemi geliştirmek, araştırmayı tek bir makine öğrenimi algoritmasının veya birden çok algoritmanın performansını tek bir saldırı kategorisi yerine dört ana saldırı kategorisinin tümüne incelemeye yönlendirir. Bu dosyada araştırmacılar tarafından kullanılan bazı algoritma ve yöntemlerden bahsedilecektir [29].

[30]'daki çalışmada, algoritmanın önce veri kümelerinden rastgele verileri seçmesi, ardından özellik seçimi için kullanılan veri ön işlemesinin yapılması önerilmektedir. Daha sonra Destek Vektör Makinesi, gereksiz veya fazla kayıtları normal veya saldırı düğümünde sınıflandırarak kaldırmak için kullanılır. Son olarak, verileri sınıflandıran ve doğruluğu, yanlış alarmları, kesinliği ve geri çağırmayı hesaplayan makine öğrenimi tekniği uygulanmıştır.

[31]'de, bu makalenin yazarı, algoritmayı daha etkili hale getirmek için veri paketlerinin özellik seçimi ve ağırlıklandırılması için gözetimli öğrenme K-NN ve gözetimsiz öğrenme K-ortalama kümeleme makine öğrenme algoritmasını önerdi.

Ayrıca, önerilen yöntemin, izinsiz giriş tespitinde bir zorluk olarak kabul edilen U2R sınıflandırmasının performansını artıran tüm saldırı kategorilerini tespit etmek için karşılaştırılabilir bir performans gösterdiğini belirtti.

[32]'te araştırmacı, önce etkili bir veri seti ön işleme prosedürü sağlayarak ağdaki anormallikleri verimli bir şekilde tanımlamak için geliştirilmiş özellik seçimine sahip bir (GA-IFS) genetik algoritma önerdi. Veri kümesi yeniden işleme tekniği ile eğitim verileri için %79.07 ve test verileri için %80.47'lik bir azalma oranı elde etmeyi başardı.

[33]'da yazar, sürekli değerli özniteliklerin karar sürecini sadeleştirmek için öznitelik ayrıklaştırma işlemi önermiştir. İzinsiz giriş tespit doğruluğu ve tespit hızı üzerindeki etkilerini analiz etmek için kullanılan birkaç makine öğrenme algoritması ve özellik seçme teknikleri vardır. Elde ettiği sonuç, bu algoritmada özellik seçimi kullanımının testin hızını artırmayı başardığını, ancak doğruluğu biraz azaltacağını göstermiştir.

[34]'de yazar, farklı saldırı türlerinin tespiti için J48 Karar Ağacı, Destek Vektör makinesi ve Naive Bayes gibi bazı teknikleri birleştiren ve ayrıca algoritmalara göre farklı doğruluk türleri içeren hibrit yaklaşımlar önermiştir. Tüm bu testler NSL-KDD veri seti üzerinde gerçekleştirilmiştir.

Bu yazıda [35], zorlu CIC IDS 2017 veri seti için Adaboost tabanlı saldırı tespit sisteminin performansını en son ve en son sürümlerde iyileştirmek için Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE), Temel Bileşen Analizi (PCA) ve Topluluk Özellik Seçimi (EFS) kullanımını ele alınmıştır. Önerilen yöntem %81,83 doğruluk, %81,83 kesinlik, %100 geri çağırma ve %90,01 F1 Skoru ile iyi performans göstermiştir.

Bu çalışmada [36], yüksek hızlı ağlarda probe ve DoS saldırılarının tespiti için çözüm önerilmiştir. Model, doğruluk ve verimliliği artırırken KDD'99'da kullanılan 41 özellik arasından en önemli özelliklerin seçilmesinden oluşuyor. İki filtre (IG ve CFS) ve dört paketleyici (NB, C4.5, RF ve REPTree) dahil olmak üzere altı özellik seçim yöntemi kullanılır. Sistem, KDD'99'un yeniden örneklenmiş bir

sürümü kullanılarak değerlendirilir. Sonuçlar, C4.5 kullanan DoS saldırıları için yaklaşık %99,6 ve %0,3 ve NB kullanan Araştırma saldırıları için %99,8 ve %2,7'lik iyi tespit ve yanlış pozitif(FP) oranlar göstermektedir. En iyi seçilmiş özellik alt kümesi kullanılarak değerlendirildiğinde işleme süresi kaydedilmiş. Bu nedenle, önerilen özellik alt kümesinin, prob tespiti için 19 özellik ve DoS tespiti için sadece 9 özellik ile yüksek hızlı ağlarda kullanılması önerilmiştir.

Bu yazıda altı farklı makine öğrenme modeli (Karar Ağaçları, Random Forest, K-en Yakın Komşu, Adaboost, Gradient Boost ve Lineer Diskriminant Analizi) güncel bir veri seti (CSE-CIC-IDS2018) kullanılarak uygulanmıştır. Dengesizlik oranını azaltmak için azınlık gruplarının veri büyüklüğü artırılarak bir veri örnekleme modeli kullanılmıştır. Deneysel sonuçlar, uygulanan modellerin son literatür çalışmaları ile karşılaştırıldığında çok iyi bir doğruluk düzeyine sahip olduğunu göstermiştir. Örneklenmiş bir veri kümesinin kullanılması, modellerin ortalama doğruluğunun %4.01 ile %30.59 arasında artmasına neden olmuştur.

2019 yılında Gao ve arkadaşları [37] uyarlanabilir bir topluluk öğrenme modeli [38] kullanarak saldırı tespit sistemi geliştirmek ve test etmek için NSL-KDD veri setini kullandı. Dört farklı algoritma kullandılar; Karar Ağacı, Rastgele Orman, K-En Yakın Komşu ve Derin Sinir Ağları. Ayrıca, kolektif uyarlamalı oylama algoritması tasarladılar. Yaklaşımlarını doğrulamak için bir NSL-KDD-Test+ dosyası kullandılar. Karar Ağacı algoritmasının doğruluğu %84,2 ve adaptif algoritmanın nihai doğruluğu %85,2'dir. Sonunda, ilgili araştırma makalelerini karşılaştırdılar ve topluluk uyarlamalı modellerinin algılama doğruluğunu iyileştirdiğini buldular.

2019 yılında Taher ve arkadaşları ağ trafiğini sınıflandırmak için denetimli bir makine öğrenme sistemi önerdi[38]. Trafiğin kötü amaçlı mı yoksa normal mi olduğunu tespit etmek istedikleri için test ve eğitim için NSL-KDD veri setini kullandılar. Bu amaçla Destek Vektör Makinesi (SVM) ve Yapay Sinir Ağı (YSA) algoritmalarını ve öznitelik seçim yöntemlerini kullanmışlardır. Özellik seçimine sahip YSA'nın SVM'den daha iyi performans gösterdiğini buldular.

Hajisalem ve arkadaşları [39] çalışmalarında Yapay Arı Kolonisi (ABC) ve Yapay Balık Sürüsü (AFS) kullanarak hibrit bir sınıflandırma yöntemi geliştirmişlerdir. Fuzzy C-Means Clustering (FCM) ve korelasyon tabanlı öznitelik seçimi (CFS) teknikleri ile öznitelik seçimi yaptılar. Son adımda ise CART tekniği ile normal ve anomali kayıtlarını ayırt etmek için If-Then kuralları oluşturmuşlardır. Bu yöntem ile geliştirilen model NSL-KDD ve UNSW-NB15 veri setlerine uygulanmış ve %99 doğruluk oranı elde edilmiştir.

Kanimozhi ve arkadaşları [40], yapay sinir ağları, RF, K-en yakın komşuluk, DVM, Adaboost, NB makine öğrenimi yöntemlerini kullanarak CSE-CIC-IDS 2018 veri setini sınıflandırdı.

Khammassi ve arkadaşları [41], Domine Edilmeyen Sıralama Genetik Algoritması II (NSGA-II) ve lojistik regresyona dayalı bir öznitelik seçim yöntemi önerdi. Önerilen yaklaşım, Baskın Olmayan Sıralamalı Genetik Algoritma Binomsal Lojistik Regresyon (NSGA2-BLR) ve Baskın Olmayan Sıralamalı Genetik Algoritma Çok Terimli Lojistik Regresyon (NSGA2-MLR) yöntemlerine göre test edilmiştir. Elde edilen en iyi alt kümeler C4.5, Random Forest (RF), Naive Bayes (NB) yöntemleri ile sınıflandırılmıştır. Çalışmada NSL-KDD, UNSW-NB15 ve CIC-IDS2017 veri setleri kullanılmıştır.

Yüzük ve arkadaşları [42], saldırı tespit sistemlerinin kapsamlı bir incelemesini yaptı. Çalışmada ağ tabanlı saldırı tespit sistemlerinin veri formatları analiz edilmiştir. Ayrıca veri setlerinin uygunluğunu değerlendirmek için 15 özellik tanımlanmıştır. Ayrıca bu özellikler 5 grupta toplanmıştır: Genel Bilgiler, Verilerin Kalitesi, Veri Hacmi, Kayıt Ortamı ve Değerlendirme.

Başka bir çalışmada Kanimozhi ve arkadaşları [40], Yapay Zeka (AI) kullanarak CSE-CIC-IDS-2018 veri setini sınıflandırdı. Sınıflandırma sonucunda %99.97 başarı sağlanmıştır.

Mustafa ve diğerleri [43], modern ağ trafiğini daha iyi simüle etmek için bir test ortamı kurmuştur. Kurduğu test ortamından UNSW-NB15 veri setini üretti. Veri seti oluşturulurken IXIA Perfect-Storm, Tcpdump, Argus ve Bro-IDS araçları

kullanılmıştır. Test ortamında dokuz güncel saldırı senaryosu oluşturuldu, Fuzzers, Analysis, IXIA aracını kullanarak Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms.

Mustafa ve arkadaşları [44] başka bir çalışmada UNSW-NB15 veri setinin karmaşıklığını inceledi. Bu amaçla ilk adımda yeterliliklerin istatistiksel analizi anlatılmaktadır. İkinci adımda, özellik korelasyonları incelenir. Son aşamada ise performans beş adet sınıf içeren veri seti ile ölçülmüş ve KDD99 veri seti ile karşılaştırılmıştır. Sonuç olarak, UNSW-NB15'in KDD99'dan daha karmaşık olduğu gözlemlendi.

Mustafa ve arkadaşları [44], özellikler ve gözlemler arasındaki mesafeler için Beta Karışım Modeli (BMM) parametrelerinden hesaplanan Trapez Alan Tahmini (TAE) tahminine dayalı Geometrik Alan Analizi (GAA) tekniğini sunmuşlardır. Bu yöntem NSL-KDD ve UNSW-NB15 veri kümelerinde test edilmiştir.

