T.C.
MİLLİ SAVUNMA ÜNİVERSİTESİ
BARBAROS DENİZ BİLİMLERİ VE MÜHENDİSLİĞİ ENSTİTÜSÜ
DENİZ BİLGİ SİSTEMLERİ MÜHENDİSLİĞİ ANABİLİM DALI
SİBER GÜVENLİK YÜKSEK LİSANS PROGRAMI

AĞ TABANLI BİLGİSAYAR SİSTEMLERİNE YÖNELİK TEHDİTLERİN/ SALDIRILARIN DENETİMLİ YAPAY ÖĞRENME İLE SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

ERCAN KURU
18100501

TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ MUSA MİLLİ
EŞ DANIŞMAN: DR. TOLGA ÖNEL

İSTANBUL
OCAK 2022

T.C.
MİLLİ SAVUNMA ÜNİVERSİTESİ
BARBAROS DENİZ BİLİMLERİ VE MÜHENDİSLİĞİ ENSTİTÜSÜ
DENİZ BİLGİ SİSTEMLERİ MÜHENDİSLİĞİ ANABİLİM DALI
SİBER GÜVENLİK YÜKSEK LİSANS PROGRAMI

AĞ TABANLI BİLGİSAYAR SİSTEMLERİNE YÖNELİK TEHDİTLERİN/ SALDIRILARIN DENETİMLİ YAPAY ÖĞRENME İLE SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

ERCAN KURU
18100501

TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ MUSA MİLLİ
EŞ DANIŞMAN: DR. TOLGA ÖNEL

İSTANBUL
OCAK 2022

T.C.
MİLLİ SAVUNMA ÜNİVERSİTESİ
BARBAROS DENİZ BİLİMLERİ VE MÜHENDİSLİĞİ ENSTİTÜSÜ
DENİZ BİLGİ SİSTEMLERİ MÜHENDİSLİĞİ ANABİLİM DALI
SİBER GÜVENLİK YÜKSEK LİSANS PROGRAMI

AĞ TABANLI BİLGİSAYAR SİSTEMLERİNE YÖNELİK TEHDİTLERİN/ SALDIRILARIN DENETİMLİ YAPAY ÖĞRENME İLE SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

ERCAN KURU
18100501

Tezin Enstitüye Verildiği Tarih: 26 Kasım 2021
Tezin Savunulduğu Tarih: 04 Ocak 2022

Tez Oy birliği / Oy çokluğu ile başarılı bulunmuştur.

Tez Danışmanı : Dr.Öğr.Üyesi Musa MİLLİ
Eş Danışman : Dr.Tolga ÖNEL
Jüri Üyeleri : Dr.Öğr.Üyesi Musa MİLLİ
Dr.Öğr.Görevlisi Sultan Nezihe TURHAN
Dr.Öğr.Görevlisi Elif BOZKAYA

İSTANBUL
OCAK 2022

ÖZGÜNLÜK RAPORU

Tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve ç) Sonuç kısımlarından oluşan toplam 71 sayfalık kısmına ilişkin, 02/12/2021 tarihinde şahsım tarafından iThenticate adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan özgünlük raporuna göre, tezimin benzerlik oranı % 16'dır.

Uygulanan filtrelemeler:

1- Kaynakça hariç
2- Alıntılar hariç
3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Millî Savunma Üniversitesi Barbaros Deniz Bilimleri ve Mühendisliği Enstitüsü Lisansüstü Tez Çalışması Özgünlük Raporu Alınması ve Kullanılması Uygulama Usul ve Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Ercan KURU
04/01/2022
İmza

ETİK BEYAN

Millî Savunma Üniversitesi Enstitüleri Lisansüstü Tez Hazırlama Kılavuzu'nda yer alan kurallara uygun olarak hazırladığım bu tez çalışmasında; tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu, bildirir; aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Bu tezdeki düşünce, görüş, varsayım, sav veya tezler bana aittir; Millî Savunma Bakanlığı, Türk Silahlı Kuvvetleri, Deniz Kuvvetleri Komutanlığı, Millî Savunma Üniversitesi ve Barbaros Deniz Bilimleri ve Mühendisliği Enstitüsü sorumlu tutulamaz.

Ercan KURU
04/01/2022
İmza

Sevgili eşim ve oğluma

ÖNSÖZ ve TEŞEKKÜR

Lisansüstü eğitimimin ilk adımı olan bu çalışma süresince gerek akademik gerekse diğer konularda beni her daim destekleyen ve kıymetli zamanları ile tecrübelerini esirgemeyen başta danışmanlarım Dr.Öğr.Üyesi Musa MİLLİ, Dr. Tolga ÖNEL ve Dr. Mehmet Bilge Kağan ÖNAÇAN'a, öğretileri ve verdikleri destekleri ile Dr.Öğr.Üyesi Ertan YAKICI ve Dr. M.Batuhan GÜNDOĞDU'ya şükranlarımı sunarım.

Ayrıca,

Çalışmam boyunca bana destek olan değerli komutanlarım ile mesai arkadaşlarıma,

Her koşulda yanımda olan ve beni motive eden hayat arkadaşım Gonca KURU'ya ve

Hayatım boyunca bana güvenen ve beni destekleyen aileme

Teşekkür ederim.

İstanbul; Ocak 2022 Ercan KURU

İÇİNDEKİLER

Sayfa

TABLOLAR LİSTESİ

Sayfa

ŞEKİLLER LİSTESİ

KISALTMALAR

ANN : Artificial Neural Network
ARFF : Attribute-Relation File Format
BFS : Best First Search
CFS : Correlation-based Feature Selection
CSV : Comma Seperated Value
DARPA : Defense Advanced Research Projects Agency
DN : Doğru Negatif
DoS : Denial of Service
DP : Doğru Pozitif
ECLAT : Equivalence Class Transformation
IDS : Intrusion Detection System
IEC : International Electrotechnical Commission
ISO : International Organization for Standardization
KDD CUP-99 : Knowledge Discovery and Data Mining Tools Competition 1999
LibSVM : Library for Support Vector Machines
MATLAB : Matrix Laboratory
NATO : North Atlantic Treaty Organization
NSL KDD : Network Security Laboratory Knowledge Discovery and Data Mining
R2L : Remote to Local
STS : Saldırı Tespit Sistemi
TDK : Türk Dil Kurumu
U2R : User to Root
URL : Uniform Resource Loader
WEKA : Waikato Environment for Knowledge Analysis
YP : Yanlış Pozitif
YN : Yanlış Negatif

ÖZ

Ağ Tabanlı Bilgisayar Sistemlerine Yönelik Tehditlerin/ Saldırıların Denetimli Yapay Öğrenme ile Sınıflandırılması

Ercan KURU

Milli Savunma Üniversitesi, Barbaros Deniz Bilimleri ve Mühendisliği Enstitüsü

İstanbul, Ocak 2022

Günümüzde gelişen teknoloji ile birlikte bilgisayar kullanımı artmaktadır. Bilgisayar kullanımındaki bu artış, bilgisayar sistemlerine yönelik saldırı sayısının artmasına ve saldırıların çeşitlenmesine sebep olmaktadır. Bu durum bilgisayarda işlenen verilerin korunmasının ve bilgi güvenliği kavramının önemini gözler önüne sermektedir. Bilgisayar sistemlerinin korunmasında önemli yer tutan saldırı tespit sistemlerinin çalışma prensibi sayesinde, bilgisayarlara ve bilgisayar ağlarına yönelik saldırılar henüz sistemlere erişmeden tespit edilebilmektedir. Artan saldırı çeşitliliği de göz önüne alındığında, saldırı tespit sistemlerinin yapay öğrenme ile geliştirilmesi son dönemde birçok araştırmaya konu olmaktadır. Denetimli ve denetimsiz yapay öğrenme, ayrı ayrı özelliklere sahip olmakla beraber, kullanıldıkları alanlara farklı katkılar sağlamaktadırlar. Bu tez kapsamında, ağ trafiğini simüle etmek maksadıyla daha önce yapılan çalışmalarda en sık kullanılan veri setlerinden birisi olan NSL KDD veri seti, WEKA uygulamasında yer alan bir takım denetimli ve denetimsiz öğrenme algoritmalarına uygulanmıştır. Çıkan sonuçlar belirli kriterler altında değerlendirildiğinde, saldırı tespitinde denetimli yapay öğrenme algoritmalarının daha doğru, denetimsiz öğrenme algoritmalarının ise daha hızlı sonuç verdiği tespit edilmiştir.

Anahtar Sözcükler: Saldırı Tespit Sistemi, Denetimli Öğrenme, Denetimsiz Öğrenme, Bilgi Güvenliği, Boyut Azaltma.

Bilim Kodu : 92431
Sayfa Sayısı : 71
Tez Danışmanı : Dr.Öğr.Üyesi Musa MİLLİ
Eş Danışman : Dr. Tolga ÖNEL

ABSTRACT

Classification of Threats/ Attacks Against Network-based Computer Systems with Supervised Machine Learning

Ercan KURU

National Defence University, Barbaros Naval Sciences and Engineering Institute

İstanbul, January 2022

With the developing technology, number of people who use computers are increasing nowadays. This increase in computer use causes an increase in the variety of attacks and the number of attacks against computer systems. This situation reveals the importance of the protection of data processed on the computers and the concept of information security. Thanks to the working principle of intrusion detection systems, which have an important place in the protection of computer systems, attacks against computers and computer networks can be detected before they effect systems. Considering the increasing variety of attacks, the development of attack detection systems with machine learning has been the subject of many studies recently. Although supervised and unsupervised machine learning have separate features, they make different contributions to the areas in which they are used. Within the scope of this study, NSL KDD data set, one of the most frequently used data sets in previous studies to simulate network traffic, was applied to a number of supervised and unsupervised learning algorithms in the WEKA application. When the results are evaluated under certain criterias, it has been determined that supervised learning algorithms give more accurate results, where unsupervised learning algorithms give faster results in the detection of attacks.

Keywords: Intrusion Detection System, Supervised Learning, Unsupervised Learning, Information Security, Dimensionality Reduction.

Science Code : 92431

Pages : 71

Supervisor : Asst.Prof. Musa MİLLİ

Co-Supervisor : Dr. Tolga ÖNEL

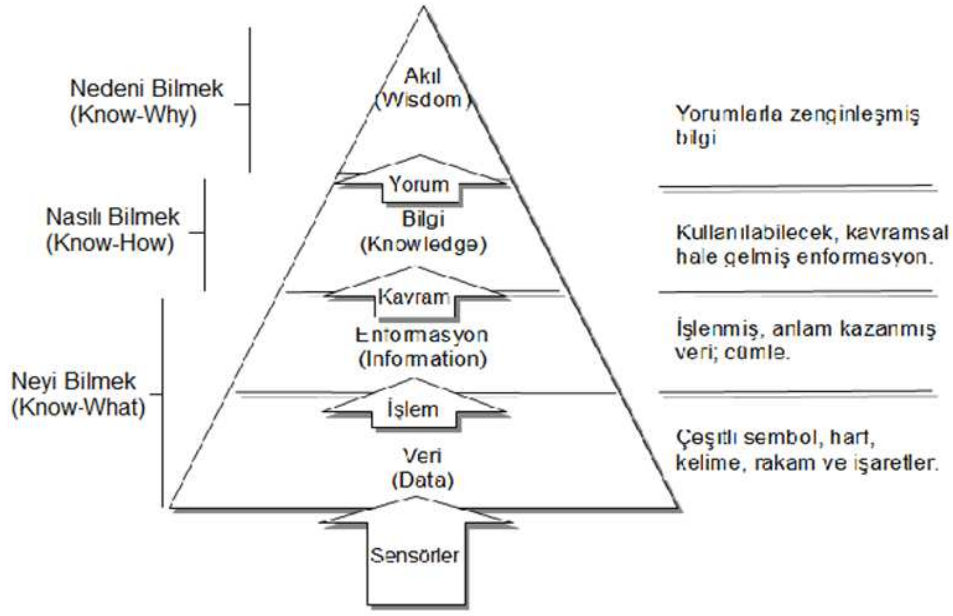
1. GİRİŞ

İnternet ve bilgisayar kullanımı gün geçtikçe her insan için ihtiyaç haline gelmekte ve hemen hemen hayatımızın her alanında bu iki tanım ile karşılaşmaktayız. 2021 yılı Mart ayı verilerine göre yeryüzünde yaşayan yaklaşık sekiz milyar insanın % 66'sı internet kullanmaktadır (World internet usage and population statistics, 2021). Kullanım alanları ve amaçlarına bakıldığında internet, insanların gündelik hayatlarında önemli bir yer kaplamaktadır.

Kuzey Atlantik Antlaşması Örgütü (NATO-North Atlantic Treaty Organization) interneti; hükümetler için kritik bir ulusal kaynak, ulusal alt yapıların hayati bir parçası ve sosyo-ekonomik büyüme ile kalkınmanın kilit itici gücü olarak tanımlamaktadır (Klimburg, 2012). Artan internet kullanımına bağlı olarak zararlı yazılımların sayısı da gün geçtikçe artmakta ve çeşitlenmektedir.

İnternet ortamında yaygınlaşan bu durum göz önüne alındığında, bilgisayarda veya bilgisayar ağlarında saklanan/ işlenen bilgiler ve bu bilgilerin güvenliği çok önemli bir husus olarak karşımıza çıkmaktadır. Dünya üzerinde birçok kurum/ kuruluş, bilgi güvenliğini sağlamak maksadıyla yazılımlar ve yöntemler geliştirmektedir. Bilgi güvenliğini sağlamak maksadıyla geliştirilen bu çözümlerin güvenilirliği ve performansı, kullanıcıların bu çözümleri tercih etmesindeki başlıca nedenlerdir.

Bilgi kavramı, İngilizce'de "data", "information" ve "knowledge" kelimeleri ile ifade edilirken, Türkçe'de ise "bilgi" kelimesi ile ifade edilmektedir. TDK sözlüğünde "bilgi" tanımı, "İnsan aklının erebileceği olgu, gerçek ve ilkelerin bütünü, bili, malumat." olarak yapılmaktadır ("Bilgi", 2021). Bilgi kavramının İngilizce'de farklı kelimeler kullanılarak ifade edilmesinin sebebi kelimelerin anlamsal açıdan farklılık içermesinden kaynaklanmaktadır. İngilizce'deki karşılıkları olarak kullanılan "veri" ve "enformasyon" kelimeleri çoğu zaman "bilgi" ile karıştırılmaktadır. Aslında bu karışıklık, dilimizde "veri" ve "enformasyon" kelimelerinin anlamlarının tam olarak oturmamış olmasından kaynaklanmaktadır.



Şekil 1.1: Bilgi Hiyerarşisi.
(Önaçan, 2015).

Şekil 1.1'de sunulan bilgi hiyerarşisi (bilgi piramidi), bilgi kavramını daha kolay anlayabilmek adına bize yardımcı olmaktadır. Bilgi hiyerarşisinin kaynağı olarak şair T.S. Eliot tarafından 1934 yılında yazılan "The Rock" isimli şiir gösterilmektedir (Sharma, 2008). Bilgi kavramının temelinde hiyerarşiden anlaşılacağı üzere veri yer almaktadır.

Veri, sensörler vasıtasıyla toplanan, çeşitli sembol, harf, rakam ve işaretlerden oluşan ve kendi başına bir anlam ifade etmeyen değerlerdir. Enformasyon, işlenmiş ve anlam kazanmış veriler topluluğudur. Bilgi, kullanılabilecek ve kavramsal hale gelmiş enformasyondan oluşturulan çıkarımlardır (Avcı ve Avcı, 2004; Bellinger ve diğ., 2004; Kocabıyık, 2005; Kurgun, 2006; İlter, 2011; Önaçan, 2015).

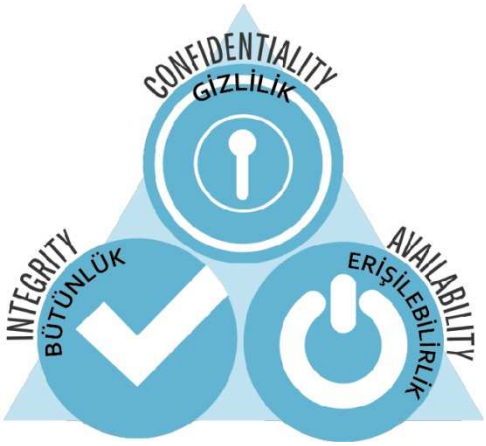
Bilgi hiyerarşisinden de anlaşılacağı üzere veri olmadan ne enformasyon olur ne de bilgi. Enformasyon, verinin işlenmiş halidir. Veri ve enformasyon bize neyi bildiğimizi gösterir. Bilgi, enformasyonun kavramsal halidir. Bize nasılı bildiğimizi gösterir. Hiyerarşinin en tepesinde yer alan kavram ise akıldır. Akıl, bilginin yorum ile zenginleştirilmesiyle ortaya çıkar ve bilgiden yapılan çıkarımı gösterir. Örneğin, on sayısı başlı başına bir anlam ifade etmeyip bir veridir. On adet bilgisayar ise enformasyondur. Bu on bilgisayarın işlem kapasitesi bilgidir. On bilgisayarın işlem kapasitesini kullanarak yapacağımız işlere karar vermek ise akıldır.

Günümüzde en değerli varlık bilgidir. Bilginin depolanmasından, işlenmesine ve güvenliğinin sağlanmasına birçok alanda faaliyet gösteren kurum ve kuruluşlar bulunmaktadır. Teknoloji çağı öncesinde sözle veya yazılı olarak nesilden nesile aktarılan bilgiler, günümüzde bilgisayarlar ve internet ile çok kolay bir biçimde toplumdan topluma aktarılabilmektedir.

Bilgiye kolay ulaşmak her ne kadar bir avantaj olsa da, doğru ve güvenilir bilgiye ulaşmak ve bilginin güvenliğini sağlamak halihazırda birçok araştırmanın odak noktası olarak yer almaktadır. Eskiden dolaplarda muhafaza edilen bilgiler günümüzde dolapların yanında bilgisayarlarda hatta bulut bilişim sayesinde devasa serverlarda muhafaza edilmektedir. Bu durum bilgisayarda ve bilgisayar ağlarında işlenen bilginin güvenliğinin ne kadar önemli olduğunu gözler önüne sermektedir. Nitekim günümüzde birçok kurum ve kuruluş, bilgi güvenliğini sağlamak maksadıyla çeşitli önlemler almaktadır.

Bilgi güvenliği, bilginin ve bilgi sistemlerinin; yetkisiz erişime, yetkisiz kullanıma, yetkisiz değiştirilmeye ve yetkisiz ortadan kaldırılmasına karşı korunmasıdır (Andress, 2011). Bilgi güvenliğinin tanımından da anlaşılacağı üzere, bilginin gerçek halini değiştirmeye ve ortadan kaldırmaya yönelik her türlü faaliyet bilgi güvenliği konusunu kapsamaktadır.

Bilgi güvenliğinin sağlanması için bilginin işlendiği her türlü platformda, gizliliğin (confidentiality) sağlanması, bütünlüğün (integrity) korunması ve bilginin olduğu haliyle erişilebilir (availability) olması gerekmektedir. Genel olarak bilgi güvenliğinin temelini oluşturan bu üç öğe, İngilizce karşılıklarının baş harflerinden oluşan CIA (Confidentiality-Integrity-Availability) üçgeni (Şekil 1.2) olarak adlandırılmaktadır (Solomon ve Chapple, 2005). Bu üç öğenin dışındaki diğer unsurlar ise; güvenilirlik (reliability), inkar edememe (non-repudiation), kimlik sınaması (authentication), yetkilendirme (authorization) ve izlenebilirlik (auditing)'tir (Şekil 1.3).

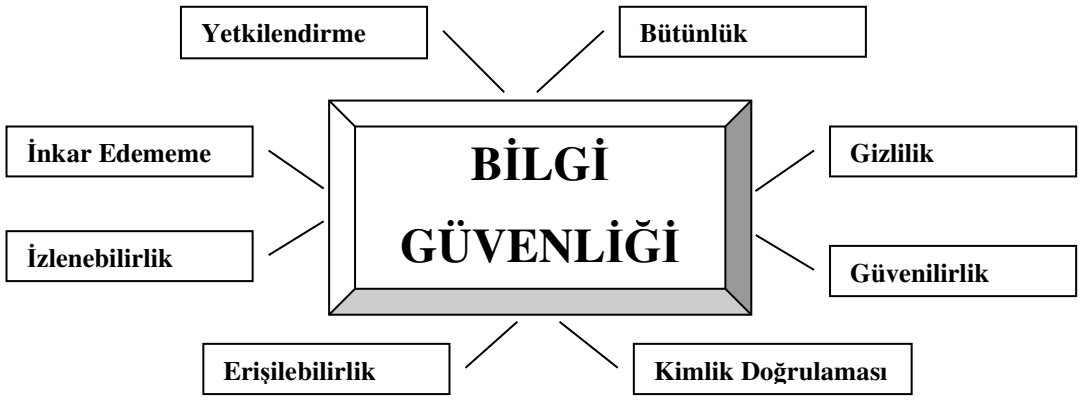


Şekil 1.2: CIA Üçgeni.

Gizlilik (Confidentiality), bilgi güvenliğinin en önemli unsurudur. Hiçbir birey kendisine ait gizli bilgilerin başkalarının eline geçmesini ve bu bilgilerin yetkisiz kişilerce kullanılmasını istemez. Gizliliğin amacı, bilginin yetkisiz kişilerin eline geçmesinin engellenmesidir.

Bütünlük (Integrity), bilginin değiştirilmemesi, ortadan kaldırılmaması ve olduğu haliyle işlenmesidir. Bütünlüğün amacı, bilgiyi olması gerektiği şekilde saklamaktır.

Erişilebilirlik (Availability), bilginin her zaman ulaşılabilir olmasını ifade eder. Erişebilirliğin amacı, kullanıcıların erişmek istedikleri veriye yetkileri dahilinde istedikleri zaman erişebilmesidir.



Şekil 1.3: Bilgi Güvenliğinin Unsurları.

Güvenilirlik (Reliability), bilgi sistemlerinin beklenen davranışı ile elde edilen sonuçlar arasındaki tutarlılık durumudur. Sistemin kendisinden beklenen görevi her yerine getirdiğinde tutarlı olması, benzer görevler arasında farklılık göstermemesidir.

İnkar Edememe (Non-repudiation), bilgiyi gönderen ve alan kullanıcı arasında ortaya çıkabilecek anlaşmazlıkları en aza indirmeyi amaçlar. İki kullanıcı arasında bir aktarım yapılmışsa ne gönderen, ne de alıcı yaptığı işlemi inkar edememelidir.

Kimlik Doğrulaması (Authentication), kullanıcının kendisini kanıtlaması için ilk yapması gereken işlemdir. Yani, kullanım sırasında cihaz veya kullanıcının kimliğinin doğrulanmasıdır. Kimlik doğrulamasında, kullanıcının sahip olduğu kullanıcı adının sistemde kayıtlı olup olmadığı kontrol edilir ve kullanıcıya verilen parola kontrol edilerek doğrulama işlemi yapılır. Doğrulama sağlanırsa kullanıcıya sisteme giriş izni verilir.

Yetkilendirme (Authorization), kimlik doğrulaması gerçekleşen kullanıcıların sisteme, programa veya ağa hangi yetkilerle erişim hakkına sahip olduklarını ifade eder. Sistemde kayıtlı kullanıcılar gruplanarak, bu gruplara çeşitli yetkiler verilir. Kullanıcı atandığı grubun bütün yetkilerine sahiptir. Güvenliğin tam olarak sağlanabilmesi için kullanıcılara gerektiği kadar yetki verilmelidir.

İzlenebilirlik (Auditing), bir sorun ile karşılaşıldığında sorunun tespitinin sağlanabilmesi için kullanılır. Sistemde bulunan kullanıcıların yaptıkları işlemler ve işlem saatleri kayıt altına alınır. Bir sıkıntı yaşandığında, bu kayıtlardan sorun anlaşılmaya ve çözülmeye çalışılır.

Bilgi güvenliği uygulamaları yukarıda bahsedilen ilkeleri sağlamak maksadıyla çalışırlar. Yani gerek antivirüs programlarında, gerek güvenlik duvarlarında, gerek ise saldırı tespit sistemlerinde amaç bilgi güvenliğine zarar gelmesinin önüne geçmektir. Ancak her uygulama bunu farklı yöntemlerle yerine getirmektedir. Örneğin, antivirüs programları bilgi sistemlerine girebilecek virüs, truva atı vb. zararlı yazılımları tespit ederek etkisiz hale getirmeye çalışırken, güvenlik duvarı, bilgisayarın yetkisiz kullanımına kısıtlama getirerek olası risklerin önüne geçmektedir (Güvenlik Duvarı Nedir/Ne İşe Yarar?, 2021). Saldırı tespit sistemleri ise, ağ trafiğinin davranışlarını inceleyerek gelen verinin normal olup olmadığına karar vermekte kullanılmaktadır. Böylelikle bilgi güvenliği zafiyeti yaratacak tehdit erkenden tespit edilebilmektedir.

