
GAZİ ÜNİVERSİTESİ
Gazi
1926

SALDIRI TESPİT SİSTEMLERİNDE MAKİNE ÖĞRENMESİ TEKNİKLERİNİN KULLANILMASI: KARŞILAŞTIRMALI PERFORMANS ANALİZİ

Çetin KAYA

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2016

Çetin KAYA tarafından hazırlanan "SALDIRI TESPİT SİSTEMLERİNDE MAKİNE ÖĞRENMESİ TEKNİKLERİNİN KULLANILMASI: KARŞILAŞTIRMALI PERFORMANS ANALİZİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Öğr. Gör. Dr. Oktay YILDIZ
Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ..……

Başkan : Prof. Dr. M. Ali AKCAYOL
Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ..……

Üye : Doç. Dr. Ebru AKÇAPINAR SEZER
Bilgisayar Mühendisliği Anabilim Dalı, Hacettepe Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ..……

Üye : Doç. Dr. Suat ÖZDEMİR
Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ..……

Üye : Yrd. Doç. Dr. Selim TEMİZER
Bilgisayar Mühendisliği Anabilim Dalı, Orta Doğu Teknik Üniversitesi
Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ..……

Tez Savunma Tarihi: 23/06/2016

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..…….

Prof. Dr. Metin GÜRÜ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına 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.

Çetin KAYA

23/06/2016

SALDIRI TESPİT SİSTEMLERİNDE MAKİNE ÖĞRENMESİ TEKNİKLERİNİN KULLANILMASI: KARŞILAŞTIRMALI PERFORMANS ANALİZİ

(Yüksek Lisans Tezi)

Çetin KAYA

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2016

ÖZET

İnternet, günlük hayatımızın vazgeçilmez bir parçasıdır. Artan web uygulamaları ve kullanıcı sayısı, veri güvenliği açısından bazı riskleri de beraberinde getirmiştir. Ağ güvenliği için önemli araçlardan biri olan saldırı tespit sistemleri (STS), güvenli iç ağlara yapılan saldırıları ve beklenmeyen erişim taleplerini tespit etmede başarılı bir şekilde kullanılmaktadır. Günümüzde, pek çok araştırmacı, daha etkin saldırı tespit sistemi gerçekleştirilmesi amacıyla çalışma yapmaktadır. Bu amaçla literatürde farklı makine öğrenmesi teknikleri ile gerçekleştirilmiş pek çok saldırı tespit sistemi vardır ancak STS'lerde saldırı türlerine göre hangi makine öğrenmesi tekniği daha başarılıdır sorusuna cevap vermemektedir. Bizim çalışmamızda ise gerçekleştirilen deneylerle saldırı tespit sistemlerinde en sık kullanılan makine öğrenmesi tekniklerinden Bayes ağları, destek vektör makinesi, karar ağaçları, yapay sinir ağları ve k en yakın komşu algoritmasının performans analizi yapılmış ve saldırı türlerine göre doğruluk, seçicilik, duyarlılık, kesinlik, F-Ölçütü değerleri incelenerek en başarılı sınıflandırıcılar belirlenmiştir. Bu çalışma ile gelecekte yapılacak makine öğrenmesi teknikleri ile saldırı tespiti çalışmalarına bir bakış açısı kazandırılması amaçlanmıştır. Deneysel çalışmalarda KDD CUP99 ve NSL-KDD verisetleri kullanılmıştır.

Bilim Kodu : 92403
Anahtar Kelimeler : STS, Makine Öğrenmesi, KDD CUP99, Bilgi Güvenliği
Sayfa Adedi : 81
Danışman : Öğr. Gör. Dr. Oktay YILDIZ

USE OF MACHİNE LEARNİNG TECHNIQUES IN INTRUSION DETECTION SYSTEMS: COMPARATIVE ANALYSIS OF PERFORMANCE

(M. Sc. Thesis)

Çetin KAYA

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2016

ABSTRACT

Internet is an indispensable part of our daily lives. Increasing web applications and number of users has brought some risks in terms of data security. An intrusion detection system is one of the important tools for network security and is used successfully to detect attacks and unexpected demands made to secure access to internal network. Today, many researchers are working in order to realize more effective intrusion detection system. For this purpose, there are many intrusion detection system was performed using different machine learning techniques in the literature but according to the types of attacks, which machine learning techniques are more successful in IDS. It does not answer this question. In our study, with the experiments performed, the most commonly used machine learning techniques in intrusion detection systems that Bayesian network, support vector machines, decision trees, neural networks and k nearest neighbor algorithm performance analysis was conducted. According to the types of attacks, accuracy, specificity, sensitivity, accuracy, F-Measure values were examined and determined the most successful classifiers. With this study, Intrusion detection with machine learning techniques for future studies aimed to gain a perspective on. KDD CUP99 and NSL-KDD datasets were used in experimental studies.

Science Code : 92403
Key Words : IDS, Machine Learning, KDD CUP99, Information Security
Page Number : 81
Supervisor : Lec. Dr. Oktay YILDIZ

TEŞEKKÜR

Tezimin her aşamasında büyük bir özveri ve sabır ile desteklerini esirgemeyen çok değerli danışmanım saygıdeğer hocam Öğr. Gör. Dr. Oktay YILDIZ'a, maddi-manevi desteklerinden dolayı nişanlım İclal DENİZ ve aileme, tez süresince yardımlarını esirgemeyip destek olan arkadaşlarıma saygı ve teşekkürlerimi sunmayı bir borç bilirim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
Dk	Dakika
Sn	Saniye

Kısaltmalar	**Açıklamalar**
DARPA	Department of Defense Advanced Research Projects Agency
DDOS	Distributed Denial of Service
DN	Doğru Negatif
DOS	Denial of Service
DP	Doğru Pozitif
DVM	Destek Vektör Makinesi
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
KDD	Knowledge Discovery and Data Mining
KNN	K-Nearest Neighbor
R2L	Remote to Local
STS	Saldırı Tespit Sistemi
TS	Türk Standartları
U2R	User to Root
WEKA	Waikato Environment for Knowledge Analysis
YN	Yanlış Negatif
YP	Yanlış Pozitif
YSA	Yapay Sinir Ağları

1. GİRİŞ

Bilişim teknolojilerinin en önemli iki elemanı olan bilgisayar ve internet, altyapı sistemleri, bankacılık, finans, sağlık, eğitim, elektronik devlet uygulamaları gibi güvenlik yönünden kritik birçok sistemde yaygın bir şekilde kullanılmaktadır. Güvenlik yönünden kritik sistemlerin internet ortamına taşınması ve günlük hayatımızın vazgeçilmez bir parçası olması, bu sistemlere yapılan saldırıları her geçen gün daha da artırmaktadır. Akamai'nin 2015 yılı son çeyreğini (Ekim, Kasım ve Aralık ayları) kapsayan internet durum güvenliği raporuna göre dağıtık hizmet aksatma saldırıları (DDOS), 2014 son çeyreğine göre %148,85, 2015 üçüncü çeyreğine göre ise %39,89 artış göstermiştir. Yine aynı raporda, 2015 son çeyreğinde web uygulamalarına yönelik saldırılarda 2015 üçüncü çeyreğine göre %28,10'luk bir artış meydana gelmiştir [1]. İnternet ortamında faaliyet gösteren uygulamalara yönelik saldırıların, bu denli artış göstermesi, beraberinde veri güvenliği sorununu ortaya çıkarmıştır. Veri güvenliği, verilerin yetkisiz veya izinsiz erişimlerden, kullanımından, ifşa edilmesinden, yok edilmesinden, değişiklik yapılmasından yâda hasar verilmesinden korunması için yapılan tüm işlemler olarak tanımlanır ve gizlilik, bütünlük, süreklilik (erişilebilirlik) olarak adlandırılan üç temel unsurdan meydana gelir. Bu üç temel unsurdan herhangi biri zarar gördüğünde güvenlik zafiyeti oluşur. TS ISO/IEC 27001:2005 bilgi güvenliği yönetim standardına göre gizlilik, bütünlük ve erişilebilirlik kavramları aşağıdaki gibi açıklanmıştır.

Gizlilik: Verilerin ya da bilgilerin yetkisiz ve izinsiz erişimlere karşı korunmasıdır.

Bütünlük: Gönderici tarafından gönderilen verinin alıcıya herhangi bir değişikliğe uğramadan tam ve doğru bir şekilde iletilebilmesidir.

Erişilebilirlik: Bilginin, bilgiye erişim yetkisi olan tüm kullanıcılar tarafından istenildiği anda ulaşılabilir ve kullanılabilir olmasıdır.

İnternet ortamından bilgisayar sistemlerine yapılan saldırıları algılayıp sistem yöneticisine haber vermek ve saldırıları engellemek amacıyla güvenlik duvarı, antivirüs yazılımları ya da saldırı tespit sistemleri gibi donanımsal ve yazılımsal tabanlı birçok sistem kullanılmaktadır. Bilgi-Sistem güvenliği uzmanları, saldırganlardan ya da hackerlerden özel verileri ve bilgisayar sistemlerini korumak için bu sistemlerden bir ya da birkaçını

birlikte kullanabilmektedir. Tek başına bir güvenlik duvarı sistemine ya da antivirüs yazılımına güvenmek bilgisayar sistemlerini ve verilerimizi korumak için yeterli değildir. Bu nedenle, güvenlik duvarı ve antivirüs yazılımlarıyla birlikte bu araçların algılamakta güçlük çektiği saldırıları da analiz edip algılayabilen saldırı tespit sistemi kullanılır.

Saldırı tespit sistemi (STS), bilgisayar sistemlerindeki ağ trafiğini denetleyerek normal ya da anormal davranışları ayırt eden ve saldırı olarak kabul ettiği anormal davranış olması durumunda sistem yöneticisine haber veren alarm mekanizmalarıdır. Güvenlik sistemindeki son savunma mekanizması olarak işlev görürler. Saldırı tespit sistemlerinin bilgisayar sistemlerinde bu denli kritik öneme sahip oluşu, bu sistemlerin etkinliğini ve performansını artırmayı gerekli kılmaktadır. Makine öğrenmesi teknikleri, saldırı tespit sistemlerinde performansı artırmak amacıyla başarılı bir şekilde kullanılmaktadır.

2. LİTERATÜR ARAŞTIRMASI

Literatürde farklı makine öğrenmesi teknikleriyle geliştirilmiş pek çok saldırı tespit sistemi mevcuttur.

Farah Jemili ve diğerleri Bayes ağları kullanarak imza temelli otomatik saldırı tespit sistemi tasarlamışlardır. Amaçları, bilgisayar sisteminde gözlenen davranışları, bilinen saldırıların imzaları ile karşılaştırıp, bilinen saldırıların imzalarını tespit etmektir. Geliştirdikleri STS'yi DARPA 99 veriseti ile test ettiklerinde, DOS saldırılarında %99,62, bilgi tarama (PROBE) saldırılarında %100, U2R saldırılarında %98,63 ve R2L saldırılarında %42,62 doğrulukta başarı elde etmişlerdir [2].

Dewan ve diğerleri Bayes ağlarını, saldırı tespit sistemlerinin önemli problemlerinden biri olan yanlış alarm sayısını azaltmak için kullanmışlardır. Geliştirdikleri sisteme, gelişmiş öz uyarlamalı Bayes algoritması (Improved Self Adaptive Bayesian Algorithm-ISABA) ismini vermişler ve anormallik tespiti temelli STS'nin güvenlik kısmına entegre etmişlerdir. KDD CUP99 verisetini kullanarak geliştirdikleri sistemi test ettiklerinde, normal davranışları ayırt etmede %99,82, DOS saldırılarında %99,49, bilgi tarama saldırılarında %99,72, U2R saldırılarında %99,47 ve R2L saldırılarında %99,35 oranında bir başarı elde etmişlerdir [3].

Destek vektör makinesi kullanılarak geliştirilen STS çalışmalarında, Yongli Zhang ve Y. Zhu 2010 yılında en küçük kareler ve DVM kullanarak tasarladıkları STS'yi greedy algoritması ile geliştirmişlerdir. KDD CUP99 veriseti kullanılarak yapılan testlerde, normal davranışları %99,32, DOS saldırılarını %93,81, bilgi tarama saldırılarını %33,67, U2R saldırılarını %39,31 ve R2L saldırılarını da %99,42 oranında doğru sınıflandırmayı başarmışlardır [4].

Jun Wang ve diğerleri yapay arı kolonisi algoritmasını, destek vektör makinesinin sınıflandırma parametrelerinin en uygun değerlerini belirlemek için kullanmışlardır. Geliştirdikleri sistemi, KDD CUP99 veri setini kullanarak test etmişler ve normal davranışları ayırt etmede %100, DOS saldırılarını tespitte %99,92, bilgi tarama saldırılarında %100, U2R saldırılarında %76 ve R2L saldırılarında %87,92 oranında başarı elde etmişleridir [5].

Qi Mu ve diğerleri ise standart DVM'in yavaş algılama hızı ve düşük saldırı tespit oranını geliştirmek için sınır artırımlı destek vektör makinesi (Boundary Incremental Support Vector Machine, B-ISVM) olarak adlandırdıkları bir sistem geliştirmişlerdir. Sistemi KDD CUP99 veri setiyle test ettiklerinde, DOS saldırılarını tespitte %100, bilgi tarama saldırılarında %100, U2R saldırılarında %100 ve R2L saldırılarında %99,11'lik bir başarı gözlemlemişlerdir [6].

Karar ağaçları ile en yüksek sınıflandırma başarısı elde eden çalışmalara baktığımızda; M. Bahrololum ve diğerleri öznitelik seçimi için karar ağaçları, esnek sinir ağacı ve parçacık sürü optimizasyonu temelli üç yöntemin performanslarını kıyaslamıştır. Kıyaslama sonucunda üç yöntemden en iyi sonucu veren karar ağaçları olmuştur. Karar ağaçları ile KDD CUP99 veri seti kullanılarak yapılan test sonucunda, normal davranışları ayırt etmede %99,96, DOS saldırılarını tespitte %99,97, bilgi tarama saldırılarında %99,66, U2R saldırılarında %88,33 ve R2L saldırılarında %99,02 oranında sınıflandırma başarısı elde etmişlerdir [7].

2012 yılında Ammar Alazab ve diğerleri saldırı tespit sistemlerinde yüksek algılama başarısı ve düşük hata oranı elde edebilmek amacıyla, bilgi kazancı temelli öznitelik seçim modeli önermişlerdir. Önerilen modeli KDD CUP99 verisetini kullanarak test ettiklerinde, normal davranışları ayırt etmede %98,2, DOS saldırılarını tespitte %97,2, bilgi tarama saldırılarında %99,6, U2R saldırılarında % 92,5 ve R2L saldırılarında %99,7 oranında başarı elde etmişlerdir [8].

Karar ağaçları kullanılarak yapılan STS çalışmalarından bir diğeri 2013 yılında Vikas Sharma ve Aditi Nema tarafından genetik algoritma kullanılarak yapılmıştır. Geliştirilen STS, KDD CUP99 veri seti kullanılarak test edilmiştir. Sharma ve Nema, DOS saldırılarını tespitte %99,98, bilgi tarama saldırılarında %88,19, U2R saldırılarında % 51 ve R2L saldırılarında %94,70 oranında başarı elde etmişlerdir [9].