Zhang ve arkadaşları [45] Çok Ölçekli Evrişimli Sinir Ağı'nı (MSCNN) Uzun Kısa Süreli Bellek (LSTM) ile birleştiren birleşik bir yöntem önerdi. Yöntemin ilk aşamasında, veri setinin uzamsal özelliklerini analiz etmek için MSCNN kullanılmıştır. Yöntemin ikinci aşamasında, geçici öznitelikleri işlemek için LSTM ağı kullanılmıştır. Modelin eğitimi ve test edilmesi için UNSW-NB15 veri seti kullanılmıştır. Yöntem, geleneksel sinir ağlarına dayalı modellerden daha iyi doğruluk, yanlış alarm oranı ve yanlış negatif hıza sahiptir.

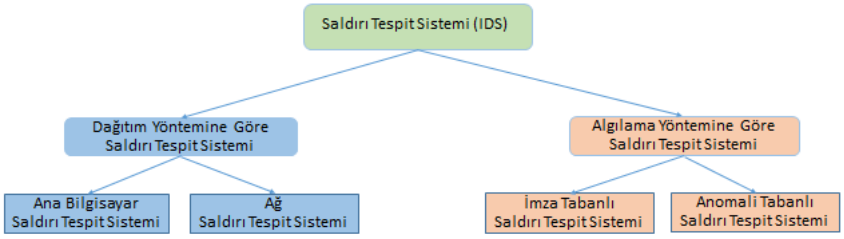
Gottwalt ve arkadaşları [46], çok değişkenli korelasyona dayalı öznitelik seçim yöntemi, çok değişkenli korelasyona dayalı ağ anomali tespiti için önerilmiştir. Yöntem, UNSW-NB15 ve NSL-KDD veri kümelerinde test edilmiştir.

[47]'in yazarları, yüksek yanlış alarm oranlarını azaltmak ve referans [48]'de önerilen sistemin algılama oranını artırmak için istifleme, torbalama ve artırma gibi toplu yöntemler kullanır. Önerilen topluluk yönteminde yapay sinir ağları, karar ağaçları, Naive Bayes, kural tümevarımı, k-en yakın komşu ve genetik algoritmalar kullanılmıştır. Bilinen saldırıları tespit etmede %99 doğruluk elde ettiklerini iddia

ettiler. Ancak, uzun yürütme süresine sahip yeni saldırılar için yalnızca %60 algılama doğruluğu elde edilebilmiştir.

2.2. SALDIRI TESPİT SİSTEMİ (STS) SINIFLANDIRILMASI

Konumlandırma ve algılama yöntemleri açısından aşağıdaki resimdeki gibi sınıflandırılabilir.



Şekil 2.1 Saldırı Tespit Sistemi Sınıflandırılması

2.2.1. Dağıtım Yöntemine Göre STS

Bulunduğu ana bilgisayar üzerindeki tüm aktiviteleri monitör etmekle görevlidir. Güvenlik politikasını ihlal eden veya şüpheli aktiviteleri tespit ederek uyarı vermektedir. Tüm ana bilgisayarlar için ayrı ayrı deploy edilmek zorunda olması dezavantajı olarak değerlendirilmektedir [3]. Bu durum her bilgisayar için işlem yüküne neden olmakta ve performansı düşürmektedir.

2.2.2. Algılama Yöntemine Göre STS

Algılama yöntemine göre saldırı tespit sistemleri "imza tabanlı saldırı tespit sistemi (SIDS)" ve "anomali tabanlı saldırı tespit sistemi (AIDS)" olarak sınıflanabilir. Bilgiye dayalı saldırı tespit sistemi olarak da bilinen imza tabanlı saldırı sisteminde, saldırılar için imza tanımı yapılması fikrine dayanmaktadır. Bu imzalar veri tabanında saklanmaktadır. Her bir veri kalıbı bu tanımlı imzalar ile eşleştirilmeye çalışılır. İmza tanımı bulunan saldırılar için yüksek algılama başarısı sağlamaktadır. Yeni saldırı yöntemlerinde imza kalıbı bulunmaması durumunda bu saldırıları algılamada yetersiz kalmaktadır. İmza veri tabanının gittikçe büyümesi durumunda her bir paketin karşılaştırılması sırasında yüksek kaynak tüketimine de neden olmaktadır. Anomali tabanlı saldırı tespit sistemleri ise normal aktiviteler için

profil tanımlaması yapar. Bu profilden sapma olması durumlarını anomali veya normal olmayan davranış olarak sınıflandırmaktadır. Bu yöntemin avantajı yeni atak tiplerini tespit edebilme yeteneğine sahip olmasıdır. Dezavantajı ise normal davranışa sahip paket ile anormal paket arasındaki sınırın belirlenmesidir. Günümüzde internet kullanımı ve IoT cihazlarının sayısının artmasına bağlı olarak çok büyük veri paylaşımı olmakta ve anomali tabanlı saldırı tespit sistemlerinin önemi gittikçe artmaktadır [3]. Literatürde bununla ilgili birçok ihlal tespit sistemi önerisi bulunmaktadır.

2.3. VERİ SETLERİ

Ağ güvenliği saldırı tespit sistemlerini geliştirilmesi amacıyla yapılan çalışmalarda veri setleri hayati bir öneme sahiptir ve bu çalışmalarda kullanılmak üzere birçok veri seti bulunmaktadır. Ferrag ve arkadaşları [49] bu veri setlerini içeriklerine göre 7 kategoriye ayırmıştır. Ağ trafiği tabanlı veri seti bu çalışmalarda faydalanılan veri setidir.

En çok kullanılan veri setlerinden bazılarından aşağıda bahsedilmiştir.

DARPA Veri Seti: IDS veri seti oluşturmak için ilk çalışmalar DARPA(Savunma İleri Araştırma Proje Ajansı) tarafından başlatıldı ve KDD98 isimli veri seti oluşturuldu. MIT Lincoln laboratuvarında daha geniş kapsamlı ve gerçekçi IDS karşılaştırma imkanı sağlaması amacıyla bu veri seti 1998 yılında üretilmiştir. ABD hava kuvvetleri ağ yapısını modelleyerek 2 aylık ağ paket dökümü elde etmek için ortam oluşturuldu. Eğitim verileri, yedi haftalık ağ tabanlı saldırıları içerir. Test verileri ayrıca iki haftalık ağ tabanlı saldırıları da içerir. IDS çalışmalarına birçok katkı sağlayan bir çalışma olmuştur. Fakat bu veri saldırı vektörüne hızlı dalgalanmaları dikkate alma ve organizasyona göre değişen ağ kapasiteleri gibi gerçek şartları dikkate alma konusunda eleştiriler almıştır [49].

KDD 99 Veri Seti: KDD Cup 99 veri seti, DARPA'98 veri seti programını temel almaktadır. DoS, Remote to Local (R2L), User to Root (U2R), Probing saldırıları simüle edilir. Veri seti 7 haftalık ağ trafiğini içerir ve yaklaşık 4.9 milyon satırdan oluşur. Bu veri seti, saldırı tespit modellerinin değerlendirilmesi için en yaygın kullanılan veri setlerinden biridir. [50,51].

NSL-KDD Veri Seti: NSL-KDD veri kümesi, KDD 99 veri kümesindeki sorunları çözmek için geliştirilmiştir. Orijinal KDD 99 veri setine göre gereksiz ve tekrarlayan kayıtlar içermez. Makul sayıda kayıt içerir. Mükerrer ve gereksiz kayıtların kaldırılması sonucunda veri seti yaklaşık 5 milyon kayıttan 150.000 kayıta düşürülmüştür. Ayrıca önceden tanımlanmış bölümlere ayrılmıştır. Saldırı tespit sistemi için eğitim ve test alt kümeleri oluşturulmuştur. NSL-KDD, KDD CUP 99 ile aynı özellikleri ve sınıfları kullanır. DoS, Remote to Local (R2L), User to Root (U2R), KDD 99 veri setinde probing saldırıları da bu veri seti için simüle edilir [42,50].

Kyoto Veri Seti: Bu veri kümesi 14 özelliğe karşılık 10 yeni özelliğin de üretilmesi 24 özellikten oluşmaktadır. KDDCup99 veri setinde bulunan 41 özellikten 14 tanesi bal küpü sistemleri kullanılarak Kyoto Üniversitesi tarafından seçilmiştir. Bu ilave olarak daha etkili bir saldırı tespit sistemi tasarımına erişmek için 10 özellik üretilmiştir [42,47].

CIDDS-001 Veri Seti: CIDDS-001 (Coburg Network IntrusionTespit Veri Kümesi) veri seti, anormallik tabanlı bir ağ saldırı tespit sisteminde oluşturulmuştur. Portscan, Pingscan, DoS ve Brute force saldırıları yapılmıştır. Veri seti 14 özellik içerir ve Normal, Saldırgan, Victim, Şüpheli, Unknown sınıflarından oluşur. Veri seti, Openstack kullanılarak birkaç eposta, web sunucusu ve istemciden oluşan küçük bir iş ortamı simülasyonu ile oluşturulmuştur. Python komutlarından faydalanılarak istemci tarafında normal kullanıcı davranışları üretilmiştir [54-57,76].

ISCX-2012 Veri Seti: Bu veri kümesi yedi günlük ağ verilerinden oluşturulmuştur. Veri kümesi, normal ve kötü niyetli ağ trafiğini içerir. Kötü amaçlı ağ trafiği, ağa içeriden sızmayı, HTTP Hizmet Reddi, Dağıtılmış Hizmet Reddi ve Brute Force SSH saldırılarını içerir. Normal ve atak sınıfları vardır [58,76].

AWID Veri Seti: AWID, IEEE veri kümesi için ağlarda oluşturulmuş bir veri kümesidir, 802.11 ağlarında sıkça meydana gelen honeypot, rogue Access point, kötü ikiz, deauthentication atak, sözlük, fragmantasyon gibi 23 farklı saldırı yöntemi kullanılmaktadır. Saldırılar daha sonra Normal, Flooding, Injection ve Impersonation (kimlik avı) etiketleriyle sınıflandırıldı [52,59,60,76].