Bilgisayar dünyasında tehditlerin artması ve çeşitlilik göstermesiyle birlikte güvenlik yazılımları da çeşitlenmiştir. Zararlı yazılımlar (malicious software), bilgisayarda çalışan diğer yazılımların davranışlarını etkileyerek olması gerekenden farklı şekilde davranmalarını sağlayan veya etkilediği yazılımlara zarar veren yazılımlardır (Kramer ve Bradfield, 2009). Güvenlik yazılımları, zararlı yazılımlara bağlı olarak çeşitlilik göstermektedir. Örneğin; virüs, truva atı gibi zararlı yazılımlara karşı antivirüs programları kullanılırken, casus yazılımlara karşı anti-spyware yazılımlar kullanılabilir. Bu örneği çeşitli kötücül yazılımlara bağlı olarak geliştirmek mümkündür. Her tehdide karşı ayrı bir yazılım kullanmak bilgisayarın performansını düşüreceğinden günümüzde üretilen güvenlik yazılımları birden fazla güvenlik tehdidine karşı koyacak şekilde geliştirilmektedir.

Birden fazla bilgisayardan oluşan bilgisayar ağlarının güvenliği ise saldırı tespit sistemleri ile sağlanmaktadır. Saldırı tespit sistemleri ağ trafiğinin davranışlarını inceleyerek gelen verinin zararlı yazılım olup olmadığını tespit etmektedir. Ağ trafiğinin davranışı geliştirilen algoritmalar ile sınıflandırılmaktadır. Bu aşamada ise yapay öğrenme devreye girmektedir. Hayatımızın birçok alanında kullanılan akıllı sistemlerin temelini oluşturan yapay öğrenme, bilgisayar sisteminde programlanan yazılımın karşılaştığı problemi, belirli bir veri setine veya daha önce edindiği tecrübelere istinaden çözümlemesi ile ifade edilebilir (Alpaydın, 2010).

Yapay öğrenme, genel olarak Denetimli (Supervised) Yapay Öğrenme, Denetimsiz (Unsupervised) Yapay Öğrenme ve Pekiştirilmiş (Reinforcement) Yapay Öğrenme olarak üç başlıkta incelenmektedir (Simeone, 2018). Probleme göre seçilen öğrenme türü kullanılarak veriler işlenmekte ve çıkan sonuçlar değerlendirilmektedir.

Bu tez kapsamında, daha iyi ve etkili bir saldırı tespit sistemi geliştirebilmek için kullanılması gereken öğrenme algoritması, KDD CUP-99 veri setinden türetilen NSL KDD veri seti ve WEKA uygulaması kullanılarak tespit edilmeye çalışılmıştır. Giriş kısmına müteakip ikinci bölümde literatürde mevcut çalışmalar, üçüncü bölümde saldırı tespit sistemleri incelenmiş, dördüncü bölümde yapılacak çalışmanın metodolojisi anlatılmış, beşinci bölümde ise yapılan uygulama ile ilgili bilgi verilmiş ve altıncı bölümde uygulama sonucu ortaya çıkan veriler değerlendirilmiştir.

2. LİTERATÜR TARAMASI

Tez çalışmasının bu kısmında, literatürde mevcut çalışmalar taranarak daha önce yapay öğrenme algoritmaları kullanılarak yapılan çalışmalar incelenmiştir.

Musa ve arkadaşları tarafından yapılan çalışmada, 2010-2015 yılları arasında yazılan araştırma makalelerinde kullanılan veri setleri ile ilgili olarak sunulan tabloda toplamda 28 araştırma makalesinden 14'ünde NSL KDD veri setinin kullanıldığı, NSL KDD veri setini 4 araştırma makalesi ile KDD CUP-99 veri setinin izlediği belirtilmiştir. Araştırmalarda kullanılan diğer veri setleri ise; CICIDS2017, UNSWNB-15, UGR-16 ve Kyoto2006+ olarak sıralanmıştır (Musa ve diğ., 2020).

Halimaa ve Sundarakantham tarafından yapılan çalışmada, geliştirilmek istenen saldırı tespit sisteminin bilinen saldırıların tespiti kadar bilinmeyen saldırıların tespit edilmesinde de yeterli olması gerektiğine vurgu yapılmış, NSL KDD veri setinden rastgele olarak seçilen 19000 veri ile Destek Vektör Makineleri ve Naive Bayes algoritmaları kullanılarak gerçekleştirilen çalışma neticesinde; gerek herhangi bir ön işleme tabi tutulmadan, gerekse ön işleme tabi tutulmuş haliyle Destek Vektör Makineleri algoritmasının daha yüksek oranda doğru sınıflandırma yaptığı tespit edilmiştir (Halimaa ve Sundarakantham, 2019).

Taher ve arkadaşları yaptıkları çalışmada; Lineer Regresyon, Destek Vektör Makineleri, Genetik Algoritma, En Yakın Komşu Algoritması, Naive Bayes ve Karar Ağacı algoritmalarının anomali tabanlı saldırı tespit sistemlerinde sık kullanılan algoritmalar olduğundan bahsetmiştir. Ayrıca özellik seçiminin (feature selection) veri boyutunu azaltmada yapay öğrenmenin bir parçası olduğunu belirterek, NSL KDD veri setinin %20'lik kısmı olan 25191 veri ile Destek Vektör Makineleri ve Yapay Sinir Ağları algoritmaları ile yaptıkları çalışmada, özellik seçici algoritma ile birlikte kullandıkları Yapay Sinir Ağları algoritmasının Destek Vektör Makineleri algoritmasına kıyasla daha yüksek oranda doğru sınıflandırma yaptığını tespit etmişlerdir (Taher ve diğ., 2019).

Perez ve arkadaşları saldırı tespit sistemlerinde kullanılmak üzere ortaya koyacakları kombine algoritma üzerine yaptıkları çalışmada, 2010-2015 yılları arasında ortaya konulan çalışmalarda; kombine algoritma kullanan 50, yeni sınıflandırıcı öneren 45, boyut azaltma kullanan 38 ve özellik seçici kullanan 34 çalışmanın olduğundan bahsetmişlerdir. Bahse konu çalışmalarda kullanılan algoritmalar incelendiğinde ise, Destek Vektör Makineleri algoritmasının 24 çalışma ile en sık kullanılan algoritma olduğu, Destek Vektör Makineleri algoritmasını, 19 çalışma ile Karar Ağacı ve 16 çalışma ile Genetik Algoritmanın izlediği, En Yakın Komşu, k-merkezli kümeleme ve Naive Bayes algoritmalarının dokuzar çalışma ile en çok çalışan algoritmalar arasında yer aldığı ve 8 çalışmada ise Yapay Sinir Ağları (Çok Katmanlı Algılayıcı) algoritmasının kullanıldığı görülmektedir (Perez ve diğ., 2017).

Yapılan bir başka çalışmada Panigrahi ve arkadaşları, saldırı tespit sistemlerinde kullanılan sınıflandırma algoritmaları ve veri setlerinden elde edilen sonuçları analiz etmişlerdir. Araştırmada NSL KDD, ISCXIDS2012 ve CICIDS2017 veri setleri ile elde edilen sonuçlar algoritma bazında tek tek gösterilmiştir. NSL KDD veri setinden alınan 7781 veri ile yapılan çalışmada, Çok Katmanlı Algılayıcı algoritmasının % 80,46, Lojistik Regresyon algoritmasının % 78,04, k-Yakın Komşu algoritmasının % 95,01 ve Karar Ağacı algoritmasının % 97,28 oranında doğru sınıflandırma yaptığı tespit edilmiştir (Panigrahi ve diğ., 2021).

Alkasassbeh ve Almseidin, KDD CUP veri setinden çıkardıkları 148758 veriden oluşan eğitim seti ve 60000 veriden oluşan test seti ile yaptıkları çalışmada; Karar Ağacı, Çok Katmanlı Algılayıcı ve Bayes algoritmalarının doğru sınıflandırma oranlarını ölçmüş ve Karar Ağacı algoritmasında % 93,11, Çok Katmanlı Algılayıcı algoritmasında % 91,9 ve Naive Bayes algoritmasında % 90,73 doğruluk oranına ulaşmışlardır (Alkasassbeh ve Almseidin, 2018).

Dhanabal1 ve Shantharajah yaptıkları çalışmada, Karar ağacı, Destek Vektör Makineleri ve Naive Bayes algoritmalarının performanslarını, NSL KDD veri setinin % 20'sini kullanarak test etmişlerdir. Yaptıkları çalışma neticesinde, Karar Ağacı algoritmasında % 97,9 ile % 99,8 arasında, Destek Vektör Makineleri algoritmasında % 91,4 ile % 98,8 arasında, Naive Bayes algoritmasında % 70,1 ile % 74,9 arasında değişen oranlarda doğru sınıflandırma oranı elde etmişlerdir (Dhanabal1 ve Shantharajah, 2015).

Kumar ve arkadaşları yaptıkları çalışmada, KDD CUP-99 veri setinden aldıkları 32640 veri ile Destek Vektör Makineleri ve k-Yakın Komşu algoritmalarının performanslarını ölçmüşlerdir. Diğer çalışmalardan farklı olarak bu çalışmada veriler hem oldukları haliyle hemde boyut azaltma işlemine tabi tutularak algoritmalarda kullanılmıştır. Elde edilen veriler neticesinde, Destek Vektör Makineleri algoritması % 86,88 doğru sınıflandırma oranına ulaşırken, boyut azaltma işlemine tabi tutulmuş verinin Destek Vektör Makineleri ile koşulması sonucu % 89,46 doğru sınıflandırma oranına ulaşıldığı görülmüştür. Çalışma kapsamındaki diğer algoritma olan k-Yakın Komşu algoritmasında ise doğru sınıflandırma oranı % 87,24 iken, boyut azaltma işlemine tabi tutulmuş olan veri ile algoritma koşulduğunda doğru sınıflandırma oranının % 90,07 olduğu gözlemlenmiştir. Araştırmacılar, boyut azaltma işlemine tabi tutulmuş veri ile kullanılan algoritmaların daha yüksek oranda doğru sınıflandırma yaptığı sonucuna ulaşmışlardır (Kumar ve diğ., 2020).

Hamid ve arkadaşları yaptıkları çalışmada, 4898430 veriden oluşan KDD CUP-99 veri setininin % 10'luk kısmını (494020 veri) ve WEKA programında yer alan bazı denetimli öğrenme algoritmalarını kullanmışlardır. Çalışmada öncelikle 494020 veri kullanılarak denetimli öğrenme algoritmaları koşulmuş, Karar Ağacı algoritmasında % 99,96, Çok Katmanlı Algılayıcı algoritmasında % 98,75, Logistik Regresyon algoritmasında % 99,94 ve k-Yakın Komşu algoritmasında % 99,94 doğru sınıflandırma oranına ulaşılmıştır. Ardından aynı veri seti kullanılarak boyut azaltma işlemi uygulanarak veri setindeki kırk bir özellik on bir özelliğe indirgenmiş, bu şekilde uygulama yapıldığında ise, Karar ağacı algoritmasında % 99,94, Çok Katmanlı Algılayıcı algoritmasında % 99,28 ve k-Yakın Komşu algoritmasında % 99,87 doğru sınıflandırma oranına ulaşılmıştır. Çalışma neticesinde, sınıflandırma algoritmalarının kırk bir özelliğin tümüne ihtiyaç duymadığını, dolayısıyla aynı veri üzerinde azaltılmış sayıda özellikle çalışarak daha iyi sonuçlar elde edebileceğini sonucuna ulaşılmıştır (Hamid ve diğ., 2016).

Ingre ve Yadav tarafından yapılan çalışmada ise, NSL KDD veri seti kullanılarak Çok Katmanlı Algılayıcı algoritmasının performans incelemesi yapılmıştır. Çalışma neticesinde kullanılan ağ yapısı ve özellik sayısına bağlı olarak % 75,3 ile % 81,2 arasında değişen doğru sınıflandırma oranına ulaşılmıştır (Ingre ve Yadav, 2015).

Devi ve arkadaşları KDD CUP-99 ve NSL KDD veri seti kullanarak yaptıkları çalışmada, denetimli ve denetimsiz öğrenme algoritmalarının çalışma prensiplerini açıklamış ve elde ettikleri sonuçları paylaşmışlardır. Çalışma neticesinde, KDD CUP-99 veri setindeki veriler ile Lojistik Regresyon algoritmasında % 79,7, Naive Bayes algoritmasında % 92,4, AdaBoost algoritmasında % 90,73 ve Çok Katmanlı Algılayıcı algoritmasında % 80,5 doğru sınıflandırma oranına ulaşmışlardır. Aynı algoritmalar ile NSL KDD veri seti üzerinde yaptıkları çalışma sonucunda ise, Lojistik Regresyon algoritmasında % 97,4, Naive Bayes algoritmasında % 89,5, AdaBoost algoritmasında % 89,3 ve Çok Katmanlı Algılayıcı algoritmasında % 88,9 doğru sınıflandırma oranına ulaşmışlardır (Devi ve Abualkibash, 2019).

Syarif ve arkadaşları yaprıkları çalışmada, NSL KDD veri seti ile denetimli öğrenme algoritmaları üzerine 10-bölmeli çapraz doğrulama yöntemi ile yaptıkları uygulamada k-Yakın Komşu algoritmasında % 99,44, Naive Bayes algoritmasında % 89,59 ve Karar Ağacı algoritmasında % 99,56 doğru sınıflandırma oranına ulaşmışlardır. Aynı algoritmaları kullanarak test seti destekli yöntem ile yaptıkları uygulamada ise k-Yakın Komşu algoritmasında % 62,84, Naive Bayes algoritmasında % 55,77 ve Karar Ağacı algoritmasında % 63,97 doğru sınıflandırma oranına ulaşmışlardır. Kümeleme algoritmaları ile yapılan uygulama neticesinde ise, k-merkezli kümeleme algoritmasında % 57,81, k-medoid algoritmasında % 76,71 ve Beklenti Maksimizasyonu (Expectation Maximization) kümeleme algoritmasında % 78,06 oranında doğru kümeleme sonucuna ulaşmışlardır. Araştırmacılar, k-merkezli kümeleme algoritmasının en hızlı, ancak en düşük doğruluk oranına sahip kümeleme algoritması olduğu sonucuna varmışlardır (Syarif ve diğ., 2012).

Deshmukh ve arkadaşları yaptıkları çalışmada, NSL KDD veri seti ile Naive Bayes, Gizli Naive Bayes ve Naive Bayes Ağacı algoritmalarının doğru sınıflandırma performanslarını karşılaştırmışlardır. Çalışma kapsamında, boyut azaltma ve özellik seçici algoritmaları kullanarak, Naive Bayes algoritmasında % 88,20, Gizli Naive Bayes algoritmasında % 93,40 ve Naive Bayes Ağacı algoritmasında % 94,60 doğru sınıflandırma oranına ulaşmışlardır. Çalışma neticesinde, yüksek sayıda özelliğe sahip veri setine uygulanan ön işlemlerin, algoritmaların performanslarını artırarak daha az sürede yüksek doğru sınıflandırma oranı elde etmesini sağlamışlardır (Deshmukh ve diğ., 2015).

Chabathula ve arkadaşları yaptıkları çalışmada, Temel Bileşenler Analizi (Principal Component Analysis) uygulanarak boyutu azaltılmış verinin sınıflandırma algoritmaları üzerindeki etkisini incelemişlerdir. KDD CUP-99 veri setinden elde ettikleri 150000 veri ile yaptıkları çalışmada, kırk bir özellik kullanılan veri setinin orijinal halinde Destek Vektör Makineleri algoritmasında % 99,63, k-Yakın Komşu algoritmasında % 99,96, Karar Ağacı algoritmasında % 99,96 ve Naive Bayes algoritmasında % 97,47 doğru sınıflandırma oranına ulaşmışlardır. Temel Bileşen Analizi algoritması kullanılarak boyut azaltma yöntemi uygulanmış altı özelliğe havi veride Destek Vektör Makineleri algoritmasında % 99,64, k-Yakın Komşu algoritmasında % 99,84, Karar Ağacı algoritmasında % 99,84 ve Naive Bayes algoritmasında % 96,66 doğru sınıflandırma oranına, on dört özelliğe havi veride ise, Destek Vektör Makineleri algoritmasında % 99,66, k-Yakın Komşu algoritmasında % 99,91, Karar Ağacı algoritmasında % 99,93 ve Naive Bayes algoritmasında % 97,05 doğru sınıflandırma oranına ulaşmışlardır. Yaptıkları çalışma neticesinde, elde ettikleri deneysel sonuçlar doğru sınıflandırma oranı açısından incelendiğinde, boyut azaltma yöntemi uygulanmış veri ile verinin orijinal hali arasında çok büyük farklılıklar olmadığı, ancak süre bakımından avantaj sağladığı sonucuna ulaşmışlardır (Chabathula ve diğ., 2015).

Duque ve Bin Omar yaptıkları çalışmada, NSL KDD veri seti ve k-merkezli kümeleme algoritmasını kullanmışlardır. Çalışma kapsamında küme sayısı (k); 11, 22, 44, 66 ve 88 olarak kabul edilerek algoritma koşulmuş ve küme sayısına bağlı olarak farklı sonuçlar elde etmişlerdir. Çalışma neticesinde k=11 için % 70,75, k=22 için % 81,61, k=44 için % 65,40, k=66 için % 61,30 ve k=88 için % 55,43 doğru sınıflandırma oranına ulaşılmıştır (Duque ve Bin Omar, 2015).

Singh ve Venkatesan yaptıkları çalışmada, denetimli ve denetimsiz öğrenme algoritmalarının kombine olarak kullanıldıkları çalışmaları incelemiş ve kendileri de benzer bir model ortaya koymuşlardır. NSL KDD veri seti ile yaptıkları çalışma neticesinde, k-Merkezli kümeleme ve Random Forest algoritmasını kullanarak %99,84 doğru sınıflandırma oranına ulaşmışlardır (Singh ve Venkatesan, 2018).

Mishra ve arkadaşları yaptıkları çalışmada, NSL KDD veri setinde saldırı olarak etiketlenen dört saldırı türünü (Hizmet Dışı Bırakma -Denial of Service (DoS)-, Ayrıcalıklı Kullanıcıya Erişim -User to Root (U2R)-, Yerel Kaynaklara Erişim -Remote to Local (R2L)-, Dinleme -Probing-) baz alarak öğrenme algoritmalarının

doğru sınıflandırma oranlarını, tekil ve kombine algoritmalar ile boyut azaltma yöntemi de dahil olmak üzere çeşitli farklı kombinasyonlarda incelemişlerdir. Çalışma neticesinde, bahse konu her dört saldırı türü için kombine algoritmaların tekil algoritmalara kıyasla daha yüksek oranda doğru sınıflandırma yaptığını, boyut azaltma tekniği kullanıldığında ise doğru sınıflandırma oranının, Hizmet Dışı Bırakma ve Dinleme saldırılarında artış, Yerel Kaynaklara Erişim ve Ayrıcalıklı Kullanıcıya Erişim saldırı türlerinde ise düşüş gösterdiğini ortaya koymuşlardır. Ayrıca çalışmada, NSL KDD veri setindeki özelliklerin hangilerinin hangi saldırı türünün tespiti için kullanılabileceğine dair bilgide verilmiştir. Bu kapsamda, Hizmet Dışı Bırakma saldırıları için, "Land", "Wrong fragment", "Service", "Duration", "Dst host same srv rate", "Same srv rate", "Dst bytes", "Flag" ve "Dst host count" özellikleri ayırt edici özellikler olarak ortaya çıkmaktayken, Ayrıcalıklı Kullanıcıya Erişim saldırıları için, "Num failed login", "Su attempted", "Is hot login", "Num shells", "Root Shell", "Num root", "Duration" ve "Service" özellikleri ayırt edici özellik olarak ortaya çıkmaktadır. Aynı şekilde, Yerel Kaynaklara Erişim saldırıları için, "Duration", "Service", "Src bytes", "Dst bytes", "Num failed login", "Is guest login", "Num compromised", "Num file creation", "Count", "Dst host count" ve "Dst host srv count" özellikleri ayırt edici özellikler olarak ortaya çıkmaktayken, Dinleme saldırıları için, "Duration", "Service, "Dst host same srv rate", "Flag", "Dst host count" ve "Dst host diff srv rate" özellikleri ayırt edici özellik olarak ortaya çıkmaktadır (Mishra ve diğ., 2018).

Yapılan literatür taraması neticesinde, farklı veri setleri ve algoritmalar kullanılarak saldırı tespit sistemlerine temel teşkil edecek uygun öğrenme algoritması/ algoritma kombinasyonunun tespit edilmeye çalışıldığı, ayrıca çalışmalarda kullanılan verilerin ön işleme tabi tutularak daha yüksek oranda doğru sınıflandırma yapan algoritma/ algoritma kombinasyonuna ulaşılmaya çalışıldığı gözlemlenmiş, araştırmacıların ağırlıklı olarak KDD CUP-99 ve NSL KDD veri setleri ile denetimli öğrenme algoritmalarını kullandıkları, boyut azaltma işlemine tabi tutulan verinin orijinal haline göre daha yüksek oranda doğru sınıflandırma yaptığı ve kombine algoritmaların tekil algoritmalara kıyasla daha iyi sonuçlar verdiği tespit edilmiştir.

3. SALDIRI TESPİT SİSTEMLERİ

3.1. Saldırı Tespit Sistemlerinin Tarihsel Gelişimi

Saldırı tespiti, bilgisayar sistemlerine ve bilgi güvenliğine yönelik herhangi bir saldırının tespit edilmesini ifade etmektedir. Saldırıdan korunma ise, saldırı tespitine ilave olarak saldırıya karşılık verilmesini ifade etmektedir. Saldırı tespiti, güvenlik açıklarından korunmanın ilk adımıdır. Saldırı tespit sistemleri, çeşitli sistemden ve ağ kaynağından bilgi toplayıp, topladıkları veriyi analiz ederek saldırıları tespit ederler (Taher ve diğ., 2019). Saldırı tespit sistemlerinin tarihine bakıldığında, bu yöndeki ilk çalışmaların 1960'lı yıllara rastladığı görülmektedir (Yost, 2016).

1970'lerin sonunda bilgisayar yöneticileri, sistem kayıtlarını yelpaze şeklinde katlanmış kağıtlara yazıyordu. Herhangi bir olağan dışı durumda, sistem yöneticileri bu kağıtları kullanarak sorunu tespit etmeye çalışıyorlardı. Depolamanın elektronik ortama geçerek ucuz hale gelmesi ve yaygınlaşması ile birlikte, veri analiz programları geliştirildi. Ancak, bu programlar yavaş çalıştığından genellikle kullanıcı yoğunluğunun az olduğu gece saatlerinde çalıştırılmaktaydı. 1990'lı yılların başında araştırmacılar veri analiz programlarını gerçek zamanlı hale getirdi ve bugün kullanılan saldırı tespit sistemlerinin ilk hali ortaya çıktı (Kemmerer ve Vigna, 2002).

Gerek kağıtların incelenmesi gerek veri analiz programlarının gece saatlerinde çalışıtırılması, sisteme karşı gerçekleştirilecek saldırıların ve olası bilgi güvenliği ihlalinin çok sonra tespit edilebildiğini ortaya koymaktadır. Nitekim gerçek zamanlı programların geliştirilmesiyle bu eksiklik giderilmiş ve başlangıçta tek bilgisayar temelinde de olsa saldırı tespit sistemi geliştirilmiştir.

3.2. Saldırı Tespit Sistemi Türleri

Saldırı tespit sistemleri, tek bilgisayar temelinde başlayıp günümüzde bünyesinde birçok cihaz bulunduran ağ sistemlerini koruyacak hale gelmiştir.

Saldırı tespit sistemlerinin gelişmesindeki bazı motivasyon kaynakları aşağıda olduğu gibidir:

- Yeni ağ sistemleri karmaşık yapıdadır ve bunun bir sonucu olarak hata vermeye meyillidirler. Bu hatalar da kötü niyetli kişilerce kullanılabilmektedir.

- Kullanılan ağ sistemleri bazı önemli savunma eksikliklerine sahiptir ve bu durum ağ sistemlerini saldırganların hedefi haline getirmektedir. Her ne kadar bu eksiklikler bazı araç ve yöntemlerle kapatılmaya çalışılsa da, tamamıyla eksikleri gidermek mümkün değildir.

- Ağ sistemlerinde, saldırıdan korunmaya yönelik sistemler olmasına rağmen tam korunma mümkün olmayabilir. Bunun bir sonucu olarak STS'lere olan ihtiyaç artmaktadır.

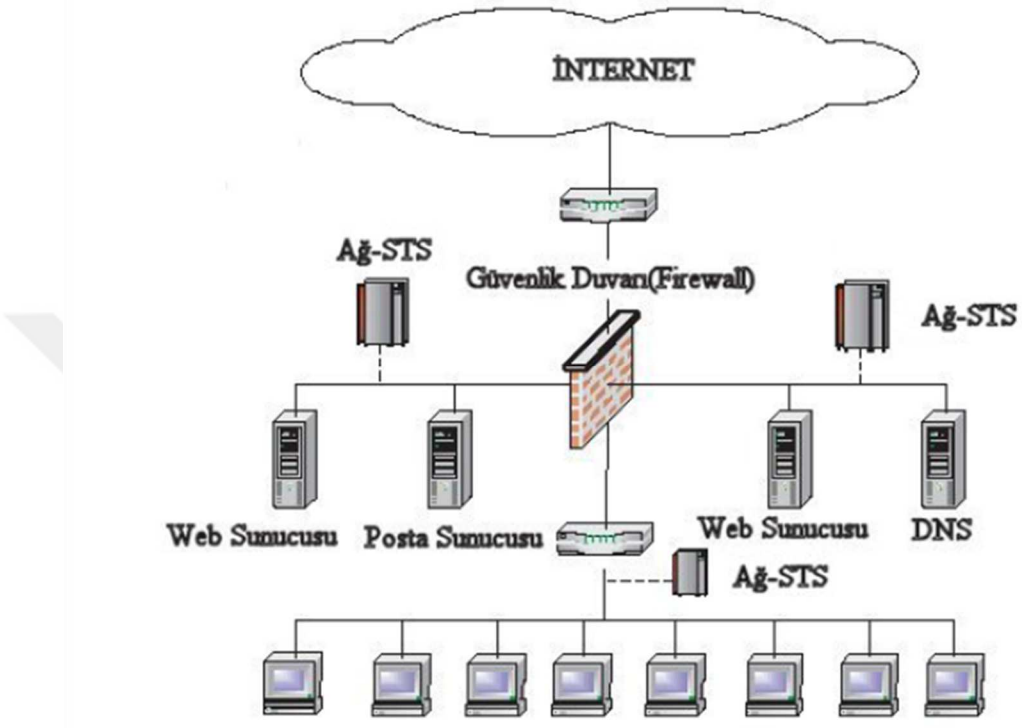
- Koruma ve tespit sistemlerine yönelik olarak devamlı olarak yeni saldırı türleri geliştirilmektedir. Böylelikle güvenlik çözümlerine yönelik olarak devamlı öğrenen ve kendini yenileyen dinamik bir yapıya ihtiyaç duyulmaktadır (Karataş ve diğ., 2018).