YSA kullanılarak gerçekleştirilen saldırı tespit sistemlerinde elde edilen sonuçlara bakıldığında, Guisong Liu ve diğerleri temel bileşen analizi ve yapay sinir ağları kullanarak (HPCANN-Hierarchical Principal Component Analysis Neural Networks) STS geliştirmişlerdir. Geliştirdikleri sistemi KDD CUP99 veri setini kullanarak test etmişler ve

normal davranışları %97,1, DOS saldırılarını %100, bilgi tarama saldırılarını %100 ve R2L saldırılarını da %97,2 başarı ile sınıflandırmışlardır [10].

2012 yılında; Xingchao Gong ve Xin Guan geri yayılımlı yapay sinir ağları ve uzman sistem kullanarak STS geliştirmişlerdir. KDD CUP99 verisetini kullanarak sistemi test ettiklerinde normalden daha az iterasyon süresi ile normal davranışları %100, DOS saldırılarını %100, bilgi tarama saldırılarını %99,76, U2R saldırılarını %99,85 ve R2L saldırılarını da %99,88 oranında doğru sınıflandırmayı başarmışlardır [11].

KNN kullanılarak yapılmış olan STS çalışmalarına baktığımızda ise şu sonuçlar çıkmaktadır. Wei-Chao Lin ve diğerleri 2015 yılında CANN (Cluster center And Nearest Neighbor) ismini verdikleri yeni bir sınıflama modeli önermişlerdir. Önerilen yöntemde, ilk olarak her bir veri örneği ve onun küme merkezi arasındaki uzaklık ölçülür, ikinci adımda, veri ve verinin aynı kümedeki en yakın komşusu arasındaki uzaklık ölçülür. Daha sonra, bu yeni ve tek boyutlu mesafe bazlı özellik, k-en yakın komşu (KNN) sınıflandırıcı tarafından saldırı tespiti için her bir veri örneğini temsil etmek için kullanılır. Önerilen model, KDD CUP99 verileriyle test edildiğinde, normal davranışları %99,68, DOS saldırılarını %99,98, bilgi tarama saldırılarını %98,49, U2R saldırılarını %17,31 ve R2L saldırılarını da %91,74 oranında doğru sınıflandırmıştır [12].

Chih-Fong Tsai ve Chia-Ying Lin ise saldırıları daha etkili tespit etmek için üçgen alanı içindeki en yakın komşuları esas alan hibrit bir STS (TANN- Triangle Area Based Nearest Neighbors) önermişleridir. Önerilen model, saldırı sınıfları ile ilgili küme merkezlerini elde etmek için k-means algoritmasını kullanır. K-means ile elde edilen iki küme merkezi ve veri setinden bir veri noktası üçgensel bölgeyi oluşturmak için kullanılır. Son olarak, k-NN sınıflandırıcı, üçgen alanlar tarafından temsil yeni özelliğe göre benzer saldırıları sınıflandırmak için kullanılır. Önerdikleri model ile yazarlar, KDD CUP99 verileriyle yaptıkları testte, normal davranışları %71,31, DOS saldırılarını %95,87, bilgi tarama saldırılarını %93,43, U2R saldırılarını %40 ve R2L saldırılarını da %85,84 oranında doğru ayırt etmeyi başarmışlardır [13].

2.1. Makine Öğrenmesi Tekniklerinin Karşılaştırılması

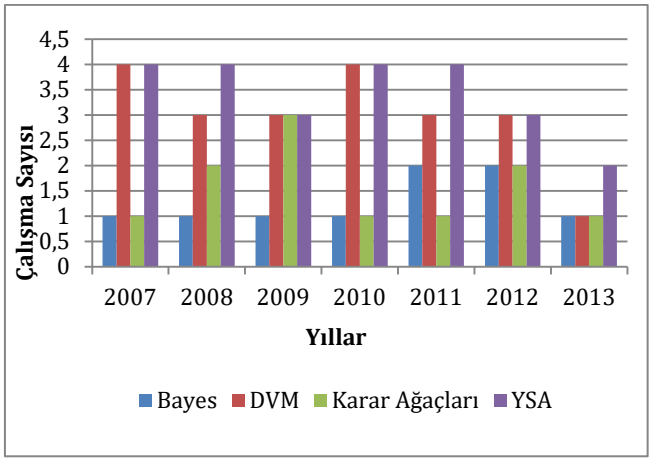
2.1.1. Sınıflandırıcılar

2007-2013 yılları arasında yapılan çalışmalar incelendiğinde bayes sınıflama, destek vektör makinesi, karar ağaçları ve yapay sinir ağlarının saldırı tespit sistemlerinde en sık kullanılan yöntemler olduğu görülmüştür. Çizelge 2.1'de görüldüğü gibi bu sınıflandırıcılar içerisinde en çok tercih edilen yöntem YSA iken DVM en çok tercih edilen ikinci yöntem olmuştur.

Çizelge 2.1. İncelenmek üzere seçilen çalışmalar ve sınıflandırıcı türleri

BAYES	DVM	KARAR AĞAÇLARI	YSA
9	**21**	**11**	**24**
F.Jemili vd. [2], Dewan Md. Farid ve Mohammad Zahidur Rahman [3], C. Xiang vd. [14], Farah Jemili vd. [15], Z. Muda vd. [16], W. Fan vd. [17], Hesham Altwaijry ve Saeed Algarny [18], Saurabh Mukherjee ve Neelam Sharma [19], Levent Koç vd. [20].	Yongli Zhang vd. [4], Jun Wang vd. [5], Qi Mu vd. [6] , Qiao Pei-li ve Chen Shi-feng [21], T. Shon ve J. Moon [22], Hua Zhou vd. [23], Yuan-Cheng Li ve Zhong-Qıang Wang [24], H. Li ve J. Wan [25], X. Ding vd. [26], Yuancheng Li vd. [27], Jing Ma vd. [28], Zhenguo Chen ve Guanghua Zhang [29], Hongle Du vd. [30], Huaping Liu vd. [31], Guan Xiaoqing vd. [32], Shi-Jinn Horng vd. [33], Preecha Somwang ve Woraphon Lilakiatsakun vd. [34], Guanghui Song vd. [35], Liu Ning ve Zhao Jianhua [36], X. Yang ve Z. Yilai [37], A.M.Chandrasekhar ve K.Raghuveer [38].	M. Bahrololum vd. [7], A. Alazab vd. [8], Vikas Sharma ve Aditi Nema [9], S. Peddabachigari vd. [39], Joong-Hee Leet vd. [40], Shina Sheen ve R Rajesh [41], Wu vd. [42], Dai Hong ve Li Haibo [43], Yongjin Liu vd. [44], P. Sangkatsanee vd. [45], Manish Kumar vd. [46].	G. Liu vd. [10], X. Gong ve X. Guan [11], Hao-Ran Deng ve Yun-Hong Wang [47], Ling Yu vd. [48], P. G. Kumar ve D. Devaraj [49], R. Beghdad [50], S.T.Powers ve Jun He [51], Tie-Jun Zhou ve Li Yang [52], H. Karimi vd. [53], X. Han [54], X. Tong vd. [55], Poojitha G. vd. [56], G. Wang vd. [57], Dong-Xue Xia vd. [58], W. Huang ve L. Ju vd. [59], M. Govindarajan ve R.M. Chandrasekaran [60], L. Xiangmei ve Q. Zhi [61], B. Zhang [62], S.Devaraju ve S.Ramakrishnan [63], D. Ippoliti ve X. Zhou [64], X. Jianga vd. [65], Nidhi Srivastav ve Rama Krishna Challa [66], Liu Ning [67], B. Zhang ve Xuesong Jin Saeed [68].

Şekil 2.1'de incelenen çalışmaların yıllar bazında dağılımı görülmektedir. Burada 2007, 2009, 2010 ve 2012 yılarında DVM ve YSA kullanım oranları eşit görülse de YSA'nın 2007-2013 yıları arasında en çok kullanılan sınıflandırıcı olduğu açıkça görülmektedir.



Şekil 2.1. İncelenen çalışmalarda kullanılan sınıflandırıcıların yıllara göre dağılımı

2.1.2. Veriseti

2007-2013 yılları arasında yapılan çalışmalar incelendiğinde Çizelge 2.2'de görüldüğü gibi DARPA98 ve UNM, en az tercih edilen veriseti iken, KDD CUP99 veriseti en sık kullanılan veriseti olmuştur. Çok az çalışmada çizelge 2.3'de geçen veriseti dışında kendi verisetini kullanan çalışma vardır. Bunun nedeni bilgisayar ağ ve sistemlerinden veri toplamanın yüksek maliyetidir. Açık (public) verisetlerinin kullanılmasında diğer bir neden ise önerilen yaklaşımların, önceki çalışmalarla kıyaslanmak istenmesidir. Sonuç olarak public verisetleri makine öğrenme teknikleri ile STS çalışmalarında sıklıkla tercih edilen standart verisetleri olarak kabul gördüğü anlaşılmaktadır.

Çizelge 2.2. İncelenen çalışmalarda kullanılan verisetlerinin yıllara göre dağılımı

	07	08	09	10	11	12	13	Toplam
KDD CUP99	9	9	9	10	9	10	5	61
DARPA98	1							1
DARPA99	1	1						2
UNM					1			1

2.1.3. Performans karşılaştırması

Çizelge 2.3'de 2007-2013 yılları arasında yapılan çalışmalarda kullanılan verisetleri ve elde edilen genel sınıflandırma başarıları görülmektedir. Çizelge 2.3'de açıkça görüldüğü gibi KDD CUP99 veriseti kullanılarak gerçekleştirilen STS'de en yüksek başarı YSA ile

%99,59 elde edilirken, en düşük başarı %93,72 ile Bayes sınıflandırıcı ile elde edilmiştir. DARPA 99 veriseti kullanılarak gerçekleştirilen STS'lerde en yüksek başarı Bayes ile %98,03 elde edilirken, en düşük başarı %87,74 ile DVM 'le elde edilmiştir.

Çizelge 2.3. İncelenen çalışmalarda kullanılan veriseti ve sınıflandırma başarısı

	KDDCUP99	**DARPA99**	**DARPA98**	**UNM**
Bayes	%93,72 [20]	%98,03 [15]	-	-
DVM	%99,44 [28]	%87,74 [22]	-	-
Karar Ağaçları	%99,33 [45]	-	%77,6 [40]	-
YSA	%99,59 [53]	-	-	%99,03 [60]

Çizelge 2.4'de incelenen çalışmalarda saldırı tiplerine göre sınıflandırıcıların başarısı görülmektedir. DOS tipi saldırılarda sınıflandırma başarıları birbirine yakın olmakla birlikte en yüksek başarı YSA ve DVM ile %100 elde edilmiştir. En düşük sınıflandırma başarı ise %99,62 ile Bayes sınıflandırıcı ile elde edilmiştir. Bilgi tarama saldırılarında, en yüksek sınıflandırma başarı oranı YSA, DVM ve Bayes sınıflandırıcı ile %100 elde edilmiştir. En düşük sınıflandırıcı ise %99,66 ile karar ağaçları olmuştur. R2L tipi saldırılarda, en yüksek sınıflandırma başarı oranı YSA ile %100 elde edilmiştir. En düşük sınıflandırıcı ise %99,35 Bayes olmuştur. U2R tipi saldırılarda, en yüksek sınıflandırma başarı oranı DVM ile %100 elde edilmiştir. En düşük sınıflandırıcı ise %92,5 karar ağacı olmuştur.

Çizelge 2.4. İncelenen çalışmalarda saldırı tiplerine göre sınıflandırıcıların başarısı

	DOS	**Bilgi tarama**	**R2L**	**U2R**
Bayes	%99,62 [15]	%100 [15]	%99,35 [3]	%99,47 [3]
DVM	%100 [6]	%100 [5]	%99,42 [4]	%100 [6]
Karar Ağaçları	%99,98 [9]	%99,66 [7]	%99,70 [8]	%92,5 [8]
YSA	%100 [10]	%100 [10]	%99,88 [11]	%99,85 [11]

Yukarıda bahsedilen çalışmalar makine öğrenmesi ile saldırı tespit sistemlerinin başarısını artırmak için geliştirilmiş önemli çalışmalardır ancak STS'lerde saldırı türlerine göre hangi makine öğrenmesi tekniği daha başarılıdır sorusuna cevap vermemektedir. Bizim

çalışmamızda ise gerçekleştirilen deneylerle saldırı tespit sistemlerinde en sık kullanılan makine öğrenmesi tekniklerinden Bayes ağları, destek vektör makinesi, karar ağaçları, yapay sinir ağları ve k en yakın komşu algoritmasının performans analizi yapılmış ve saldırı türlerine göre doğruluk, seçicilik, duyarlılık, kesinlik, F-Ölçütü değerleri incelenerek en başarılı sınıflandırıcılar belirlenmiştir. Bu çalışma ile gelecekte yapılacak makine öğrenme teknikleri ile saldırı tespiti çalışmalarına bir bakış açısı kazandırılması amaçlanmıştır.

3. BİLGİ VE BİLGİ GÜVENLİĞİ

Bilgi, birbiriyle henüz bağlantısı kurulmamış olan verilerin işlenerek belirli bir anlam ifade edecek şekilde düzenlenmiş halidir [69]. Günümüz dünyasında bilgi sahibi olduğu insanlara değer katan ve onları diğer canlılar karşısında avantajlı duruma getiren çok değerli bir varlıktır. Bilginin bu denli önemli bir katma değer olması ve sahibine avantajlar sağlaması, bilgiye erişmek ve sahip olmak için insanları birbirleriyle yarışır hale getirmiştir. Bilgiye sahip olmak için sürekli devam eden bu yarış nedeniyle bilgiyi korumak ve güvenliğini sağlamak çok büyük önem taşımaktadır. Bilgi güvenliği konusuna girmeden önce birkaç temel kavramdan bahsetmek önem arz etmektedir.

Risk: Eksik ya da yanlış yazılım tasarımı veya donanımın arızalanmasına nedeniyle çalışan sistemlerin ve bilgilerin istenmedik ve beklenmeyen bütünlük ihlaline maruz kalması [70].

Tehdit (Threat): Bir sisteme, sistemin güvenlik açıklarından faydalanarak, yetkisiz kullanıcıların erişip bilgi çalması ve sistemi güvenilmez ya da kullanılmaz yapabilmesi ihtimalidir [70].

Sistemlere ve kaynaklara erişim durumuna göre tehditler Anderson tarafından geldikleri yere göre Çizelge 3.1'de gösterildiği gibi 3 gruba ayrılarak incelenmiştir.

Çizelge 3.1. Sistemlere ve kaynaklara erişim durumuna göre tehdit sınıfları [71]

		Veriye/Kaynağa Erişim Durumu	
		Yetkili	Yetkisiz
Sisteme Erişim Durumu	Yetkili	(A) Dış Kaynaklı Tehdit	
	Yetkisiz	(B) İç Kaynaklı Tehdit	(C) Kötüye Kullanım

(A) grubundaki tehditler, sistemlere ve kaynaklara erişim yetkisi olmayan kişiler tarafından sisteme ve kaynaklara erişim yapabilmek için yapılan hareket ve durumları içerir. (B) grubundaki tehditler, sisteme kısıtlı erişim yetkisi olup, yetkisiz oldukları kaynaklara erişmek isteyenler tarafından gerçekleştirilen hareket ve durumlardır. (C) grubu tehditler

ise sisteme ve kaynaklara erişim yetkisi varken, bu yetkisini, yetkili olmayan başka kişilere kullandıran kullanıcılardan kaynaklanan tehdit unsurudur.