CSE-CIC-IDS 2017 Veri Seti: Bu veri seti, İletişim Güvenliği Kuruluşları (CSE) ve Kanada Siber Güvenlik Enstitüsü (CIC) tarafından 2017 yılında oluşturulmuştur. Veri seti oluşturulması için hazırlanan test ortamında saldırgan ve mağdur ağları oluşturulmuştur. Veri setini oluşturmak için saldırgan ve kurban ağlarının bulunduğu bir laboratuvar ortamı oluşturulmuştur. Saldırıların yapıldığı ağda bir switch, Kali Linux işletim sistemli bir bilgisayar ve Windows 8 işletim sistemli üç bilgisayar bulunuyordu. Hedef ağda bir adet Windows Server 2016, bir adet Ubuntu 16 işletim sistemli sunucu, bir adet Ubuntu 12 işletim sistemli sunucu, bir adet yönlendirici ve bir adet güvenlik duvarı bulunmaktadır. Windows Server 2016 üzerinde aktif dizin özelliği açılmıştır ve kurban ağındaki tüm cihazlar bu etki alanındadır. Buna ek olarak kurban ağdaki tüm trafiği dinlemek için yönlendiricinin (router) yukarı bağlantı portu yansıtılır [61,62,76].

CSE-CIC-IDS veri setinde, normal trafik oluşturmak için Java-B-profil sistemine dayalı bir ajan yazılmıştır. Bu aracı ile HTTP, HTTPS, FTP, SSH ve e-posta gibi bazı protokol tabanlı öznitelikler, makine öğrenmesi ve istatistiksel yöntemler kullanılarak yeniden üretildi. Saldırı trafiği verilerini oluşturmak için Patator, slowloris, Slowhttps, Metasploit, Ares gibi bazı araçlar kullanılmıştır. Brute force, heartbleed saldırısı, botnet, DoS, DDoS, Web saldırısı, Sızma saldırısı gibi saldırılar düzenlendi. Veri setinde toplam 14 farklı saldırı tipi etiketlenmiştir. Etiketlenen saldırı türleri şunlardır: DoS Golden Eye, Heartbleed, DoS hulk, DoS Slow http, DoS Slowloris, DDoS, SSH-Patator, FP, Patator, Brute force, XSS, Botnet, sızma, PortScann, SQL enjeksiyon. Ayrıca veri seti oluşturulurken CICFlowMeter kullanılmış ve yakalanan trafikten 80 adet özellik oluşturulmuştur [61,62,76].

CSE-CIC-IDS 2018 Veri Seti: 2017 yılında oluşturulan veri seti ile aynı özellikleri göstermektedir. Ancak saldırıların daha iyi modellenebilmesi için test ortamında daha fazla cihaz kullanılmıştır. Bu veri setini oluşturmak için saldırgan ağının altyapısı 50 makine, mağdur ağı 420 makine ve 30 sunucu içerir. Ayrıca mağdur ağı 6 bölüme ayrılmıştır: Ar-Ge departmanı (Dep1), Yönetim Departmanı (Dep2), Teknisyen departmanı (Dep3), Sekreter ve operasyonlar departmanı (Dep4), IT departmanı (Dep5) ve sunucu odaları. Ayrıca IT departmanı dışındaki tüm

departmanlara farklı Windows işletim sistemleri (Windows 8.1 ve Windows 10) kurulumu yapılmıştır. IT bölümündeki tüm bilgisayarlarda Ubuntu işletim sistemi kuruludur. Sunucu odası için Windows server 2012 ve Windows Server 2016 gibi farklı sunucu işletim sistemleri kurulmuştur.

Böylece gerçek dünya ağlarına benzer makine çeşitliliğine sahip bir topoloji oluşturulmuştur. Bu veri setini oluşturmak için Brute force, heartbleed saldırısı, botnet, DoS, DDoS, Web saldırısı, Sızma saldırısı gibi 7 farklı saldırı senaryosu uygulandı [61,63,76].

UNSW-NB15 Veri Seti: Bu veri kümesi, Avustralya Siber Güvenlik Merkezi (ACCS) tarafından oluşturulmuştur. Veri seti oluşturulurken IXIA Perfect-Storm, Tcpdump, Argus ve Bro-IDS araçları kullanılmıştır. Normal ve anormal trafik oluşturucu olarak kullanılan IXIA aracı, üç sanal sunucu üzerine kuruludur. IXIA aracı kullanılarak 9 saldırı senaryosu oluşturuldu: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance(Keşif), shellcode(Kabuk kodu), Solucanlar. IXIA aracı, modern bir tehdit ortamı oluşturmak için CVE sitesini kullanır. Test ortamında kullanılan yönlendiriciler güvenlik duvarına bağlanır. Güvenlik duvarı, tüm trafiği normal veya anormal şekilde geçirecek şekilde yapılandırılmıştır. Yönlendiricilerden biri tcp dump çalıştırıldı ve simülasyon sırasında alınan veri paketleri kaydedildi. Argus, Bro-IDS araçları ve C#'daki on iki algoritma kullanılarak elde edilen veriler 49 özniteliğe çıkarılmıştır [43,44,76].

2.4. ATAK TİPLERİ

Burada saldırı tespit sistemleri araştırmaları sırasında karşılaşılan bazı atak tiplerinden bahsedilecektir.

Hizmet Reddi (DoS): Hedef sistemin RAM ve CPU gibi kaynak kapasitelerinden yararlanılarak yapılan saldırı türüdür. Bu saldırı türünde hedef sistem devre dışı bırakılır ve sistem hizmet veremez [64].

Dağıtılmış Hizmet Reddi (DDoS): Hedef sistemi çok daha hızlı devre dışı bırakmaya yönelik saldırılardır. Saldırganlar ağı tarar ve saldırıda kullanmak üzere güvenlik açığı bulunan cihazlar tespit eder. Daha sonra bu cihazlar kullanılarak DoS

saldırıları aynı anda birden fazla kaynaktan gerçekleştirilir. DDoS saldırılarında hedef bant genişliği, CPU gibi kaynaklardır. Bu kaynakları hedef alarak büyük hacimli ağ trafiği veya birçok sayıda bağlantı isteği göndererek sistemi devre dışı bırakmayı amaçlarlar. Böylece sistem gelen bağlantılara yanıt veremez hale gelir. Bu saldırı senaryosu ICMP taşması, HTTP taşması, TCP taşması gibi birçok yöntemle yapılmaktadır [65,66,76].

Brute Force Atak: Deneme yanılma yöntemini kullanarak hedef sistem üzerinde tam yetkili olmayı amaçlayan saldırı türüdür. Erişim yetkisi bulunmadığı sistemlere kullanıcı adı ve şifre bilgilerinin denenerek kullanıcı adı ve şifre bilgilerine ulaşarak erişim sağlamaya çalışılmaktadır. Bu saldırı türü, Telnet, SSH, RDP, FTP, HTTP gibi protokolleri kullanan sistemlere sıklıkla uygulanmaktadır [67,68].

Exploit: Yetki yükseltmek için hedef sisteme yapılan bir saldırı türüdür. Sistemdeki yazılım ve donanım güvenlik zafiyetlerinden faydalanmak üzere hazırlanan kod parçaları ile sistem ele geçirilmektedir. Birçok istismar türü vardır. En tehlikelilerinden olan sıfır gün saldırısı (Zero Day Exploit), güvenlik açıkları tespit edildiği anda yazılır fakat sistem yöneticisi buradaki saldırının farkına varamamaktadır.

SQL Enjeksiyonu: Genellikle web uygulamalarında veri tabanına erişim yöntemindeki zafiyetleri hedef alan veri tabanındaki tüm verilerin güvenliğini tehlikeye atılmasına sebebiyet veren bir saldırı yöntemidir. Atak yapanın tüm gizli bilgilere erişmesine neden olabilmektedir. Bu saldırı türü, web uygulama güvenliği ile ilgilenen OWASP'ta (Open Web Application Security Project) ilk sırada yer almaktadır [69,70,76].

Bu saldırı türlerinin yanı sıra çalışmalarda sıklıkla kullanılan DARPA, KDD99 ve NSL-KDD veri setlerinde Remote to Local (R2L), User to Root (U2R), Probing gibi saldırı türleri bulunmaktadır. U2R saldırı türünde, saldırgan sistemdeki normal bir kullanıcı hesabını ele geçirir ve ardından sisteme root hakkı ile erişimi elde etmek için güvenlik açıklarından yararlanır. R2L saldırı türünde, saldırgan ağ üzerinden hedef makineye bir paket gönderir ve bu makinenin kullanıcısı olarak yerel erişim

elde etmeye çalışır. Son olarak, hedef ağdaki güvenlik açıklarını tespit etmek için yoklama saldırıları kullanılır [71].

ÜÇÜNCÜ BÖLÜM

3. STS İÇİN DİFERANSİYEL EVRİM ALGORİTMASI

Saldırı tespit sistemi tasarımı aşamasında NSL-KDD veri setine ait öznitelik seçiminde diferansiyel evrim algoritmasından faydalanılmıştır. DE'nin, bu çalışmada geliştirilen saldırı tespit sisteminde sürekli değerli fonksiyonların optimizasyonu için başarılı sonuç vermesi tercih edilmesinde etkili olmuştur.

Bölüm 1.2.3'de bahsedilen DE'nin akışı takip edilerek bu çalışmada popülasyon boyu, çaprazlama oranı, mutasyon oranı, iterasyon sayısı gibi başlangıç parametreleri literatürdeki çalışmalar gözönünde bulundurularak belirlenmiş ve sonuçları çıkarılmıştır. Elde edilen sonuçlara göre kullanılan parametrelerin etkisi elde edilmeye çalışılmıştır. Popülasyon büyüklüğünü birbirinden farklı 3 bireyin seçilmesi şartından dolayı en az 4 adet olacak şekilde belirlenmiştir. Burada popülasyon büyüklüğü değerlerini seçerken kabul edilebilir sürelerde sonuç üretmediği için çok büyük değerlerden kaçınılmıştır. Çaprazlama oranı ve mutasyon oranı değerleri 0-1 arasında değerler alacak şekilde belirlenir. İterasyon sayısı her defasında daha iyi sonuçlar elde etme ihtimaline karşılık makul bir değer olarak atanır. Aday çözümler 41 adet özellik içeren NSL-KDD veri setinin label encoding ve one hot encoding işlemi uygulandıktan sonra elde edilen 122 elemandan oluşmaktadır ve $[0,1]$ aralığında değerler alır.