Saldırı tespit sistemleri, kullanıcıların davranışlarını analiz ve tahmin ederek, davranışların bir saldırı veya normal bir davranış olup olmadığına karar verir. Saldırı tespit sistemleri genel olarak ağ tabanlı saldırı tespit sistemleri (Network Based Intrusion Detection System) ve bilgisayar tabanlı saldırı tespit sistemleri (Host Based Intrusion Detection System) olarak iki sınıfta incelenmektedirler.

Ağ tabanlı saldırı tespit sistemleri (Şekil 3.1), şüpheli durumları tespit etmek üzere temel ağ paketlerini kullanarak ağ trafiğini incelerler. İncelenen ağ paketleri üç yöntemle sınıflandırılır. Cümlecik imzaları (String signatures) yönteminde, paket verilerinde meydana gelebilecek olayla ilgili verilere bakılırken, kapı imzaları (Port signatures) yönteminde ilgili kapılardan farklı ağ trafiği olup olmadığına bakılır. Başlık imzaları (Header signatures) yönteminde ise, gelen ağ paketlerinin başlıkları incelenerek mantıksız veya muhtemel tehlikeli bir isteğin olup olmadığı kontrol edilir (Liu, 2014).

Ağ tabanlı saldırı tespit sistemlerinin avantaj ve dezavantajlarını şöyle sıralamak mümkündür. Bu tür sistemler, saldırı tespiti maksadıyla ağ paketlerini kullandığından ve ağ verisi, internet protokolü (IP) paketlerinin bir formu olduğundan platform ve işletim sisteminden bağımsız olarak çalışabilmektedir. Saldırı tespitinde ağ

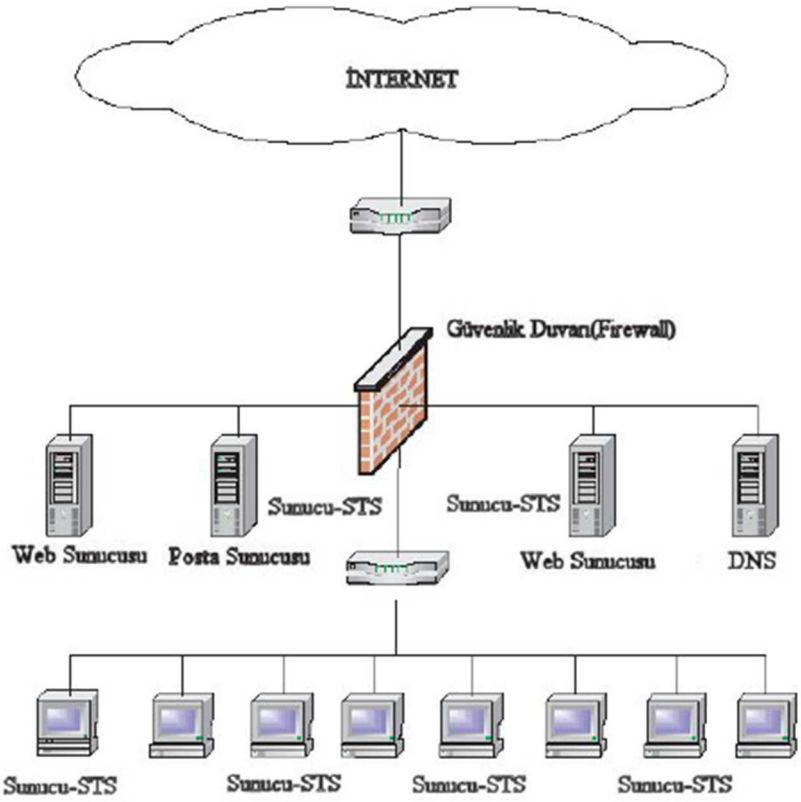
paketlerinin kullanılması, muhtemel saldırı durumunda erken ve hızlı tespiti beraberinde getirmektedir. Ayrıca bulunduğu sistemdeki bilgisayar performansına etki etmemektedir. Ağ tabanlı saldırı tespit sistemleri, olağan bir aşırı ağ trafiği durumunda etkin çalışmamakta ve şifrelenmiş veri paketlerinden oluşan ağ trafiğinde tespit yapmakta zorlanabilmektedir.



Şekil 3.1: Ağ Tabanlı Saldırı Tespit Sistemi Mimarisi.
(Aydın, 2005).

Bilgisayar tabanlı saldırı tespit sistemleri (Şekil 3.2) ise, tek bir bilgisayardaki verileri denetlemektedir. Denetlenen verilere örnek olarak işletim sistemi çağrıları, olaylar, kaynak kullanımları ve sistem kayıtları verilebilir. Bu verilerde oluşabilecek herhangi bir uyumsuzluk veya olağandışı davranış tespit edilmeye çalışılmaktadır (Liu, 2014).

Bilgisayar tabanlı saldırı tespit sistemlerinin avantaj ve dezavantajlarını incelediğimizde ise, bu sistemler gerçekleşen saldırıların başarılı olup olmadığı konusunda fikir sahibi olmamızı, kullanıcı/ dosyalara erişim aktivitelerinin ve sistem dosyalarında meydana gelebilecek değişikliklerin daha kolay kontrol edilebilmesini sağlamaktadır. Buna rağmen, bilgisayar tabanlı saldırı tespit sistemleri saldırılara gerçek zamanlı karşılık verme konusunda ve büyük çaplı saldırılara karşı daha zayıftır.



Şekil 3.2: Bilgisayar Tabanlı Saldırı Tespit Sistemi Mimarisi.
(Aydın, 2005).

Bu iki sınıflandırmanın dışında saldırı tespit sistemleri karar verme yöntemlerine, yerleşim durumlarına, tespit yöntemlerine ve gösterdikleri reaksiyonlara göre de sınıflandırılmaktadır (Bijone, 2016). Şekil 3.3'te saldırı tespit sistemlerine ait sınıflandırma sunulmuştur. Saldırı tespit sistemlerinin gelişmeye devam ettiği göz önünde bulundurulduğunda bu sınıflandırmanın her geçen gün değişebileceği akıldan çıkarılmamalıdır.

Karar verme tabanlı saldırı tespit sistemleri sınıfına giren yardımlı saldırı tespit sistemleri, genellikle mobil özel ağlarda kullanılır ve ağa gelen bir saldırı birden fazla karar vericinin ortak çalışması ile tespit edilir (Santosh ve diğ., 2008). Otonom saldırı tespit sistemlerinde ise saldırıya karar veren mekanizma otonom çalışmaktadır. Mekanizma tarafından sürekli olarak gelen veri incelenerek değerlendirilmekte ve olası önlemlere karar verilmektedir (Kholidy ve diğ., 2013).

Dağıtık ve merkezi saldırı tespit sistemleri, yerleşimlerine göre isimlendirilen saldırı tespit sistemlerindendir. Dağıtık saldırı tespit sistemlerinde gelen veri ayrık yerlerde analiz edilmektedir (Snapp ve diğ., 1992). Merkezi saldırı tespit sistemlerinde ise verilerin analiz edildiği yerler toplu halde bulunmaktadır (Lunt ve diğ., 1992). Gerek

dağıtık saldırı tespit sistemlerinde, gerek merkezi saldırı tespit sistemlerinde önemli olan veri toplanan sensörlerin ayrık veya toplu olması değil, analiz edilen yerlerin ayrık veya toplu olmasıdır.

Gösterdiği reaksiyona göre saldırı tespit sistemleri ise, aktif saldırı tespit sistemi ve pasif saldırı tespit sistemi olarak adlandırılmaktadır. Pasif saldırı tespit sistemlerinde, muhtemel güvenlik ihlali kayıt altına alınarak kullanıcıya uyarı gönderilir. Saldırıyı önlemeye yönelik girişimde bulunulmaz. Aktif saldırı tespit sistemlerinde ise, olası güvenlik ihlalinde sistem harekete geçerek, saldırgan ve saldırı hakkında bilgi toplar ve saldırıya otomatik olarak karşılık verir. Aktif saldırı tespit sistemi, sistem yapılandırması ve ağ cihazlarının (router, güvenlik duvarı vb.) ayarlarında değişiklik yaparak da saldırganı durdurmayı dener.

Son olarak, tespit yöntemlerine göre saldırı tespit sistemleri üçe ayrılmaktadır. Bunlar, olağan dışı davranış (anomali) tabanlı saldırı tespit sistemleri, imza/yanlış kullanım tabanlı saldırı tespit sistemleri ve durum protokolü analizi tespiti tabanlı saldırı tespit sistemleridir. Olağan dışı davranış tabanlı saldırı tespit sistemleri, normal sistem davranışına aykırı görünen olayları tespit etmeye yönelik çalışırlar. Olağan dışı davranış tabanlı saldırı tespit sistemlerinin en cazip özelliği, yeni geliştirilen ve daha önceden kullanılmamış saldırıları tespit edebilmesidir. Bu sistemlerin en büyük dezavantajı ise, temelinde olağan dışı hareketi algılamak olduğundan dolayı, olağan dışılık göstermeyen saldırıları tespit edememesidir. İmza/yanlış kullanım tabanlı saldırı tespit sistemleri ise, daha önce gerçekleştirilmiş saldırıların gerçekleşmekte olan saldırı ile kıyaslanarak tespit edilmesi temeline dayanmaktadır. Bu tür saldırı tespit sistemleri, daha önce karşılaşılan saldırılara ait bilgileri bir veri tabanında tutarak kıyaslama yapmakta ve saldırını türüne karar vermektedir. Virüs önleme yazılımlarında olduğu gibi bu tür saldırı tespit sistemlerinde de sistemin başarısı kullandığı veri tabanının yeterliliğine bağlıdır. Durum protokolü analizi tespiti tabanlı saldırı tespit sistemleri ise, daha önceden tanımlanmış olan bir protokolün gerçekleştirdiği işlemler ile gerçekleşen olayın davranışını kıyaslayarak saldırıyı tespit etmektedir. Gerçekleştirdiği bu işlem derinlemesine paket analizi (deep packet inspection) olarakta adlandırılmaktadır (Whitman ve Mattord, 2009).

Şekil 3.3: Saldırı Tespit Sistemleri.
(Bijone, 2016).

4. METODOLOJİ

4.1. Yapay Öğrenme

Klasik anlamda hayatımızda yer alan birçok cihaza yapay öğrenme tekniklerinin uygulanması neticesinde, bu cihazlar "akıllı cihaz" olarak karşımıza çıkmaktadır. Yapay öğrenme, basit anlamda bir bilgisayar programının karşılaştığı problemi, daha önce edindiği tecrübeler ve programa tanımlanan veri setlerini kullanarak çözmesi olarak tanımlanabilir. Bir problemin yapay öğrenme yöntemleriyle çözülüp, çözülemeyeceğine;

- İyi tanımlanmış girdilerden, belirli sonuçlara giden fonksiyonlara ihtiyaç duyulması,
- Problemin çözümü için çok büyük veri setlerine ihtiyaç olması,
- Net olarak tanımlanabilen hedefleri ve verileri içeren geri beslemelere ihtiyaç duyulması,
- Sonuca ulaşmak için kararın nasıl verildiğine dair detaylı açıklama gerekmesi,
- Problemin çözümü, hata için toleranslı ve kanıtlanabilir en uygun çözüme ihtiyaç duymaması ve
- Problemin çözümü için özel el becerisi, fiziksel beceri veya hareketlilik ihtiyacı olmaması gibi kriterler sorgulanarak karar verilmektedir (Brynjolfsson ve Mitchell, 2017).

Genel olarak yapay öğrenme, öğrenme yöntemine göre Denetimli (Supervised) Yapay Öğrenme, Denetimsiz (Unsupervised) Yapay Öğrenme ve Pekiştirmeli (Reinforcement) Yapay Öğrenme olarak üç başlık altında incelenmektedir. Denetimli yapay öğrenmede etiketlenmiş (labelled) giriş değerleri ile istenen çıkış değerleri arasında bağıntı kuran bir fonksiyon kullanılır. Denetimsiz yapay öğrenmede etiketlenmemiş (unlabelled) veri kullanılarak bilinmeyen bir yapıyı öğrenmeye yönelik bir fonksiyon kullanılır. Pekiştirmeli yapay öğrenmede ise öğrenen etken,

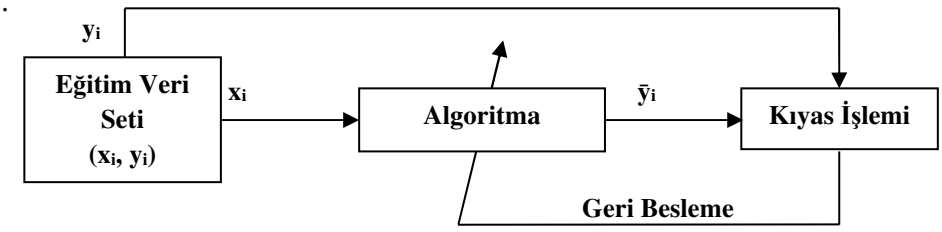
ortamla etkileşerek aldığı geri bildirime göre çıktıyı maksimuma ulaştıracak hareket tarzını bulmaya çalışır.

Pekiştirmeli yapay öğrenme, karar vermek için yapılan tercih neticesinde gerçekleşen olaya bağlı olarak verilen ceza (punishment) ve ödül (reward) mekanizmasına bağlı olarak çalışmaktadır (Simeone, 2018). Örneğin pekiştirmeli yapay öğrenme kullanarak satranç oynayan bir kod geliştirdiğimizi, her satranç taşının bir puan değeri olduğunu ve oyuncunun puanı sıfıra düştüğünde oyunun sona erdiğini düşünelim. Geliştirdiğimiz kodu kullanarak satranç oynayacak bir bilgisayar, her puan kaybında oyunun bitmeye yaklaştığını anlayacak ve hamlesini taş kaybetmeyecek şekilde yapacaktır.

4.1.1. Denetimli (Supervised) Yapay Öğrenme

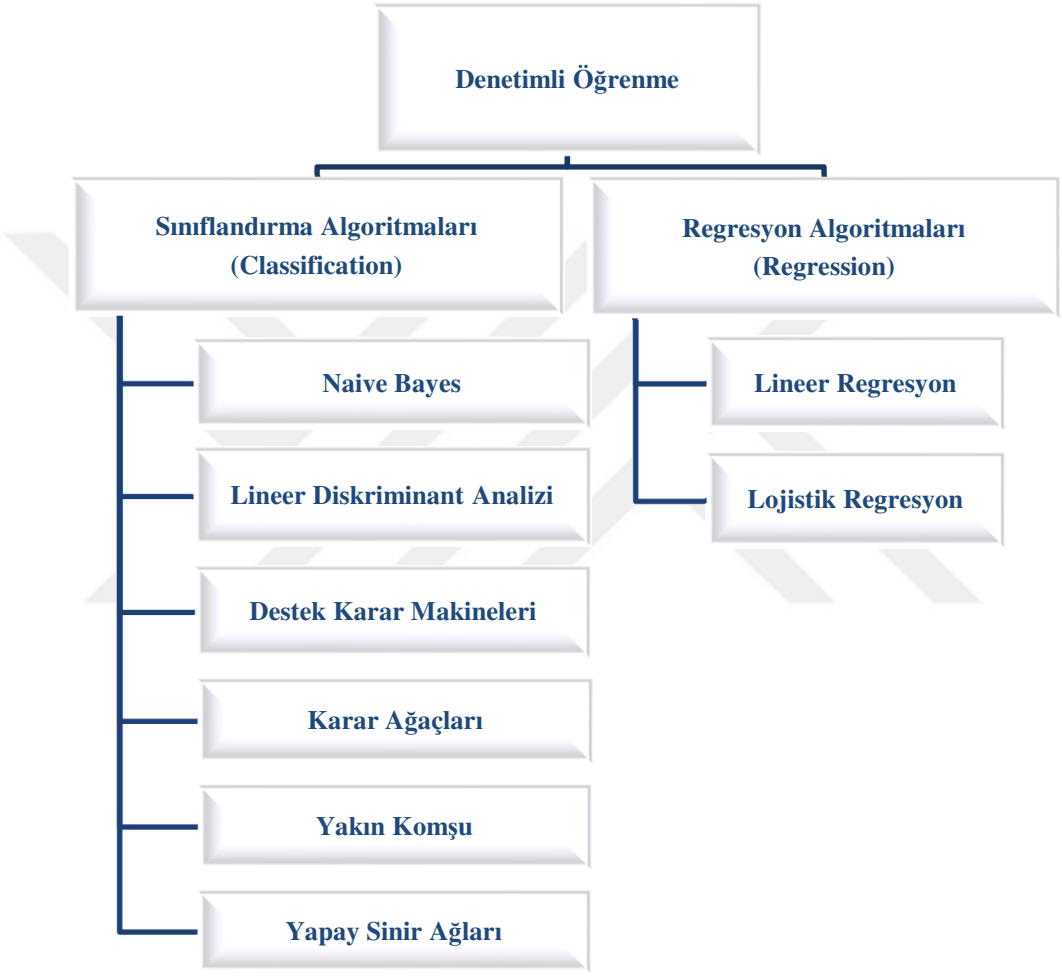
Denetimli yapay öğrenme, bir öğrencinin bilmediği bir konuyu öğretmeni yardımıyla öğrenmesi örneğiyle açıklanabilir. Öğretmen, öğretilecek konuyu ve öğrencisinin neyi öğreneceğini bilmektedir. Denetimli yapay öğrenme de temelde bu mantıkla çalışmaktadır. Eğitilecek olan algoritmaya etiketli veri seti çalıştırılarak, algoritmanın girdisi ile çıktısı arasındaki bağıntıyı kurması "öğretilir" ve eğitilmiş algoritma müteakip işlemlerini öğrendiği veri ışığında gerçekleştirir.

Eğitilecek algoritmaya girdi olarak verilen eğitim setinin, algoritmanın öğrenme sürecinde bir öğretmen gibi yapacağı işe destek olması bu öğrenme türüne adını vermiştir. Eğitilen algoritma yaptığı her işlem sonucunda, girdi olarak kullandığı veri setindeki cevaplarla kıyaslanarak daha doğru sonuçlar vermesi için eğitilir (Şekil 4.1). Nihayetinde algoritmanın çıktıları kabul edilebilir doğruluk seviyesine ulaştığında eğitim sonlandırılır (Brownlee, 2017). Bu aşamadan sonra algoritmaya koşulan veriler, algoritmanın daha önceden öğrendiği şekilde işleme tabi tutularak kategorize edilir.



Şekil 4.1: Denetimli Öğrenme.

Denetimli öğrenme algoritmaları genel olarak iki başlıkta incelenir (Şekil 4.2). Bunlar sınıflandırma (classification) ve regresyon (regression) problemlerini çözmekte kullanılan algoritmalardır. Sınıflandırma algoritmaları, kendisine girdi olarak verilen veriyi sınıflandırarak, çıktı olarak bu verinin hangi sınıfa veya kategoriye ait olduğuna karar verir. Regresyon algoritmaları ise, girdi olarak verilen veriler ile sonuç arasında bir ilişki kurarak girdi verisinin sonuca etkisine karar verir.



Şekil 4.2: Denetimli Öğrenme Algoritmaları.

Yukarıdaki şekilde verilen algoritmalar yapay öğrenmede en sık kullanılan denetimli öğrenme algoritmalarıdır. Naive Bayes, bir olayın gerçekleşme koşulunu diğer olay ile ilişkilendirerek açıklayan Bayes teoremini baz alan bir sınıflandırıcıdır. Lineer diskriminant analizi, bir veriye ait özniteliklerin doğrusal birleşimini bularak veriyi sınıflara ayıran bir sınıflandırıcıdır. Destek karar makineleri (Support Vector Machines), verileri sınıflandırmak için düzlemsel veya parabolik "karar sınırı"

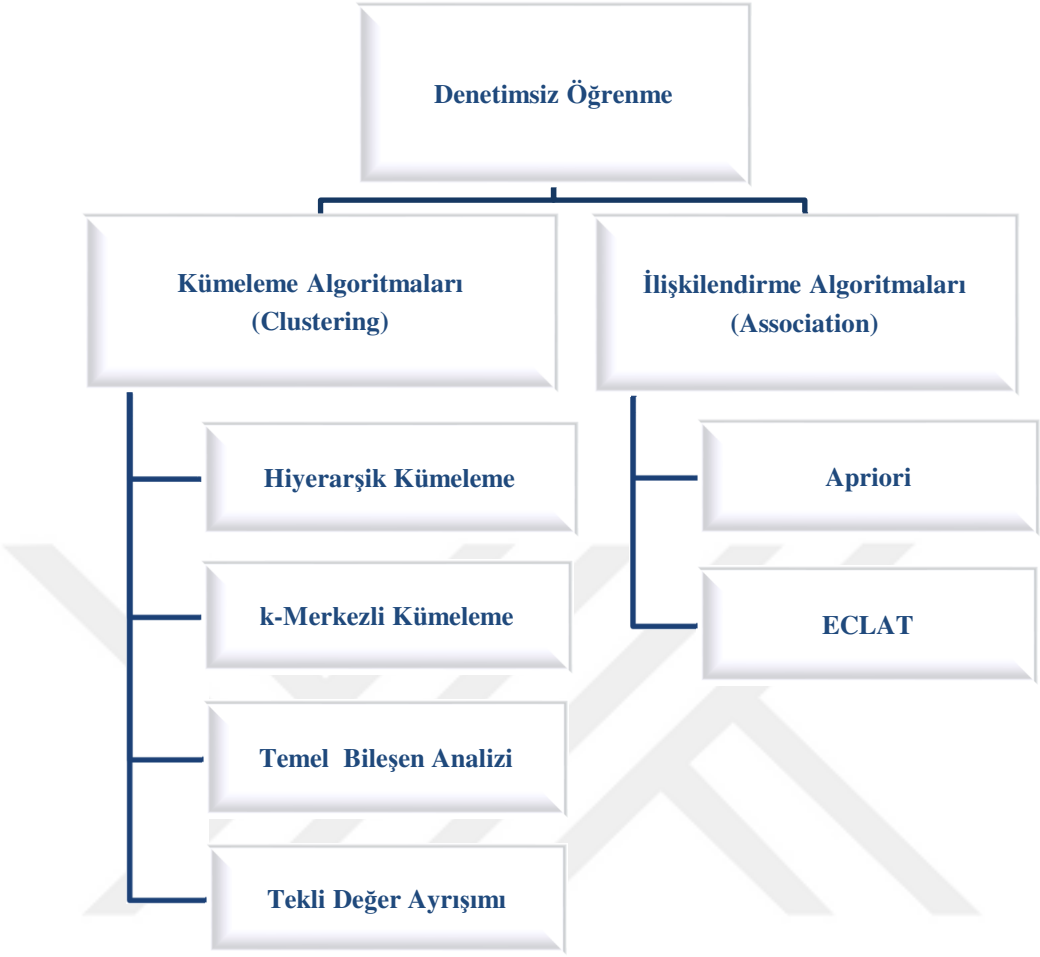
kullanan bir sınıflandırıcıdır. Karar ağaçları (Decision Trees) ise, verinin özelliklerini kullanarak ağaç yapısı oluşturan bir sınıflandırıcıdır. Böylelikle benzer özellikleri taşıyan veriler, bu ağaç yapısı içerisinde düğüm (node) ve yapraklar (leaf) altında sınıflandırılır. Yakın komşu (Nearest Neighborhood) algoritması, verileri özelliklerine göre sınıflandırarak, yeni veriyi en yakın özelliklere sahip sınıfa ayırma mantığıyla çalışan bir sınıflandırıcıdır. Yapay sinir ağları (Artificial neural networks), insan beyninde bulunan sinir sisteminin çalışma şekli taklit edilerek oluşturulmuş bir sınıflandırıcıdır.

Regresyon algoritmalarında ise girdi ve çıktı arasındaki ilişki kullanılır. İki veri arasındaki ilişki doğrusal ise bu lineer regresyondur. Lineer regresyon, çıktı olarak sonsuz sayıda değişen içerebilir. Lojistik regresyon ise, girdi ile çıktı arasındaki ilişkinin sonlu sayıda değişken içermesi istenen durumlar veya olasılıksal sonuçlar için kullanılır.