Güvenlik Açığı (Vulnerability): Bir sistemin işlem döngüsünde, yazılımda ya da donanımda ortaya çıkan, sistemin delinmesine ya da bilgilerin açığa çıkmasına neden olabilen bilinen ya da şüphelenilen bir kusurdur [70].

Saldırı (Attack): Bir tehdidi gerçekleştirmek için bir planın yürütümü ya da özel bir formülasyonu [70].

Saldırgan: Bilerek ve isteyerek, bilgisayar sistemlerinin güvenlik açıklarından faydalanarak, sistem üzerindeki verilere, hizmetlere veya sistemin işleyişine zarar verebilecek hareket veya durumları oluşturan kişidir.

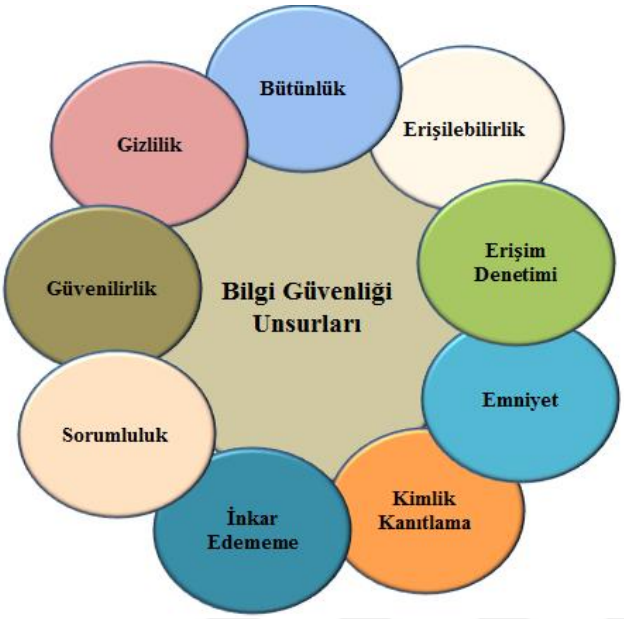
Penetrasyon (Penetration): Başarılı bir saldırı olup, bir sistemin kontrolünü ele geçirme, programlara ve dosyalara yetkisiz erişim hakkı elde etme yeteneğidir [70].

Yetkisiz Kullanıcı: Bilgisayar sistemlerine izinsiz giriş yapan kullanıcıdır.

3.1. Bilgi Güvenliği Unsurları

Bilginin güvenliğini sağlayabilmek, bilginin, gizliliğini (confidentiality), bütünlüğünü (integrity) ve erişilebilirliğini (availability) her türlü tehdide karşı koruyabilmekle mümkündür. Bilgi güvenliği temel olarak bu üç unsuru korumayı hedefler. Bunların dışında kimlik kanıtlama (authentication), inkâr edememe (non-repudiation), sorumluluk (accountability), güvenilirlik (reliability), erişim denetimi (access control) ve emniyet (safety) kavramları da bilgi güvenliğini destekleyen yan unsurlardır. TSE 17799 "Bilgi Güvenliği Yönetim Standardı" belgesine göre bu kavramlara bakacak olursak gizlilik, bilginin sahibinden başka tüm yetkisiz kişilerin erişimine ve kullanımına kapatılmasıdır. Bütünlük, bilginin yetkisiz kişilerce silinmesi, değiştirilmesi ya da ne şekilde olursa olsun her türlü tahrip edilme tehlikesine karşı içeriğinin muhafaza edilmesidir. Erişilebilirlik, bilginin sadece yetkili olduğu kişiler tarafından, yetkili olduğu zaman diliminde, her ihtiyaç duyulduğu anda, ulaşılabilir ve kullanılabilir halde bulunmasıdır. Kimlik kanıtlama, Bir sistemde, geçerli kullanıcı ve işlemlerin, hangi sistem kaynaklarına erişebileceğinin belirlenip, sisteme erişim olduğunda, geçerli kullanıcı ve işlemlerin tanınıp

doğrulanabilmesidir. İnkâr edememe, bir bilgiyi alan tarafın aldığını, gönderen tarafın ise gönderdiğini inkâr edememesini sağlamaktır. Sorumluluk, bir eylemin yapılmasından, kimin ve ya neyin sorumlu olduğunu belirleme yeteneğidir.



Şekil 3.1. Bilgi güvenliği unsurları

Güvenilirlik, bir sistemin tasarım gereksinimlerine uygun olarak tam ve düzgün bir şekilde çalışabilmesidir. Erişim denetimi, bir sisteme, kaynağa ya da bilgiye erişmek için yetkili kişilerin belirlenip izinlerin verilmesi ya da yetkilerin iptal edilmesidir. Emniyet, bir sisteme ya da bilgiye yönelik olarak her türlü riski ve tehlikeyi göz önüne alıp gerekli tedbirlerin alınmasıdır.

3.2. Güvenlik Süreçleri

Bilgi güvenliği çerçevesinde, kurulacak olan güvenlik sisteminin altyapısı belirlenirken, korunacak olan bilgilerin ve güvenlik politikalarının, doğru ve eksiksiz bir şekilde planlanması gerekmektedir. Güvenlik yönetimi, kurulmuş ya da kurulacak olan bilgisayar sistemlerinin, güvenlik açıklarını kapatıp, sisteme yönelik olarak gelebilecek tehditleri önceden öngörüp tedbir alarak güvenlik riskini minimize etmeyi amaçlamaktadır. Risk yönetimi yapılırken korunacak olan sistemlerin, bilgilerin ya da varlıkların belirlenip; bu varlıkların kişi ya da kurumlar için ne derece değerli olduğu analiz edilmeli ve bu

varlıklara yönelik olarak gelebilecek her türlü tehdidin boyutu saptanarak yapılacaklar belirlenmelidir.

Risk yönetim planlaması sonucunda korunacak olan bilginin değeri ve tehdit durumuna bağlı olarak, kurulacak ve işletilecek güvenlik sistemlerinin maliyeti de dikkate alınmalıdır. Hiçbir sistemin %100 güvenli olmayacağı ilkesi düşünülmekle birlikte, bilgi güvenliğinin ideal yapılandırılması üç süreç ile gerçekleştirilir. Bunlar önleme (prevention), saptama (detection) ve karşılık vermedir (response ya da reaction). Bu kavramlara bakacak olursak.

3.2.1. Önleme

Bilgisayar sistemlerine yönelik, yapılabilecek her türlü tehdit ve saldırıya karşı, sistemi güvenli hale getirmek için alınmış olan tedbirler önleme faaliyeti olarak adlandırılır. Örneğin kişisel bilgisayar güvenliği ile ilgili olarak, virüs tarama programlarının kurulu olması, kurulu programların ve işletim sistemlerinin en güncel hali ile çalıştırılması, bilgisayarın açılışından önce şifre girilmesi, şifrelerin gizli tutulup belirli aralıklarla değiştirilmesi ve internet üzerinden elde edilen dosya veya verilerin kullanılmadan önce virüs tarama programlarıyla taranması önleme faaliyetleridir. Ancak alınabilecek bunca önleme karşın tam bir güvenlik söz konusu değildir. Bu nedenle saptama ve karşı koyma tedbirlerine de başvurulması gerekmektedir.

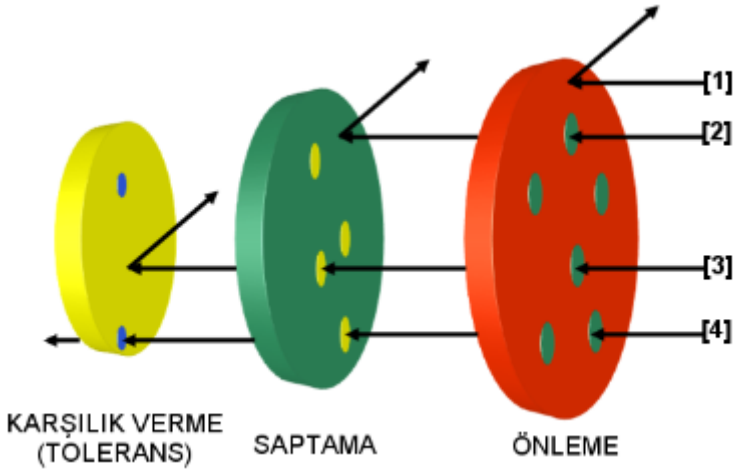
3.2.2. Saptama

Saptamada amaç sistemin bütün durumunu analiz etmek ve sistemdeki her türlü faaliyeti gözlemleyip kayıt altına almak ve bu kayıtların saklanmasıdır. Sistemdeki faaliyetlerin kayıt altına alınmasıyla hem delil elde edilmiş olup hem de bu kayıtlar incelenerek ilerisi için benzer saldırıların önüne geçmek amacıyla tedbir alınır. Güvenlik duvarları, saldırı tespit sistemleri, virüs programları, ağ trafiği izleyiciler saptama aşamasında kullanılan en temel araçlardır.

3.2.3. Karşı koyma

Karşılık verme aşamasında, önleme süreciyle engellenemeyen ve saptama süreciyle tespit edilmiş olan saldırılara karşı en hızlı şekilde cevap verecek şekilde eylemler geliştirilip

uygulamaya konulur. Saptama sürecinde kullanılan saldırı tespit sistemleri, tespit ettiği duruma cevap verecek bir sistemin olması ile işlevlik kazanır. Aksi halde saldırıyı engelleyemedikten sonra tespit etmenin bir önemi kalmayacaktır. Bu nedenle karşılık verme süreci bilgi güvenliği süreçlerinin önemli bir adımıdır [69].



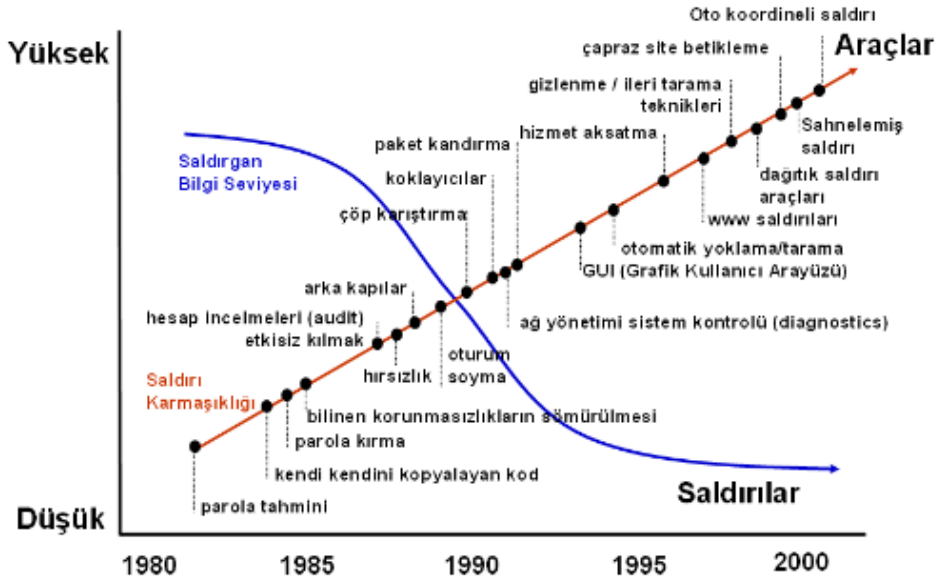
Şekil 3.2. Bilgi güvenliği unsurları ve saldırılara tepkileri [69]

Tüm bu süreçlerden sonra Şekil 3.2'de [4] numaralı saldırıda olduğu gibi herhangi bir saldırı engellenemese bile, sistemin yeniden ayağa kaldırılması için, saldırılar hakkında bilgi toplama için, saldırıya sebep olan güvenlik açıklarını tespit etmek için ve saldırganların yakalanması için önleme, saptama ve karşılık verme süreçleri önem arz etmektedir.

4. SALDIRI VE SALDIRI TİPLERİ

4.1. Saldırı Tanımı

Bilgisayar sistemlerini ele geçirip yetkili olmadığı bilgilere erişmek, sistemin işleyişini aksatmak ya da sistemi hizmet veremez hale getirmek amacıyla kötü niyetli kişiler tarafından gerçekleştirilen tüm faaliyetler saldırı olarak tanımlanır. Saldırganlar, amaçlarına ulaşmak için, her geçen gün daha da çeşitlilik gösteren birçok yöntem kullanırlar. Genel olarak bir saldırının anatomisi şu şekildedir. Saldırganlar, bir sistemi ele geçirmek amacıyla öncelikle hedef sistem hakkında hedef sistemin web sayfası, whois kayıtları, arama motorları, sosyal ağlar gibi kaynaklardan bilgi toplarlar. Hedef sisteme ait bilgiler, hedef sistemin ip aralığı, çalışmakta olan işletim sistemleri ve aktif servisler v.d. bilgilerdir. 2. aşamada hedef ağdaki aktif sistemlerin, açık portların, servislerin, belirlenmesi amacıyla sistemde tarama yapılır. 3. aşamada saldırgan, topladığı bilgileri kullanarak hedef sisteme girmek amacıyla çalışır. 4. aşamada sisteme giren saldırgan, sistem içinde varlığını sürdürmek amacıyla rootkit, Truva atı gibi zararlı yazılımlar kurarak sistemde kalıcı olmaya çalışır. 5. ve son aşamada ise saldırgan sisteme erişim izlerini yok ederek delil bırakmamaya çalışır.



Şekil 4.1. Yıllara göre saldırıların karmaşıklığı ve saldırgan teknik bilgisi [69]

Şekil 4.1'de de görüldüğü gibi gelişen teknoloji ile beraber zamanla saldırılar farklılık göstermeye başlamıştır. Teknolojinin gelişmesi saldırganlara daha az teknik bilgi ile karmaşık araçlar kullanarak daha başarılı saldırılar yapabilme imkânı vermiştir. Bu durum, saldırgan sayısını artırırken, sistem yöneticilerinin saldırganlarla baş edebilmesinin gittikçe zorlaşmasına neden olmaktadır.

4.2. Saldırı Sebepleri

Saldırganların amacı genel olarak maddi menfaat sağlama, politik, ticari ve ekonomik yönden rakiplerine karşı avantaj sağlama, sahip olamadığı ek kaynaklara sahip olma isteği, kişisel öfke ya da intikam duygusu, kurumsal veya ulusal çıkar elde etme isteği, merak ve ya öğrenme isteği gibi çok çeşitli konuları kapsamaktadır.

4.3. Saldırı Türleri

Günümüzde bilgisayar sistemlerinin her alanda yaygınlaşması ve sistemlerin birbirlerine bilgisayar ağlarıyla bağlamasıyla beraber, sistemlere yapılan saldırılar da günden güne artmıştır. Teknolojinin gelişmesi, sistemlere yapılan saldırı türlerinin artmasına ve saldırıların çeşitlilik göstermesine neden olmuştur. Dünya genelindeki saldırı davranışları göz önüne alındığında, günümüzde, ağ üzerinden yapılan saldırılar en sık karşılaşılan problemlerdir.