Her bir aday çözümün uygunluk değerinin hesaplanmasında NSL-KDD veri seti dengesiz dağılımlı veri seti olması nedeniyle ağırlıklandırılmış $f1$-skor değeri kullanılmıştır. Ayrıca amaç daha az sayıda öz nitelik olduğundan dolayı uygunluk değerini hesaplarken daha az sayıda öznitelik seçilmesi göz önünde bulundurulmuştur. Buna göre, diferansiyel evrim algoritmasının uygunluk değer hesaplaması Denklem 3.1'de verilmiştir.

$$k = \text{Ö}znitelik\ seçiminde\ seçilen\ \text{ö}zellik\ sayısı \qquad \text{Denklem 3.1}$$

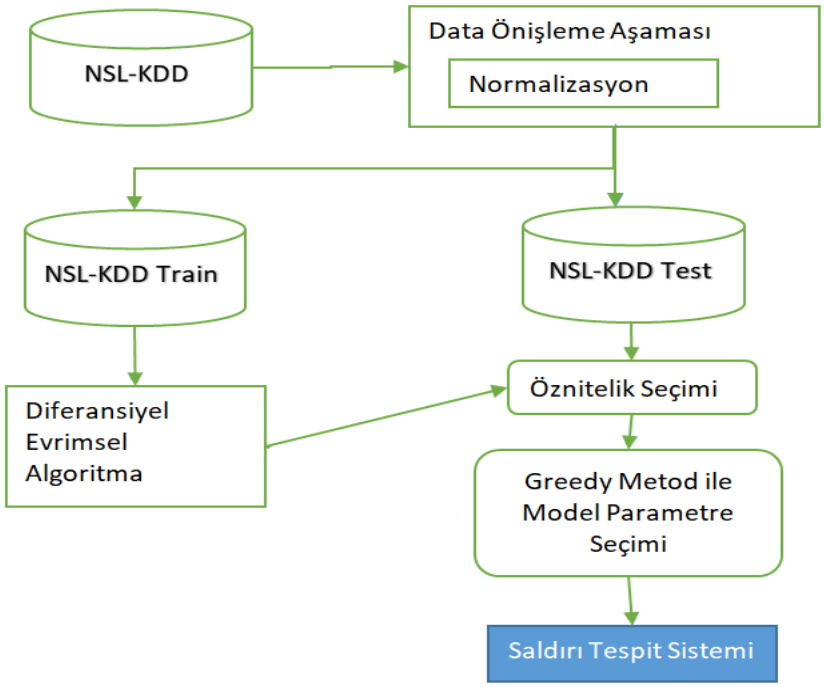
$N = Veri\ setinde\ bulunan\ toplam$ özellik $sayısı$

$w = 0.6\ (ağırlık\ değeri)$

$Uygunluk = w * f1 - skor + (1.0 - w) * (1 - k/N)$

Popülasyon boyutu ve sınır değerleri dikkate alınarak rastgele başlangıç popülasyonu oluşturulur. Daha sonra herbir popülasyon bireyi için uygunluk değeri hesaplaması yapılır. Saldırı tespit sistemini tasarımı maksimizasyon problemi olarak ele alındığı için burada hesaplanan değerlerden en büyüğü sonraki çözümlerle karşılaştırılmak üzere en iyi çözüm olarak belirlenir. Çaprazlama ve mutasyon işlemleri sonrası yeni bireyler elde edilir ve bir sonraki nesil için bireyler seçilir. Mutasyon sonucu elde edilen 122 uzunluktaki çözüm vektörünün her bir alanı için [0,1] aralığında olması kontrolü yapılır. Aralık dışındaki değerler [0,1] aralığına çekilir.

Oluşturulan bireyin 0.5 ve üzeri değere sahip kromozom yapısı için o öznitelik seçilir ve seçilen öznitelik kümesine göre greedy yöntemle hiperparametreleri belirlenen Adaboost algoritması uygulanarak bireyin performans hesaplaması yapılır. Şekil 3.1'de algoritma akışı verilmiştir.



Şekil 3.1 Algoritma Akışı

DÖRDÜNCÜ BÖLÜM

4. DENEYSEL ÇALIŞMALAR

Bu tezde, NSL-KDD veri seti kullanılarak ön işleme sonrası diferansiyel evrim algoritması kullanılarak öznitelik ve model seçimi yapılmaktadır.

4.1. VERİ SETİ

Bu çalışma kapsamında gereksiz kayıtların temizlenmiş olduğu en yaygın veri setlerinden olan NSL-KDD veri seti tercih edilmiştir.

NSL-KDD veri seti Tablo 4.1 ‘de verilen 41 adet özellik, 1 adet sınıf değeri ve zorluk seviyesini gösteren level alanı olmak üzere 43 kolondan oluşmaktadır.

Tablo 4.1 NSL-KDD Veri Seti Özellikleri [72]

Sıra No	Özellik	Tanım
1	duration	Bağlantı süresinin uzunluğu.
2	protocol_type	Kullanılan protokol tipi.
3	service	Kullanılan hedef şebeke servisi
4	flag	Bağlantı durumu (Normal ya da Hata)
5	src_bytes	Tek bağlantıda kaynaktan hedefe aktarılan veri bayt sayısı
6	dst_bytes	Tek bağlantıda hedeften kaynağa aktarılan veri bayt sayısı
7	Land	Kaynak IP, hedef IP adresleri ve bağlantı noktası numaraları eşitse, bu değişken 1 veya 0 değerini alır
8	wrong_fragment	Bu bağlantıdaki toplam yanlış parça sayısı
9	urgent	Bu bağlantıdaki acil paket sayısı. Acil paketler, acil biti etkinleştirilmiş paketlerdir.
10	Hot	İçerikteki "sıcak" göstergelerin sayısı, örneğin: içerik dizinine girme, program oluşturma ve programları yürütme
11	num_failed_logins	Başarısız oturum açma girişimlerinin sayısı
12	logged_in	Oturum açma durumu: Başarılı iken 1, başarılı değilken 0

13	num_compromised	"Tehlikeye atılmış/Güvenliği ihlal edilmiş" koşulların sayısı
14	root_shell	Kök kabuktan elde edilen ise 1, değilse 0
15	su_attempted	"su root" komutu denenmişse 1, değilse 0
16	num_root	Kök erişim sayısı veya bağlantıda kök olarak gerçekleştirilen işlem sayısı
17	num_file_creations	Bağlantıdaki dosya oluşturma işlemlerinin sayısı
18	num_shells	Kabuk istemlerinin sayısı
19	num_access_files	Erişim denetimi dosyalarındaki işlem sayısı
20	num_outbound_cmds	Ftp oturumunda giden komut sayısı
21	is_host_login	Oturum açma sıcak listeye yani admin ya da root ise 1, değilse 0
22	is_guest_login	Misafir oturum açma ise 1, değilse 0
23	Count	Son iki saniyedeki geçerli bağlantıyla aynı hedef ana bilgisayara bağlantı sayısı
24	srv_count	Son iki saniyedeki geçerli bağlantıyla aynı servise bağlantı sayısı (bağlantı noktası numarası)
25	serror_rate	Count(23)'ta toplanan bağlantılar arasında s0,s1,s2 veya s3 flag(4)'ını etkinleştiren bağlantıların yüzdesi
26	srv_serror_rate	Srv_coun(24)'ta toplanan bağlantılar arasında s0, s1, s2 veya s3 flag(4) ını etkinleştiren bağlantıların yüzdesi
27	rerror_rate	Count(23)'ta toplanan bağlantılar arasında REJ flag(4)'ını etkinleştiren bağlantıların yüzdesi
28	srv_rerror_rate	Srv_count(24)'ta toplanan bağlantılar arasında REJ flag(4)'ını etkinleştiren bağlantıların yüzdesi
29	same_srv_rate	Count(23)'ta toplanan bağlantılar arasında aynı hizmet olan bağlantıların yüzdesi
30	diff_srv_rate	Count(23)'ta toplanan bağlantılar arasında farklı hizmet olan bağlantıların yüzdesi
31	srv_diff_host_rate	Srv_count'ta (24) toplanan bağlantılar arasında farklı hedef makinelere olan bağlantıların yüzdesi
32	dst_host_count	Aynı hedef ana bilgisayar IP adresine sahip bağlantı sayısı
33	dst_host_srv_count	Aynı bağlantı noktası numarasına sahip bağlantı sayısı
34	dst_host_same_srv_rate	dst_host_count (32) içinde toplanan bağlantılar arasında aynı hizmet olan bağlantıların yüzdesi
35	dst_host_diff_srv_rate	dst_host_count (32) içinde toplanan bağlantılar

		arasında farklı hizmetlere olan bağlantıların yüzdesi
36	dst_host_same_src_port_rate	dst_host_srv_count (33) içinde toplanan bağlantılar arasında aynı kaynak bağlantı noktasına olan bağlantıların yüzdesi
37	dst_host_srv_diff_host_rate	dst_host_srv_count(33) içinde toplanan bağlantılar arasında farklı hedef makinelere olan bağlantıların yüzdesi
38	dst_host_serror_rate	dst_host_count'ta (32) toplanan bağlantılar arasında s0, s1, s2 veya s3 flag(4)'ını etkinleştiren bağlantıların yüzdesi
39	dst_host_srv_serror_rate	dst_host_srv_count'ta (33) toplanan bağlantılar arasında s0, s1, s2 veya s3 flag(4)'ını etkinleştiren bağlantıların yüzde olarak oranı
40	dst_host_rerror_rate	dst_host_count'ta (32) toplanan bağlantılar arasında REJ flag(4)'ını etkinleştiren bağlantıların yüzdesi
41	dst_host_srv_rerror_rate	dst_host_srv_count(33) içinde toplanan bağlantılar arasında REJ flag(4)'ını etkinleştiren bağlantıların yüzde olarak oranı
42	attack	Atak türleri
43	level	Atağın Zorluk seviyesi

NSL-KDD veri seti aşağıdaki gibi 4 sınıf atak tipi ve normal kayıtlardan oluşmaktadır.

U2R sınıfı: Saldırganın önce sistemin kullanıcı kimliğini elde ettiği, ardından sistemin köküne erişim elde etmek için savunmasız kısımlardan yararlandığı saldırı sınıfıdır.

R2L sınıfı: Saldırganın ağ üzerinden veri paketleri göndererek kullanıcı olarak sisteme girişini yapar ve buna yerel erişim sağlar.

DOS sınıfı: Bu saldırı türünde bir saldırgan, bellek ve bilgi işlem kaynakları gibi sistemin farklı kaynaklarını meşgul ederek sistem performansını tamamen bozar. Bu nedenle sistem yasal taleplere cevap verememektedir.

Probe sınıfı: Bu saldırı türünde, bir saldırgan bilgi toplamak ve güvenlik açıklarını tespit etmek için ağı araştırır. Bundan sonra ağdaki makine ve servislere göre bilgi sahibi olunarak ağın davranışlarını kolaylıkla tespit edebilmektedir [73].

NSL-KDD veriseti içinde bulunan atak tipleri ve bu atakların dahil olduğu atak sınıfı Tablo 4.2'de özetlenmiştir.