4.1.2. Denetimsiz (Unsupervised) Yapay Öğrenme

Denetimsiz yapay öğrenme, denetimli yapay öğrenmenin aksine "öğretilen" ve "etiketli" verinin bulunmadığı, algoritmanın tamamen kendi kendisine öğrenmeyi gerçekleştirdiği durumlarda karşımıza çıkmaktadır. Denetimsiz yapay öğrenmede, algoritma girdi olan veri setindeki özellikleri kullanarak kendisine bir öğrenme paterni oluşturmakta ve öğrenme sürecinde denetimli öğrenme örneğindeki gibi bir öğretmene ihtiyaç duymamaktadır.

Denetimsiz öğrenme algoritmaları genel olarak iki başlık altında incelenmektedir (Şekil 4.3). Bunlar kümeleme (clustering) ve ilişkilendirme (association) algoritmalarıdır. İsimlerinden de anlaşılacağı üzere, kümeleme algoritmaları, girdi olarak verilen veri setini özelliklerine göre kendi içerisinde kümelemekte, ilişkilendirme algoritmaları ise veri setindeki özellikleri ilişkiler kurarak ayırmaktadır.



Şekil 4.3: Denetimsiz Öğrenme Algoritmaları.

Yukarıdaki şekilde verilen algoritmalar yapay öğrenmede en sık kullanılan denetimsiz öğrenme algoritmalarıdır. Kümeleme algoritmalarından Hiyerarşik kümeleme (Hierarchical Clustering) algoritması, adından da anlaşılacağı üzere girdi olarak verilen veri setini hiyerarşik yapıda kümelere ayırmak üzere tasarlanır. Algoritma sonucu ortaya çıkan kümeler kendi içlerinde benzer özelliklere sahipken, diğer kümelere özellikleri bakımından farklılık göstermektedir. k-Merkezli kümeleme (k-Means clustering) algoritması, veri setindeki verileri kullanarak k adet küme oluşturmak üzere hareket eder. Algoritma öncelikle k adet kümenin merkezini veri setindeki özellikleri kullanarak belirleyerek diğer örnekleri uzaklıklarına göre bu kümelere dahil eder. Ardından kümelerin merkezlerini güncelleyerek algoritma kararlı hale gelene kadar kümeleme işlemine devam eder. Temel bileşen analizi (Principal Component Analysis) algoritması, veri setindeki özellikleri temel

bileşenlere “indirgeyerek” verileri kümelere ayrıştırmaktadır. Temel bileşen analizi algoritmasının veri setine uygulanması neticesinde verilerin düzlemsel uzayda doğrular etrafında kümelenmediğini düşünmek bu algoritmayı anlamamıza yardımcı olmaktadır. Tekli değer ayrışımı (Singular Value Decomposition) algoritması da, temel bileşen analizi algoritmasında olduğu gibi veri setindeki verileri “indirgeyerek”, tekil değerlere göre kümeleme işlemi yapmaktadır.

Apriori algoritması, veri setinde satırlarda bulunan özellikleri inceleyerek, satırlar/ değişkenler arasındaki ilişkileri tespit etmektedir. Algoritma her satırı tek tek inceleyerek, daha önce incelediği satırlar ile kıyaslayarak aralarındaki ilişkiyi tespit eden iteratif bir yapıya sahiptir ve sığ öncelikli arama (Breadth-First Search) yaklaşımını kullanır. ECLAT (Equivalence Class Transformation - Eşdeğerlik Sınıf Dönüşümü) algoritması ise, temelde Apriori algoritması ile aynı yaklaşımı kullanmaktadır. İki algoritma arasındaki fark ise, ECLAT algoritmasının sığ öncelikli arama yerine derinlik öncelikli arama (Depth-First Search) kullanmasıdır.

4.1.3. Denetimli ve Denetimsiz Yapay Öğrenmenin Kıyaslanması

Önceki kısımlarda verilen denetimli ve denetimsiz öğrenme algoritmaları, günümüzde en sık kullanılan yapay öğrenme algoritmalarındandır. Gelişen teknoloji ve yapay öğrenmeye duyulan ihtiyaç göz önünde bulundurulduğunda, gerek denetimli öğrenme algoritmaları, gerekse denetimsiz öğrenme algoritmaları gün geçtikçe gelişmekte ve farklılaşmaktadır.

Öğrenme algoritmalarının gelişimi ve farklılaşması ortaya çıkan yeniliklere ve ihtiyaçlara bağlı olduğundan, algoritmaların özellikleri ve aralarındaki farklılıklar önem kazanmaktadır. Örneğin, daha doğru sonuçlara ihtiyaç duyulan projelerde denetimsiz öğrenme algoritmaları yerine denetimli öğrenme algoritmaları tercih edilmekte iken, girdi olarak verilen veri setindeki bağıntıları araştırmak isteyenler ise, denetimsiz öğrenme algoritmalarını kullanarak amaçlarına ulaşmaktadır.

Verilen örneklerden de anlaşılacağı üzere, denetimli ve denetimsiz öğrenme algoritmalarının yapıları gereği avantajlı ve dezavantajlı yönleri bulunmaktadır. Tablo 4.1’de denetimli ve denetimsiz öğrenme algoritmaları, tanımları ve uygulamaları ile sonuçları açısından kıyaslanmıştır.

Tablo 4.1: Denetimli ve Denetimsiz Öğrenme Algoritmalarının Kıyaslanması.

Sıra No	Parametre/ Kıyaslamaya Konu Özellik	Denetimli Öğrenme Algoritması	Denetimsiz Öğrenme Algoritması
1	Girdi Verisi	Etiketli veri	Etiketsiz veri
2	Amaç	Eğitim setinden farklı olarak verilen verinin çıktısını tahmin edebilen fonksiyon elde etmek	Girdi veri setindeki olası yapıları ve gizli modelleri bulmak
3	Hesaplama Karmaşıklığı (Computational Complexity)	Basit	Karmaşık
4	Veri Kullanımı	Girdi ve çıktılar arasında bağ kurar	Çıktı verisini kullanmaz
5	Sonuçların Doğruluğu	Yüksek güvenilirlik ve doğruluk	Düşük güvenilirlik ve doğruluk
6	Sınıf Sayısı	Kullanılan sınıf sayısı belirlidir	Kullanılan sınıf sayısı belirsizdir
7	Kullanım alanları	Resim ve ses dosyalarında örüntü tanıma, finansal analizler, sinir ağlarının eğitimi	Ham verileri işleme, veri çözümleme, denetimli öğrenme algoritmalarının ön eğitimi

4.2. Önerilen Metod

Bu tez çalışması kapsamında yapılan uygulamada, ön işleme tabi tutulmuş ve tutulmamış veriler kullanılarak iki farklı test metodu ile dört adet denetimli öğrenme ve iki adet denetimsiz öğrenme algoritması denenmiştir. Yapılan uygulamaya ait sonuçlar beşinci bölümde sunulmuştur.

5. DENEYSEL SONUÇLAR

Ağ tabanlı bilgisayar sistemlerine yönelik tehditlerin/ saldırıların denetimli yapay öğrenme ile sınıflandırılması kapsamında daha iyi ve etkili bir saldırı tespit sistemi geliştirebilmek için kullanılması gereken öğrenme algoritması, KDD CUP-99 verisetinden türetilen NSL KDD veri seti ve WEKA uygulaması kullanılarak tespit edilmeye çalışılmıştır.

Uygulamalar, üçüncü nesil Intel Core i5 (2.6 GHz) işlemciye ve Windows 10 64 bit işletim sistemine sahip, 8 GB bellek, 750 GB hard disk kapasitesi olan dizüstü bilgisayarı ile gerçekleştirilmiştir.

5.1. Veri Setinin İncelenmesi

KDD CUP-99 veri seti, DARPA'nın 1998 yılında geliştirdiği veri setinin 1999 yılında düzenlenen Uluslararası Bilgi Keşfi ve Veri Madenciliği Araçları Yarışması (International Knowledge Discovery and Data Mining Tools Competition)'nda geliştirilmiş halidir (Ferrag ve diğ., 2020). NSL KDD veri seti ise, KDD CUP-99 veri setinin derlenmiş halidir. Bahse konu tüm veri setleri, zararlı trafikte dahil olmak üzere ağdaki veri trafiğini temsil etmektedir. NSL KDD veri seti günümüzde araştırmacılar tarafından sıklıkla kullanılmakta olup, "KDDTest", "KDDTest-21", "KDDTrain_20Percent", "KDDTrain" alt setlerinden oluşmaktadır.

"KDDTest" alt seti isminden de anlaşılacağı üzere test verilerini, "KDDTrain" alt seti eğitim verilerini içermektedir. "KDDTest-21" ve "KDDTrain_20Percent" alt setleri ise "KDDTest" ve "KDDTrain" alt setlerinden türetilmiştir. "KDDTrain_20Percent" alt seti tüm eğitim verilerinin yüzde 20'sini, "KDDTest-21" alt seti ise en zorlayıcı trafik verilerini içermeyen test verilerini içermektedir (Dhanabal ve Shantharajah, 2015).

NSL KDD veri setini KDD CUP-99 veri setinden ayıran ve kullanıcıların tercih etmesine neden olan üç ana özellik bulunmaktadır. Bunların birincisi KDD CUP-99 veri setinde bulunan ve sınıflandırma algoritmalarını yanıltan verilerin, NSL KDD

veri setinde azaltılmış olmasıdır. Böylelikle sınıflandırma algoritmaları koşulurken yanılma payları azalmaktadır. İkincisi, NSL KDD veri setinde yer alan ve saldırı tespiti açısından farklı zorluk seviyesinde bulunan verilerin, KDD CUP-99 veri setindeki veriler ile ters orantılı olmasıdır. Bu özellik, NSL KDD veri seti ile uygulanan farklı yapay öğrenme algoritmalarının sınıflandırma oranlarının geniş bir aralığa yayılmasına neden olmaktadır ve bu durum kullanıcılara farklı algoritmaların sonuçlarının doğru değerlendirilmesi açısından fayda sağlamaktadır. Üçüncü ve son özellik ise, NSL KDD veri setindeki eğitim ve test verisi sayısının, KDD CUP-99 veri setine göre azaltılmış sayıda olmasıdır. Bu özellik ise, kullanıcıların veri setinin herhangi bir kısmını seçmeye gerek kalmadan tüm veri seti üzerinde çalışma yapabilmesine olanak sağlamaktadır (Chae ve diğ., 2013).

NSL KDD veri setini oluşturan kırk bir adet özelliğe ait bilgiler Tablo 5.1'de sunulmuştur.

Tablo 5.1: NSL KDD Veri Seti Özellikleri.

Özellik No	Özellik	Açıklama	Veri Tipi
1	Duration	Bağlantı süresi	Nümerik
2	Protocol_Type	Bağlantı protokolü	Nominal
3	Service	Hedef ağda kullanılan servis	Nominal
4	Flag	Bağlantı durumu (Normal veya Hata)	Nominal
5	Src_Bytes	Bağlantıda hedefe gönderilen veri sayısı (byte)	Nümerik
6	Dst_Bytes	Bağlantıda hedeften gelen veri sayısı (byte)	Nümerik
7	Land	Hedef ve kaynak IP adresi ve port kontrol biti	Binary
8	Wrong_Fragment	Bağlantıdaki yanlış fragman sayısı	Nümerik
9	Urgent	Bağlantıdaki önemli paket sayısı	Nümerik
10	Hot	İçerikteki 'hot'indikatör sayısı	Nümerik
11	Num_failed_logins	Başarısız bağlantı sayısı	Nümerik
12	Logged_in	Bağlantı durum biti	Binary
13	Num_compromised	Uyumlu şart sayısı	Nümerik

Tablo 5.1 - devam.

Özellik No	Özellik	Açıklama	Veri Tipi
14	Root_shell	Root shell kontrol biti	Binary
15	Su_attempted	Super user erişim kontrol biti	Binary
16	Num_root	Bağlantıda root'ta gerçekleştirilen işlem sayısı	Nümerik
17	Num_file_creations	Bağlantıda oluşturulan dosya sayısı	Nümerik
18	Num_shells	'Shell' komut sayısı	Nümerik
19	Num_access_files	Erişim kontrol dosyalarında yapılan işlem sayısı	Nümerik
20	Num_outbound _cmds	Bir FTP oturumda giden komut sayısı	Nümerik
21	Is_host_login	'Host' giriş kontrol biti (root, admin vb.)	Binary
22	Is_guest_login	Misafir giriş kontrol biti	Binary
23	Count	Mevcut bağlantıda son iki saniyede aynı hedef bilgisayara yapılan bağlantı sayısı	Nümerik
24	Srv_count	Mevcut bağlantıda son iki saniyede aynı hizmete yapılan bağlantı sayısı	Nümerik
25	Serror_rate	Hedef bilgisayara yapılan bağlantılardan s0, s1, s2 veya s3 kontrol bitlerini aktive eden bağlantı oranı	Nümerik
26	Srv_serror_rate	Aynı hizmete yapılan bağlantılardan s0, s1, s2 veya s3 kontrol bitlerini aktive eden bağlantı oranı	Nümerik
27	Rerror_rate	Hedef bilgisayara yapılan bağlantılardan REJ kontrol bitini aktive eden bağlantı oranı	Nümerik
28	Srv_rerror_rate	Aynı hizmete yapılan bağlantılardan REJ kontrol bitini aktive eden bağlantı oranı	Nümerik
29	Same_srv_rate	Hedef bilgisayara yapılan bağlantılardan aynı hizmete yapılan bağlantı oranı	Nümerik
30	Diff_srv_rate	Hedef bilgisayara yapılan bağlantılardan farklı hizmetlere yapılan bağlantı oranı	Nümerik
31	Srv_diff_host_rate	Aynı hizmete yapılan bağlantılardan farklı hedef bilgisayarlara yapılan bağlantıların oranı	Nümerik
32	Dst_host_count	Aynı hedef bilgisayar IP adresine sahip bağlantı sayısı	Nümerik
33	Dst_host_srv_count	Aynı port numarasına sahip bağlantı sayısı	Nümerik
34	Dst_host_same_srv _rate	Aynı hedef bilgisayar IP adresine sahip bağlantılardan aynı hizmete yapılan bağlantı oranı	Nümerik
35	Dst_host_diff_srv _rate	Aynı hedef bilgisayar IP adresine sahip bağlantılardan farklı hizmetlere yapılan bağlantı oranı	Nümerik
36	Dst_host_same_src _port_rate	Aynı port numarasına sahip bağlantılardan aynı kaynak portuna yapılan bağlantı oranı	Nümerik
37	Dst_host_diff_src _port _rate	Aynı port numarasına sahip bağlantılardan farklı hedef bilgisayara yapılan bağlantı oranı	Nümerik

Tablo 5.1 - devam.

Özellik No	Özellik	Açıklama	Veri Tipi
38	Dst_host_serror _rate	Aynı hedef bilgisayar IP adresine sahip bağlantılardan s0, s1, s2 veya s3 kontrol bitlerini aktive eden bağlantı oranı	Nümerik
39	Dst_host_srv_serror _rate	Aynı port numarasına sahip bağlantılardan s0, s1, s2 veya s3 kontrol bitlerini aktive eden bağlantı oranı	Nümerik
40	Dst_host_rerror _rate	Aynı hedef bilgisayar IP adresine sahip bağlantılardan REJ kontrol bitini aktive eden bağlantı oranı	Nümerik
41	Dst_host_srv_rerror _rate	Aynı port numarasına sahip bağlantılardan REJ kontrol bitini aktive eden bağlantı oranı	Nümerik
42	Label	Veri paketini saldırı veya normal olarak tanımlayan etiket	-

Choudhary ve Kesswani, 2020.

Tablo 5.1'den de anlaşılacağı üzere NSL KDD veri setinde tanımlı bir trafik verisine ait kırk bir özellik veriye ait temel nitelikleri, içeriği ve trafik bilgisini tanımlarken, kırk ikinci özellik trafik verisinin normal veya saldırı olup olmadığını ifade etmektedir. NSL KDD veri setinde bulunan ve saldırı olarak etiketlenen trafik verileri, toplamda dört sınıfta değerlendirilen otuz dokuz adet saldırı türünden oluşmaktadır. NSL KDD veri setinde saldırı olarak etiketlenen trafik verilerinin ilki hizmet dışı bırakma (Denial of Service (DoS)) saldırısı, ikincisi ayrıcalıklı kullanıcıya erişim (User to Root (U2R)) saldırısı, üçüncüsü uzaktan yerel kaynaklara erişim (Remote to Local (R2L)) saldırısı ve dördüncüsü ise, dinleme (probing) saldırısıdır.

Hizmet dışı bırakma saldırısı, bilgisayar kaynaklarının normalden fazla kullanılmasını hedeflemektedir ve bu saldırıya maruz kalan bilgisayarlar kullanıcıların taleplerine cevap veremez hale gelmektedir. Saldırganlar, ayrıcalıklı kullanıcıya erişim saldırısında, normal bir kullanıcı olarak girdiği sistemde ayrıcalıklı kullanıcı (root, administrator vb.) olmayı, uzaktan yerel kaynaklara erişim saldırısında ise ağdan üzerinden gönderdikleri veriler ile yerel makinede açık yaratarak bu açığı kullanmayı hedeflemektedir. Dinleme saldırılarında ise, ağ trafiği incelenerek bilgisayarlar hakkında veri toplanmakta ve tespit edilen zayıf noktalara uygun saldırı geliştirilmektedir (Thomas ve Pavithran, 2018). Bahse konu saldırı türlerine ait saldırı yöntemleri Tablo 5.2'de, NSL KDD veri setindeki örneklem sayıları Tablo 5.3'te olduğu gibidir (Latah ve Toker, 2018).

Tablo 5.2: NSL KDD Veri Setinde Bulunan Saldırı Yöntemleri.

Saldırı Kategorisi	Saldırı Yöntemi
Hizmet Dışı Bırakma (Denial of Service (DoS))	Apache2, Smurf, Neptune, Back, Teardrop, Pod, Land, Mailbomb, Processtable, UDPstorm
Ayrıcalıklı Kullanıcıya Erişim (User to Root (U2R))	WarezClient, Guess_Password, WarezMaster, Imap, Ftp_Write, Named, MultiHop, Phf, Spy, Sendmail, SnmpGetAttack, SnmpGuess, Worm, Xsnoop, Xlock
Yerel Kaynaklara Erişim (Remote to Local (R2L))	Buffer_Overflow, Httptuneel, Rootkit, LoadModule, Perl, Xterm, Ps, SQLattack
Dinleme (Probing)	Satan, Saint, Ipsweep, Portsweep, Nmap, Mscan

Tablo 5.3: NSL KDD Veri Seti Örneklem Sayıları.

Veri Seti	Toplam Örneklem Sayısı	Normal	DoS	R2L	U2R	Probe
KDD Train	125973	67343 (%53.46)	45927 (%36.46)	995 (%0.79)	52 (%0.04)	11656 (%9.25)
KDD Test	22544	9711 (%43.07)	7458 (%33.08)	2754 (%12.22)	200 (%0.89)	2421 (%10.74)

5.2. WEKA Programının Tanıtılması

WEKA (Waikato Environment for Knowledge Analysis), Yeni Zelanda'da bulunan Waikato Üniversitesi tarafından geliştirilen Java tabanlı bir veri işleme ve analiz programıdır. Program, 1993 yılında Yeni Zelanda hükümetinin desteğiyle geliştirilmeye başlanmış ve ilk olarak 1999 yılında dünya çapında kullanıma sunulmuştur. WEKA programının modüler ve geliştirilebilir yapısı, kullanıcıların farklı yapay öğrenme yöntemlerini, farklı veri kümeleri ile hızlı bir şekilde denemelerine ve karşılaştırmalarına olanak sağlamaktadır (Witten ve diğ., 2009).

WEKA programına, veri tabanından, internet üzerinden (URL) ve dosyadan veri yükleme işlemi yapılabilmektedir. Program kendisi için üretilen ARFF formatı ile CSV ve LibSVM gibi birçok dosya formatını desteklemektedir. Ayrıca sunduğu

görsel arayüzü sayesinde kullanıcılar yaptıkları işlemleri grafikler ile sergileyebilmektedir.

5.3. Sonuçların Değerlendirilmesinde Kullanılacak Veriler

Denetimli öğrenme algoritmalarının değerlendirmesi, karmaşıklık matrisindeki değişkenler kullanılarak yapılmaktadır. Hangi denetimli öğrenme algoritmasının daha iyi olduğuna karar vermek için hesaplanan verilerin temelini, Doğru Pozitif (True Positive), Doğru Negatif (True Negative), Yanlış Pozitif (False Positive) ve Yanlış Negatif (False Negative) olarak nitelendirilen dört adet değişken oluşturmakta olup, bu dört değişkenin nasıl hesaplandığı Tablo 5.4'te gösterilmiştir (Nguyen ve Armitage, 2008). Karmaşıklık matrisindeki diyagonal hücreler doğru tespit edilen veri sayısını, diğer hücreler ise hatalı tespit sayısını göstermektedir (Deshmukh ve diğ., 2015).

Tablo 5.4: Karmaşıklık Matrisi Değerlendirme Kriterleri.

Karmaşıklık Matrisi		**Tahmin Edilen Sınıf**	
		A	**Ā**
Gerçek Sınıf	**A**	Doğru Pozitif	Yanlış Negatif
	Ā	Yanlış Pozitif	Doğru Negatif

Yukarıdaki tablodan da anlaşılacağı üzere;

- *Doğru Pozitif (DP)* değeri, gerçekte A sınıfına ait olan ve A sınıfına ait olduğu tahmin edilen veri sayısını,

- *Doğru Negatif (DN)* değeri, gerçekte A sınıfına ait olmayan ve A sınıfına ait olmadığı tahmin edilen veri sayısını,

- *Yanlış Pozitif (YP)* değeri, gerçekte A sınıfına ait olmayan ancak A sınıfına ait olduğu tahmin edilen veri sayısını,

- *Yanlış Negatif (YN)* değeri, gerçekte A sınıfına ait olan ancak A sınıfına ait olmadığı tahmin edilen veri sayısını göstermektedir.

Karmaşıklık matrisinde karşımıza çıkan bu dört değişken kullanılarak, Doğruluk (Accuracy), Kesinlik (Precision), Duyarlılık (Recall) ve F-ölçütü (F-measure) gibi

veriler hesaplanıp denetimli öğrenme algoritmalarının performans değerlendirmesi yapılmaktadır (Kaya, 2016; Yiğidim, 2012).

Doğruluk, doğru olarak tahmin edilen verilerin toplam veriye oranı olarak ifade edilmektedir. Sınıflandırma algoritmasının performansını ortaya koyan önemli bir ölçüttür. Eşitlik olarak ifadesi aşağıda olduğu gibidir.

$$\text{Doğruluk} = \frac{\text{DP} + \text{DN}}{\text{DP} + \text{YP} + \text{DN} + \text{YN}} \tag{5.1}$$

Kesinlik, doğru olarak tahmin edilen verilerin doğru tahmin edilen toplam veri sayısına oranıdır. Eşitlik olarak ifadesi aşağıda olduğu gibidir.

$$\text{Kesinlik} = \frac{\text{DP}}{\text{DP} + \text{YP}} \tag{5.2}$$

Duyarlılık, doğru olarak tahmin edilen verilerin gerçekte o sınıfa ait verilerin sayısına oranıdır. Algoritmanın veriyi hangi oranla doğru tahmin ettiğini göstermektedir. Eşitlik olarak ifadesi aşağıda olduğu gibidir.

$$\text{Duyarlılık} = \frac{\text{DP}}{\text{DP} + \text{YN}} \tag{5.3}$$

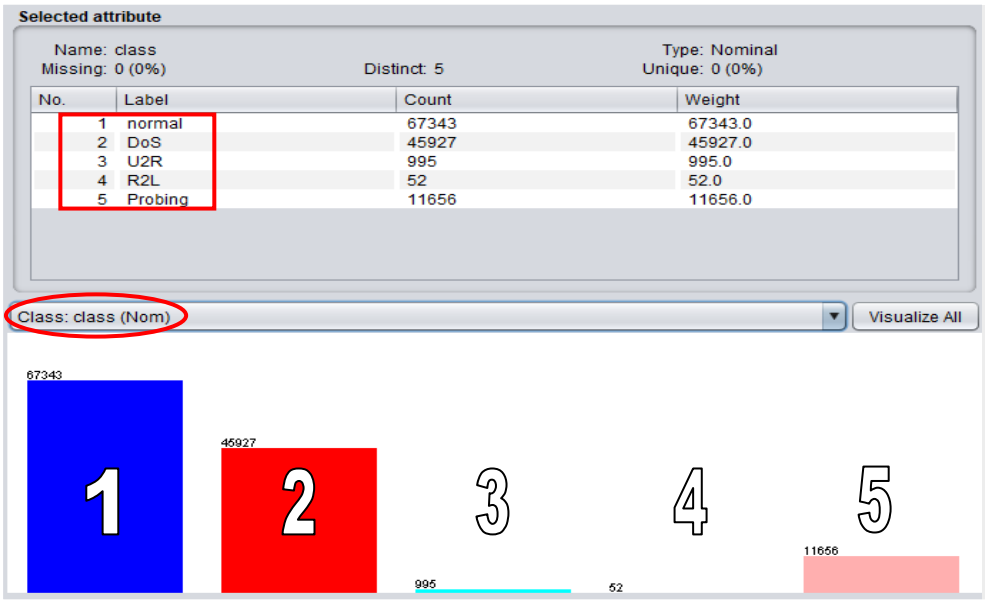
F-ölçütü, Kesinlik ve Duyarlılık verilerinin harmonik ortalaması alınarak bulunmaktadır. Bu nedenle her iki veriyi ayrı ayrı kullanmak yerine bu veri kullanılarak, denetimli öğrenme algoritmalarının kıyaslaması yapılabilir. Eşitlik olarak ifadesi aşağıda olduğu gibidir.

$$\text{F-ölçütü} = \frac{2 \times \text{Duyarlılık} \times \text{Kesinlik}}{\text{Duyarlılık} + \text{Kesinlik}} \tag{5.4}$$

5.4. Denetimli ve Denetimsiz Öğrenme Algoritmaları ile Elde Edilen Veriler

Denetimli ve denetimsiz öğrenme algoritmalarının WEKA programı ile koşulabilmesi maksadıyla öncelikle veri seti üzerinde bazı işlemler gerçekleştirmek gerekmektedir. Gerçekleştirilen bu işlemler algoritmaların doğru işlenmesi ve yapılacak çalışmaların doğru sonuç vermesi açısından önem arz etmektedir.