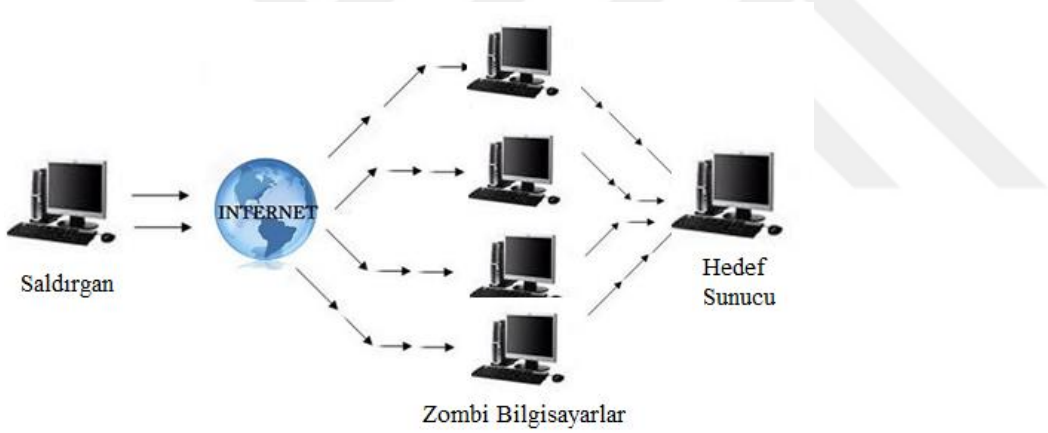
Ağ üzerinden gerçekleştirilen saldırılar temel olarak 4 grupta incelenir.

4.3.1. Hizmet engelleme saldırıları (Denial of Service-DOS)

Bir sisteme, TCP/IP protokolünün yapısından kaynaklanan açıklardan faydalanılarak, sistemin tüm kaynaklarını tüketip, hizmet veremez hale getirmek amacıyla, arka arkaya yapılan düzenli saldırılardır. Hizmet aksatma saldırılarında, bir veya birden fazla noktadan, hedefteki bilgi sistemleri üzerine, gereğinden fazla yükler bindirilerek, sistemler üzerindeki asli hizmetlerin aksaması ve bu aksama anında zayıflayan sistemlere sızabilmek amaçlanır. Disk alanlarının doldurulması, işlemci tüketimi, yerel alan ağlarındaki merkezi anahtarlara trafik yüklemek, internet yönlendiricilerine gereksiz trafik yükleyerek yetkisiz erişim elde etmek, hizmet aksatma saldırılarına örnek olarak verilebilir. DOS saldırıları ağlar için en tehlikeli saldırılardan bir tanesidir. Çünkü ağ trafiğine bakarak DOS saldırılarının

anlaşılması ve normal ağ trafiğinden ayırt edilmesi zordur. Bu saldırıların tespit edilebilmesi amacıyla anormallik tespiti tabanlı saldırı tespit sistemleri kullanılabilir [72].

DOS saldırıları, sistem erişimini kısıtlayarak, yetkili kullanıcıların hizmet almasını engellemek amacıyla çeşitlilik gösterir. Sistem zayıflıklarından faydalanarak yapılan, ağ tabanlı DOS saldırıları, savunmasızlık (vulnerability) ve taşkın (flooding) olmak üzere iki çeşittir [73]. Savunmasızlık yönteminde, kurban olarak seçilen bir sistemde çalışan uygulama programlarının ya da mevcut bazı ağ protokollerinin güvenlik açıklarından faydalanılarak, sistem çalışamaz hale getirilir. Bu saldırı türü, aşırı bellek tüketimi, fazladan CPU işleme, sistemin yeniden başlatılması veya genel sistem yavaşlaması gibi sistemin çalışmasını aksatan problemlere neden olur. Taşkın yönteminde ise hedef olarak seçilen kurbana, kaldırabileceğinden fazla miktarda iş yüklenerek, sistem hizmet veremez hale getirilmeye çalışılır.



Şekil 4.2. DOS saldırısı

DOS saldırıları gerçekleştirilirken, Şekil 2.4'de olduğu gibi, saldırgan tarafından, daha önce ele geçirilmiş olan zombi bilgisayarlar kullanılır. Saldırgan internet üzerinden kontrol ettiği zombi bilgisayarları kullanarak hedef sunucuya çok sayıda erişim isteği gönderip, sunucuyu hizmet vereme hale getirir.

DOS saldırılarının, sistem açıklarını kullanma şekli ve yapılış yöntemleri incelendiğinde çeşitli tipleri ve teknikleri mevcuttur. Bilinen DOS saldırısı çeşitleri aşağıdaki gibidir.

TCP/SYN (Senkronize) taşma saldırısı

Saldırgan zombi olarak seçtiği bilgisayarları kullanarak, hedefteki kurban sunucuya sahte SYN paketleri gönderir ve sistem kaynaklarını doldurur. Sistem kaynakları dolan sunucu, hizmet veremez ve istemciler sunucu ile bağlantı kuramaz [74].

Ping taşma saldırısı (Ping flood attack)

Bu teknikte, saldırgan, hedef sunucuya büyük boyutlu ICMP paketleri yollar ve sunucunun bant genişliğini doldurarak, iletişimi aksatır. Ölümcül ping saldırısı, bu teknikte bir saldırıdır. PING uygulamasını kullanarak IP tespitinde izin verilen maksimum 65535 bayt'lık veri limitini aşan IP paketleri gönderilir. Gereğinden fazla miktarda veri gönderildiği için sunucu durur, çöker ya da yeniden başlatılmak zorunda kalır [74].

Land saldırısı

Bu teknikte saldırgan, hedefteki sunucunun IP adresini elde edip kaynak IP adresi olarak göstererek, sunucuyu SYN paketleri ile istila eder. Bu durumda sunucu bilgisayar sürekli kendi kendine yanıt vermeye çalıştığı için hizmet veremez hale gelir.

Gözyaşı saldırısı (Teardrop attack)

Gözyaşı saldırısı, IP paketlerinin yeniden birleştirilmesi sırasında oluşan güvenlik açığından faydalanır. Veri, ağlar üzerinde iletilirken genel olarak küçük parçalara ayrılır. Her bir parça, orijinal bir paket gibi görünür. Ancak, bunların haricinde bir de ofset alan bulunmaktadır. Teardrop programı paket parçalarından oluşan bir küme meydana getirir. Bu parçaların genellikle birbirleriyle eşleşen ofset alanları bulunmaktadır. Söz konusu parçalar hedef sistem bünyesinde nihayet bir araya getirildiğinde sistemler bozulabilir, durabilir veya yeniden başlayabilirler [2].

Kaba kuvvet saldırısı (Brute-Force attack)

Kaba kuvvet saldırısında, hedef bilgisayar ağı, gereksiz verilerle işgal edilir. Bu atağı kullanan saldırgan, paketlerin hedef adresini ağın yayın (broadcast) adresi olarak seçer. Bu durumda yönlendirici (router) ağdaki bütün host'lara ICMP echo isteği gönderecektir. Eğer

ağ bünyesinde çok sayıda host varsa, bu çok sayıda ICMP echo istek paketi oluşturacaktır. Broadcast seli mevcut bant genişliğini tüketecektir. Bu durumda haberleşme imkânsız hale gelecektir [75].

4.3.2. Bilgi tarama saldırıları (Probe)

Bilgi tarama saldırıları, bir sunucunun ya da herhangi bir makinenin geçerli ip adreslerini, ağdaki bilgisayar sayısını, bilgisayardaki kullanıcı sayısını ve kullanıcı bilgilerini, aktif giriş kapılarını (port) veya işletim sistemini öğrenmek için yapılırlar [72].

Bilinen probe saldırılarına örnek olarak ipsweep ve portsweep verilebilir. Ipsweep saldırısı, belirli bir portu sürekli tarama saldırısıdır. Portsweep ise bir sunucu üzerindeki hizmetleri bulmak için tüm portları sürekli tarar.

4.3.3. Kullanıcı hesabının yönetici hesabına yükseltilmesi (user to root-U2R)

Kullanıcı hesabının yönetici hesabına yükseltilmesi saldırısı; sisteme erişim yetkisi olan fakat yönetici yetkilerine sahip olmayan bir kullanıcının yönetici haklarını elde etmesidir. Genellikle sistem açıklarını kullanarak gerçekleştirilir [72]. U2R saldırılarına eject ve sqlattack saldırıları örnek verilebilir.

Eject saldırısı, solaris üzerinde eject programı ile tampon taşmasına (buffer flow) yol açıp, yönetici haklarına sahip olunmasıdır. Sqlattack saldırısı ise sql veritabanı kurulu linux makinalarda, sunucuya bağlanan kullanıcının, belirli komutlarla yönetici hakları ile komut satırı elde etmesidir.

4.3.4. Yönetici hesabı ile yerel oturum açma (remote to local-R2L)

Kullanıcı yetkisine sahip olunmadığı halde, hedef ağdaki bilgisayara bazı paketler gönderilerek misafir ya da başka bir kullanıcı olarak bilgisayara erişim yetkisi kazanılmasıdır [72]. R2L saldırılarına Sshtrojan ve guest saldırıları örnek verilebilir.

Sshtrojan saldırısı, unix işletim sistemi üzerinde çalışan bir trojan saldırısıdır. Guest saldırısı ise tahmin etmesi kolay şifreleri bulup sisteme girmeyi amaçlar.

5. SALDIRI TESPİT SİSTEMLERİ (STS)

Saldırı tespit sistemi fikri ve ilk STS konsepti, 1980 yılında "bilgisayar güvenlik tehditleri izleme ve gözetim" başlıklı makalede, Anderson tarafından önerilmiştir [71]. James Anderson, kullanıcı davranışlarını anlamada ve kötüye kullanımı izlemede, denetim izlerinin (log kayıtları) hayati öneme sahip bilgiler içerebileceği fikrini ortaya atmıştır. Bu fikir sayesinde denetim verilerinin öneminin anlaşılması, hemen hemen her işletim sisteminin alt sistemlerinin denetlenmesinde yeni gelişmelere yol açmıştır. Böylece Anderson'un hipotezi gelecekteki saldırı tespit sistemi tasarımı ve geliştirilmesi çalışmalarına temel oluşturmuştur [76].

5.1. Saldırı Tespit Sisteminin Tanımı ve Görevi

Saldırı tespit sistemleri (STS), güvenlik duvarı ve antivirüs yazılımları ile beraber kullanılabilen, bilgisayar sistemlerini ve ağ kaynaklarını izleyerek bu sistemlerin algılamakta güçlük çektiği saldırıları da analiz edip algılayabilen yazılımsal veya donanımsal tabanlı sistemlerdir.

Diğer bir tanımda;

Saldırı tespit sistemi, bilgisayar sistemi ya da bilgisayar ağında meydana gelen olaylarla ilgili bilgileri toplayıp analiz eden, güvenlik politikası ihlallerini tespit eden ve saldırı izlerini algılayan güvenlik mekanizmasıdır.

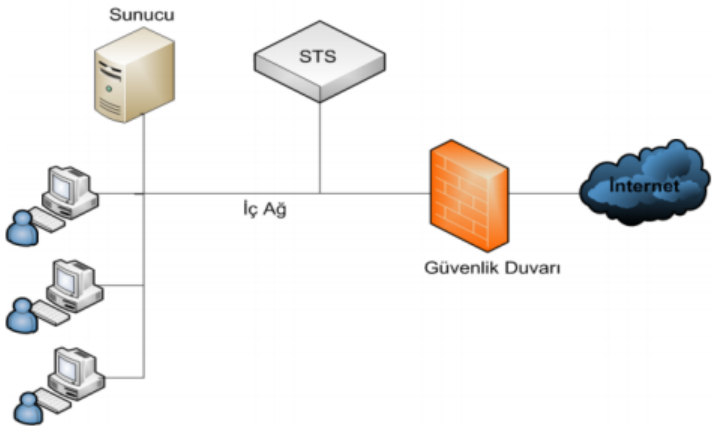
Saldırı tespit sisteminin, bilgisayar sistemleri ya da ağlarındaki ana rolü, ağ saldırılarıyla uğraşmak ve bilgisayar sistemlerine yardım etmektir [76].

Saldırı tespit sisteminin görevleri aşağıdaki gibidir.

- Hem kullanıcı hem de sistem faaliyetlerini izleme ve analiz etme
- Sistem yapılandırılmasını ve güvenlik açıklarını analiz etme
- Sistemi ve dosya bütünlüğünü değerlendirme
- Saldırıların tipik örüntülerini tanıyabilme
- Anormal olayların izlerini analiz edebilme

- Kullanıcıların güvenlik politikası ihlallerini izleme

Saldırı tespit sisteminin amacı, saldırılar ile başa çıkmak için bilgisayar sistemlerine yardımcı olmaktır. STS'ler, bilgisayar sistemleri ve ağları içinde birkaç farklı kaynaktan bilgi toplar ve bu bilgileri saldırı ya da zayıflık olup olmadığını ayırt etmek için daha önce elde ettiği örüntülerle karşılaştırır. Şekil 5.1'de güvenlik önlemi olarak bir saldırı tespit sistemi görülmektedir.



Şekil 5.1. Saldırı tespit sistemi

STS'ler saldırı önleme sistemi değildirler. Bilgisayar sistemlerindeki normal ya da anormal davranışları ayırt edip sistem yöneticisine haber veren alarm mekanizmalarıdır. Güvenlik sistemindeki son savunma mekanizması gibi çalışırlar.

5.2. Saldırı Tespit Sisteminin Tarihsel Gelişimi

Saldırı tespit sistemlerinin amacı, ağdaki anormal ve kötü amaçlı davranışları algılamak için ağa ait tüm bileşenleri izlemektir. STS kavramı yaklaşık 20 yıldan beri var olmasına karşın, son zamanlarda popülerliği artmış ve tüm bilgi güvenliği altyapılarına dâhil edilmiştir. İlk STS fikri, 1980 yılında "bilgisayar güvenlik tehditleri izleme ve gözetim" başlıklı makalede, Anderson tarafından önerilmiştir. Anderson'un çalışması gelecekteki STS tasarımlarına ve geliştirilmesine temel oluşturmuştur.

1983 yılında, SRI International firması ve Dr. Dorothy Denning yeni bir STS geliştirilmesi için devlet destekli olarak çalışmaya başladı. Çalışmanın amacı, devlete ait mainframe

bilgisayarlardaki denetim kayıtlarını analiz etmek ve kullanıcıların aktivitelerine göre kullanıcı profili oluşturmaktır. Bir yıl sonra, Dr. Denning'in yardımıyla ilk STS modeli geliştirilmiştir.

1984 yılında, SRI firması bir izleme aracı geliştirmiş ve ARPANET'deki kullanıcıların kimlik doğrulama (authentication) bilgilerini içeren denetim kayıtlarını analiz etmiştir. Daha sonra, Dr. Denning, araştırmalarını ve SRI firmasında yaptığı çalışmaları yayınlamış ve ticari saldırı tespit sistemlerinin geliştirilmesi için gereken bilgiyi ortaya koymuştur.

1989 yılında, ticari bir şirket olan Haystack Labs firması tarafından son nesil teknoloji ürünü olan Stalker geliştirilmiştir. Stalker, manuel ve otomatik denetim verileri sorgulamak için, güçlü arama yeteneklerine sahip host tabanlı, örüntü eşleme sistemidir.