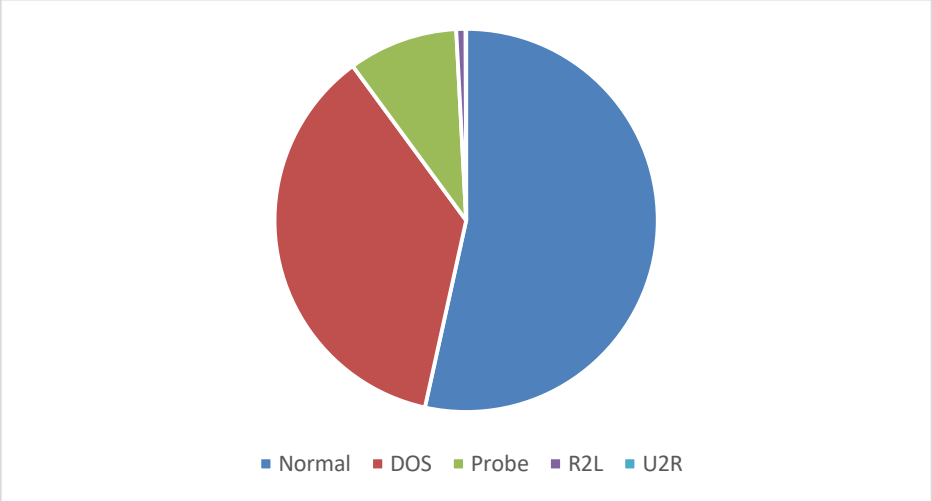
Tablo 4.2 Atak Kategorileri [40,73]

Sıra No	Atak Kategorisi	Atak İsmi
1	DOS (Denial of Service attack) Sisteme gönderilen meşru istekleri reddetme	apache2, back, land, neptune, mailbomb, pod, processtable, smurf, teardrop, udpstorm, worm
2	Probe attack Bilgi toplama saldırısı	ipsweep, mscan, nmap, portsweep, saint, satan
3	U2R (user to root attack) Lokal admin ve root kullanıcı hakkına yetkisiz erişimi	buffer_overflow, loadmdoule, perl, ps, rootkit, sqlattack, xterm
4	R2L(remote to local attack) Ağdaki bilgisayara yetkisiz erişim	ftp_write, guess_passwd, http_tunnel, imap, multihop, named, phf, sendmail, snmpgetattack, snmpguess, spy, warezclient, warezmaster, xclock, xsnoop

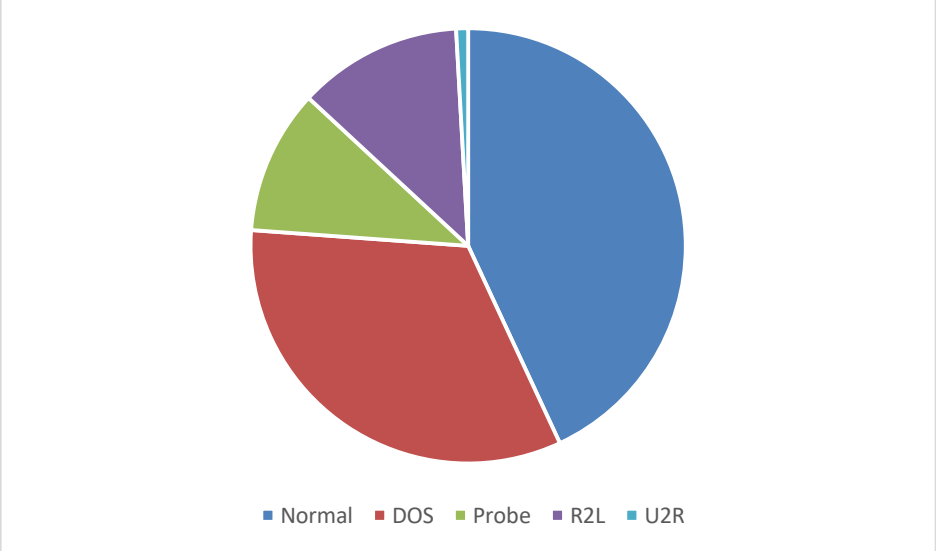
NSL-KDD eğitim ve test veri setlerinde bulunan verilerin sayısal dağılımı Tablo 4.3 özetlenmiştir.

Tablo 4.3 Eğitim ve Test Veri Seti Atak Dağılımı [74]

Eğitim Veri Seti		Test Veri Seti	
Sınıf	Adet	Sınıf	Adet
Normal	67343	Normal	9711
DOS	45927	DOS	7458
Probe	11656	Probe	2421
R2L	995	R2L	2754
U2R	52	U2R	200
Toplam	125973	Toplam	22544



Şekil 4.1 NSL-KDD Eğitim Veri Seti Atak Sınıfı Dağılımı



Şekil 4.2 NSL-KDD Test Veri Seti Atak Sınıfı Dağılımı

4.2. VERİ ÖNİŞLEME

NSL-KDD veri seti nümerik ve nümerik olmayan veriler içermektedir. Nümerik olmayan alanlar makine öğrenmesi algoritmaları için uygun değildir. Numerik olmayan protocol_type, service ve flag gibi özellikler numerik değere dönüştürülür. Nümerik olmayan bu alanlar için label encoding işlemi uygulanarak nümerik değerler elde edilir. Daha sonra onehot encoding işlemi uygulanarak yeni kolonlar elde edilir.

Eğitim ve test veri setinde bulunan nümerik olmayan kolonlarda bulunan birbirinden farklı her bir değer kolon olarak eklenir. Bu kolonlarda ilgili değer varsa 1 yoksa 0 olarak doldurulur.

Tablo 4.4 Eğitim ve Test Veri Seti Nümerik Olmayan Kolondaki Farklı Değer Sayısı

Eğitim Veri Seti		**Test Veri Seti**	
Öznitelik	**Sayı**	**Öznitelik**	**Sayı**
protocol_type	3	protocol_type	3
Service	70	Service	64
Flag	11	Flag	11
Toplam	**84**	**Toplam**	**78**

Eğitim ve test veri seti için onehot encoding uygulanmasıyla elde edilecek kolon bilgisi tablo 4.5 de sunulmuştur.

Tablo 4.5 Eğitim ve Test Veri Seti Yeni Üretilen Kolonlar

Protocol_type			
Protocol_type_icmp	Protocol_type_tcp	Protocol_type_udp	
Service			
service_IRC	service_eco_i	service_imap4	service_nntp
service_ssh	service_X11	service_ecr_i	service_iso_tsap
service_ntp_u	service_sunrpc	service_Z39_50	service_efs
service_klogin	service_other	service_supdup	service_aol
service_exec	service_kshell	service_pm_dump	service_systat
service_auth	service_finger	service_ldap	service_pop_2
service_telnet	service_bgp	service_ftp	service_link
service_pop_3	service_tftp_u	service_courier	service_ftp_data
service_login	service_printer	service_tim_i	service_csnet_ns
service_gopher	service_mtp	service_private	service_time
service_ctf	service_harvest	service_name	service_red_i
service_urh_i	service_daytime	service_hostnames	service_netbios_dgm
service_remote_job	service_urp_i	service_discard	service_http
service_netbios_ns	service_rje	service_uucp	service_domain
service_http_2784	service_netbios_ssn	service_shell	service_uucp_path
service_domain_u	service_http_443	service_netstat	service_smtp
service_vmnet	service_echo	service_http_8001	service_nnsp
service_sql_net	service_whois		
Flag			
flag_OTH	flag_RSTO	flag_RSTR	flag_S1
flag_S3	flag_REJ	flag_RSTOS0	flag_S0

flag_S2	flag_SF	flag_SH

Tablo 4.4'de eğitim veri setinde bulunup test veri setinde bulunmayan alanlar için tüm değerleri 0 olacak şekilde doldurulur.

Veri setindeki 43. numaradaki level kolonu da kaydın atak olup olmadığını tespit kapsamında değerlendirilmeyen bir alan olduğu için ilgili alan veri setinden kaldırılmıştır. Ataklar 4 ana atak sınıfı ve normal olmak üzere 5 sınıfa dahil olacak şekilde değer ataması yapılır.

4.3. PARAMETRE ATAMALARI VE MODEL OLUŞTURMA

Diferansiyel evrim algoritması kullanılarak Adaboost parametrelerinin ve veri setine ait özniteliklerin seçimi yapılmaktadır. Diferansiyel evrim algoritmasına ait başlangıç parametreleri aşağıdaki gibi seçilmiştir.

Popülasyon Boyutu: Farklı değerlerde popülasyon boyutu değeri set edilerek tasarlanan sistemin performansı incelenmiştir. Bu çalışmada popülasyon boyutu 15 alınmıştır.

İterasyon Sayısı: Diferansiyel evrim algoritması içinde popülasyonun en iyi değerini elde etme sürecinin kaç defa tekrarlanacağını ifade eder. Bu çalışmada iterasyon sayısı 10 alınmıştır.

Mutasyon Oranı (F) : Mutasyon oranını ifade etmektedir. Bu çalışmada mutasyon oranı 0.6 olarak alınmıştır.

Çaprazlama Oranı (CR): Cr çaprazlama oranını ifade eder. Çaprazlama oranından büyük değer gelmesi durumunda ilgili indeks için mutasyona uğramış değer seçilir değilse populasyondaki mevcut değeri alınır. Çaprazlama oranı olarak 0.7 değeri alınmıştır.

4.3.1. Adaboost Parametreleri

Adaboost için geçerli olan 3 parametre için DE algoritması kullanılarak değer ataması yapılmaktadır. Daha sonra bu parametreler kullanılarak oluşturulan model,

eğitim veri seti ile eğitilmekte ve uygunluk değeri hesaplanmaktadır. Adaboost parametrelerinin seçimi aşağıdaki maddelerde verilmiştir.

1- **Temel Tahmin Edici:** base_estimator parametresine karşılık gelmekte ve aşağıdaki tablodaki kontroller dikkate alınarak belirlenmektedir.

   Max_depth parametresi, ağacın maksimum derinliğinin ne kadar olacağını ifade eden değerdir. Hiçbiri ise, tüm yapraklar saf olana kadar düğümler genişletilir.
   Criterion parametresi, bir bölünmenin kalitesini ölçme işlevi için kullanılacak kritere karşılık gelmektedir. Varsayılan değeri gini'dir.

   Bu çalışmada default tahmin edici olan karar ağacı sınıflandırıcısı kullanılmıştır.

2- **Tahmin Edici Sayısı:** n_estimators parametresine karşılık gelmekte ve bu parametre Tablo 4.5'deki değerleri alabilmekte ve bu çalışmada optimize edilmek üzere kullanılmıştır.