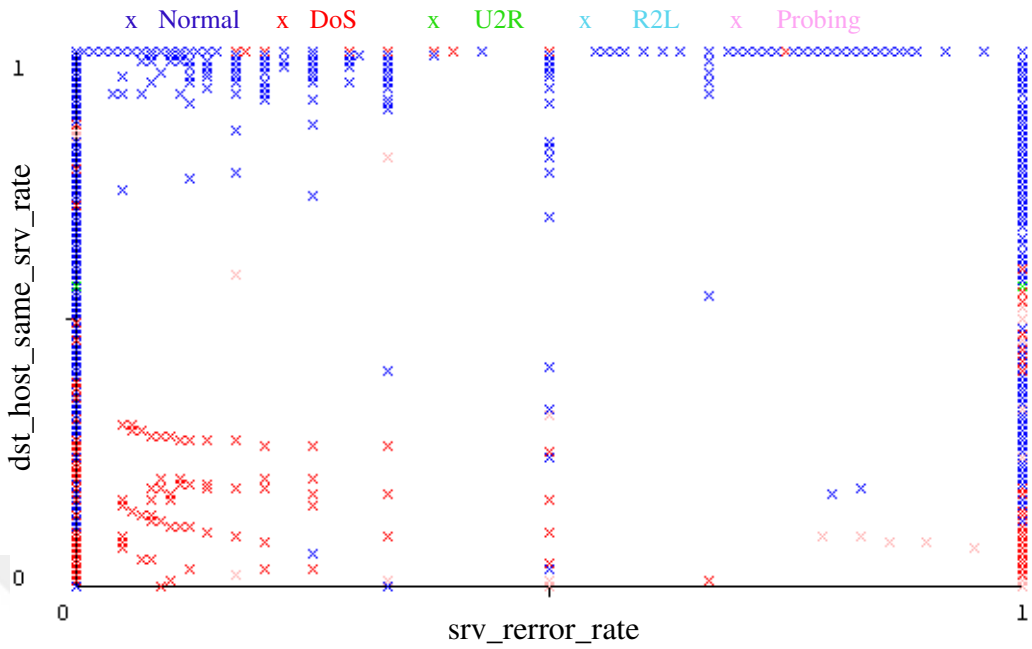
Nitekim NSL KDD veri seti, orijinal halinde üç tür veri (Nominal, Nümerik ve Binary), etiket verisi (kırk ikinci özellik) olarak ise sadece “Normal” ve “Anomaly” olarak iki değer içermektedir. Yapılacak çalışmada gerçek duruma yakın sonuçlar elde etmek üzere bu etiket verileri, hem eğitim hem de test verileri için veri seti içerisinde bulunan sınıf değerleri, “Normal”, “DoS”, “U2R”, “R2L” ve “Probing” olarak beş değere çevirilmiştir (Şekil 5.1). Aynı şekilde, veri setindeki farklı özelliklere ait veri türleri kullanılan algoritmaya uygun veri türlerine çevirilmiştir.



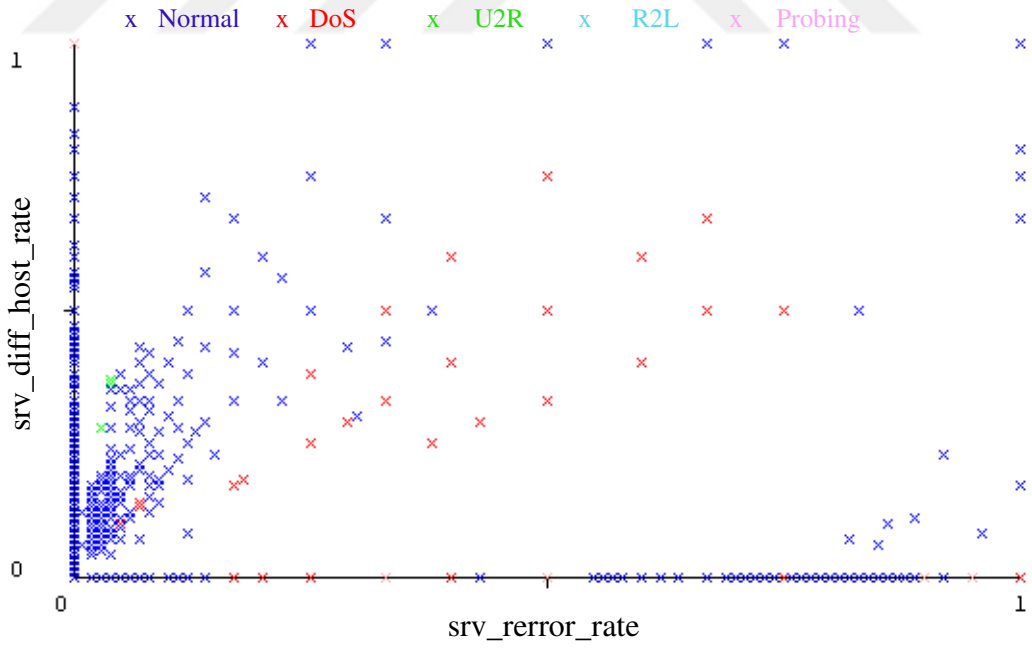
Şekil 5.1: Veri Seti Etiket Değerleri.

Algoritma performanslarının NSL KDD veri setinin tamamı ve azaltılmış veri üzerindeki etkisini incelemek amacıyla, veri seti boyut azaltma işlemine tabi tutulmuş ve veri setinin sahip olduğu kırk bir özellik altı özelliğe (dst_host_srv_serror_rate, srv_rerror_rate, logged_in, dst_host_count, dst_host_same_srv_rate ve srv_diff_host_rate) indirgenmiştir. Bahse konu altı özelliğin varyansı 0,90'dır. Varyansın 0,90 olması, veri setinde bulunan 125973 verinin 0,90 oranında altı özellik ile ifade edilebildiğini anlatmaktadır.

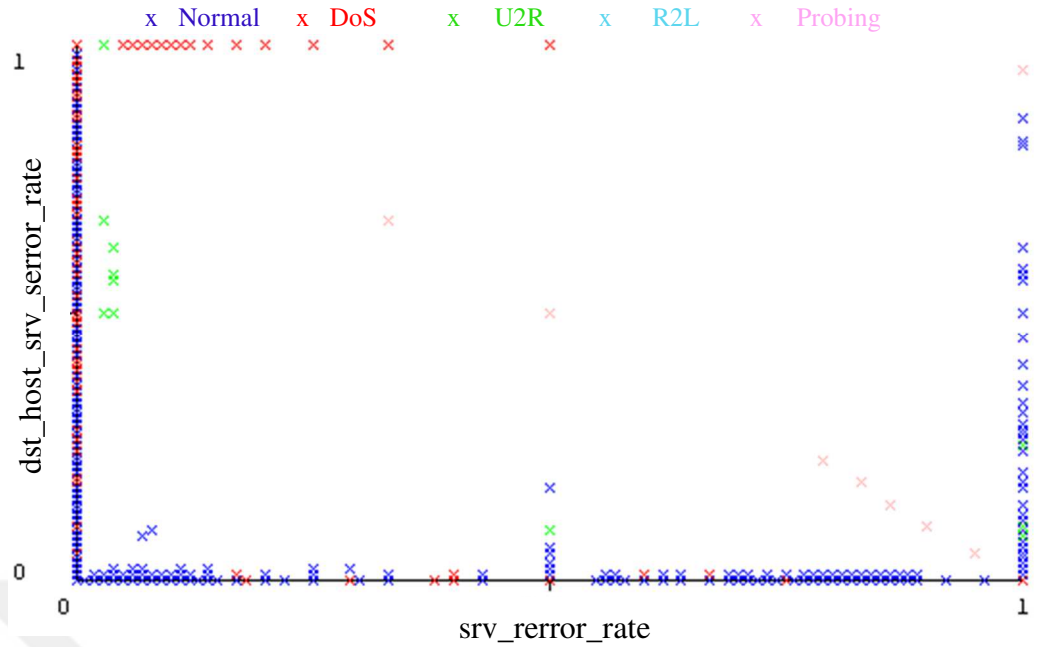
WEKA programı aracılığıyla, boyut azaltma (Temel Bileşenler Analizi (Principal Component Analysis) algoritması) ve verinin iki boyutlu sergilenmesi işlemleri gerçekleştirilmiştir. Seçilen altı özelliğin boyut azaltma işlemine tabi tutulmadan önce grafik üzerindeki dağılımları Şekil 5.2, Şekil 5.3, Şekil 5.4, Şekil 5.5 ve Şekil 5.6'da görülmektedir. Grafiklerden de anlaşılacağı üzere rastgele seçilen srv_rerror_rate özelliği yatay eksende sabit tutularak diğer özellikler değiştirilmek suretiyle iki boyutta özellikler ile sınıflar arasındaki ilişki gösterilmiştir. Grafik eksenleri, oran ifade eden dört özellik ve logged_in özelliği için 0 ile 1, dst_host_count özelliği için 0 ile 255 arasında olacak şekilde oluşturulmuştur. Grafiklerde bulunan mavi işaretler normal veri trafiğini, kırmızı işaretler DoS saldırılarını, yeşil işaretler U2R saldırılarını, turkuaz işaretler R2L saldırılarını ve pembe işaretler Probing saldırılarını temsil etmektedir.



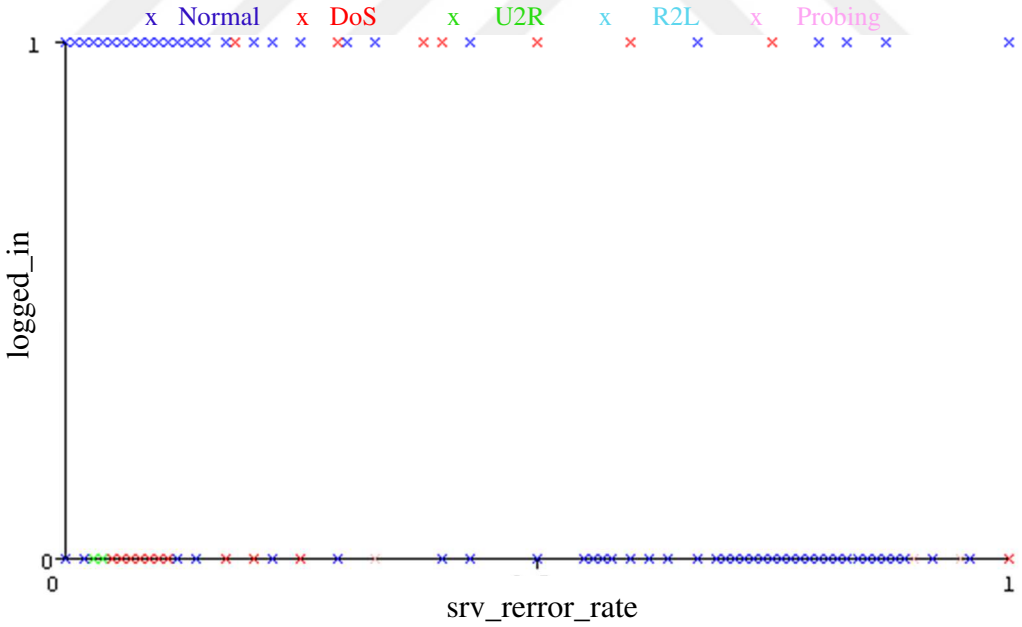
Şekil 5.2: Boyut Azaltma İşlemi Öncesi Veri (dst_host_same_srv_rate) Dağılımı.



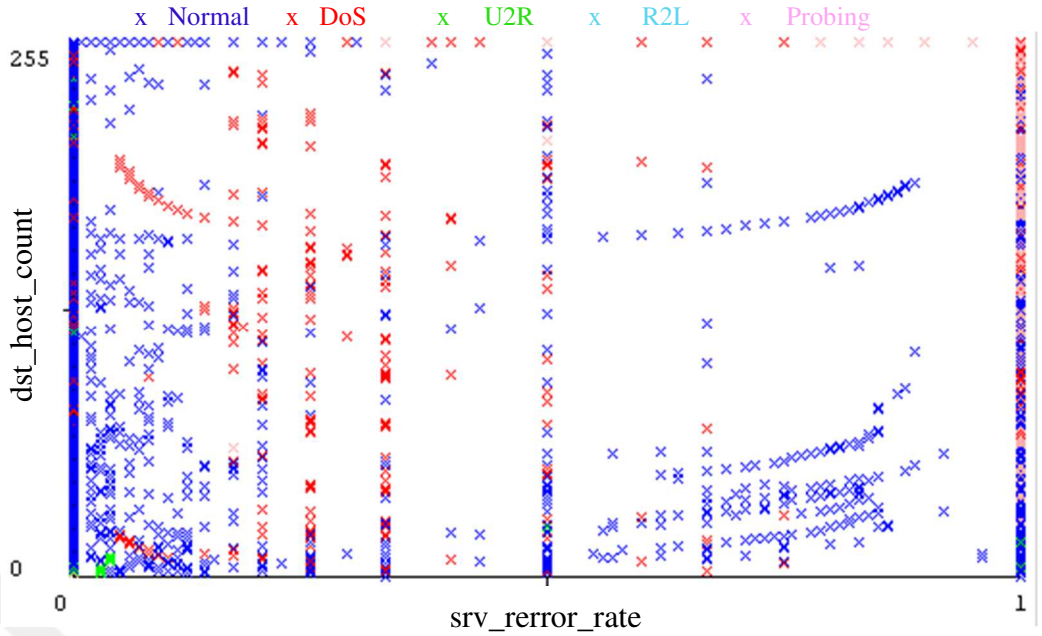
Şekil 5.3: Boyut Azaltma İşlemi Öncesi Veri (srv_diff_host_rate) Dağılımı.



Şekil 5.4: Boyut Azaltma İşlemi Öncesi Veri (dst_host_srv_serror_rate) Dağılımı.

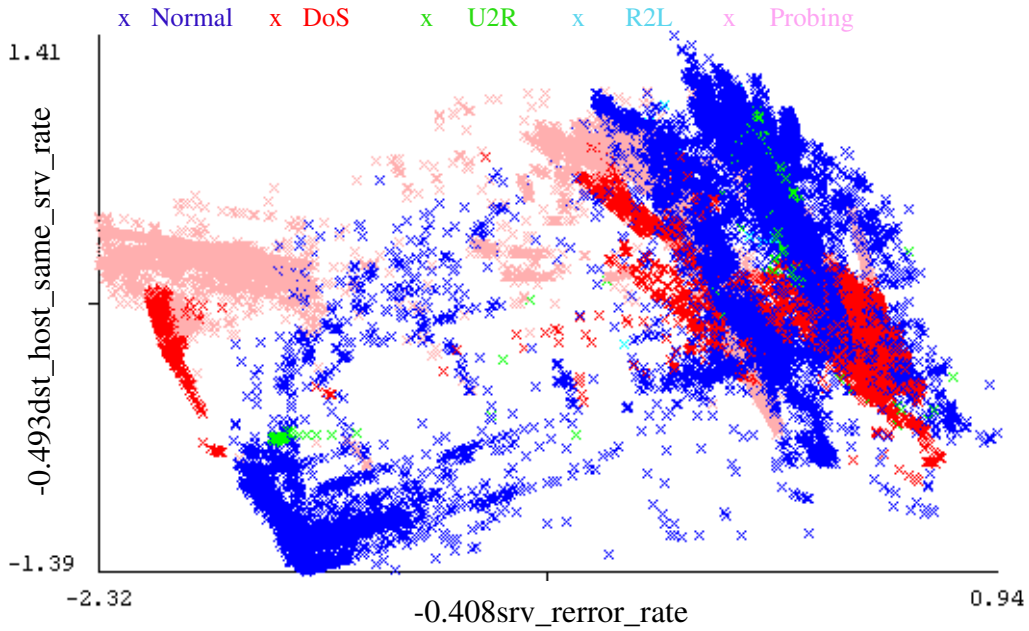


Şekil 5.5: Boyut Azaltma İşlemi Öncesi Veri (logged_in) Dağılımı.

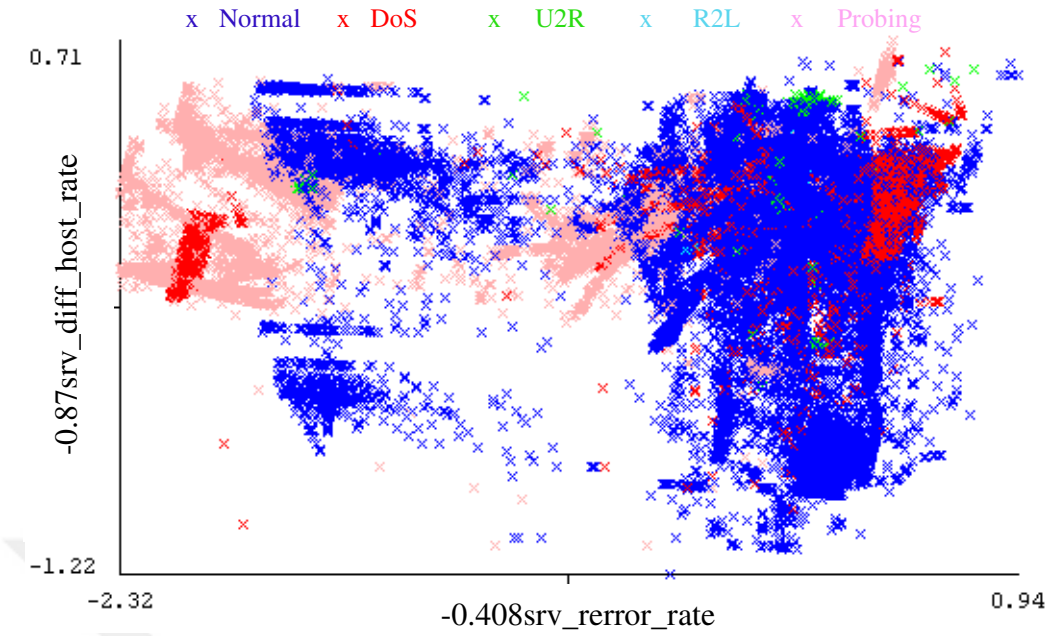


Şekil 5.6: Boyut Azaltma İşlemi Öncesi Veri (dst_host_count) Dağılımı.

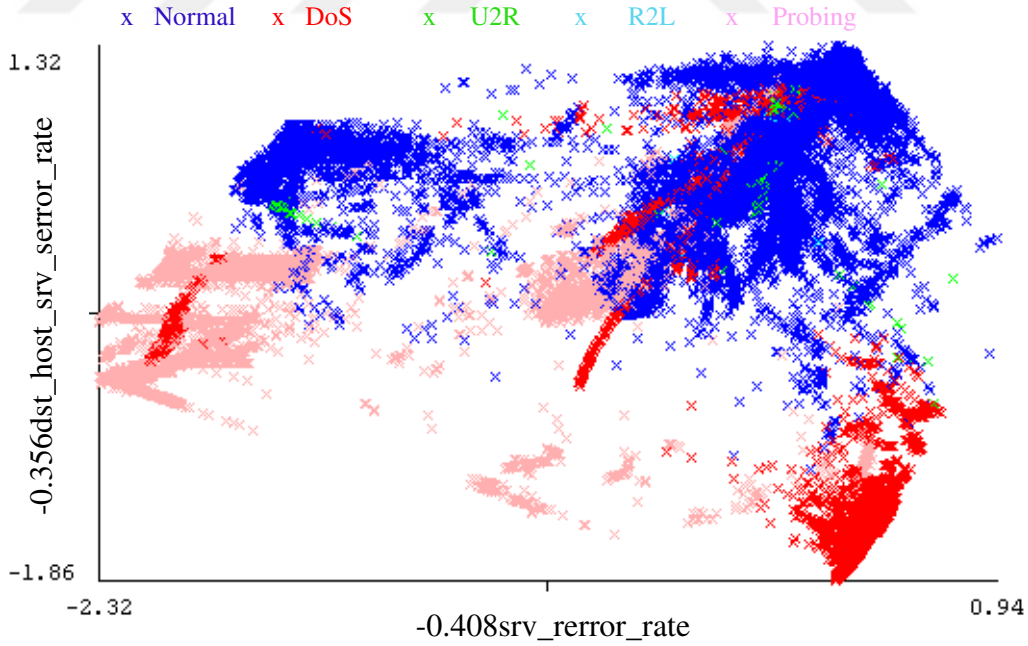
Şekil 5.7, Şekil 5.8, Şekil 5.9, Şekil 5.10 ve Şekil 5.11'de ise, aynı verilerin boyut azaltma işlemine tabi tutulduktan sonraki dağılımları görülmektedir. Özelliklerin başındaki ondalık değerler boyut azaltma işlemi sonucunda oluşan temel bileşenlerin (principal component) yatay ve dikey eksene oturtulmak üzere ne kadar döndürüldüğünü ifade etmektedir. Eksenlerde bulunan değerler ise, her bir örneğin temel bileşenlerin merkez noktasına olan uzaklığının dağılımını ifade etmektedir.



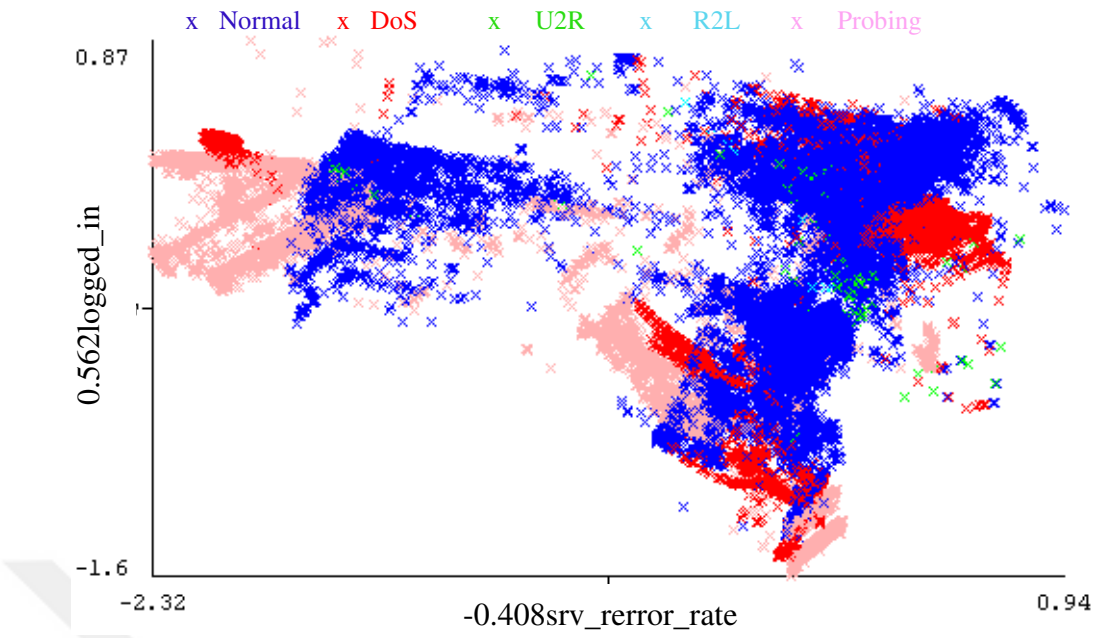
Şekil 5.7: Boyut Azaltma İşlemi Sonrası Veri (dst_host_same_srv_rate) Dağılımı.



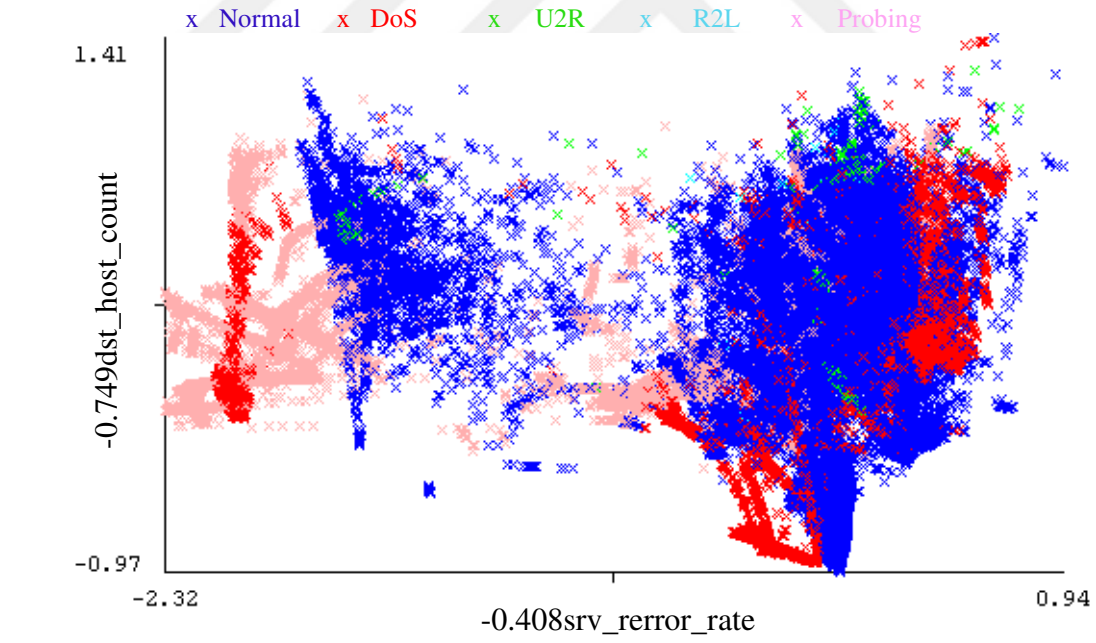
Şekil 5.8: Boyut Azaltma İşlemi Sonrası Veri (srv_diff_host_rate) Dağılımı.