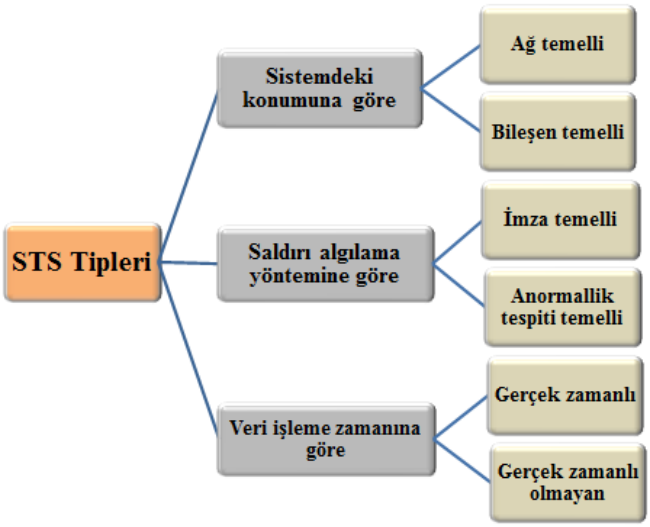
Saldırı tespit teknolojisindeki ticari gelişmeler 1990 yıllarında başlamıştır. Haystack Labs, host tabanlı ürünü Stalker ile STS araçlarının ilk ticari satıcısıdır. Daha sonra SAIC (The Science Applications International Corporation) tarafından "bilgisayar kötüye kullanım algılama sistemi" olarak adlandırılan host tabanlı STS geliştirilmiştir. Bu sistemle eş zamanlı olarak Amerika hava kuvvetleri kriptoloji destek merkezi tarafından, Amerika hava kuvvetleri iletişim ağındaki trafiği görüntülemek amacıyla otomatik güvenlik ölçüm sistemi (Automated Security Measurement System-ASIM) geliştirilmiştir.

STS ürünlerinin ticarileşmesi 1997 yıllarının ortalarında önemli aşama kaydetmiştir. Internet security systems (ISS) firması, bu yıllarda Real Secure olarak adlandırdığı STS ürünü ile marketin lideri olmuştur.

Günümüzde ise STS'ler en çok kullanılan güvenlik araçlarından biridir [77].

5.3. Saldırı Tespit Yaklaşımları

Saldırı tespit sistemleri, güvenlik duvarı kullanmak, antivirüs programı yüklemek gibi alınabilecek tüm güvenlik önlemlerine rağmen yapılan saldırıları tespit etmek ve sistem yöneticisini uyarmak üzere geliştirişmiş alarm mekanizmalarıdır. Saldırı tespit sistemleri, genel olarak, saldırıları algılama yöntemine göre, sistemde bulunduğu konuma göre ve veri işleme zamanına göre 3 grupta incelenir.



Şekil 5.2. STS tipleri

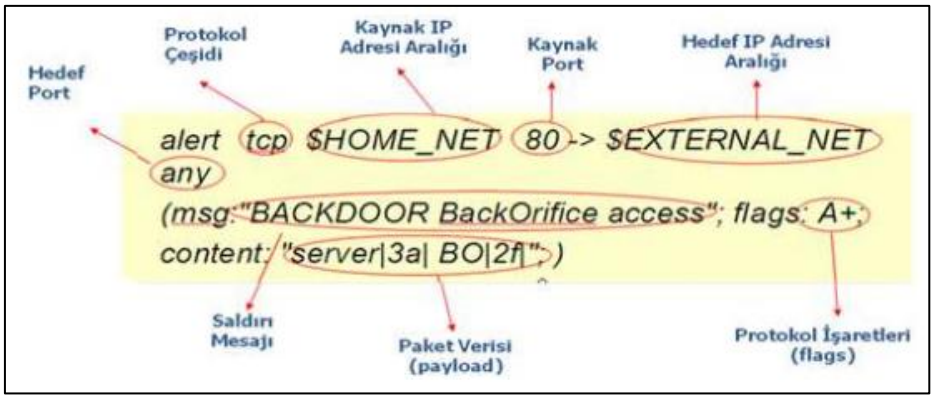
5.3.1. Saldırı algılama yöntemine göre STS

Saldırı algılama yöntemine göre STS'ler iki grupta incelenir. Bunlar imza temelli ve anormallik tespiti temelli yaklaşımlardır.

İmza temelli STS

İmza temelli STS, yetkisiz erişimlerin ve bilinen saldırıların örüntülerini aramak için tasarlanmıştır. Kötüye kullanım temelli STS ya da bilgi temelli STS olarak da adlandırılır. Bilinen saldırıların imza kaydını tutar ve yeni gelen bir isteği kayıtlarını kontrol ederek "saldırı" ya da "normal davranış" olarak sınıflandırır. Bu imzalar, sistemin şimdiki zamana kadar tespit etmiş olduğu saldırıların, karakteristik özellikleridir. Ağ paketlerinin protokol çeşidi, protokol işaretleri (flags), kaynak ya da hedef IP adresleri, port numaraları ve paketin paket verisi (payload) kısmı imzaların karakteristik özellikleridir.

Bu sistemlerin en zayıf yönü, yeni saldırı türlerini algılayabilme yeteneğinin olmamasıdır. Yeni bir saldırı türüne ait kayıtlar imza veritabanına eklenmediği için sistem bu saldırıyı tespit edemez. Bu nedenle imza temelli STS'ler yeterli bir çözüm olamadığı için araştırmacılar, anormallik tespiti temelli STS'yi geliştirmişlerdir [77].



Şekil 5.3. Saldırı imzası [78]

Anormallik tespiti temelli STS

Anormallik tespiti temelli STS'lerde, bilgisayar sistemleri ya da iletişim ağlarının, beklenen ya da normal davranışlarını temsil eden bir model oluşturulur. Sistemde, beklenmeyen bir davranış olduğunda, STS alarm üreterek sistem yöneticisini uyarır. Sistemde tespit edilen yetkisiz erişimler ve saldırı aktiviteleri, anormal davranışlar kümesinin bir parçası olarak varsayılır. Bu sistemlerde arzu edilen, tüm kötü niyetli aktivitelerin tanımlanmış olan anormal davranışlar kümesinde yer almasıdır. Bu durumda, yanlış pozitif ve yanlış negatif hatası olmayacağı için sistem hiç yanlış alarm üretmez. Sistemin alarm verip vermeyeceği ile ilgili olarak olabilecek 4 durum vardır. Bunlar;

Saldırı aktivitesi olan ancak normal algılanan: Bu durum yanlış negatif olarak bilinir ve tehlikelidir. Sistem yetkisiz erişimi ya da saldırıyı tespit edemez.

Saldırı olmayan ve normal aktiviteler: Pozitif- negatif durum olarak adlandırılır. Normal ve saldırı olmayan aktiviteleri içerir. STS alarm üretmez.

Saldırı olmayan ve anormal algılanan aktiviteler: Yanlış pozitif olarak adlandırılır. Sistem alarm verir.

Saldırı olan ve anormal aktiviteler: Sistemin bir saldırı ya da güvenlik ihlalini doğru algıladığı ideal durumdur.

Bu yöntemin zorluğu, normal sistem özelliklerinin belirlenmesidir. Bu yöntemde yanlış veya yetersiz modelleme nedeniyle, normal işlemler yanlışlıkla saldırı olarak kabul edilebilir. Bu yöntemin avantajı ise yeni saldırıların tespit edilebilmesidir [79].

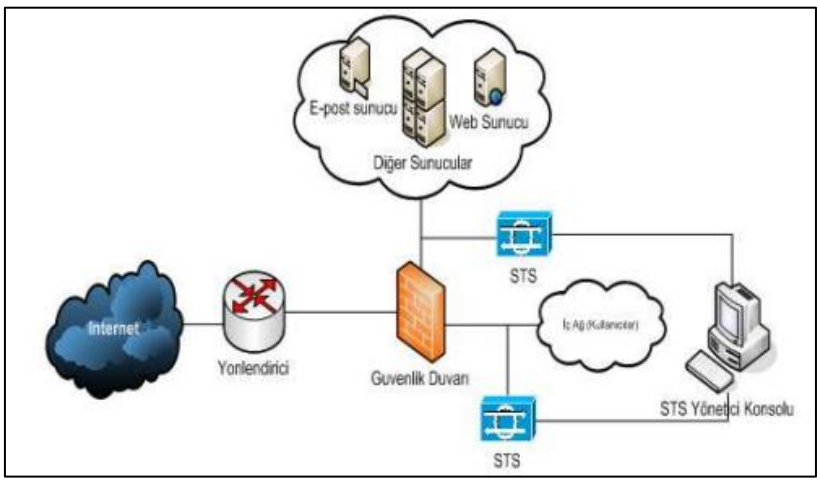
5.3.2. Sistemdeki konumuna göre STS

Bilgilerin görüntülenme ve analiz edilme kaynağına göre sınıflandırılmasıdır. Ağ temelli ve bileşen (host) temelli olmak üzere 2 gruba ayrılır. Her iki STS türü de problem formülasyonu ve çözümü bakımından kendine özgüdür.

Ağ Temelli STS

Ağ temelli STS, birden fazla ağ elemanından ya da kendi ağ segmentinden toplanan veriyi görüntüler. Ağ temelli STS, veri toplayıcı, yönetici ve veri alıp, verinin analiz sonuçlarını iletmekle görevli olan, iletişim modülü olmak üzere temel olarak üç üniteden oluşur.

Veri toplayıcı, ağ üzerine dağıtılmış olan sensörlere sahiptir ve sensörler vasıtasıyla hem veri yakalayıp hem de ağ trafiğini analiz edip biçimlendirmekten sorumludur. Buradan ağ temelli STS'nin, veri önişleme ve başlangıçtaki ağ trafiğini analiz etmekten sorumlu olduğu sonucuna varabiliriz. Veri toplayıcıda, toplanan giriş bilgisi işlenir ve normalize edilir. Daha sonra genellikle standart anormal davranış profilleri ile karşılaştırılmak için kullanılır. Standartlaştırılan veriye yöneticide sınıflandırma mekanizması uygulanarak normal ya da anormal olarak iki sınıfa ayrılır.



Şekil 5.4. Ağ temelli temelli STS [78]

Ağ temelli STS'lerin en önemli avantajları,

- Ağ performansına herhangi bir olumsuz etkisi yoktur.
- Saldırıları gerçek zamanlı olarak algılayıp, sistem yöneticisine hemen haber verebilir.
- Henüz tanımlanmamış şüpheli saldırıları yakalayabilir.
- İşletim sisteminden bağımsızdır.

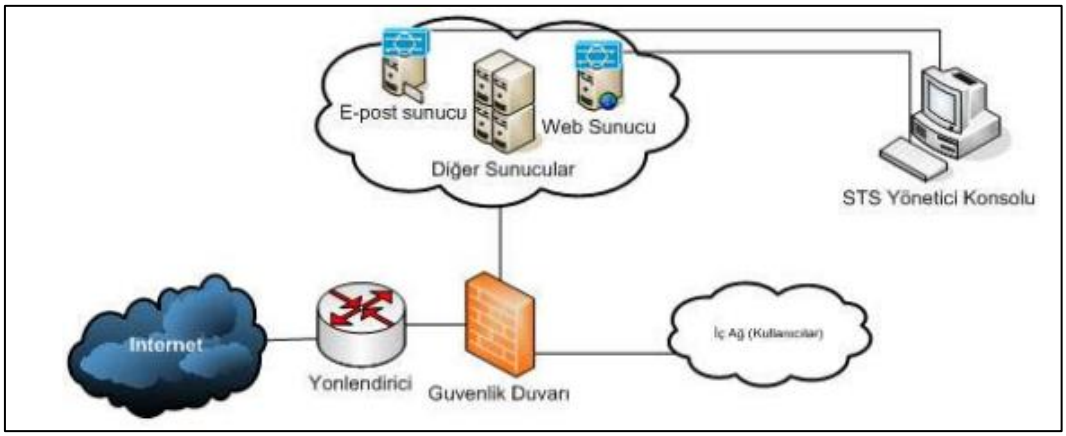
Ağ temelli STS'lerin en önemli dezavantajları ise,

- Belirli uygulamalar için karmaşık protokoller gerektirebilir.
- Görüntülemeye karşı şifrelenmiş trafiği izleyemez.
- Ağ trafik izlerinin hacmi sistemin toplayabileceği kapasiteyi aştığında işlev göremez.

Bileşen (host) temelli STS

Bileşen temelli STS, sistemin dinamik durumunu ya da davranışını görüntülemek için kullanılır. Denetim ajanlarından aldığı verilere ya da log analiz dosyalarına başvurur ve dosya sistemlerindeki değişimleri, kullanıcı erişim kontrolünü, sistem proseslerinin davranışlarını ve diğer kaynakların kullanımını, algılar ve değerlendirir. Bileşen temelli STS, ağa bağlı makinelere gelen paketlerin kısa ve basit olarak denetimini gerçekleştirir.

Bileşen temelli STS'lerin diğer bir özelliği ise dosya sistemlerinin hash değerini tutmasıdır. Bileşen temelli STS, önemli olarak varsaydığı dosyaların hash değerlerini tutan bir veritabanı oluşturur. Veritabanında kayıtlı veride, herhangi bir güvenlik ihlali olduğunu algıladığında, verinin hash değeri yeniden hesaplanır ve veritabanında kayıtlı olan önceki değeri ile karşılaştırılır. İki hash değeri arasında tutarsızlık olduğunda veride değişiklik olduğunu tespit eder. Hash değeri doğrulaması zararlı verilere karşı çok etkili bir yöntemdir [77].



Şekil 5.5. Bileşen (host) temelli STS [78]

Bileşen temelli STS'lerin en önemli avantajları,

- Programlar ve kullanıcılar arasında güçlü bir bağ kurar.
- Saldırıların potansiyel hedeflerini doğru anlayabilmek amacıyla, saldırı esnasında değerli bilgiler elde eder.
- Kurulum için ilave bir donanıma ihtiyaç yoktur.

Bileşen temelli STS'lerin en önemli dezavantajları ise,

- Sistemin ölçeklenebilirliği ve yönetimi (yapılandırma açısından) oldukça karmaşıktır.
- Derin veri analizi ve detaylı bilgi toplama işlemleri, host görüntüleme performansını negatif etkileyebilir.

5.3.3. Veri işleme zamanına göre STS

STS'ler için veri işleme zamanı, tespit edilen bir saldırı ya da güvenlik ihlalinin algılanıp sistem yöneticisine haber verilmesine kadar geçen süreyi ifade eder. STS'ler veri işleme zamanına göre gerçek zamanlı (aktif) ve gerçek zamanlı olmayan (pasif) olmak üzere 2 gruba ayrılır.

Gerçek zamanlı (aktif) STS

Gerçek zamanlı saldırı tespit sistemlerinde, algılanan saldırının bildiriminden hemen sonra STS varolan saldırı tehdidine karşı aktif reaksiyon gösterir. Bu tip STS'ler üzerine günümüzde çalışmalar yapılmakta ve bu sistemler saldırı engelleme sistemi (Intrusion Prevention System-IPS) olarak adlandırılmaktadırlar. Aktif olarak hemen cevap üretilmesi gereken ticari uygulamalarda tek çözüm olan sistemlerdir. Gerçek zamanlı STS'lerin, ağ trafiğinin yoğun olduğu sistemlerde uygulanması, performans açısından zayıf ve maliyetlidir.