Tablo 4.6 Tahmin Edici Sayısı

Sıra	Değer
1	10
2	15
3	20
4	50

3- **Öğrenme Oranı:** learning_rate parametresine karşılık gelmekte ve şartlara göre aşağıdaki değerleri alabilmektedir.

Tablo 4.7 Öğrenme Oranı

Sıra	Değer
1	1
2	0.1
3	0.001

4- **Algoritma:** SAMME veya SAMME.R olabilir. SAMME ve SAMME.R algoritmalarının performansı karşılaştırılır. SAMME.R, toplama modelini

güncellemek için olasılık tahminlerini kullanırken, SAMME yalnızca sınıflandırmaları kullanır. SAMME.R algoritması tipik olarak SAMME'den daha hızlı yakınsayarak, daha az artırma yinelemesi ile daha düşük bir test hatası elde eder. Hata 1/2'nin üzerine çıktıktan sonra SAMME.R bozuluyor. Bununla birlikte, hata 1/2'den büyük veya 1/2'ye eşit olsa da, SAMME için durum böyle değildir, tahmin edicinin ağırlığı hala pozitiftir; bu nedenle, yanlış sınıflandırılmış eğitim örnekleri daha fazla ağırlık alır ve test hatası azalmaya devam eder.

Tablo 4.8 Algoritma

Sıra	**Değer**
1	SAMME
2	SAMME.R

4.4. PERFORMANS METRİKLERİ

Bu bölüm, saldırı tespit sistemleri için makine öğrenmesi yöntemlerinin performansını ölçmek için en sık kullanılan değerlendirme metriklerini açıklar. Tüm değerlendirme metrikleri, gerçek ve öngörülen sınıf hakkında bilgi sağlayan iki boyutlu bir matris olan karışıklık matrisinde kullanılan farklı öznitelikleri temel alır [3]. [75]'ye göre karışıklık matrisi, sınıflandırıcının farklı sınıfların demetlerini tanımada iyi olup olmadığını analiz etme işlevine sahip bir tablo olarak değerlendirilebilir.

Tablo 4.9 Karışıklık Matrisi

		Tahmin Edilen	
		Saldırı	**Normal**
Gerçek	**Saldırı**	TP	TN
	Normal	FP	FN

- **Gerçek Pozitif(TP):** Sınıflandırıcı tarafından doğru bir şekilde saldırı olarak tahmin ediln veri örnekleri.

- **Yanlış Negatif (FN):** Normal örnekler olarak yanlış tahmin edilen veri örnekleri.

- **Yanlış Pozitif (FP):** Yanlış bir şekilde Saldırı olarak sınıflandırılan veri örnekleri.

- **Gerçek Negatif (TN):** Normal örnekler olarak doğru şekilde sınıflandırılan örnekler.

Karışıklık matrisinin köşegeni doğru tahminleri gösterirken köşegen olmayan elemanlar belirli bir sınıflandırıcının yanlış tahminleridir. Tablo 4.8, karışıklık matrisinin bu özelliklerini göstermektedir. Ayrıca, son çalışmalarda kullanılan farklı değerlendirme ölçütleri,

Kesinlik (Precision): Doğru tahmin edilen saldırıların, saldırı olarak tahmin edilen tüm örneklere oranıdır.

$$Kesinlik = TP / (TP + FP)$$

Geri Çağırma (Recall): Saldırı olarak doğru bir şekilde sınıflandırılan tüm örneklerin, gerçekten Saldırı olan tüm örneklere oranıdır. Aynı zamanda Algılama Oranı olarak da adlandırılır.

$$Geri\ Çağırma = TP / (TP + FN)$$

Yanlış Alarm Oranı (FAR): Yanlış pozitif oranı olarak da adlandırılır ve yanlış tahmin edilen Saldırı örneklerinin Normal olan tüm örneklere oranı olarak tanımlanır.

$$FAR = FP / (FP + TN)$$

Gerçek Negatif Oran (TNR): Doğru sınıflandırılan normal örnek sayısının normal olan tüm örneklere oranı olarak tanımlanır.

$$TNR = TN / (TN + FP)$$

Doğruluk (Accuracy): Doğru sınıflandırılmış örneklerin toplam örnek sayısına oranıdır. Algılama Doğruluğu olarak da adlandırılır ve yalnızca bir veri kümesi dengelendiğinde yararlı bir performans ölçüsüdür.

$$Doğruluk = (TP + TN) / (TP + TN + FP + FN)$$

F1-Skor (F1-score): Kesinlik ve Geri Çağırmanın harmonik ortalaması olarak tanımlanır. Başka bir deyişle, sistemin hem kesinliğini hem de geri çağrılmasını dikkate alarak bir sistemin doğruluğunu incelemeye yönelik istatistiksel bir tekniktir.

$$F1 - skor = 2(Kesinlik\ x\ Geri\ Çağırma) / (Kesinlik + Geri\ Çağırma)$$

BEŞİNCİ BÖLÜM

5. BULGULAR VE TARTIŞMA

Bu bölümde geliştirilen algoritma için verilen parametrelere göre sonuçlar ve tartışma verilmiştir.

Veri seti DoS atak ve normal veriden oluşacak şekilde sınıflandırılmış ve öznitelik seçimi yapılmadan Adaboost sınıflandırıcı model uygulandığında Tablo 5.1 deki sonuçlar elde edilmiştir.

Tablo 5.1 DoS Atak ve normal verilerden oluşan öznitelik seçimi yapılmadan sınıflandırma

Tekrar Sayısı	10					
Sıra No		**Algoritma**	**Öğrenme Oranı**	**Tahmin Edici Sayısı**	**F1-Skor**	**Doğruluk**
1	AdaBoost	SAMME.R	1	50	0.90162	0.90327
	DT				0.81451	0.82284
	RF				0.50442	0.61010
	NB				0.75189	0.76909
	SVM				0.90472	0.90595
	MLP				0.88660	0.88731

DoS ve normal veriden oluşan veri setine DE algoritması ile öznitelik seçimi yapıldı ve Adaboost sınıflandırıcı uygulandığında Tablo 5.2'deki sonuç elde edilmiştir.

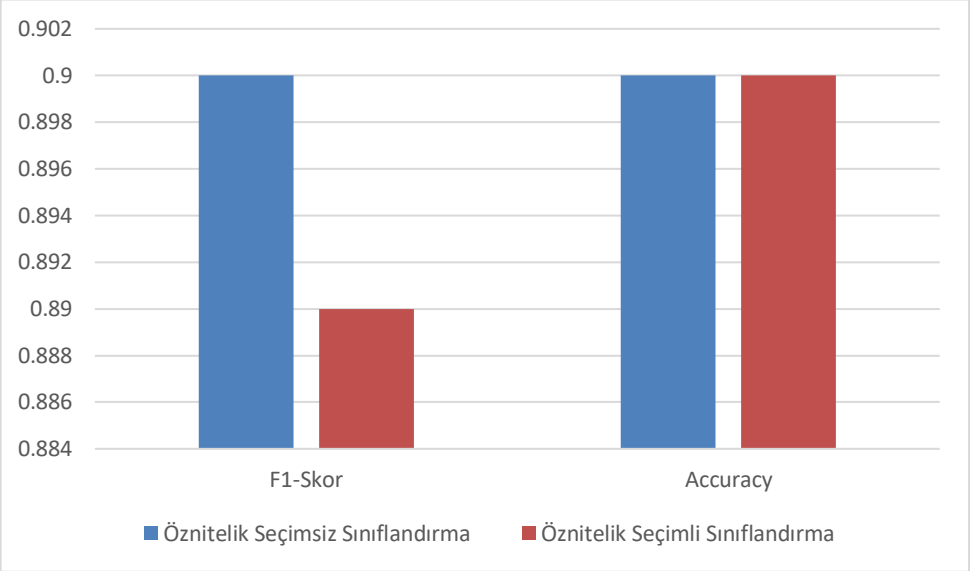
Tablo 5.2 DoS atak ve normal verilerden oluşan öznitelik seçimi yapılarak sınıflandırma

Sıra No		**Algoritma**	**Öğrenme Oranı**	**Tahmin Edici Sayısı**	**F1-Skor**	**Doğruluk**	**Öznitelik Sayısı**
1	AdaBoost	SAMME.R	1	50	0.89283	0.89511	44
	DT				0.90772	0.90921	
2	AdaBoost	SAMME.R	1	50	0.71079	0.73781	43
	DT				0.86332	0.86757	

3	AdaBoost	SAMME.R	1	50	0.90832	0.90991	46
	DT				0.84964	0.85545	
4	AdaBoost	SAMME.R	1	50	0.89761	0.89954	39
	DT				0.86162	0.86599	
5	AdaBoost	SAMME.R	1	50	0.63832	0.68907	38
	DT				0.87582	0.87916	
6	AdaBoost	SAMME.R	1	50	0.66743	0.70497	42
	DT				0.62103	0.66018	
7	AdaBoost	SAMME.R	1	50	0.85147	0.85708	48
	DT				0.83958	0.84614	
8	AdaBoost	SAMME.R	1	50	0.86614	0.86955	51
	DT				0.83778	0.84491	
9	AdaBoost	SAMME.R	1	50	0.68122	0.70695	46
	DT				0.85962	0.86180	
10	AdaBoost	SAMME.R	1	50	0.71152	0.74014	36
	DT				0.93721	0.93780	

Tablo 5.3 DoS atak ve normal verilerden oluşan öznitelik seçimi yapılarak sınıflandırma istatistikleri

İstatistik	**F1-Skor**	**Doğruluk**	**Öznitelik Sayısı**
Standart Sapma	0.10767768	0.091838	4.588523
Ortalama	0.71151721	0.740143	44
Max	0.90832149	0.909906	51
Min	0.63831898	0.689069	36



Şekil 5.1 DoS atak ve normal verilerden oluşan küme için öznitelik seçimi yapılarak ve öznitelik seçimi yapılmadan yapılan sınıflandırma

Veri seti DoS atak, diğer atak ve normal veriden oluşacak şekilde sınıflandırılmış ve öznitelik seçimi yapılmadan Adaboost sınıflandırıcı model uygulandığında Tablo 5.4 'deki sonuçlar elde edilmiştir.