Şekil 5.9: Boyut Azaltma İşlemi Sonrası Veri (dst_host_srv_serror_rate) Dağılımı.



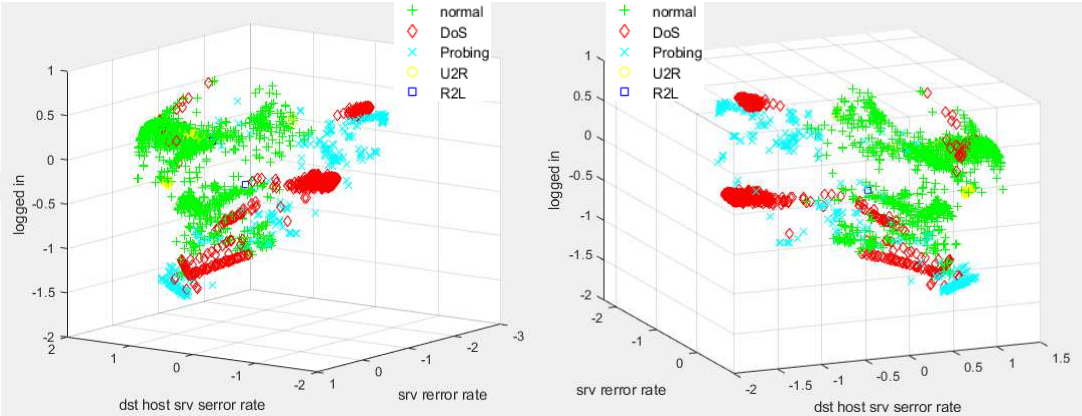
Şekil 5.10: Boyut Azaltma İşlemi Sonrası Veri (logged_in) Dağılımı.



Şekil 5.11: Boyut Azaltma İşlemi Sonrası Veri (dst_host_count) Dağılımı.

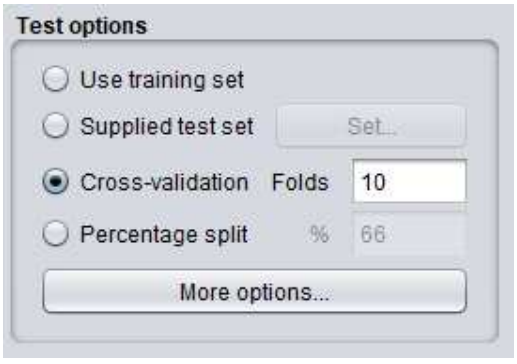
Boyut azaltma işlemi öncesi özelliklerin sınıflarla ilişkisi için çok fazla bir yorum yapılamazken, boyut azaltma işleminden sonra meydana gelen veri dağılımı, tehdit olarak etiketlenen veri hakkında fikir vermekte, sınıf ayıraçlarının tespit edilebilmesine olanak sağlamaktadır.

Sınıf ayıraçlarının tespit edilebilmesi ve iki boyutlu grafikte iç içe geçmiş durumda bulunan sınıf etiketlerinin daha iyi anlaşılabilmesi maksadıyla MATLAB programı kullanılarak veri üç boyutlu şekilde görselleştirilmiştir (Şekil 5.12). Boyut azaltma işlemine tabi tutularak elde edilen altı özelliğin içerisinde en yüksek varyansa sahip üç özellik seçilmiş ve veri setinin içerisinden beş bin adet örneklem rastgele seçilerek oluşturulan üç boyutlu grafik, verinin ve sınıf ayıraçlarının daha iyi anlaşılabilmesi amacıyla iki farklı açıdan sunulmuştur.



Şekil 5.12: Verinin Üç Boyutta Sergilenmesi.

WEKA programı, model oluşturma ve test işlemini, eğitim seti kullanma (use training set), test seti destekli (supplied test set), k-bölmeli çapraz doğrulama (k-fold cross-validation) ve yüzde bölme (percentage split) olmak üzere dört farklı yöntem ile yapabilmektedir (Şekil 5.13). Bu dört yöntemden en sık tercih edilen yöntem k-bölmeli çapraz doğrulama yöntemidir (Kohavi, 1995).



Şekil 5.13: Model Oluşturma ve Test İşlemleri.

Bu yöntem ile WEKA programına verilen eğitim veri seti k bölüme ayrılarak bir bölümü test için, diğer bölümleri ise model oluşturmak için kullanılır ve işlem k defa tekrarlanır. Bu tez kapsamında yapılan çalışmalar esnasında, algoritmalar koşulurken k-bölmeli çapraz doğrulama ve test seti destekli yöntemler ile kırk bir özelliğe sahip veri seti ve boyut azaltma işlemi neticesinde elde edilen altı özelliğe sahip veri seti kullanılmıştır. Elde edilen sonuçlar kıyaslamaların kolay olması açısından tek bir tabloda sunulmuştur.

5.4.1. Yakın Komşu Algoritması

Yakın komşu algoritmasında, sınıflandırılacak veri, k sayıda yakın komşuları göz önünde bulundurularak uzaklık hesabı yapılması ile sınıflandırılmaktadır. Uzaklık hesabında en sık kullanılan fonksiyonlar, Öklid ve Manhattan fonksiyonlarıdır (Zhang, 2016). WEKA programında yer alan yakın komşu algoritmasının ilksel uzaklık fonksiyonu Öklid fonksiyonudur.

Yakın komşu algoritmasının performans değerlendirmesi maksadıyla yapılan uygulama k=1 ve k=5 (1 ve 5 yakın komşular) olmak üzere iki farklı değer seçilerek gerçekleştirilmiştir. WEKA programı ile her iki örneklem için test seti destekli test yöntemi (Tablo 5.5, Tablo 5.6, Tablo 5.9 ve Tablo 5.10) ve 10-bölmeli çapraz doğrulama yöntemi (Tablo 5.7, Tablo 5.8, Tablo 5.11 ve Tablo 5.12) uygulanmıştır. Tablo satırlarında üst hücredeki veriler kırk bir özellik kullanılarak elde edilen sonuçları, alt hücredeki veriler ise altı özellik kullanılarak elde edilen sonuçları göstermektedir.

Test seti destekli yöntem uygulanırken veri seti olarak “Normal”, “DoS”, “U2R”, “R2L” ve “Probing” etiketli KDDTrain ve KDDTest veri setleri, 10-bölmeli çapraz doğrulama yöntemi uygulanırken “Normal”, “DoS”, “U2R”, “R2L” ve “Probing” etiketli KDDTrain veri seti kullanılmıştır. Uygulama sonucu ortaya çıkan veriler altıncı bölümde değerlendirilmiştir.

Test seti destekli test yöntemi ile 1-Yakın Komşu Algoritması, kırk bir özelliği kullanarak 676,71 saniye sürede % 77,09, altı özelliği kullanarak 410,16 saniye sürede % 39,66 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5.5: Test Seti Destekli 1-Yakın Komşu Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		Normal	DoS	U2R	R2L	Probing
Normal	41 özellik	9342	57	3	2	307
	6 özellik	8887	2	95	329	398
DoS	41 özellik	1139	6050	189	10	70
	6 özellik	7329	6	2	121	0
U2R	41 özellik	2523	2	184	40	5
	6 özellik	2470	0	13	271	0
R2L	41 özellik	170	0	2	20	8
	6 özellik	162	0	2	36	0
Probing	41 özellik	453	174	8	2	1784
	6 özellik	1891	24	159	347	0

Tablo 5.6: Test Seti Destekli 1-Yakın Komşu Algoritması Değerlendirme Kriterleri.

Değerlendirme Kriteri / Sınıf Etiketi		Kesinlik	Duyarlılık	F-Ölçütü
Normal	41 özellik	0,686	0,962	0,801
	6 özellik	0,429	0,915	0,584
DoS	41 özellik	0,963	0,811	0,881
	6 özellik	0,188	0,001	0,002
U2R	41 özellik	0,477	0,067	0,117
	6 özellik	0,048	0,005	0,009
R2L	41 özellik	0,270	0,100	0,146
	6 özellik	0,033	0,180	0,055
Probing	41 özellik	0,821	0,737	0,776
	6 özellik	0	0	0

10-bölmeli çapraz doğrulama yöntemi ile 1-Yakın Komşu Algoritması, kırk bir özelliği kullanarak 2998,7 saniye sürede % 99,72, altı özelliği kullanarak 2670,33 saniye sürede % 99,39 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5.7: 10-Bölmeli Çapraz Doğrulama Yöntemi 1-Yakın Komşu Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		Normal	DoS	U2R	R2L	Probing
Normal	41 özellik	67179	50	43	12	59
	6 özellik	66954	169	105	17	98
DoS	41 özellik	43	45881	0	0	3
	6 özellik	128	45792	0	0	7
U2R	41 özellik	58	1	932	3	1
	6 özellik	98	3	891	3	0
R2L	41 özellik	19	0	6	27	0
	6 özellik	27	0	3	22	0
Probing	41 özellik	50	9	0	0	11597
	6 özellik	96	12	0	0	11548

Tablo 5.8: 10-Bölmeli Çapraz Doğrulama Yöntemi 1-Yakın Komşu Algoritması Değerlendirme Kriterleri.

Değerlendirme Kriteri / Sınıf Etiketi		Kesinlik	Duyarlılık	F-Ölçütü
Normal	41 özellik	0,997	0,998	0,998
	6 özellik	0,995	0,994	0,995
DoS	41 özellik	0,999	0,999	0,999
	6 özellik	0,996	0,997	0,997
U2R	41 özellik	0,950	0,937	0,943
	6 özellik	0,892	0,895	0,894
R2L	41 özellik	0,643	0,519	0,574
	6 özellik	0,524	0,423	0,468
Probing	41 özellik	0,995	0,995	0,995
	6 özellik	0,991	0,991	0,991

Test seti destekli test yöntemi ile 5-Yakın Komşu Algoritması, kırk bir özelliği kullanarak 677,51 saniye sürede % 76,9, altı özelliği kullanarak 390,13 saniye sürede % 41,37 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5.9: Test Seti Destekli 5-Yakın Komşu Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		Normal	DoS	U2R	R2L	Probing
Normal	41 özellik	9352	57	3	3	296
	6 özellik	9325	1	88	0	297
DoS	41 özellik	1129	6081	0	0	48
	6 özellik	7453	1	4	0	0
U2R	41 özellik	2627	2	121	0	4
	6 özellik	2754	0	0	0	0
R2L	41 özellik	165	0	1	17	17
	6 özellik	200	0	0	0	0
Probing	41 özellik	431	182	43	0	1765
	6 özellik	2277	10	134	0	0

Tablo 5.10: Test Seti Destekli 5-Yakın Komşu Algoritması Değerlendirme Kriterleri.

Değerlendirme Kriteri / Sınıf Etiketi		Kesinlik	Duyarlılık	F-Ölçütü
Normal	41 özellik	0,673	0,963	0,792
	6 özellik	0,424	0,960	0,588
DoS	41 özellik	0,962	0,815	0,883
	6 özellik	0,083	0	0
U2R	41 özellik	0,720	0,044	0,083
	6 özellik	0	0	0
R2L	41 özellik	0,850	0,085	0,155
	6 özellik	0	0	0
Probing	41 özellik	0,829	0,729	0,776
	6 özellik	0	0	0

10-bölmeli çapraz doğrulama yöntemi ile 5-Yakın Komşu Algoritması, kırk bir özelliği kullanarak 2267,13 saniye sürede % 99,57, altı özelliği kullanarak 1840,23 saniye sürede % 99,18 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5-11: 10-Bölmeli Çapraz Doğrulama Yöntemi 5-Yakın Komşu Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		Normal	DoS	U2R	R2L	Probing
Normal	41 özellik	67142	76	74	7	44
	6 özellik	66850	257	111	9	116
DoS	41 özellik	61	45855	0	0	8
	6 özellik	181	45727	1	0	18
U2R	41 özellik	92	2	899	2	0
	6 özellik	129	3	859	3	1
R2L	41 özellik	28	0	1	22	1
	6 özellik	36	0	2	13	1
Probing	41 özellik	115	34	0	0	11507
	6 özellik	145	19	0	0	11492

Tablo 5.12: 10-Bölmeli Çapraz Doğrulama Yöntemi 5-Yakın Komşu Algoritması Değerlendirme Kriterleri.

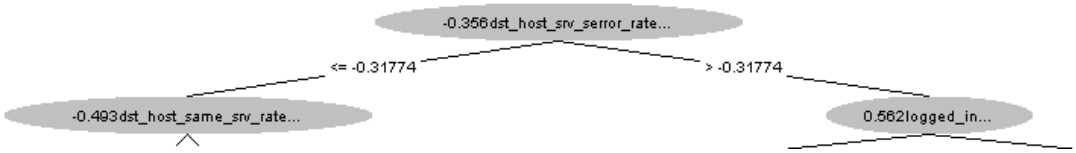
Değerlendirme Kriteri / Sınıf Etiketi		Kesinlik	Duyarlılık	F-Ölçütü
Normal	41 özellik	0,996	0,997	0,996
	6 özellik	0,993	0,993	0,993
DoS	41 özellik	0,998	0,998	0,998
	6 özellik	0,994	0,996	0,995
U2R	41 özellik	0,923	0,904	0,913
	6 özellik	0,883	0,863	0,873
R2L	41 özellik	0,710	0,423	0,530
	6 özellik	0,520	0,250	0,338
Probing	41 özellik	0,995	0,987	0,991
	6 özellik	0,988	0,986	0,987

5.4.2. Karar Ağacı Algoritması

Karar ağacı algoritmasında, eğitim veri setinden oluşturulan karar ağacı kullanılarak sınıflandırılacak verinin sınıfı belirlenir. Karar ağacı oluşturulurken öncelikle kök düğümü (root node) belirlenir. Kök düğümü belirlenirken örnekleri en iyi ayıran özellik seçilir. Ardından bu işlem yaprak düğümlerde (leaf node) tekrarlanarak ağacın yapısı belirlenir (Aksu ve Doğan, 2019).

Karar aracı algoritmasının performans değerlendirmesi maksadıyla yapılan uygulamada J48(C4.5) karar ağacı algoritması kullanılarak, WEKA programı ile test seti destekli test yöntemi (Tablo 5.13 ve Tablo 5.14) ve 10-bölmeli çapraz doğrulama yöntemi (Tablo 5.15 ve Tablo 5.16) uygulanmıştır. Tablo satırlarında üst hücredeki veriler kırk bir özellik kullanılarak elde edilen sonuçları, alt hücredeki veriler ise altı özellik kullanılarak elde edilen sonuçları göstermektedir. Uygulama neticesinde kırk bir özellik kullanıldığında, 14 düğüm (1 kök ve 13 yaprak düğüm) derinlikli ve 422 düğüme sahip karar ağacı, altı özellik kullanıldığında ise 32 düğüm (1 kök ve 31 yaprak düğüm) derinlikli ve 475 düğüme sahip karar ağacı elde edilmiştir. Kırk bir özellik için karar ağacının başlangıç düğümleri Şekil 5.14'te, altı özellik için Şekil 5.15'te gösterilmiştir.

Şekil 5.14: Kırk Bir Özellik ile Karar Ağacı.



Şekil 5.15: Altı Özellik ile Karar Ağacı.

Test seti destekli yöntem uygulanırken veri seti olarak "Normal", "DoS", "U2R", "R2L" ve "Probing" etiketli KDDTrain ve KDDTest veri setleri, 10-bölmeli çapraz

doğrulama yöntemi uygulanırken “Normal”, “DoS”, “U2R”, “R2L” ve “Probing” etiketli KDDTrain veri seti kullanılmıştır. Uygulama sonucu ortaya çıkan veriler altıncı bölümde değerlendirilmiştir.

Test seti destekli test yöntemi ile Karar Ağacı Algoritması, kırk bir özellik kullanarak 37,33 saniye sürede % 75,26, altı özellik kullanarak 5,97 saniye sürede % 45,58 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5.13: Test Seti Destekli Karar Ağacı Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		**Normal**	**DoS**	**U2R**	**R2L**	**Probing**
Normal	41 özellik	9420	82	3	1	205
	6 özellik	9701	0	0	0	10
DoS	41 özellik	1598	5772	3	0	85
	6 özellik	6894	563	1	0	0
U2R	41 özellik	2279	0	180	3	292
	6 özellik	2754	0	0	0	0
R2L	41 özellik	181	1	2	9	7
	6 özellik	198	0	0	0	2
Probing	41 özellik	620	216	0	0	1585
	6 özellik	2396	4	0	0	12

Tablo 5.14: Test Seti Destekli Karar Ağacı Algoritması Değerlendirme Kriterleri.

Değerlendirme Kriteri / **Sınıf Etiketi**		**Kesinlik**	**Duyarlılık**	**F-Ölçütü**
Normal	41 özellik	0,668	0,970	0,791
	6 özellik	0,442	0,999	0,613
DoS	41 özellik	0,951	0,774	0,853
	6 özellik	0,993	0,075	0,140
U2R	41 özellik	0,957	0,065	0,122
	6 özellik	0	0	0
R2L	41 özellik	0,692	0,045	0,085
	6 özellik	0	0	0
Probing	41 özellik	0,729	0,655	0,690
	6 özellik	0,500	0,005	0,010

10-bölmeli çapraz doğrulama yöntemi ile Karar Ağacı Algoritması, kırk bir özellik kullanarak 457,15 saniye sürede % 99,76, altı özellik kullanarak 79,02 saniye sürede % 99,09 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5.15: 10-Bölmeli Çapraz Doğrulama Yöntemi Karar Ağacı Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		Normal	DoS	U2R	R2L	Probing
Normal	41 özellik	67231	32	20	8	52
	6 özellik	66844	255	97	13	134
DoS	41 özellik	25	45894	0	0	8
	6 özellik	164	45724	1	1	37
U2R	41 özellik	50	2	941	2	0
	6 özellik	180	3	811	1	0
R2L	41 özellik	21	1	0	28	2
	6 özellik	35	1	2	14	0
Probing	41 özellik	63	10	3	0	11580
	6 özellik	184	40	1	0	11431

Tablo 5.16: 10-Bölmeli Çapraz Doğrulama Yöntemi Karar Ağacı Algoritması Değerlendirme Kriterleri.

Değerlendirme Kriteri / Sınıf Etiketi		Kesinlik	Duyarlılık	F-Ölçütü
Normal	41 özellik	0,998	0,998	0,998
	6 özellik	0,992	0,993	0,992
DoS	41 özellik	0,999	0,999	0,999
	6 özellik	0,994	0,996	0,995
U2R	41 özellik	0,976	0,946	0,622
	6 özellik	0,889	0,815	0,851
R2L	41 özellik	0,737	0,538	0,994
	6 özellik	0,483	0,269	0,346
Probing	41 özellik	0,995	0,993	0,998
	6 özellik	0,985	0,981	0,983

5.4.3. Yapay Sinir Ağları Algoritması

Yapay sinir ağları algoritması, girdi katmanı, ara katman (gizli katman) ve çıkış katmanı olmak üzere en az üç katmandan oluşan bir yapıya sahiptir. Ara katman en az bir katman olabileceği gibi belirlenecek sayıda birden fazla katmandan da oluşabilir. Yapay sinir ağında öğrenme, geri yayılım (back propagation) ve threshold fonksiyonu ile sağlanır. Algoritma ayrıca, öğrenmenin yerel en iyileri aşması ve kendisini toparlamasında kullanılan momentum katsayısı ve ağırlıkların güncellenmesinde kullanılan öğrenme oranı değişkenlerini içermektedir (Arı ve Berberler, 2017).

Yapay sinir ağları algoritmasının performans değerlendirmesi maksadıyla yapılan uygulamada WEKA programında mevcut algoritmanın ilksel değerleri (gizli katman sayısı 23, sigmoid sayısı 67, momentum 0,2 ve öğrenme oranı 0,3) kullanılmış ve çok katmanlı algılayıcı algoritmasına, WEKA programı ile test seti destekli test yöntemi (Tablo 5.17 ve Tablo 5.18) ve 10-bölmeli çapraz doğrulama yöntemi (Tablo 5.19 ve Tablo 5.20) uygulanmıştır. Tablo satırlarında üst hücredeki veriler kırk bir özellik kullanılarak elde edilen sonuçları, alt hücredeki veriler ise altı özellik kullanılarak elde edilen sonuçları göstermektedir.

Test seti destekli yöntem uygulanırken veri seti olarak “Normal”, “DoS”, “U2R”, “R2L” ve “Probing” etiketli KDDTrain ve KDDTest veri setleri, 10-bölmeli çapraz doğrulama yöntemi uygulanırken “Normal”, “DoS”, “U2R”, “R2L” ve “Probing” etiketli KDDTrain veri seti kullanılmıştır. Uygulama sonucu ortaya çıkan veriler altıncı bölümde değerlendirilmiştir.

Test seti destekli test yöntemi ile Yapay Sinir Ağları Algoritması, kırk bir özellik kullanarak 10318,14 saniye sürede % 75,54, altı özellik kullanarak 91,92 saniye sürede % 43,08 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5.17: Test Seti Destekli Yapay Sinir Ağları Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		Normal	DoS	U2R	R2L	Probing
Normal	41 özellik	9016	447	1	0	247
	6 özellik	9711	0	0	0	0
DoS	41 özellik	1560	5817	1	0	80
	6 özellik	7458	0	0	0	0
U2R	41 özellik	2458	1	283	0	12
	6 özellik	2754	0	0	0	0
R2L	41 özellik	195	0	3	0	2
	6 özellik	200	0	0	0	0
Probing	41 özellik	274	193	33	7	1914
	6 özellik	2421	0	0	0	0

Tablo 5.18: Test Seti Destekli Yapay Sinir Ağları Algoritması Değerlendirme Kriterleri.

Değerlendirme Kriteri / Sınıf Etiketi		Kesinlik	Duyarlılık	F-Ölçütü
Normal	41 özellik	0,668	0,928	0,777
	6 özellik	0,431	1,000	0,602
DoS	41 özellik	0,901	0,780	0,836
	6 özellik	0	0	0
U2R	41 özellik	0,882	0,103	0,184
	6 özellik	0	0	0
R2L	41 özellik	0	0	0
	6 özellik	0	0	0
Probing	41 özellik	0,849	0,791	0,819
	6 özellik	0	0	0

10-bölmeli çapraz doğrulama yöntemi ile Yapay Sinir Ağları Algoritması, kırk bir özellik kullanarak 82045,11 saniye sürede % 99,02, altı özellik kullanarak 993,96 saniye sürede % 96,39 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5.19: 10-Bölmeli Çapraz Doğrulama Yöntemi Yapay Sinir Ağları Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		Normal	DoS	U2R	R2L	Probing
Normal	41 özellik	66930	169	157	0	87
	6 özellik	66523	421	14	0	385
DoS	41 özellik	360	45563	0	0	4
	6 özellik	1404	44322	0	6	195
U2R	41 özellik	269	0	726	0	0
	6 özellik	940	8	45	0	2
R2L	41 özellik	44	0	8	0	0
	6 özellik	50	1	0	0	1
Probing	41 özellik	130	4	1	0	11521
	6 özellik	917	207	0	0	10532

Tablo 5.20: 10-Bölmeli Çapraz Doğrulama Yöntemi Yapay Sinir Ağları Algoritması Değerlendirme Kriterleri.

Değerlendirme Kriteri / Sınıf Etiketi		Kesinlik	Duyarlılık	F-Ölçütü
Normal	41 özellik	0,988	0,994	0,991
	6 özellik	0,953	0,988	0,970
DoS	41 özellik	0,996	0,992	0,994
	6 özellik	0,986	0,965	0,975
U2R	41 özellik	0,814	0,730	0,769
	6 özellik	0,763	0,045	0,085
R2L	41 özellik	0	0	0
	6 özellik	0	0	0
Probing	41 özellik	0,992	0,988	0,990
	6 özellik	0,948	0,904	0,925

5.4.4. Lojistik Regresyon

WEKA programında lojistik regresyon algoritmasının hesaplanmasında logit modeli kullanılmaktadır. Değişkenler arasındaki ilişki, logit model sayesinde doğrusal olmayan "S" şeklinde bir eğri ile ifade edilir. Bahse konu eğri, değişkenlerin eğriye olan uzaklıkları logaritmik olarak hesaplanarak çizilmektedir (Ürük, 2007).

Denetimli öğrenme algoritmalarından lojistik regresyonun performans değerlendirmesi maksadıyla WEKA programı ile test seti destekli test yöntemi (Tablo 5.21 ve Tablo 5.22) ve 10-bölmeli çapraz doğrulama yöntemi (Tablo 5.23 ve Tablo 5.24) uygulanmıştır. Tablo satırlarında üst hücredeki veriler kırk bir özellik kullanılarak elde edilen sonuçları, alt hücredeki veriler ise altı özellik kullanılarak elde edilen sonuçları göstermektedir.