Gerçek zamanlı olmayan (pasif) STS

Gerçek zamanlı olmayan (pasif) STS'ler herhangi bir saldırı örüntüsü algıladığında bildirim oluşturur ve bu bildirimleri sistem yöneticisine gönderir. Pasif sistemler, depola ve yolla mantığı ile çalışırlar. Belirlenen saldırılar ve olası tehditleri kayıt ederler ve daha sonraki bir tarihte incelenmek üzere saklarlar.

6. MAKİNE ÖĞRENMESİ

Yapay zekânın bir branşı olan makine öğrenmesi, veriden karmaşık örüntünün tespit edilmesi ve akılcı karar verme için istatistik ve bilgisayarın hesaplama gücünden faydalanır. Makine öğrenme teknikleri sınıflandırma problemlerinde başarılı bir şekilde kullanılmaktadır [80-83].

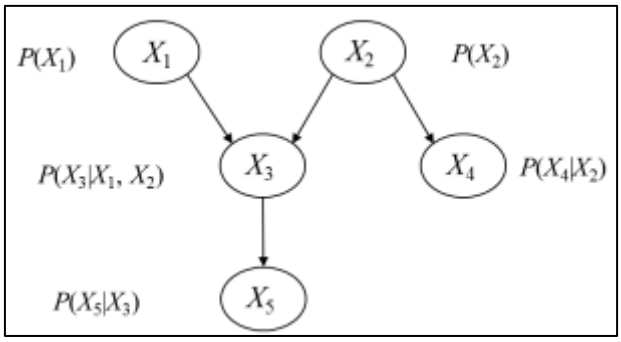
Yapılan bu çalışmada, Bayes sınıflama, k en yakın komşu algoritması, yapay sinir ağları, destek vektör makinesi ve karar ağaçlarının saldırı tespit sistemlerindeki başarısı incelenmiş ve sonuçlar karşılaştırmalı olarak sunulmuştur.

6.1. Bayes Ağları

Bayes ağları, Thomas Bayes tarafından geliştirilmiştir ve makine öğrenmesinde öğreticili öğrenme alt başlığı altında incelenir [84].

Bayes ağları; rastgele değişkenlerin düğümler ve değişkenler arası, olasılıksal bağımlılık ilişkilerinin ise yönlü oklar ile gösterildiği, yönlü dönüşsüz graflardır. Genel olarak bir Bayes ağı, düğümler ve oklar yardımıyla değişkenler ve değişkenler arasındaki olasılıksal ilişkilerin gösterildiği grafiksel model ve değişkenlerin koşullu olasılık değerlerini içeren tablolardan oluşur. Ağda iki düğüm birbirine ok ile bağlandığında okun başlangıcında bulunan düğüm ebeveyn düğüm, okun bitişinde bulunan düğüm ise çocuk düğüm olarak adlandırılır.

Şekil 5.1'de X_1, X_2, X_3, X_4 ve X_5 olmak üzere beş değişkenden oluşan bir Bayes ağı mimarisi gösterilmiştir. Bu ağda X_1 ve X_2 değişkenleri X_3 değişkeninin ebeveyni, X_3 değişkeni ise X_5 değişkeninin ebeveynidir. Ayrıca şekilde görüldüğü üzere X_4 değişkeni X_2 değişkeninin çocuk değişkenidir. Şekilde, değişkenlere ait koşullu olasılık dağılımları, $P(X_1)$, $P(X_2)$, $P(X_3| X_1, X_2)$, $P(X_4| X_2)$ ve $P(X_5| X_3)$ olarak gösterilmiştir [85].



Şekil 6.1. Beş değişkenli bir Bayes ağı mimarisi

Bayes teoremi, birden fazla etkenin olduğu bir olayın meydana gelmesinde, olayda hangi etkenin payının yüksek olduğunun hesaplanması temeline dayanır. Bayes teoremi aşağıda Eşitlik 6.1'de gösterildiği gibi ifade edilebilir.

$$P(X_4|X_2) = \frac{P(X_2|X_4)\,P(X_4)}{P(X_2)} \tag{6.1}$$

Denklemde verilen,

$P(X_4)$ = X_4 olayının önsel olasılığını

$P(X_2)$ = X_2 eğitim verisinin önsel olasılığını

$P(X_2|X_4)$ = X_4 olayı verildiğinde X_2'nin koşullu olasılığını

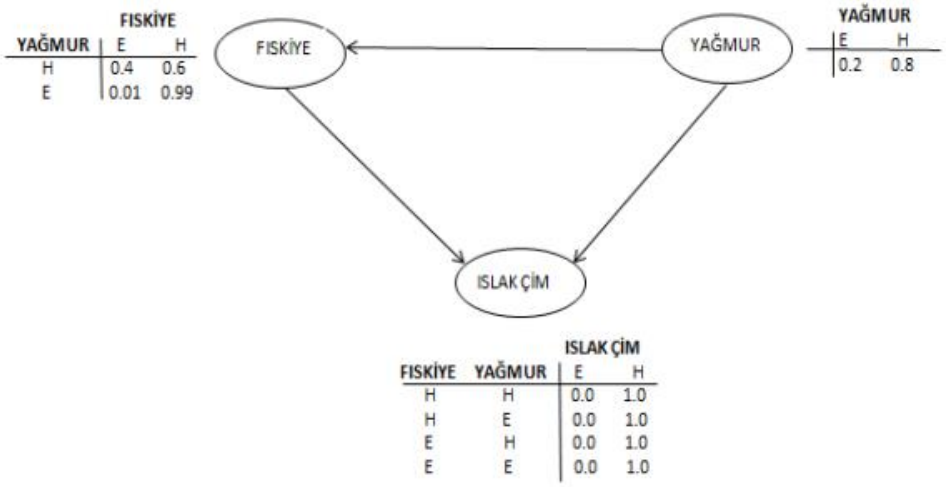
$P(X_4|X_2)$ = X_2 eğitim verisi verildiğinde X_4'ün koşullu olasılığını ifade eder.

Bayes sınıflandırma işleminde genel olarak elde bir örüntü (pattern) vardır ve bu örüntü daha önceden tanımlanmış olan sınıfları tespit eder. Bayes ağları, bilinmeyen verileri sınıflandırmadan önce, eğitim setinden, sınıf etiketleri ve nitelikleri arasındaki ilişkileri öğrenir [45].

Bayes sınıflandırıcılar, saldırı tespit sistemlerinde sıklıkla kullanılan bir yöntemdir. Literatürde Bayes ağları ile geliştirilmiş birçok STS bulunmaktadır [14-17].

6.1.1. Bayes ağları ile karar verme

Bayes ağları ile karar vermeyi anlayabilmek için literatürde sıkça karşılaşılan Pearl'in fıskiye problemi örnek olarak verilebilir. Örnekte çimlerin ıslak olmasını yağmurun yağması ve fıskiyenin açık olması etkilemektedir. Ayrıca fıskiye ile yağmur arasında da koşullu olasılık bulunmaktadır.



Şekil 6.2. Bayes ağı fıskiye örneği

Örneğin Bayes ağı ile gösterimi Şekil 6.2 de görüldüğü gibidir. Düğümler yağmur, fıskiye, ıslak çim değişkenlerini tutmaktadır. Şekildeki tablolar içinde düğümlerin koşullu olasılıkları bulunmaktadır. Bu örneğe göre çimler ıslak olduğunda yağmurun yağmış olma olasılığının bulunması aşağıda anlatılmıştır.

Ç: Islak Çim, F: Fıskiye, Y: Yağmur

$P(Ç,F,Y) = P(Ç|F,Y) \times P(F|Y) \times P(Y)$

$P(Y=E|Ç=E)$: Çimler ıslaksa yağmurun yağmış olma olasılığı

$P(Ç=E,Y=E)$: Çimlerin ıslak ve yağmurun yağmış olduğu durumlar

$P(Ç=E)$: Çimlerin ıslak olma olasılığı

$$P(Y=E|Ç=E) = P(Ç=E,Y=E) / P(Ç=E)$$

$$P(Ç=E,Y=E) = P(Ç=E,F=H,Y=E) + P(Ç=E,F=E,Y=E)$$

$$= 0,8 * 0,99 * 0,2 +0,99 * 0,01 * 0,2$$

$$= 0,1584 + 0,00198$$

$$= 0,16038$$

$$P(Ç=E) = P(Ç=E,F=H,Y=E) + P(Ç=E,F=E,Y=E) + P(Ç=E,F=H,Y=H) + P(Ç=E,F=E,Y=H)$$

$$= 0,8 * 0,99 * 0,2 +0,99 * 0,01 * 0,2 + 0 * 0,6 * 0,8 + 0,9 * 0,4 * 0,8$$

$$= 0,1584 + 0,00198 + 0 + 0,288$$

$$= 0,44838$$

$$P(Y=E|Ç=E) = P(Ç=E,Y=E) / P(Ç=E) = 0,16038 / 0,44838 \approx 35,77\%$$

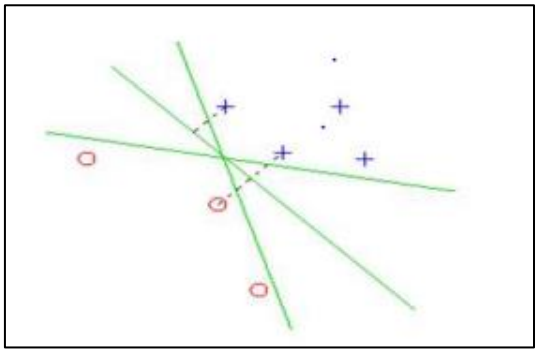
Bu sonuca göre çimler ıslaksa yağmurun yağmış olma olasılığı %35,77'dir.

6.2. Destek Vektör Makinesi

Destek Vektör Makinesi (DVM), Vapnik tarafından 1998 yılında önerilmiş güçlü bir sınıflandırıcıdır. Temeli istatistiksel yöntemlere dayanır. DVM, öğrenme alanında, elde edilen örüntüleri tanıma ve analiz etmede, sınıflama ve regresyon analizini kullanan denetimli bir öğrenme modelidir [86].

DVM, etiketli bir giriş veri setine ihtiyaç duyar. İki sınıftan oluşan veri setinde, girilen giriş veri setinden çıkış olarak iki sınıf oluşturur. Girilen eğitim örnekleri, iki kategoriden birine dâhil edilir. DVM eğitim algoritması, yeni gelen bir örneği kategorilendirmek için bir model kurar. DVM modeli, uzayda noktalar gibi örneklerin temsilidir. Kategorilere ayrılan örnekler, Şekil 6.2'de olduğu gibi mümkün olduğu kadar geniş, net bir hiperdüzlem

ile ayrılır. Yeni örnekler aynı uzaya dâhil edilir ve hangi kategoriye ait oldukları tahmin edilir.



Şekil 6.3. Sınıflandırılacak verilerin en uygun hiperdüzlem ile ayrılması

Veri setine gelen yeni örnekleri en uygun sınıfa dâhil edebilmek amacıyla en uygun hiperdüzlemi bulabilmek için, her iki sınıfın en uygun hiperdüzlemine en yakın veri noktalarından geçen hiperdüzlemler çizilir ve bu iki hiperdüzlem birbirine paraleldir. DVM, farklı sınıflara ait destek vektörleri arasındaki uzaklığı maksimize eden ayırma hiperdüzleminin bulunmasını amaçlar [87].

Doğrusal olarak ayrılabilen iki sınıflı bir sınıflandırma probleminde DVM'nin eğitimi için k sayıda örnekten oluşan X eğitim verisinin $\{x_i, y_i\}$, i=1,.....,k olduğu kabul edilirse, optimum hiperdüzleme ait eşitsizlikler aşağıdaki şekilde olur

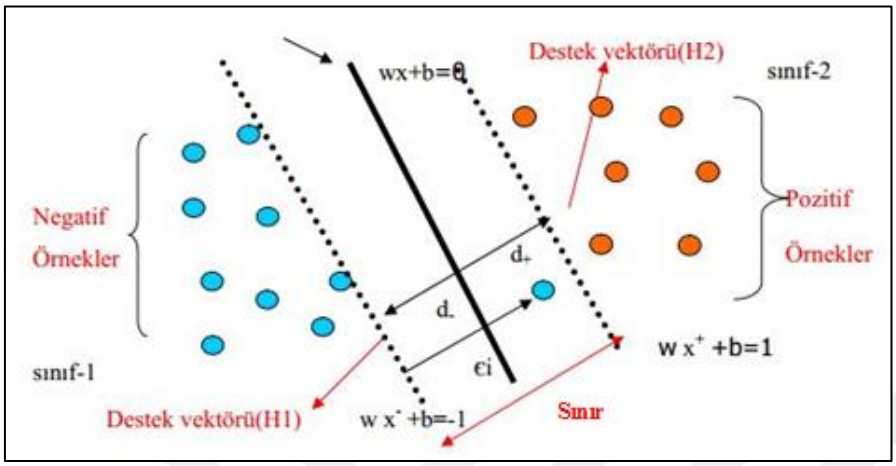
$$w.x_i + b \geq +1 \qquad \text{her y=+1 için} \qquad (6.2)$$

$$w.x_i + b \leq +1 \qquad \text{her y=-1 için} \qquad (6.3)$$

Burada $x \in R^N$ olup N-boyutlu bir uzayı, $y \in \{-1, +1\}$ ise sınıf etiketlerini, w ağırlık vektörünü (hiperdüzlemin normali) ve b eğilim değerini göstermektedir. Optimum hiperdüzlemin belirlenebilmesi için, bu düzleme paralel ve sınırlarını oluşturacak iki hiperdüzlemin belirlenmesi gerekir (Şekil 6.3). Bu hiperdüzlemleri oluşturan noktalar destek vektörleri olarak adlandırılır ve bu düzlemler $w.x_i+b=\pm1$ denklemi ile ifade edilirler [88].

Verilen X örneğini sınıflandırmak için öncelikle en uygun hiperdüzlem bulunur. Bu

hiperdüzlem taraflarından biri negatif sınıfı, diğeri ise pozitif sınıfı temsil eder. X örneği DVM yöntemi ile formüle edilir ve eğer f(x) fonksiyonu sıfırdan büyük çıkarsa pozitif sınıfa, negatif çıkarsa negatif sınıfa atanır.



Şekil 6.4. Doğrusal olarak ayrılabilen veri setleri için hiperdüzlemin belirlenmesi

Destek vektör yöntemi hiperdüzleme en yakın pozitif ve negatif örnekler arasındaki mesafenin (sınır genişliğinin) en yüksek olduğu bir hiperdüzlem bulmaya çalışır. Sınır genişliği (M) Eşitlik 6.7'daki gibi hesaplanır.

$$w.x^{+} + b = +1 \tag{6.4}$$

$$w.x^{-} + b = -1 \tag{6.5}$$

$$w.(x^{+} - x^{-}) = 2 \tag{6.6}$$

$$M = \frac{(x^{+}-x^{-}).w}{|w|} = \frac{2}{|w|} \tag{6.7}$$

"w" değeri ne kadar küçülürse sınır genişliği o kadar artar.

Destek vektör makinesi saldırı tespit sistemlerinde araştırmacılar tarafından en sık kullanılan tekniklerdendir. Literatürde DVM ile geliştirilmiş birçok çalışma yer almaktadır [22, 23, 27, 34].