Tablo 5.4 DoS atak, diğer atak ve normal verilerden oluşan öznitelik seçimi yapılmadan sınıflandırma

Tekrar Sayısı	10					
Sıra No		**Algoritma**	**Öğrenme Oranı**	**Tahmin Edici Sayısı**	**F1-Skor**	**Doğruluk**
1	AdaBoost	SAMME.R	0.1	15	0.539742	0.596256
	DT				0.396187	0.49277
	RF				0.471024	0.549814
	NB				0.032754	0.108188
	SVM				0.684325	0.734342
	MLP				0.647661	0.689496

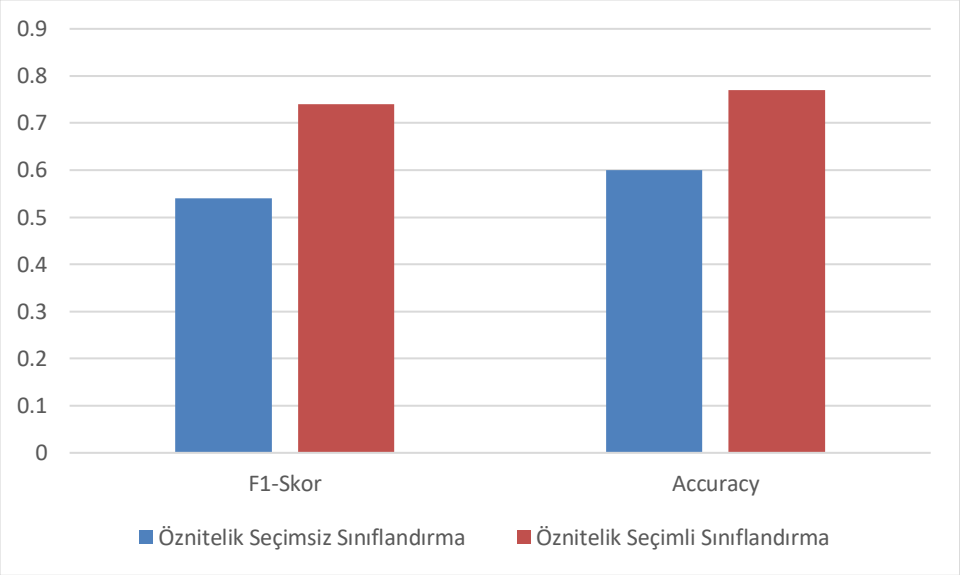
DoS atak, diğer ataklar ve normal veriden oluşan verisetine DE algoritması ile öznitelik seçimi yapıldı ve Adaboost sınıflandırıcı uygulandığında Tablo 5.5'deki sonuçlar elde edilmiştir.

Tablo 5.5 DoS atak, diğer atak ve normal verilerden oluşan veriseti için öznitelik seçimi yapılarak sınıflandırma

Sıra No	Sınıflandırıcı	Algoritma	Öğrenme Oranı	Tahmin Edici Sayısı	F1-Skor	Doğru-luk	Öznite-lik Sayısı
1	AdaBoost	SAMME	1	50	0.7152	0.7351	42
	DT				0.7214	0.7339	
2	AdaBoost	SAMME.R	1	50	0.6956	0.7133	42
	DT				0.7219	0.7386	
3	AdaBoost	SAMME.R	1	20	0.7405	0.7661	46
	DT				0.7193	0.7363	
4	AdaBoost	SAMME.R	1	50	0.7363	0.7433	43
	DT				0.7316	0.7478	
5	AdaBoost	SAMME.R	1	50	0.6836	0.6986	41
	DT				0.6420	0.6605	
6	AdaBoost	SAMME.R	1	50	0.7004	0.7186	44
	DT				0.5890	0.6089	
7	AdaBoost	SAMME.R	1	50	0.7295	0.7467	45
	DT				0.7334	0.7393	
8	AdaBoost	SAMME.R	1	50	0.6987	0.7175	41
	DT				0.6783	0.6925	
9	AdaBoost	SAMME.R	1	15	0.6923	0.7074	44
	DT				0.6317	0.6505	
10	AdaBoost	SAMME.R	1	50	0.7334	0.7530	42
	DT				0.7526	0.7679	

Tablo 5.6 DoS atak, diğer atak ve normal verilerden oluşan öznitelik seçimi yapılarak sınıflandırma sonucu istatistiksel bilgiler

İstatistik	F1-Skor	Doğruluk	Öznitelik Sayısı
Standart Sapma	0.05394	0.04816	2.0136
Ortalama	0.70036	0.71864	43
Max	0.74055	0.76615	47
Min	0.54621	0.58610	41



Şekil 5.2 DoS atak, diğer atak ve normal verilerden oluşan küme için öznitelik seçimi yapılarak ve öznitelik seçimi yapılmadan yapılan sınıflandırma

Tablo 5.1'de herhangi öznitelik seçimi yapılmadan DoS ve normal verilerden oluşan verisetine, algoritma SAMME.R, öğrenme oranı 1 ve tahmin edici sayısı 50 verilerek oluşturulan Adaboost modeli uygulanarak yapılan sınıflandırmada f1-skor değeri 0.90162 ve doğruluk değeri 0.90327 olarak en yüksek oran elde edilmiştir.

Tablo 5.2'de diferansiyel evrim algoritması uygulanarak onehot encoding işlemi uygulanarak elde edilen 122 öznitelikten 46 tanesi seçilen veriseti için algoritma SAMME.R, öğrenme oranı 1 ve tahmin edici sayısı 50 verilerek oluşturulan Adaboost modeli uygulanmıştır. Yapılan sınıflandırmada f1-skor değeri 0.90832 ve doğruluk değeri 0.90991 olarak en yüksek oran elde edilmiştir.

Tablo 5.4'de herhangi öznitelik seçimi yapılmadan DoS, diğer atak ve normal verilerden oluşan verisetine, algoritma SAMME.R, öğrenme oranı 0.1 ve tahmin edici sayısı 15 verilerek oluşturulan Adaboost modeli uygulanarak yapılan sınıflandırmada f1-skor değeri 0.539742 ve doğruluk değeri 0.596256 olarak en yüksek oran elde edilmiştir.

Tablo 5.5'de diferansiyel evrim algoritması uygulanarak onehot encoding işlemi uygulanarak elde edilen 122 öznitelikten 46 tanesi seçilen veriseti için algoritma SAMME.R, öğrenme oranı 1 ve tahmin edici sayısı 20 verilerek

oluşturulan Adaboost modeli uygulanmıştır. Yapılan sınıflandırmada f1-skor değeri 0.7405 ve doğruluk değeri 0.7661 olarak en yüksek oran elde edilmiştir.

SONUÇ

Bu çalışmada NSL-KDD veri seti üzerinde diferansiyel evrim algoritması kullanılarak öznitelik ve model seçimi yapılarak saldırı tespit sistemi algoritması geliştirilmiştir. Veri seti olarak saldırı tespit sistemi geliştirilmesinde en fazla tercih edilen veri setlerinden biri olan KDD'99 veri setinin gereksiz kayıtlardan arındırılmasıyla elde edilen NSL-KDD veri seti kullanılmıştır. 41 adet öznitelik, 1 adet atak tipi ve 1 adet atak seviyesini ifade eden level alanı olmak üzere 43 adet özellikten oluşmaktadır.

Saldırı tespit sistemi geliştirilmesinde 41 adet öznitelikten nümerik olmayan alanlar için önişleme aşamasında one hot encoding uygulanır. Böylece elde edilen toplam 122 özellik arasından DE algoritması ile seçim yapılmakta ve greedy yöntemle seçilen parametreler ile oluşturulan Adaboost algoritmasının eğitilmesi sağlanmaktadır. Daha sonra uygunluk değeri en yüksek olan model ve öznitelik seçimi test veri seti için tatbik edilmiştir. Test veri setine uygulanan en uygun çözümün, eğitim veri setine göre uygunluk değerinin daha düşük olduğu gözlemlenmiştir.

DoS atak ve normal verilerden oluşan iki sınıflı veri kümesi için 46 adet öznitelik seçimi ve Adaboost hiper parametreleri olarak belirlenen seçenekler arasında algoritma parametresi olarak SAMME.R, öğrenme oranı olarak 1 ve tahmin edici 50 olarak oluşturulan algoritma akışının daha başarılı sonuç verdiği görülmüştür.

DoS atak, diğer ataklar ve normal verilerden oluşan üç sınıflı veri kümesi için 46 adet öznitelik seçimi ve Adaboost hiper parametreleri olarak belirlenen seçenekler arasında algoritma parametresi olarak SAMME.R, öğrenme oranı olarak 1 ve tahmin edici 20 olarak oluşturulan algoritma akışının daha başarılı sonuç verdiği görülmüştür.

Test veri seti için uygunluk değerinin yükseltilmesi için bu algoritmadaki DE'nin öznitelik ve model parametrelerinin belirlenmesindeki kullanımda

değişikliğe gidilmiştir, DE'nin öznitelik ve model seçimi için farklı aşamalarda kullanımı değerlendirilecektir.

Algoritmanın başarısının yükseltilmesi için çaprazlama oranı, mutasyon oranı ve uygunluk değeri hesaplamasındaki parametre değerinin değiştirilmesi gibi çalışmalar yapılması planlanmaktadır.

Çalışmanın iki sınıflı ve üç sınıflı başarısı artırıldıktan sonra 4 adet atak sınıfı ve normal kayıtlardan oluşacak şekilde 5 sınıfı kapsayacak çalışmanın yapılması planlanmaktadır.

KAYNAKÇA

[1] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150.

[2] Thomas, R., & Pavithran, D. (2018). A survey of intrusion detection models based on NSL-KDD data set. 2018 Fifth HCT Information Technology Trends (ITT), 286-291.

[3] Xin Y, Kong L, Liu Z, et al. Machine learning and deep learning methods for cybersecurity. IEEE Access. 2018; 6: 35365- 35381

[4] Chary S, Rama B. A survey on comparative analysis of decision tree algorithms in data mining, International Journal of Advanced Scientific Technologies, Engineering and Management Sciences; vol. 3, 2017:91-95.

[5] Sahani R, Rout C, Badajena JC, Jena AK, Das H. Classification of intrusion detection using data mining techniques. Progress in Computing, Analytics and Networking. New York, NY: Springer; 2018: 753- 764.

[6] Safavian, S.R. and Landgrebe, A survey of decision tree classifier methodology. IEEE trans. sys man cybernetics, 1 ¯ 991, 21(3), 660–674.

[7] Kasongo, S.M. Genetic Algorithm Based Feature Selection Technique for Optimal Intrusion Detection. Preprints 2021, 2021060710 (doi: 10.20944/preprints202106.0710.v1).)