Test seti destekli yöntem uygulanırken veri seti olarak "Normal" ve "Anomaly" etiketli KDDTrain ve KDDTest veri setleri, 10-bölmeli çapraz doğrulama yöntemi uygulanırken "Normal" ve "Anomaly" etiketli KDDTrain veri seti kullanılmıştır. Uygulama sonucu ortaya çıkan veriler altıncı bölümde değerlendirilmiştir.

Test seti destekli test yöntemi ile Lojistik Regresyon Algoritması, kırk bir özellik kullanarak 58,69 saniye sürede % 75,61, altı özellik kullanarak 4,6 saniye sürede % 44,22 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5.21: Test Seti Destekli Lojistik Regresyon Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		Normal	Anomaly
Normal	41 özellik	8988	723
	6 özellik	9449	262
Anomaly	41 özellik	4776	8057
	6 özellik	12313	520

Tablo 5.22: Test Seti Destekli Lojistik Regresyon Algoritması Değerlendirme Kriterleri.

Değerlendirme Kriteri / Sınıf Etiketi		Kesinlik	Duyarlılık	F-Ölçütü
Normal	41 özellik	0,653	0,926	0,766
	6 özellik	0,434	0,973	0,600
Anomaly	41 özellik	0,918	0,628	0,746
	6 özellik	0,665	0,041	0,076

10-bölmeli çapraz doğrulama yöntemi ile Lojistik Regresyon Algoritması, kırk bir özellik kullanarak 664,63 saniye sürede % 97,5, altı özellik kullanarak 47,51 saniye sürede % 92,35 oranında doğru sınıflandırma gerçekleştirmiştir.

Tablo 5.23: 10-Bölmeli Çapraz Doğrulama Yöntemi Lojistik Regresyon Algoritması Karmaşıklık Matrisi.

Karmaşıklık Matrisi		Normal	Anomaly
Normal	41 özellik	66096	1247
	6 özellik	62168	5175
Anomaly	41 özellik	1908	56722
	6 özellik	4464	54166

Tablo 5.24: 10-Bölmeli Çapraz Doğrulama Yöntemi Lojistik Regresyon Algoritması Değerlendirme Kriterleri.

Değerlendirme Kriteri / Sınıf Etiketi		Kesinlik	Duyarlılık	F-Ölçütü
Normal	41 özellik	0,972	0,981	0,977
	6 özellik	0,933	0,923	0,928
Anomaly	41 özellik	0,978	0,967	0,973
	6 özellik	0,913	0,924	0,918

5.4.5. 2-merkezli Kümeleme Algoritması

k-merkezli kümeleme algoritmasında, öncelikle kümelerin merkezini oluşturan k adet nesne seçilir. Ardından diğer nesnelerin merkez nesnelere uzaklıkları genellikle öklid mesafe fonksiyonu ile hesaplanır. Yapılan hesaplama neticesinde kümeler oluşturulur ve oluşan kümelerin yeni merkezleri belirlenir. Algoritma, kümelerin merkez güncelleme işlemi sona erene kadar iteratif olarak devam ettirilir (Na ve diğ., 2010).

Denetimsiz öğrenme algoritmalarından k-merkezli kümeleme algoritmasının performans değerlendirmesi maksadıyla KDDTrain veri seti kullanılarak algoritma koşulmuştur.

k-merkezli kümeleme algoritmasının uygulanması esnasında KDDTrain veri setinin kırk ikinci özelliği olan etiket verisi kullanılmamış, "Normal" ve "Anomaly" ağ trafiği düşünülerek k=2 seçilmiştir. Uygulama sonucu ortaya çıkan veriler altıncı bölümde değerlendirilmiştir.

2-merkezli kümeleme algoritması, kırk bir özellik kullanıldığında 6,05 saniye sürede, altı özellik kullanıldığında 4,99 saniye sürede kümeleme işlemini tamamlamıştır. Uygulama sonucunda (Tablo 5.25), kırk bir özellikte 24039 veri "Küme 0 (Sıfır)-(Anomaly)", 101934 veri "Küme 1 (Bir)-(Normal)" olarak, altı özellikte 27345 veri "Küme 0 (Sıfır)", 98628 veri "Küme 1 (Bir)" olarak değerlendirilmiştir.

Tablo 5.25: 2-merkezli Kümeleme Algoritması Uygulama Sonuçları.

Sınıf Etiketi \ **Küme Değeri**		**Küme 0 (Anomaly)**	**Küme 1 (Normal)**
Normal	41 özellik	14007	53336
	6 özellik	17211	50312
Anomaly	41 özellik	10032	48598
	6 özellik	10134	48496
Toplam Veri ⇨		24039	101934
		27345	98628

Denetimli öğrenme algoritmaları ile kıyaslama açısından oluşan kümelerdeki verilerin etiket verileri incelendiğinde, kırk bir özellik kullanıldığında doğru kümeleme oranının % 50,3, altı özellik kullanıldığında doğru kümeleme oranının % 52,1 olduğu görülmüştür. Tablo satırlarında üst hücredeki veriler kırk bir özellik kullanılarak elde edilen sonuçları, alt hücredeki veriler ise altı özellik kullanılarak elde edilen sonuçları göstermektedir.

5.4.6. Apriori Algoritması

Apriori algoritması, tümevarım mantığıyla çalışan bir ilişkilendirme algoritmasıdır. Algoritma, öncelikle veri setinde bulunan verilerin kullanım sıklığını belirlemekte, en sık kullanılan veriler arasında ilişkilendirme yapmaktadır. İlişkilendirme kuralının oluşması için minimum destek ve minimum güven kriterlerinin sağlanması gerekmektedir (Al-Maolegi ve Arkok, 2014). İlişkilendirme algoritmalarını diğer öğrenme algoritmalarından ayıran en büyük özellik ise, matematiksel verilerin yanında kategorik veriler ile de başarılı bir şekilde çalışmasıdır.

Denetimsiz öğrenme algoritmalarından Apriori ilişkilendirme algoritmasının performans değerlendirmesi maksadıyla KDDTrain veri seti kullanılarak algoritma koşulmuştur. Apriori algoritmasının doğru çalışabilmesi maksadıyla WEKA programının, özellik seçici (feature selection) ve filtreleme yetenekleri kullanılarak KDDTrain veri seti bir takım ön işleme tabi tutulmuştur. Bu kapsamda, öncelikle KDDTrain veri setinde bulunan kırk bir özellik, özellik seçici algoritmalardan CFSSubsetEval (Korelasyon tabanlı alt küme seçici) ve BFS (En iyi en önce) arama algoritmaları kullanılarak Tablo 5.26'da gösterilen altı özelliğe indirgenmiş, ardından filtreleme özelliği kullanılarak anılan özelliklerin veri tipi "Nominal" olarak değiştirilmiştir.

Tablo 5.26: Apriori Algoritmasında Kullanılan Özellikler.

Özellik No	Özellik	Veri Tipi
4	Flag	Nominal
5	Src_Bytes	Nümerik
6	Dst_Bytes	Nümerik
12	Logged_in	Binary
26	Srv_serror_rate	Nümerik
30	Diff_srv_rate	Nümerik

Apriori algoritması sonucunda bulunan altı özellik ile, boyut azaltma (Temel Bileşenler Analizi) işlemi sonucunda bulunan altı özelliğin farklılık göstermesi, Apriori algoritmasında kullanılan CFSSubsetEval özellik seçici algoritmasının verilerin kullanım sıklığını belirleyerek, sık kullanılan verilen arasında korelasyon yapmasından kaynaklanmaktadır. Temel Bileşenler Analizinde ise algoritma adından da anlaşılacağı üzere verinin tüm özelliklerini inceleyip, temel özellikleri bulmakta ve tüm özellikler yerine azaltılmış sayıda özellik ile işlem yapılmasına imkan vermektedir.

Yapılan bu işlemin ardından kırk ikinci özelliği olan etiket verisi (Normal ve Anomaly) ve Tablo 5.26'da gösterilen altı özellik, Apriori algoritmasında koşulmuş, algoritma 600,38 saniye sürede ilişkilendirme işlemini tamamlamıştır.

Uygulama sonucunda en yüksek güvenilirlik (confidence) oranını veren on ilişki değerlendirmeye alınmış olup, algoritma neticesinde ortaya çıkan ilişkiler Tablo 5.27'de olduğu gibidir.

Tablo 5.27: Apriori Algoritması Uygulama Sonuçları.

İlişki No	Algoritma Tarafından Tespit Edilen Kriterler ve Değerleri	Kriterleri Sağlayan Veri Sayısı	Verilerin Ağırlıklı Etiket Verisi (Sayısı)	Güvenilirlik Oranı
1	(dst_bytes=0) ve (logged_in=0)	63589	Anomaly (56405)	% 89
2	(flag=SF) ve (diff_srv_rate=0)	68930	Normal (58900)	% 85
3	(flag=SF) ve (srv_serror_rate=0) ve (diff_srv_rate=0)	67497	Normal (57487)	% 85
4	(flag=SF)	74945	Normal (63393)	% 85
5	(flag=SF) ve (srv_serror_rate=0)	73511	Normal (61979)	% 84
6	(dst_bytes=0)	67967	Anomaly (57009)	% 84
7	(srv_serror_rate=0) ve (diff_srv_rate=0)	73181	Normal (60479)	% 83
8	(diff_srv_rate=0)	76217	Normal (62773)	% 82
9	(logged_in=0)	76121	Anomaly (56635)	% 74
10	(srv_serror_rate=0)	88754	Normal (65017)	% 73

Tablodan da anlaşılacağı üzere, Apriori algoritması en sık kullanılan özellikler ve değerlerini tespit edip, birbirleri ile ilişkilendirmiş ve ilişkilendirilmiş özelliklerin sınıf etiketini kullanarak, veri trafiğinin “Normal” veya “Anomaly” olduğuna karar vermiştir. Örneğin, dst_bytes=0 ve logged_in=0 durumlarını sağlayan toplam veri sayısı 63589 iken, bu ilişkiyi sağlayan verilerin 56405 adedi “Anomaly” veri trafiği olarak etiketlenmiştir. Bu durum, bahse konu özelliğe ve değere sahip verinin % 89 oranında zararlı veri trafiği olacağı ilişkisini ortaya koymaktadır. Uygulama sonucu ortaya çıkan veriler altıncı bölümde değerlendirilmiştir.

6. SONUÇ

Bu tez çalışması kapsamında yapılan uygulama sonuçları, algoritmaların çalışma süreleri, doğruluk ve F-ölçütü kriterleri kıyaslanarak değerlendirilmiştir.

Tablo 6.1: Test Seti Destekli Yöntem Uygulama Sonuçları Değerlendirmesi.

Algoritma		Süre (sn)		Doğruluk (%)	
		41 özellik	6 özellik	41 özellik	6 özellik
Yakın Komşu	1-Yakın Komşu	676,71	410,16	77,09	39,66
	5-Yakın Komşu	677,51	390,13	76,90	41,37
Karar Ağacı	J48 (C4.5)	37,33	5,97	75,26	45,58
Yapay Sinir Ağları	Çok Katmanlı Algılayıcı	10318,14	91,92	75,54	43,08
Regresyon	Lojistik Regresyon	58,69	4,6	75,61	44,22

Test seti destekli yöntem, saldırı modellemesi açısından gerçek hayatta karşılaşılabilecek durumu yansıtmaktadır. Nitekim test seti destekli yöntemde KDDTrain veri seti ile eğitilmiş modele uygulanan KDDTest veri seti ile sınıflandırma işlemi gerçekleştirilmektedir. Yapılan işlem sonucunda Tablo 6.1'de görüleceği üzere algoritmaların birbirlerine çok yakın doğruluk oranında sınıflandırma yaptığı, ancak bazı algoritmaların çalışma sürelerinin çok fazla olduğu görülmüştür. Diğer taraftan boyut azaltma işlemine tabi tutulan verinin istenilen seviyede doğru sınıflandırma yapamadığı, ancak kırk bir özelliğe sahip veriye göre daha hızlı işlem gerçekleştirdiği tespit edilmiştir. Gerçek zamanlı saldırı tespiti yapan saldırı tespit sistemleri için saldırının mümkün olduğu kadar kısa sürede tespit edilmesi gerekliliği göz önüne alındığında süre-doğruluk orantısının çok iyi sağlanması gerekmektedir. Bu yöntemde, gerek kırk bir özellik gerekse altı özellik kullanıldığında karar ağacı algoritmasının en kısa işlem süresine sahip olduğu görülmektedir. Kırk bir özellik kullanıldığında en yüksek doğruluk oranı veren yakın komşu algoritmaları incelendiğinde ise, süre ve doğruluk oranı açısından çok büyük farklılıklar görülmezken, altı özellik kullanıldığında beş yakın komşu seçildiğinde

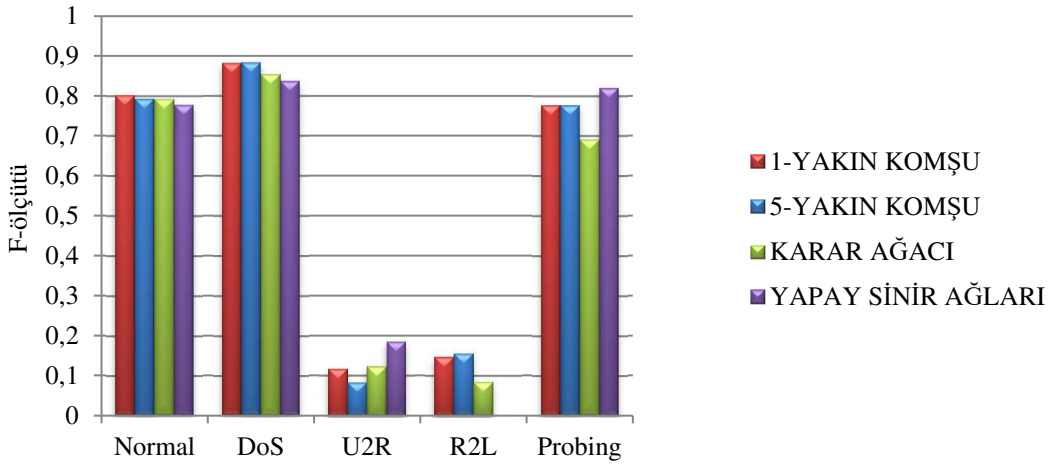
(k=5) hem sürenin kısaldığı, hemde bir yakın komşulara göre daha yüksek oranda doğru sınıflandırma gerçekleştiği anlaşılmaktadır. Ayrıca altı özellik kullanıldığında, çok katmanlı algılayıcı algoritmasının kırk bir özelliğe göre çok daha hızlı bir sürede sınıflandırma yaptığı tespit edilmiştir.

Tablo 6.2: 10-Bölmeli Çapraz Doğrulama Yöntemi Uygulama Sonuçları Değerlendirmesi.

Algoritma		Süre (sn)		Doğruluk (%)	
		41 özellik	6 özellik	41 özellik	6 özellik
Yakın Komşu	1-Yakın Komşu	2998,7	2670,38	99,72	99,39
	5-Yakın Komşu	2267,13	1840,23	99,57	99,18
Karar Ağacı	J48 (C4.5)	457,15	79,02	99,76	99,09
Yapay Sinir Ağları	Çok Katmanlı Algılayıcı	82045,11	993,96	99,02	96,39
Regresyon	Lojistik Regresyon	664,63	47,51	97,50	92,35

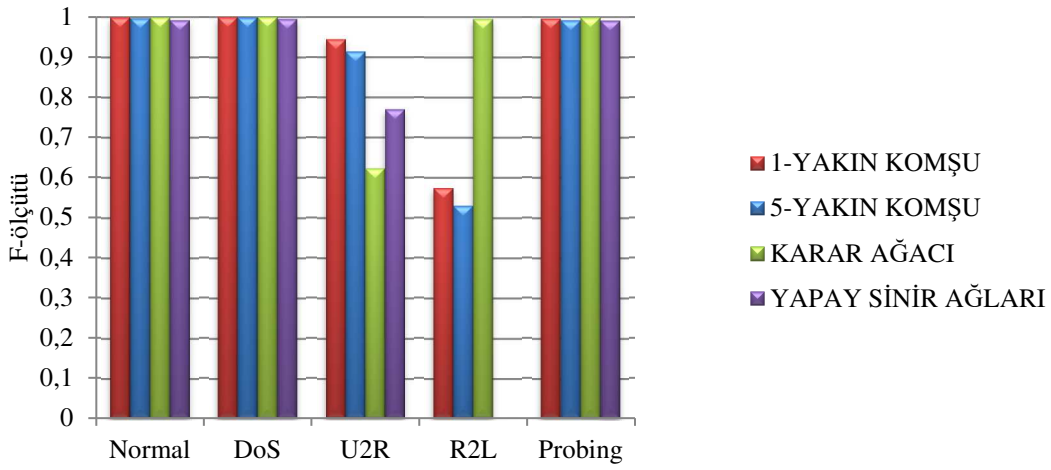
10-Bölmeli Çapraz Doğrulama yöntemi, KDDTrain veri seti içerisinde bölmelendirme yaparak aynı değerler ile sınıflandırma yaptığından çok yüksek oranda doğru sınıflandırma yapmaktadır. İşlem sürelerinin hem kırk bir özellik için, hemde altı özellik için test seti destekli yönteme göre daha uzun olduğu görülmektedir. Ancak, altı özellik kullanılarak elde edilen sonuçlar incelendiğinde, test seti destekli yöntemin aksine doğru sınıflandırma oranlarının kırk bir özellik kullanılarak elde edilen sonuçlara yakın olduğu görülmektedir. Tablo 6.2 incelendiğinde en kısa işlem süresi ve en yüksek doğruluk verisine sahip algoritmanın karar ağacı algoritması olduğu görülmektedir.

Yapılan çalışmanın değerlendirilmesi kapsamında kullanılan bir diğer kriter olan F-ölçütünün, “Normal”, “DoS” ve “Probing” olarak sınıflandırılan verilerde yüksek oranlarda sonuç verdiği tespit edilmiştir. Test seti destekli yöntem (Şekil 6.1 ve Şekil 6.4) ve 10-Bölmeli Çapraz Doğrulama yöntemi (Şekil 6.2 ve Şekil 6.5) ile gerçekleştirilen işlemler neticesinde, F-ölçütü değerinin doğruluk verisinde olduğu gibi, test seti destekli yöntemde daha düşük gerçekleştiği gözlemlenmiştir.



Şekil 6.1: Test Seti Destekli Yöntem F-ölçütü Verileri (41 Özellik).

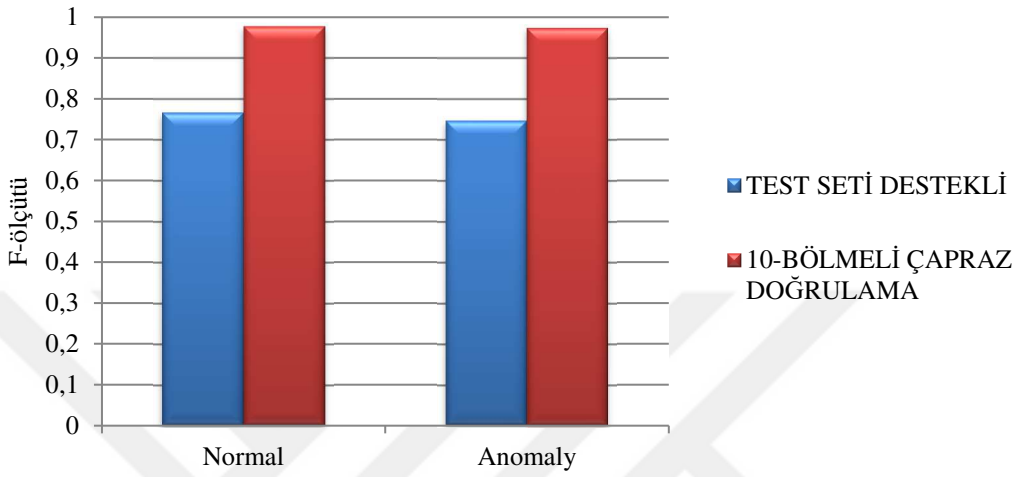
Şekil 6.1'de görüldüğü üzere, test seti destekli yöntemde kullanılan üç farklı algoritma da "U2R" ve "R2L" etiket verisine sahip verilerin sınıflandırılmasında düşük oranlarda başarı göstermişlerdir. Diğer yandan 10-Bölmeli Çapraz Doğrulama yönteminde (Şekil 6.2), "U2R" ve "R2L" etiket verisine sahip verilerin sınıflandırılmasındaki başarı oranının yükseldiği, ancak diğer üç etiket verisine sahip verilerde olduğu kadar yüksek oranlar (Karar ağacı algoritması-"R2L" hariç) olmadığı tespit edilmiştir.



Şekil 6.2: 10-Bölmeli Çapraz Doğrulama Yöntemi F-ölçütü Verileri (41 Özellik).

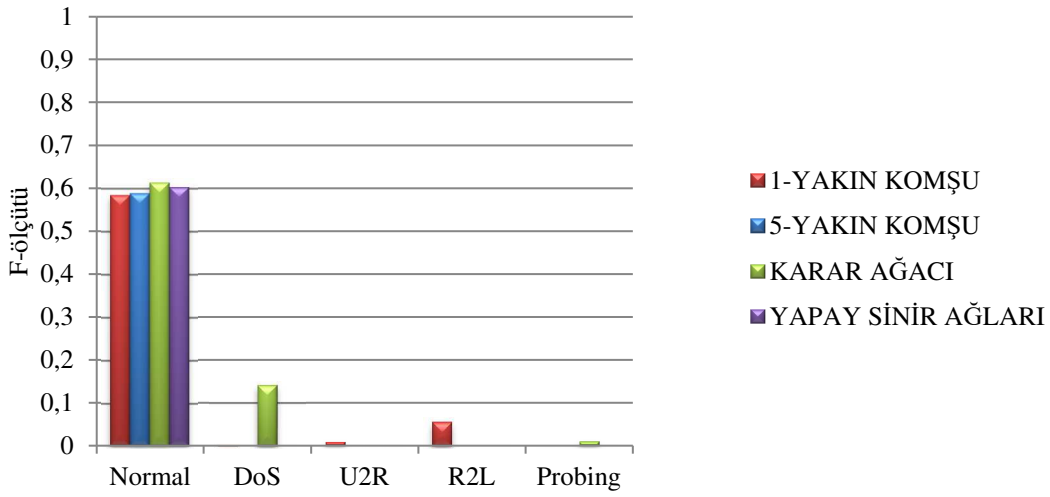
Her iki şekilde de yapay sinir ağları algoritmasına ait "R2L" F-ölçütü verilerinin olmadığı görülmektedir. Bunun sebebi sınıflandırıcının "R2L" etiket verisine sahip verileri doğru sınıflandıramamasından kaynaklanmaktadır.

Diğer bir denetimli öğrenme algoritması olan Lojistik Regresyon algoritmasına ait F-ölçütü değerleri incelendiğinde (Şekil 6.3), test seti destekli yöntemde ortaya çıkan verilerin diğer denetimli öğrenme algoritmalarına kıyasla daha düşük gerçekleştiği, 10-Bölmeli Çapraz Doğrulama yönteminde ise hemen hemen aynı oranlarda sınıflandırma yaptığı görülmüştür.



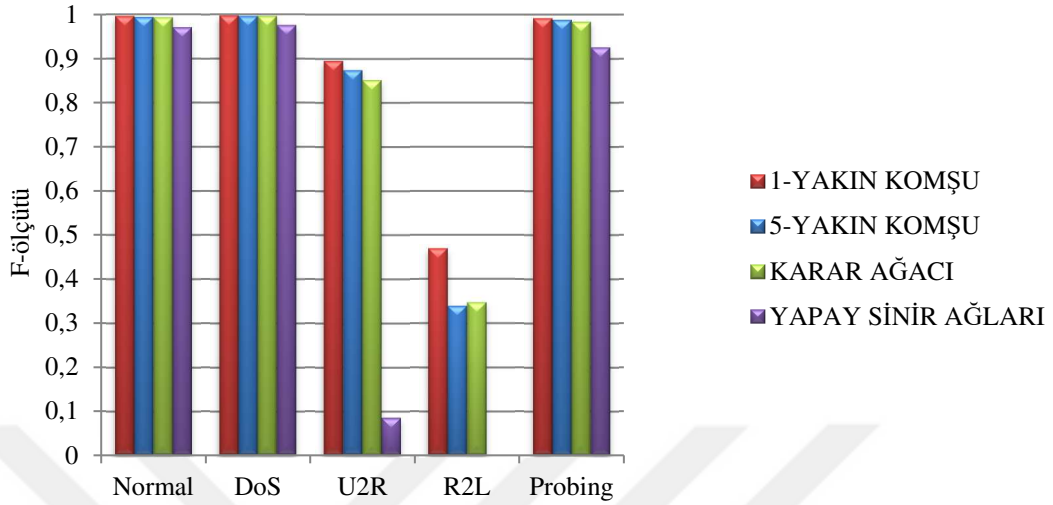
Şekil 6.3: Lojistik Regresyon Algoritması F-ölçütü Verileri (41 Özellik).

Boyut azaltma işlemi sonucunda elde edilen altı özellik ile gerçekleştirilen sınıflandırma işlemleri neticesinde ortaya çıkan F-ölçütü verileri Şekil-6.4, Şekil 6.5 ve Şekil 6.6'da olduğu gibidir. Altı özellik kullanılarak test seti destekli yöntem ile koşulan algoritmalar, doğruluk verisinde olduğu gibi F-ölçütü verisinde de oldukça düşük sonuçlar ortaya koymuştur. Şekil 6.4 incelendiğinde, F-ölçütü verilerinin "Normal" etiket verisine sahip verilerde kısmi sonuç verdiği görülmektedir.