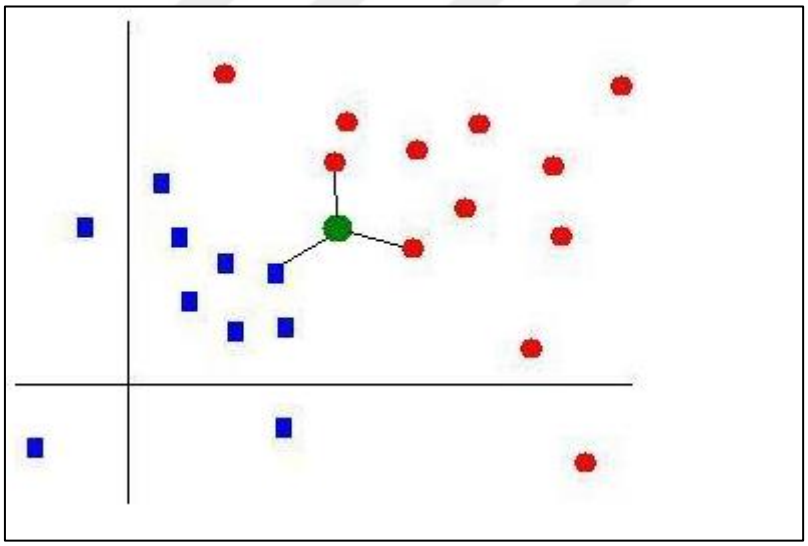
6.3. K En Yakın Komşu Algoritması

K en yakın komşu algoritması denetimli öğrenme algoritmasıdır. Sınıflandırılacak verilerin, eğitim kümesindeki diğer verilere benzerlikleri hesaplanarak, en yakın olduğu düşünülen k verinin ortalamasıyla belirlenen eşik değere göre sınıflara atamaları yapılır.

KNN yönteminde önce test verisi değerleriyle eğitim veri kümesindeki veri değerleri arasındaki Öklid uzaklıkları hesaplanır. Hesaplanan uzaklıklara göre test verisine en yakın mesafedeki k komşu sınıf belirlenir ve karar buna göre en fazla örnek taşıyan sınıfa göre verilir. X={X_1, X_2, X_3, …, X_n} ve Y={Y_1, Y_2, Y_3, …Y_n} arasındaki Öklid uzaklığı eşitlik 6.8'deki gibi hesaplanır.

$$D(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{6.8}$$

Bilinmeyen örnek A, en yakın k-komşularının arasında en sık bulunan sınıfa atanır. Şekil 6.4'te, k değerinin 3 olduğu duruma örnek gösterilmektedir.



Şekil 6.5. k=3 için en yakın komşuluk

KNN algoritmasının adımları;

1. Test kümesindeki her verinin X={X1, X2, X3, …, Xn}, öğrenme kümesindeki verilere D ={d1, d2, d3, …, dn} yakınlığı hesaplanır.

2. $\mathrm{sim}(x_i, d_l) = \frac{x_i \,.\, d_l}{||x_i|| \,.\, ||d_l||}$ (6.9)
3. (i={1,2,3,…,n}, l={1,2,3,…,n})
4. Her verinin öğrenme kümesindeki verilere olan yakınlıkları sıralanıp ilk "k" tanesi alınarak ortalamaları hesaplanır.
5. $\mathrm{sim\_avg}(x_i) = \frac{\max(\sum_{l=1}^{k} \mathrm{sim}(x_i, d_l))}{k}$ (6.10)
6. Ortalama değerleri, belirlenen eşik değerinden büyük olanlar normal, küçük olanlar ise anormal olarak sınıflandırılır.

Algoritmanın performansını etkileyen kriterler;

- K en yakın komşu sayısı (benzerlik ölçümü için seçilecek komşu sayısı: k)
- Eşik değer (etiketleme işlemi için verinin k en yakın komşuya olan benzerliklerinden hesaplanan ortalama değerinin kıyaslanmasında kullanılır)
- Benzerlik ölçümü
- Öğrenme kümesindeki normal davranışların yeterli sayıda olması (öğrenme kümesi yeterli çeşitlilikte ve sayıda normal davranış verisi içermiyorsa, test kümesinde yer alan yeni normal davranış verileri anormal olarak algılanabilir.)

KNN yöntemi basit olmasına karşın eğitim verilerinin tamamıyla tek tek kıyaslama gerektirdiğinden hesaplama ve depolama yönünden maliyetlidir. Literatürde KNN yöntemi STS çalışmalarında başarılı bir şekilde kullanılmaktadır [89-91].

6.4. Yapay Sinir Ağları

Yapay Sinir Ağları (YSA), biyolojik sinir hücrelerini (nöron) modelleyen, güçlü bir sınıflandırma aracıdır. Ağı oluşturan her bir elemana yapay sinir (nöron) adı verilmektedir. Yapay sinir ağı çeşitli ağırlıklandırmalar sayesinde birbirine bağlanmış birçok yapay sinir hücresinden oluşmaktadır.

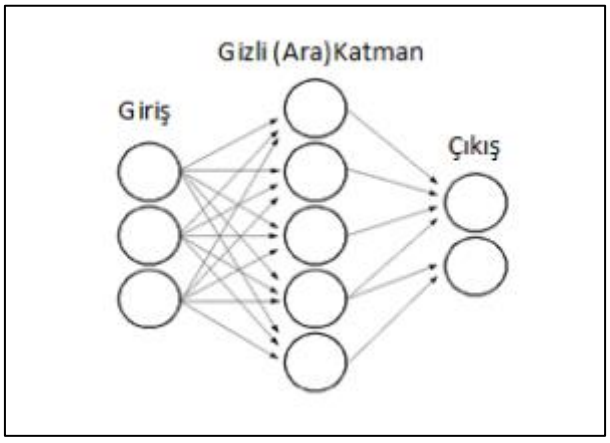
Normalde tek bir nöron sadece doğrusal problemleri çözebilir. Diğer birçok sınıflandırma problemini çözebilmek için çok katmanlı yapay sinir ağları, (Multi Layer Perceptron-MLP) kullanılmaktadır. Çok katmanlı YSA, fonksiyon uydurma, sınıflandırma ve eşleşme

problemlerinde sıkça kullanılmaktadır. Sınıflandırmadaki başarısından dolayı STS'lerde sıklıkla kullanılmıştır [92].

Çok Katmanlı Algılayıcılar (ÇKA)'dan günümüzde en sık kullanılanı back propagation ağlardır. Bugün özellikle sınıflandırma işlemlerinde en çok kullanılan yöntemlerin başında gelmektedir. Back propagation ağlarda öğrenme fonksiyonu olarak delta öğrenme kuralı kullanılmaktadır. Delta öğrenme kuralı aşağıda gösterildiği gibi Eşitlik 6.9 ile ifade edilir.

$$w_{i,j}(new) = w_{i,j}(old) + (\mu * [t - f(y_{in})] * f'(y_{in})) \quad (6.9)$$

ÇKA'lar; girdi katmanı, ara katmanlar ve çıktı katmanı olmak üzere 3 katmandan oluşmaktadır. Bilgiler girdi katmanından ağa tanıtılır, ara katmanlardan çıktı katmanına ulaşır ve çıktı katmanından dış dünyaya aktarılır [93]. ÇKA mimarisi Şekil 6.5'de gösterilmiştir [83]. Yapay sinir ağı öğrenme sürecinde, gerçek hayattaki probleme ilişkin veri ve sonuçlardan diğer bir deyişle örneklerden faydalanılır. Probleme ilişkin değişkenler yapay sinir ağının girdi dizisini, bu değişkenlerle elde edilmiş gerçek sonuçlar ise yapay sinir ağının ulaşması gereken hedef çıktılar dizisini oluşturur.



Şekil 6.6. Standart üç katmanlı YSA yapısı

Öğrenme sürecinde, seçilen öğrenme yaklaşımına göre ağırlıklar değiştirilir. Ağırlık değişimi, öğrenmeyi ifade eder. YSA' da ağırlık değişimi yoksa öğrenme işlemi de durmuştur. Bu nedenle eğitimde kullanılacak eğitim veri setinin oluşturulmasında çok dikkatli olunmalıdır. Verisetinin gerçekten de ilgili olayların motiflerini içerdiğinden emin olmak gerekir [94].

YSA kullanılarak gerçekleştirilen saldırı tespit sistemlerinde oldukça başarılı sonuçlar elde edilmiştir [47, 49, 53].

6.5. Karar Ağaçları

Karar ağaçları, eğitim ve testinin hızlı olması, sonuçlarının daha kolay yorumlanabilmesi ve etkin olması sebebiyle sınıflandırmada sıklıkla kullanılan yöntemlerden biridir [95, 96]. Karar ağaçları ile sınıflandırma iki adımda gerçekleştirilir. İlk adımda ağaç oluşturulur. İkinci adımda ise bu ağaç yapısından sınıflandırma kuralları elde edilir. Genel olarak sınıflandırma işlemi şöyle ifade edilebilir: $D=\{t_1, t_2, \dots, t_n\}$ bir veri tabanı olsun ve her bir kayıt t_i ile temsil edilsin. $C=\{C_1, C_2, \dots, C_m\}$ ise m adet sınıftan oluşan sınıflar kümesini temsil etsin. Her bir C_j ayrı bir sınıftır ve her bir sınıf kendisine ait kayıtları içerir. Yani, $C_j=\{t_i|=C_j, 1\leq i\leq n$ ve $t_i \in D\}$' dir. Veritabanındaki her bir kayıt için alanlar ise $\{A_1, A_2, \dots, A_n\}$ 'den oluşsun. Bu tanıma ilaveten her bir kayıt $C=\{C_1, C_2, \dots, C_m\}$ sınıflarından birine ait ise karar ağacı şöyle tanımlanabilir: Her bir düğüm A_i alanı ile isimlendirilir. Kök düğüm ile yaprak arasındaki düğümler birer sınıflandırma kuralıdır.

Karar ağaçları oluşturulurken kullanılan algoritmanın ne olduğu önemlidir. Kullanılan algoritmaya göre ağacın yapısı değişebilir. Değişik ağaç yapıları farklı sınıflandırma sonuçları verebilir [97].

Çizelge 6.1. Bazı karar ağacı algoritmaları ve özellikleri [98]

KARAR AĞACI ALGORİTMASI	ÖZELLİKLER
C&RT	Gini'ye dayalı ikili bölme işlemi mevcuttur. Son veya uç olmayan her bir düğümde iki adet dal bulunmaktadır. Budama işlemi ağacın karmaşıklık ölçüsüne dayanır. Sınıflandırma ve regresyonu destekleyici bir yapıdadır. Sürekli hedef değişkenleri ile çalışır. Verinin hazırlanmasına gereksinim duyar.
C4.5 ve **C5.0** (ID3 karar ağacı algoritmasının ileri sürümleri)	Her düğümden çıkan çoklu dallar ile ağaç oluşturur. Dalların sayısı tahmin edicinin kategori sayısına eşittir. Tek bir sınıflayıcı da birden çok karar ağacını birleştirir. Ayırma işlemi için bilgi kazancı kullanır. Budama işlemi her yapraktaki hata oranına dayanır.
CHAID (Chi-Squared Automatic. Interaction Detector)	Ki-kare testleri kullanarak bölme işlemini gerçekleştirir. Dalların sayısı iki ile tahmin edicinin kategori sayısı arasında değişir.
SLIQ (Supervised Learning in Quest)	Hızlı ölçeklenebilir bir sınıflayıcıdır. Hızlı ağaç budama algoritması mevcuttur.
SPRINT (Scalable Parallelizable Induction of Decision Tree)	Büyük veri kümeleri için idealdir. Bölme işlemi tek bir niteliğin değerine dayanır. Tüm bellek sınırlamaları üzerinde nitelik listesi veri yapısı kullanarak işlem yapar.

Karar ağaçlarına dayalı olarak geliştirilen birçok algoritma vardır. Bu algoritmalar birbirlerinden kök, düğüm ve dallanma kriterine göre farklı kategorilere ayrılırlar. Yaygın olarak bilinen algoritmalar ID3, C4.5 ve C5dir.

Karar ağaçları, saldırı tespit sistemlerinde sıklıkla kullanılan yöntemlerden bir tanesidir. Literatürde karar ağaçları ile geliştirilmiş birçok STS bulunmaktadır [39-41].

7. KULLANILAN VERİ KÜMESİ VE DENEYSEL ÇALIŞMA

7.1. KDD CUP99

Saldırı tespit sistemlerinin performansını belirlemek için en zorlu aşama geçerli ve uygun veri setlerinin elde edilmesidir. İnternet ortamından elde edilen veriler saldırının var olup olmadığına dair genel bir bilgi içermez. Saldırı için belirleyici özellik veya bilgi ağın gözlemlenmesi yoluyla elde edilebilir. Genel olarak ağın gözlemlenmesi masraflı ve gereksiz bir iş olarak görülebilir. Ancak ağ veya bilgisayar sistemlerinin çalışabilmesi için, ağdan veya bilgisayar sistemlerinden veri toplama, artık günümüzde kaçınılmaz bir süreçtir. Bu süreç biraz maliyetli olduğundan dolayı bazı ağ mühendisleri yapay veriler kullanarak ağ veya sistemlerini sorunsuz çalıştırmayı istemektedirler. Ancak yapay verinin internet trafiğine benzediğini kanıtlamak zordur. Genel olarak; gerçek veri, saldırı türleri belli olan veri setleri bulmak ve ağ trafiğini tanımlamak ve benzetim yapmak zordur. Yukarıda belirtilen zorluklara rağmen saldırı tespit sistemlerini test etmek için geçerli veri kümelerine ihtiyaç vardır.

Genel olarak bir ağ trafiği bir koklayıcı (sniffer) kullanılarak gözlemlenebilir. Ancak sadece ağ paketlerini gözlemlemek ağ trafiği hakkında genel bir bilgi vermeyebilir. Bu dezavantajlara rağmen, saldırı tespit sistemlerinin testleri için geliştirilen birkaç veri seti bulunmaktadır. Bunlardan bazıları KDD CUP99, DARPA 1998, DARPA 1999, UNM, SSCNNJU, CUCS, Windows sistem ve network tcpdump data veri setleridir [99]. KDD CUP99 ve DARPA veri seti saldırı tespit çalışmalarında en çok tercih edilen veri setidir [100].

KDD CUP99 veriseti, basit, içerik, zaman tabanlı trafik ve host tabanlı trafik adı altında 4 farklı tipte saldırı türü içeren, 41 nitelik ile temsil edilen, yaklaşık beş milyon kayıttan oluşmaktadır. KDD ve DARPA veriseti Amerikan Hava Kuvvetleri (US Air Force) network ağına benzer bir yapıya sahip olması düşünülerek tasarlanmış, bir benzetim veri setidir. Verisetindeki saldırı tipleri aşağıdaki gibidir.

Hizmet Engelleme Saldırıları (Denial of Service-DOS): TCP/IP protokolünün yapısından kaynaklanan açıklardan ya da, işletim sistemi veya uygulamada bulunan zayıflıklardan faydalanarak, bir sunucu, servis veya ağın servis veremez hale getirilmesidir [72].

Bilgi Tarama Saldırıları (Probe): Bir sunucunun ya da herhangi bir ağın geçerli ip adreslerini, ağdaki bilgisayar sayısını, bilgisayardaki kullanıcı sayısını ve kullanıcı bilgilerini, aktif giriş kapılarını (port) veya işletim sistemini öğrenmek için yapılan saldırılardır [72].