[8] Chen WH, Hsu SH, Shen HP. Application of SVM and ANN for intrusion detection. Comput Oper Res. 2005; 32(10): 2617- 2634

[9] Roopa Devi E, Suganthe R., Enhanced transductive support vector machine classification with grey wolf optimizer cuckoo search optimization for intrusion detection system. Concurr Comput Pract Exp. 2020; 32(4):e4999

[10] Yan B, Han G. Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. IEEE Access. 2018; 6: 41238- 41248

[11] Ghanem K, Aparicio-Navarro FJ, Kyriakopoulos KG, Lambotharan S, Chambers JA. Support vector machine for network intrusion and cyber-attack

detection. Paper presented at: Proceedings of the Sensor Signal Processing for Defence Conference (SSPD). London, UK: IEEE; 2017:1-5.

[12] Ayhan, S., & Erdoğmuş, Ş. (2014). Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi. Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi, 9(1), 175-201.

[13] https://medium.com/analytics-vidhya/image-classification-using-machine-learning-support-vector-machine-svm-dc7a0ec92e01, erişim tarihi 10/06/2022

[14] Shen Y, Zheng K, Wu C, Zhang M, Niu X, Yang Y. An ensemble method based on selection using bat algorithm for intrusion detection. Comput J. 2018; 61(4): 526- 538

[15] Gao X, Shan C, Hu C, Niu Z, Liu Z. An adaptive ensemble machine learning model for intrusion detection. IEEE Access. 2019; 7: 82512- 82521

[16] Jaw, E., & Wang, X. (2021). Feature selection and ensemble-based intrusion detection system: an efficient and comprehensive approach. Symmetry, 13(10), 1764.

[17] Syarif, I., Zaluska, E., Prugel-Bennett, A.; Wills, G. Application of Bagging, Boosting and Stacking to Intrusion Detection. In Machine Learning and Data Mining in Pattern Recognition. MLDM 2012. Lecture Notes in Computer Science; Perner, P., Ed.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7376.

[18] Li, Y.; Chen, W. A Comparative Performance Assessment of Ensemble Learning for Credit Scoring. Mathematic 2020, 8, 1756

[19] Aburomman, A.; Reaz, M.B.I. A survey of intrusion detection systems based on ensemble and hybrid classifiers. Comput. Secur. 2017, 65, 135–152.

[20] Jaw, E., & Wang, X. (2021). Feature selection and ensemble-based intrusion detection system: an efficient and comprehensive approach. Symmetry, 13(10), 1764.

[21] https://www.datacamp.com/community/tutorials/adaboost-classifier-python, erişim tarihi 16/05/2022

[22] https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30 erişim tarihi 10/06/2022

[23] Zaman, S., El-Abed, M., & Karray, F. (2013, January). Features selection approaches for intrusion detection systems based on evolution algorithms. In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication (pp. 1-5).

[24] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.

[25] Storn, R., Price, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997).

[26] Zou D., Liu H., Gao L., Li S., Computers & Mathematics with Applications Volume 61, Issue 6, March, 2011 pp 1608–1623

[27] Madavan, N. K. (2003). Turbomachinery airfoil design optimization using differential evolution. In Computational Fluid Dynamics 2002 (pp. 585-590). Springer, Berlin, Heidelberg.

[28] Yildiz, A. R. (2013). Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations. Applied Soft Computing, 13(3), 1433-1439.

[29] Abdullah, M., Alshannaq, A., Balamash, A., & Almabdy, S. (2018). Enhanced intrusion detection system using feature selection method and ensemble learning algorithms. International Journal of Computer Science and Information Security (IJCSIS), 16(2), 48-55.

[30] Solanki S. , Gupta C., Rai K., A Survey on Machine Learning based Intrusion Detection System on NSL-KDD Dataset, International Journal of Computer Applications (0975 – 8887) Volume 176 – No. 30, June 2020

[31] XIAOYAN WANG, HANWEN WANG, A High Performance Intrusion Detection Method Based on Combining Supervised and Unsupervised Learning at IEEE Smart World, Ubiquitous Intelligence $ Computing Advanced $ Trusted Computing, Scalable Computing, Internet of People and Smart City Innovations in 2018.

[32] ELMER C. MATEL, ARIEL M.SISAN, Optimization of Network Intrusion Detection System using Genetic Algorithm with Improved Feature Selection Technique at Technological Institute of the Pilippines Quezon City, Phillipines 2019.

[33] LUKMAN HAKIM, RAHILLA FATMA NOVRIANDI Influence Analysis of Feature Selection to Network Intrusion Detection System Performance Using NSL-KDD Dataset at ICOMITEE 2019, October 16th-17th 2019, Jember, Indonesia in 2019.

[34] Bhumgara A., Pitale A., "Detection of Network Intrusions Using Hybrid Intelligent System" at International Conferences on Advances in Information Technology in 2019.

[35] Yulianto, A., Sukarno, P., & Suwastika, N. A. (2019). Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset. Journal of Physics: Conference Series, 1192, 12018.

[36] Tchakoucht A., T., & Mostafa, E. (2018). Building A Fast Intrusion Detection System For High-Speed-Networks: Probe and DoS Attacks Detection. Procedia Computer Science, 127, 521–530.

[37] Karatas, G., Demir, O., & Sahingoz, O.K. (2020). Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. IEEE Access, 8, 32150-32162.

[38] 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.

[39] Hajisalem, V., & Babaie, S. (2018). A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. Computer Networks, 136, 37-50.

[40] 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, 4(06), 209-213.

[41] Khammassi, C., & Krichen, S. (2017). A GA-LR wrapper approach for feature selection in network intrusion detection. computers & security, 70, 255-277.

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

[43] Moustafa, N., & Slay, J. (2015, November). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.

[44] Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1-3), 18-31.

[45] Zhang, J., Ling, Y., Fu, X., Yang, X., Xiong, G., & Zhang, R. (2020). Model of the intrusion detection system based on the integration of spatial-temporal features. Comput. Secur., 89.

[46] Gottwalt, F., Chang, E., & Dillon, T. (2019). CorrCorr: A feature selection method for multivariate correlation network anomaly detection techniques. Computers & Security, 83, 234-245.

[47] Thaseen, I.S.; Kumar, C.A. Intrusion detection model using fusion of chi-square feature selection and multi class SVM. J. King Saudi Univ. Comput. Inf. Sci. 2017, 29, 462–472.

[48] Syarif, I., Zaluska, E., Prugel-Bennett, A., Wills, G. Application of Bagging, Boosting and Stacking to Intrusion Detection. In Machine Learning and Data Mining in Pattern Recognition. MLDM 2012. Lecture Notes in Computer Science; Perner, P., Ed.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7376.

[49] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.

[50] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). Ieee.

[51] https://kdd.ics.uci.edu/databases/kddcup99/, erişim tarihi 13/06/2022

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

[53] Song J., Takakura H., Okabe Y., Description of Kyoto University Benchmark Data, (2010) 10–12.

[54] Ring M., Wunderlich S., Technical report CIDDS-001 data set, 16 (2017) 361–369.

[55] Ring M., Wunderlich S., Grüdl D., Landes D., Hotho A., Flow-based benchmark data sets for intrusion detection, in: Eur. Conf. Inf. Warf. Secur. ECCWS, 2017, pp. 361–369.

[56] Verma, A., & Ranga, V. (2018). Statistical analysis of CIDDS-001 dataset for network intrusion detection systems using distance-based machine learning. Procedia Computer Science, 125, 709-716.

[57] Ranga, V. (2018). On evaluation of Network Intrusion Detection Systems: Statistical analysis of CIDDS-001 dataset using Machine Learning Techniques

[58] Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. computers & security, 31(3), 357-374.

[59] Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2015). Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials, 18(1), 184-208.

[60] Thanthrige, U. S. K. P. M., Samarabandu, J., & Wang, X. (2016, May). Machine learning techniques for intrusion detection on public dataset. In 2016 IEEE Canadian conference on electrical and computer engineering (CCECE) (pp. 1-4). IEEE.

[61] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp, 1, 108-116.

[62] Intrusion Detection Evaluation Dataset (CICIDS2017), (n.d.). https://www.unb.ca/cic/datasets/ids-2017.html, erişim tarihi 13/06/2022

[63] CSE-CIC-IDS2018 on AWS, (n.d.). https://www.unb.ca/cic/datasets/ids-2018.html, erişim tarihi 13/06/2022

[64] Zhang, C. L., Yang, G. H., & Lu, A. Y. (2021). Resilient observer-based control for cyber-physical systems under denial-of-service attacks. Information Sciences, 545, 102-117.

[65] Virupakshar, K. B., Asundi, M., Channal, K., Shettar, P., Patil, S., & Narayan, D. G. (2020). Distributed denial of service (DDoS) attacks detection system for OpenStack-based private cloud. Procedia Computer Science, 167, 2297-2307.

[66] Barki, L., Shidling, A., Meti, N., Narayan, D. G., & Mulla, M. M. (2016, September). Detection of distributed denial of service attacks in software defined networks. In 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 2576-2581). IEEE.

[67] Najafabadi, M.M., Khoshgoftaar, T.M., Kemp, C., Seliya, N., & Zuech, R. (2014). Machine Learning for Detecting Brute Force Attacks at the Network Level. 2014 IEEE International Conference on Bioinformatics and Bioengineering, 379-385.

[68] Stiawan D. , Idris M.Y., Malik R.F., Nurmaini S., Alsharif N., Budiarto R., Investigating brute force attack patterns in IoT network, J. Electr. Comput. Eng.(2019), https://doi.org/10.1155/2019/4568368.

[69] Boyd S.W., Keromytis A.D., SQLrand: preventing SQL injection attacks, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). (2004), https://doi.org/10.1007/978-3-540-24852-1_21.

[70] Halfond, W. G., Viegas, J., & Orso, A. (2006, March). A classification of SQL-injection attacks and countermeasures. In Proceedings of the IEEE international symposium on secure software engineering (Vol. 1, pp. 13-15). IEEE.

[71] Mukkamala, S., Janoski, G., & Sung, A. (2002, May). Intrusion detection using neural networks and support vector machines. In Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290) (Vol. 2, pp. 1702-1707). IEEE.

[72] https://www.kaggle.com/code/kbrabestezcan/final-final-project, erişim tarihi 23/04/2022

[73] Maher, M., Ibrahim, Z., & Al-Safi, A. (2021). Using A Hybrid Algorithm and Feature Selection for Network Anomaly Intrusion Detection. Journal of Mechanical Engineering Research and Developments, 44, 253–262.

[74] Ingre, B., & Yadav, A. (2015, January). Performance analysis of NSL-KDD dataset using ANN. In 2015 international conference on signal processing and communication engineering systems (pp. 92-96). IEEE.

[75] Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering (Vol. 5). Technical report, Ver. 2.3 EBSE Technical Report. EBSE.

[76] Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks,188,10784