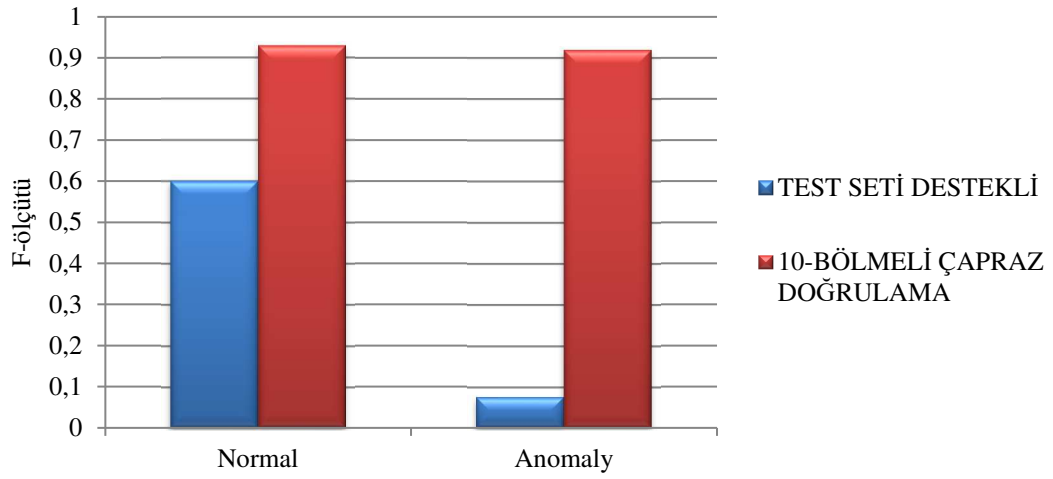
Şekil 6.4: Test Seti Destekli Yöntem F-ölçütü Verileri (6 Özellik).

Şekil 6.5 incelendiğinde ise, altı özellik kullanılarak 10-Bölmeli Çapraz Doğrulama yönteminde elde edilen sonuçların yüksek olmakla birlikte kırk bir özellik kullanılarak elde edilen sonuçlara kıyasla daha düşük olduğu görülmektedir.



Şekil 6.5: 10-Bölmeli Çapraz Doğrulama Yöntemi F-ölçütü Verileri (6 Özellik).

Altı özellik kullanılarak koşulan Lojistik Regresyon algoritmasına ait F-ölçütü değerleri incelendiğinde (Şekil 6.6), test seti destekli yöntemde ortaya çıkan verilerin diğer denetimli öğrenme algoritmalarına kıyasla daha düşük gerçekleştiği, 10-Bölmeli Çapraz Doğrulama yönteminde ise hemen hemen aynı oranlarda sınıflandırma yaptığı görülmüştür. Altı özellik kullanılarak koşulan Lojistik Regresyon algoritmasında da diğer algoritmalarda olduğu gibi, kırk bir özellik kullanılarak elde edilen sonuçlara kıyasla daha düşük oranda başarı sağlandığı tespit edilmiştir.



Şekil 6.6: Lojistik Regresyon Algoritması F-ölçütü Verileri (6 Özellik).

Denetimsiz öğrenme algoritmaları incelendiğinde ise, kırk bir özellik kullanılarak koşulan 2-merkezli kümeleme algoritmasının, KDDTrain veri seti ile gerçekleştirilen uygulama sonuçları neticesinde 6,05 saniye sürede % 50,3 oranında, altı özellik kullanılarak koşulan 2-merkezli kümeleme algoritmasının ise, 4,99 saniye sürede % 52,1 oranında kümeleme işlemi yaptığı görülmektedir. Kırk bir özellik kullanılarak koşulan 2-merkezli kümeleme algoritması, her ne kadar denetimli öğrenme algoritmaları ile kıyaslandığında süre olarak avantaj sağlasada, doğruluk oranının çok düşük olduğu görülmektedir. Diğer taraftan, altı özellik kullanılarak koşulan 2-merkezli kümeleme algoritması ile elde edilen doğru sınıflandırma oranının, altı özellik kullanılarak test seti destekli yöntem ile koşulan denetimli öğrenme algoritmalarında elde edilen doğru sınıflandırma oranlarından yüksek olduğu tespit edilmiştir.

Apriori algoritması ise uygulama öncesinde özellik seçici algoritma ile desteklenmesi ve filtreleme işlemine tabi tutulmasından dolayı diğer öğrenme algoritmalarından ayrılmakla birlikte 600,38 saniye sürede % 70 ile % 90 arasında değişen oranlarda güvenilirlik seviyesinde ilişkilendirme yapmıştır. Diğer algoritmalardan farklı olarak, Apriori algoritmasında ilişkilendirme yapılacak veriler arasındaki güvenilirlik oranının daha fazla olması sağlanabilmektedir. Bir başka deyişle, örnek uygulamada % 70 ile % 90 arasında olan güvenilirlik oranı, daha düşük veya daha yüksek olacak şekilde belirlenebilmektedir. Ancak bu durum daha yüksek güvenilirlik oranı için daha fazla süreye tekabül etmektedir.

Sonuç olarak, bu tez çalışması kapsamında incelenen tüm öğrenme algoritmalarına ait sonuçlar incelendiğinde; en hızlı çalışan ancak en düşük sınıflandırma oranına sahip öğrenme algoritmasının, denetimsiz öğrenme algoritmalarından kümeleme algoritması olduğu, diğer bir denetimsiz öğrenme algoritması olan Apriori algoritmasının istenilen doğruluk oranlarında ilişkilendirme yapabildiği ancak bu işlem öncesinde bir takım ön işlemlere tabi tutulması gerektiği, süre açısından değerlendirildiğinde ise, denetimli öğrenme algoritmalarına yakın sürelerde işlem yaptığı görülmüştür. Denetimli öğrenme algoritmalarından 5-Yakın Komşu algoritmasının test seti destekli yöntemde diğer algoritmalara kıyasla daha iyi sınıflandırma yaptığı, 10-bölmeli çapraz doğrulama yönteminde ise, karar ağacı algoritmasının daha iyi performans gösterdiği tespit edilmiştir. Genel olarak, denetimli öğrenme algoritmalarının birbirlerine yakın doğruluk oranlarında

sınıflandırma yaptığı ancak süre açısından yukarıda bahsedilen algoritmaların diğerlerine kıyasla daha hızlı olduğu görülmüştür.

NSL KDD veri setinin kırk bir özellik içeren şekli ile boyut azaltma işlemine (Temel Bileşenler Analizi) tabi tutulmuş altı özellik içeren şekli kullanılarak gerçekleştirilen sınıflandırma işlemi neticesinde benzer sonuçların ortaya çıktığı gözlemlenmiş, uygulanan test metodları göz önüne alındığında tutarlı veriler elde edilmiştir. Boyut azaltma işlemine tabi tutulmuş verinin, test seti destekli yöntem ile algoritmalara uygulanması sonucu elde edilen çıktıların istenilen düzeyde sınıflandırma yapamamasını günümüzde gelişen farklı saldırı teknikleri ile açıklamak mümkündür. Nitekim öğrenme algoritmasının işleyişinde "öğrendiği" saldırı türleri gün geçtikçe farklı türlerde karşımıza çıkmaktadır. Yine 10-bölmeli çapraz doğrulama yöntemi ile elde edilen sonuçlarda boyut azaltma işlemi neticesinde süreden sağlanacak tasarrufu ortaya koyması açısından olumlu bir sonuç teşkil etmektedir.

Görüldüğü üzere, denetimli ve denetimsiz öğrenme algoritmalarının kıyaslanması maksadıyla kullanılabilecek kısıtlı sayıda veri bulunmaktadır. Her ne kadar bu durum algoritmalar arasında tercih işlemini zorlaştırsada, ağ tabanlı bilgisayar sistemlerine yönelik tehditlerin/ saldırıların tespit edilmesi amacıyla kullanılacak öğrenme algoritmasının, denetimli öğrenme algoritmaları arasından tercih edilmesinin ve işlenecek verinin bir ön işleme tabi tutulmasının (Boyut azaltma vb.), süre ve doğru sınıflandırma açısından yüksek performans ortaya koyacağı değerlendirilmektedir.

Günümüzde gelişen teknoloji ve artan internet kullanımı ile birlikte bilgisayar sistemlerinin güvenlik ihtiyacı sürekli olarak artış göstermektedir. Artan güvenlik ihtiyacına cevap vermek maksadıyla geliştirilen saldırı tespit sistemlerinde yapay öğrenme kullanılması büyük önem taşımaktadır.

Nitekim bilgisayar sistemlerine olan saldırılar;

- Saldırganların sisteme yetkisiz erişmek istemesi,

- Sistemde yetkili konumda bulunan kullanıcıların, yetkilerinin olmadığı konularda ilave ayrıcalık kazanmak istemesi ve

- Yetkili kullanıcıların kendilerine tanınan ayrıcalığı hatalı kullanımı,

nedenleriyle gerçekleşmektedir (Zhang ve diğ., 2012). Saldırıların gerçekleştirilme nedenleri göz önüne alındığında, her geçen gün yeni tür saldırılar ile karşı karşıya

kalmak kaçınılmaz bir hal almaktadır. İşte bu nedenle, yapay öğrenmenin saldırı tespit sistemlerinde kullanılması, saldırı tespit sistemlerinin işlevselliğini arttırmakta ve gerçekleşebilecek yeni tür saldırıların en kısa sürede tespit edilebilmesine imkan sağlamaktadır.

Bu tez çalışması kapsamında, saldırı tespit sistemleri ve yapay öğrenme konularında inceleme yapılarak, denetimli ve denetimsiz öğrenme algoritmalarının performans değerlendirmesi gerçekleştirilmiştir. Denetimli ve denetimsiz öğrenme algoritmalarının koşulması maksadıyla, WEKA programı ve literatürde en sık kullanılan veri seti olan KDD CUP-99 veri setinden türetilmiş NSL KDD veri seti kullanılmıştır (Kaya ve Yıldız, 2014).

Gerçekleştirilen uygulamalar sonucunda, denetimli öğrenme algoritmaları ile ağ tabanlı bilgisayar sistemlerine yönelik tehditlerin/ saldırıların tespit edilme ihtimalinin daha yüksek olduğu tespit edilmiştir. Herhangi bir ön işleme tabi tutulmamış NSL KDD veri seti kullanılarak, denetimli öğrenme algoritmaları ile yapılan iki farklı test yönteminde de, algoritmaların sınıflandırma oranlarının birbirlerine yakın olduğu ancak işlem sürelerinin farklılık gösterdiği görülmüştür.

Diğer yandan denetimsiz öğrenme algoritmalarının, işlem süresi açısından hızlı olduğu ancak düşük doğruluk oranına sahip olduğu tespit edilmiştir. Denetimsiz öğrenme algoritmalarının bir diğer olumsuz tarafı ise sınıflandırma işlemi yapamamasıdır. Her ne kadar verileri kullanarak kümeleme veya ilişkilendirme işlemi gerçekleştirselerde, denetimsiz öğrenme algoritmalarının çıktılarının yorumlanması maksadıyla mutlaka bir kez daha işleme tabi tutulması gerekmektedir.

Ön işleme (Boyut azaltma) tabi tutulmuş NSL KDD veri seti kullanılarak icra edilen denemelerde ise, test seti destekli yöntemin çok düşük doğru sınıflandırma oranına sahip olduğu görülmüştür. 10-bölmeli çapraz doğrulama yönteminde ise, ön işleme tabi tutulmamış veriler ile yapılan uygulamalara benzer oranda sonuçlar verdiği, ancak algoritma çalışma sürelerinin daha az olduğu tespit edilmiştir.

Ağ tabanlı bilgisayar sistemlerine yönelik tehditlerin/ saldırıların tespit edilmesi kapsamında gelecekte icra edilecek çalışmalarda, denetimsiz ve denetimli öğrenme algoritmalarının birlikte çalışabileceği kombine bir algoritmanın incelenmesinin, ayrıca algoritmalara girdi olacak verilere boyut azaltma (dimensionality reduction)

ve/veya özellik seçimi (feature selection) gibi ön işlemlerin uygulanmasının faydalı olacağı değerlendirilmektedir.

Bu tez çalışması kapsamında uygulanan Apriori algoritması örneğinde olduğu gibi, denetimsiz öğrenme algoritmasından çıkan sonuçlar denetimli öğrenme algoritmasına girdi olarak uygulanırsa meydana gelecek kombine algoritmanın, tek başına denetimli öğrenme algoritmasının vereceği sonuçlara nazaran daha olumlu sonuç vereceği ve algoritma seçimine bağlı olarak tehdit/ saldırının daha kısa sürede tespit edilebileceği değerlendirilmektedir.

Aynı şekilde, algoritmalara girdi olacak veri setine boyut azaltma (dimensionality reduction) ve/veya özellik seçimi (feature selection) gibi ön işlemlerin uygulanmasının da performansı arttırabileceği, algoritmaların işlem süresini kısaltabileceği ve daha kesin sonuçlara götürebileceği değerlendirilmektedir.

KAYNAKÇA

Aksu, G., & Doğan, N. (2019). "Comparison of decision trees used in data mining". *Pegem Eğitim ve Öğretim Dergisi*, *9*(4), 1183–1208. doi: 10.14527/pegegog.2019.039.

Alkasassbeh, M., & Almseidin, M. (2018). "Machine learning methods for network intrusion detection". *International Conference on Computing, Communication (ICCCNT) ArXiv*, abs/1809.02610.

Al-Maolegi, M., & Arkok, B. (2014). "An improved apriori algorithm for association rules". *International Journal on Natural Language Computing*, *3*(1), 21–29. doi: 10.5121/ijnlc.2014.3103.

Alpaydin, E. (2010). *Introduction to machine learning*. The MIT Press.

Andress, J. (2011). *The basics of information security: Understanding the fundamentals of infosec in theory and practice*. Syngress.

Arı, A. & Berberler, M.E. (2017). "Yapay sinir ağları ile tahmin ve sınıflandırma problemlerinin çözümü için arayüz tasarımı". *Acta Infologica*, *1*(2), 55-73.

Avcı, U., & Avcı, M. (2004). "Örgütlerde bilginin önemi ve bilgi yönetimi süreci". *Mevzuat Dergisi*. http://www.mevzuatdergisi.com/2004/02a/01.htm. [Erişim Tarihi: 20.11.2021].

Aydın, M. A. (2005). *Bilgisayar ağlarında saldırı tespiti için istatistiksel yöntem kullanılması* (Yüksek Lisans Tezi). İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

Bilgi. (2021). In *Türk Dil Kurumu Sözlüğü*. Alındığı yer: https://www.sozluk.gov.tr. [Erişim tarihi: 20.11.2021].

Bellinger, G., Castro, D. & Mills, A. (2004). Data, Information, Knowledge, and Wisdom, http://www.systems-thinking.org/dikw/dikw.htm. [Erişim Tarihi: 20.11.2021].

Bijone, M. (2016). "A survey on secure network: Intrusion detection & prevention approaches". *American Journal of Information Systems*, *4*(3), 69-88. doi: 10.12691/ajis-4-3-2.

Brownlee, J. (2017). *Master machine learning algorithms: Discover how they work and implement them from scratch*. Machine Learning Mastery.

Brynjolfsson, E., & Mitchell, T. (2017). "What can machine learning do? Workforce implications". *Science*, *358*(6370), 1530–1534. doi: 10.1126/science.aap8062.

Chabathula, K. J., Jaidhar, C. D., & Ajay Kumara, M. A. (2015). "Comparative study of principal component analysis based intrusion detection approach using machine learning algorithms". *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*. doi: 10.1109/icscn.2015.7219853.

Chae, H., Jo, B., Choi, S., & Park, T. (2013). "Feature selection for intrusion detection using NSL-KDD". *Recent Advances in Computer Science*, 184-187.

Choudhary, S., & Kesswani, N. (2020). "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT". *Procedia Computer Science*, *167*, 1561-1573. doi: 10.1016/j.procs.2020.03.367.

Deshmukh, D. H., Ghorpade, T., & Padiya, P. (2015). "Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset". *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, 1–6. doi: 10.1109/iccict.2015.7045674.

Dhanaball, L. & Shantharajah, S. P. (2015). "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms". *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(6), 446-452.

Duque, S. & Bin Omar, M. N. (2015). "Using data mining algorithms for developing a model for intrusion detection system (IDS)". *Procedia Computer Science*, *61*, 46-51. doi: 10.1016/j.procs.2015.09.145.

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. doi: 10.1016/j.jisa.2019.102419.

Güvenlik Duvarı Nedir / Ne İşe Yarar? (2021). Alındığı yer: http://www.bilgimikoruyorum.org.tr/?b411_guvenlik-duvari-nedir. [Erişim Tarihi: 11.06.2021].

Halimaa A., A., & Sundarakantham, K. (2019). "Machine learning based intrusion detection system". *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. doi: 10.1109/icoei.2019.8862784

Hamid, Y., Sugumaran, M., & Journaux, L. (2016). "Machine learning techniques for intrusion detection: A comparative analysis". *Proceedings of the International Conference on Informatics and Analytics*. doi: 10.1145/2980258.2980378.

Ingre, B. & Yadav, A. (2015). "Performance analysis of NSL-KDD dataset using ANN". *International Conference on Signal Processing and Communication Engineering Systems*, 92-96. doi: 10.1109/SPACES.2015.7058223.

İlter, H. K. (2011). Bilgeliğe giden yol mideden geçer mi?. *PiVOLKA*, *20*(6), 3-7.

Karataş, G., Demir, O., & Şahingöz, O. K. (2018). "Deep learning in intrusion detection systems". *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT).* doi: 10.1109/ibigdelft.2018.8625278.

Kaya, Ç. (2016). *Saldırı tespit sistemlerinde makine öğrenmesi tekniklerinin kullanılması: Karşılaştırmalı performans analizi* (Yüksek Lisans Tezi). Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Kaya, Ç. & Yıldız, O. (2014). "Makine öğrenmesi teknikleriyle saldırı tespiti: Karşılaştırmalı analiz". *Marmara Fen Bilimleri Dergisi*, *26*(3), 108. doi: 10.7240/mufbed.24684.

Kemmerer, R. A. & Vigna, G. (2002). "Intrusion detection: a brief history and overview". *Computer*, *35*(4), 27-30. doi: 10.1109/MC.2002.1012428.

Kholidy, H. A., Erradi, A., Abdelwahed S. & Baiardi, F. (2013). "HA-CIDS: A hierarchical and autonomous IDS for cloud systems". *Fifth International Conference on Computational Intelligence, Communication Systems and Networks*, 179-184. doi: 10.1109/CICSYN.2013.9.

Klimburg, A. (2012). *National cyber security framework manual*. NATO Cooperative Cyber Defense Center of Excellence Publication.

Kocabıyık, L. (2005). *Information and knowledge management in the military domain* (Yüksek Lisans Tezi), Vrije Universiteit Brussel, Faculty of Economic, Social and Political Sciences, Brüksel.

Kohavi, R. (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". *International Joint Conference on Artificial Intelligence*, *14*(12), 1137-1143.

Kramer, S., & Bradfield, J. C. (2009). "A general definition of malware". *Journal in Computer Virology*, *6*(2), 105–114. doi: 10.1007/s11416-009-0137-1.

Kumar, I., Mohd, N., Bhatt, C., & Sharma, S. K. (2020). "Development of IDS using supervised machine learning". *Advances in Intelligent Systems and Computing*, 565-577. doi: 10.1007/978-981-15-4032-5_52.

Kurgun, O. A. (2006). "Bilgi yönetim sistemlerinin yapılandırılması", *Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, *8*(1), 274-291.

Latah, M. & Toker, L. (2018). "Towards an efficient anomaly-based intrusion detection for software-defined networks". *IET Networks*, *7*(6), 453-459. doi: 10.1049/iet-net.2018.5080.

Liu, G. G. (2014). "Intrusion detection systems". *Applied Mechanics and Materials*, *596*, 852–855. doi: 10.4028/www.scientific.net/amm.596.852.

Lunt, T. F., Tamaru, A., Gilham, F., Jagannathan, R., Neumann, P. G., Javitz, H. S., Valdes, A. & Garvey, T. D. (1992). "A real-time intrusion-detection expert system (IDES)". Alındığı yer: http://www.csl.sri.com/papers/9sri/9sri.pdf. [Erişim tarihi: 20.11.2021].

Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2019). "A detailed investigation and analysis of using machine learning techniques for intrusion detection". *IEEE Communications Surveys & Tutorials*, 21(1), 686-728. doi: 10.1109/comst.2018.2847722.

Musa, U. S., Chhabra, M., Ali, A., & Kaur, M. (2020). "Intrusion detection system using machine learning techniques: A review". *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. doi: 10.1109/icosec49089.2020.9215333.

Na, S., Xumin, L., & Yong, G. (2010). "Research on k-means clustering algorithm: An improved k-means clustering algorithm". *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, 63–67. doi: 10.1109/iitsi.2010.74.

Nguyen, T. T., & Armitage, G. (2008). "A survey of techniques for internet traffic classification using machine learning". *IEEE Communications Surveys & Tutorials*, *10*(4), 56–76. doi: 10.1109/surv.2008.080406.

Önaçan, M. B. K. (2015). *Organizasyonlar için bilgi yönetimi çerçevesi ve bilgi yönetim sistemi mimarisi önerisi: Doblyn (Doküman ve bilgi yönetimi)* (Doktora Tezi). Ankara Üniversitesi, Sosyal Bilimler Enstitüsü, Ankara.

Panigrahi, R., Borah, S., Bhoi, A. K., Ijaz, M. F., Pramanik, M., Jhaveri, R. H., & Chowdhary, C. L. (2021). "Performance assessment of supervised classifiers for designing intrusion detection systems: A comprehensive review and recommendations for future research". *Mathematics*, 9(6), 690–721. doi: 10.3390/math9060690.

Perez, D., Astor, M. A., Abreu, D. P., & Scalise, E. (2017). "Intrusion detection in computer networks using hybrid machine learning techniques". *2017 XLIII Latin American Computer Conference (CLEI)*. doi: 10.1109/clei.2017.8226392.

Devi, R., & Abualkibash, M. (2019). "Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets - A review paper". *International Journal of Computer Science and Information Technology*, 11(03), 65–80. doi: 10.5121/ijcsit.2019.11306.

Santosh, N., Saranyan, R., Senthil, k. P. & Vetriselvi V. (2008). "Cluster based co-operative game theory approach for intrusion detection in mobile Ad-Hoc grid". *16th International Conference on Advanced Computing and Communications*, 273-278. doi: 10.1109/ADCOM.2008.4760460.

Sharma, N. (2008). The origin of the "Data information knowledge wisdom" hierarchy. Alındığı yer: https://www.researchgate.net/publication/292335202_The_Origin_of_Data_Information_Knowledge_Wisdom_DIKW_Hierarchy. [Erişim tarihi: 20.11.2021].

Simeone, O. (2018). "A brief introduction to machine learning for engineers". *Foundations and Trends in Signal Processing*, *12*(3–4), 200–431. doi: 10.1561/2000000102.

Singh, P., & Venkatesan, M. (2018). "Hybrid approach for intrusion detection system". *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*. doi: 10.1109/icctct.2018.8551181.

Snapp, S. R., Smaha, S., Teal, D. M. & Grance T. (1992). "The DIDS (Distributed Intrusion Detection System) prototype", *USENIX Summer 1992 Technical Conference*.

Solomon, M. G. & Chapple, M. (2005). *Information security illuminated*. Jones and Bartlett.

Syarif, I., Prugel-Bennett, A., & Wills, G. (2012). "Unsupervised clustering approach for network anomaly detection". *Networked Digital Technologies*, 135–145. doi: 10.1007/978-3-642-30507-8_13.

Taher, K. A., Jisan, B. M. Y. & Rahman M. M. (2019). "Network intrusion detection using supervised machine learning technique with feature selection". International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 643-646. doi: 10.1109/ICREST.2019.8644161.

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. doi: 10.1109/ctit.2018.8649498.

Ürük, E. (2007). *İstatistiksel uygulamalarda lojistik regresyon analizi* (Yüksek Lisans Tezi). Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Whitman M. E. & Mattord, H. J. (2009). *Principles of information security*. Cengage Learning.

Witten, I. H., Hall, M., Frank, E., Holmes, G., Pfahringer, B., & Reutemann, P. (2009). "The WEKA data mining software". *ACM SIGKDD Explorations Newsletter*, *11*(1), 10–18. doi: 10.1145/1656274.1656278.

World internet usage and population statistics. (2021). Alındığı yer: https://www.internetworldstats.com/stats.htm. [Erişim tarihi: 20.11.2021].

Yiğidim, H. A. (2012). *Makine öğrenme algoritmalarını kullanarak ağ trafiğinin sınıflandırılması* (Yüksek Lisans Tezi). TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Yost, J. R. (2016). “The march of IDES: Early history of Intrusion-Detection expert systems”. *IEEE Annals of the History of Computing*, *38*(4), 42–54. doi: 10.1109/mahc.2015.41.

Zhang, X., Jia, L., Shi, H., Tang, Z., & Wang, X. (2012). “The application of machine learning methods to intrusion detection”. *2012 Spring Congress on Engineering and Technology*, 1–4. doi: 10.1109/scet.2012.6341943.

Zhang, Z. (2016). “Introduction to machine learning: K-nearest neighbors”. *Annals of Translational Medicine*, *4*(11), 218. doi: 10.21037/atm.2016.03.37.