Kullanıcı hesabını yönetici hesabına yükseltme (User to Root-U2R): Yönetici haklarına sahip olmayıp sadece sisteme erişim yetkisi olan bir kullanıcının yönetici haklarını ele geçirmek amacıyla yaptığı saldırılardır [72].

Yönetici hesabını ele geçirerek yerel ağda oturum açma (Remote to Local-R2L): Sistemde kullanıcı haklarına sahip olunmadığı halde, hedef ağdaki bilgisayarlarda misafir ya da başka bir kullanıcı olarak erişim yetkisi kazanmak için yapılan saldırılardır [72].

KDD CUP99 veri setinde kullanılan özniteliklerin, açıklamaları ve tipi Çizelge 7.1'de gösterildiği gibidir.

Turuncu olarak işaretlenmiş öznitelikleri elde etmek için sunucu seviyesinde bilgi gerekmektedir. Bilgisayar ağından elde edilen bilgiler ile bu öznitelikler oluşturulamamaktadır. Ayrık tipteki öznitelikler tek bir ağ paketi bilgisinden elde edilebilirken sürekli tiptekiler için bir veya birden fazla oturum bilgisine ihtiyaç vardır. Sürekli tipteki öznitelikler, tek bir oturum bilgisi ile belirlenebileceği gibi daha önceki oturumlara ait bilgilere de ihtiyaç duyabilirler. Daha önceki oturum bilgilerini tutmak için iki farklı bellek yapısı kullanılmaktadır. İlk bellek yapısında, *t* saniye içerisinde gerçekleşen oturumlar tutulmaktadır. İkinci yapıda ise en son gerçekleşen *n* adet oturum bilgisi saklanmaktadır. Bu bellekler, sistemin bilgisayar ağı trafiği hakkında daha genel bilgi elde etmesini sağlarlar [101].

7.2. NSL-KDD Veriseti

NSL-KDD veriseti KDD CUP99 veri setinin doğasında yer alan bazı sorunların çözümü için önerilmiştir. KDD CUP99 veriseti anormallik tespiti için en yaygın kullanılan veri setidir. Ancak Tavallaee ve diğerleri bu veri kümesi üzerinde istatistiksel bir analiz yapmış ve değerlendirilen sistemlerin performansını etkileyen ve anormallik tespiti yaklaşımlarının çok kötü sonuçlar üretmesine neden olan iki önemli sorun tespit

etmişleridir. Bu iki sorunun çözümü için, tamamı KDD CUP99 veri seti içindeki kayıtlardan seçilen yeni bir veri seti olan NSL-KDD veri setini önermişlerdir [102].

NSL-KDD veri setinin KDD CUP99 verisetine göre avantajları şunlardır;

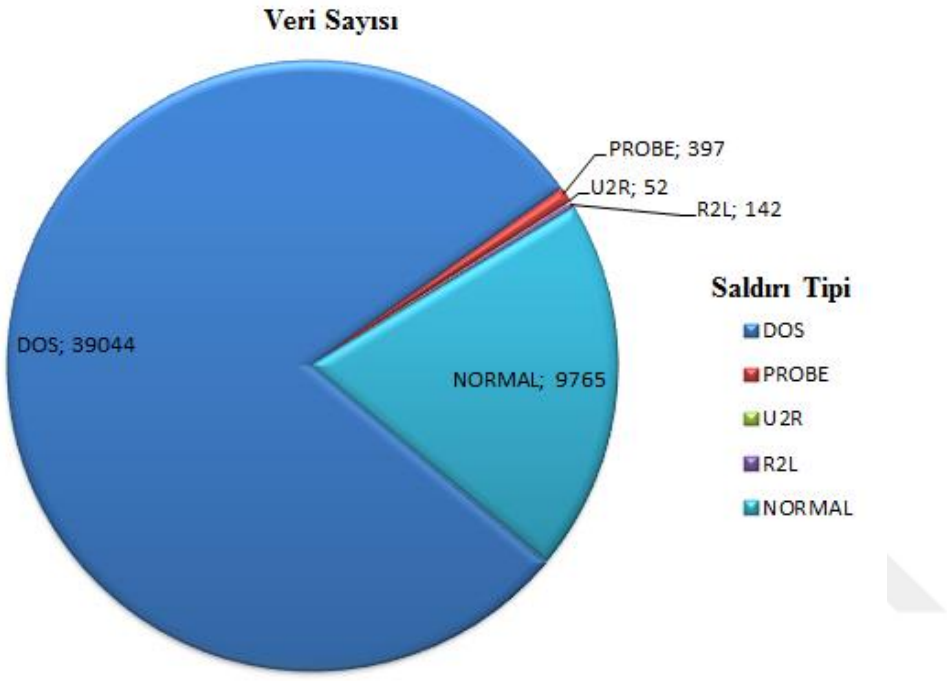
Birincisi, eğitim setinde lüzumsuz kayıt yer almaz, bu nedenle sınıflandırıcılar daha sık kayıtlara karşı önyargılı olmayacaktır. İkinci olarak ise her bir zorluk seviyesi grubundan seçilen kayıtların sayısı, orijinal KDD veri setindeki kayıtların yüzdesi ile ters orantılıdır. Sonuç olarak farklı makine öğrenmesi tekniklerinin sınıflandırma oranları daha geniş bir aralıkta değişkenlik gösterir ve buda farklı öğrenme tekniklerinin daha doğru ve etkili bir değerlendirme yapılmasına olanak tanır. Üçüncü olarak, eğitim ve test setlerindeki kayıt sayıları makuldür. Bu sayede deneyler yaparken eğitim ve test amacıyla tüm veri setinden küçük parçalar almaya gerek kalmaz.

NSL-KDD veriseti içinde 41 nitelik ile temsil edilen, 4 saldırı türü ve normal davranışlar olmak üzere 5 sınıftan oluşan örnekler yer almaktadır. Hizmet aksatma saldırıları (DOS), bilgi tarama saldırıları (Probe), kullanıcı hesabını yönetici hesabına yükseltme saldırıları (User to Root-U2R) ve yönetici hesabını ele geçirerek yerel ağda oturum açma saldırıları (Remote to Local-R2L) veriseti içindeki saldırı türleridir.

Çizelge 7.1. KDD CUP99 veri setindeki nitelikler [101]

Özellik	***Açıklama***	***Tip***
duration	bağlantı süresi (sn.)	Sürekli
protocol_type	protokol tipi, örn. tcp, udp, vb.	Ayrık
service	hedef ağ servisi, örn., http, telnet, vb.	Ayrık
src_bytes	kaynaktan hedefe iletilen veri miktarı (byte)	Sürekli
dst_bytes	hedeften kaynağa iletilen veri miktarı (byte)	Sürekli
flag	bağlantının durumu: normal veya hata	Ayrık
land	bağlantının kaynak ve hedefi aynı sunucu/port ise 1; diğer 0	Ayrık
wrong_fragment	hatalı parça sayısı	Sürekli
urgent	acil paket sayısı	Sürekli
hot	“sıcak” göstergelerin sayısı	Sürekli
num_failed_logins	hatalı giriş sayısı	Sürekli
logged_in	başarılı giriş 1; diğer 0	Sürekli
num_compromised	“tehlikeli” durum sayısı	Sürekli
root_shell	root shell ele geçirildiyse 1; diğer 0	Ayrık
su_attempted	``su root'‘ komutu çalıştırıldıysa 1; diğer 0	Ayrık
num_root	“root” erişim sayısı	Sürekli
num_file_creations	yaratılan dosya sayısı	Sürekli
num_shells	açık shell sayısı	Sürekli
num_access_files	erişim kontrol dosyalarındaki işlem sayısı	Sürekli
num_outbound_cmds	ftp oturumundaki giden komut sayısı	Sürekli
is_hot_login	“sıcak” listeden bir giriş ise 1; diğer 0	Ayrık
is_guest_login	“misafir” girişi ise 1; diğer 0	Ayrık
count	aynı sunucuya t sn içinde yapılan bağlantı sayısı	Sürekli
serror_rate	``SYN'‘ hatası alınan bağlantı yüzdesi	Sürekli
rerror_rate	``REJ'‘ hatası alınan bağlantı yüzdesi	Sürekli
same_srv_rate	aynı servise yapılan bağlantı yüzdesi	Sürekli
diff_srv_rate	farklı servislere yapılan bağlantı yüzdesi	Sürekli
dst_host_count	aynı sunucuya n adet bağlantı içinde yapılan bağlantı sayısı	Sürekli
dst_host_serror_rate	``SYN'‘ hatası alınan bağlantı yüzdesi	Sürekli
dst_host_reror_rate	``REJ'‘ hatası alınan bağlantı yüzdesi	Sürekli
dst_host_same_srv_rate	aynı servise yapılan bağlantı yüzdesi	Sürekli
dst_host_diff_srv_rate	farklı servislere yapılan bağlantı yüzdesi	Sürekli
dst_host_same_src_port_rate	aynı kaynak portundan gelen bağlantı yüzdesi	Sürekli
srv_count	aynı servise t saniye içinde yapılan bağlantı sayısı	Sürekli
srv_serror_rate	``SYN'‘ hatası alınan bağlantı yüzdesi	Sürekli
srv_rerror_rate	``REJ'‘ hatası alınan bağlantı yüzdesi	Sürekli
srv_diff_host_rate	farklı sunuculara yapılan bağlantı yüzdesi	Sürekli
dst_host_srv_count	aynı servise n adet bağlantı içinde yapılan bağlantı sayısı	Sürekli
dst_host_srv_serror_rate	``SYN'‘ hatası alınan bağlantı yüzdesi	Sürekli
dst_host_srv_rerror_rate	``REJ'‘ hatası alınan bağlantı yüzdesi	Sürekli
dst_host_srv_diff_host_rate	farklı sunuculara yapılan bağlantı yüzdesi	Sürekli

Gerçekleştirilen testlerde iki adet veriseti kullanılmıştır. Birinci verisetinde KDD CUP99 verisetinden %1'lik kısım rastgele örnekleme yöntemiyle seçilmiştir. İkinci veriseti ise NSL-KDD verisetidir. Seçilen verisetlerinde normal olarak sınıflandırılan veriler ve DOS, PROBE, U2R, R2L saldırı verileri yer almaktadır. Birinci verisetinde saldırı tipine göre veriseti içindeki örneklerin sayısal dağılımı Şekil 7.1'de gösterildiği gibidir.



Şekil 7.1. KDD CUP99 verisetinde saldırı tipine göre verisetindeki örneklerin sayısal dağılımı

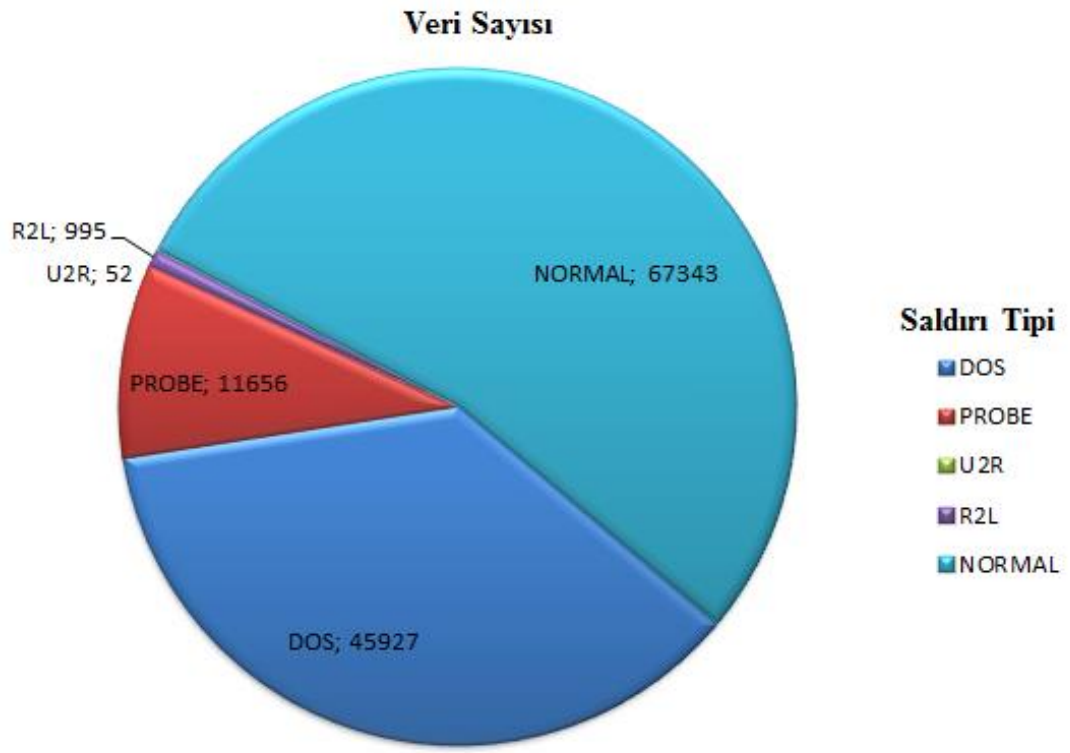
Şekil 7.1'e göre kullanılan verisetinde, 9765 adet normal olarak kabul edilen davranış örneği, 39044 adet DOS saldırısı örneği, 397 adet PROBE saldırısı örneği, 142 adet R2L saldırı örneği ve 52 adet U2R saldırı örneği yer almaktadır. Verisetinde yer alan örnekler, sistemlere yapılan saldırıların sıklığıyla orantılı olacak şekilde dağılım göstermektedir.

Kullanılan birinci veriseti KDD CUP99 içindeki örneklerin saldırı tipi ve saldırı türlerine göre sayısal dağılımı Çizelge 7.2'deki gibidir.

Çizelge 7.2. KDD CUP99 verisetinde saldırı tipi ve saldırı türlerine göre örneklerin sayısal dağılımı

Saldırı Tipi	Saldırı Türü	Veri Sayısı
Normal		9765
DOS (39044)	back	194
	land	3
	neptune	10626
	pod	29
	smurf	28096
	teardrop	96
PROBE (397)	ipsweep	119
	nmap	26
	portsweep	94
	satan	158
U2R (52)	buffer_overflow	30
	loadmodule	9
	perl	3
	rootkit	10
R2L (142)	ftp_write	5
	guess_passwd	8
	phf	4
	multihop	7
	imap	12
	spy	1
	warezclient	101
	warezmaster	4
TOPLAM		49400

İkinci veriseti NSL-KDD'de saldırı tipine göre veriseti içindeki örneklerin sayısal dağılımı Şekil 7.2'de gösterildiği gibidir.



Şekil 7.2. NSL-KDD verisetinde saldırı tipine göre verisetindeki örneklerin sayısal dağılımı

Şekil 7.2'ye göre kullanılan verisetinde, 67343 adet normal olarak kabul edilen davranış örneği, 45927 adet DOS saldırısı örneği, 11656 adet PROBE saldırısı örneği, 995 adet R2L saldırı örneği ve 52 adet U2R saldırı örneği yer almaktadır. Verisetinde yer alan örnekler, sistemlere yapılan saldırıların sıklığıyla orantılı olacak şekilde dağılım göstermektedir.

Kullanılan ikinci veriseti NSL-KDD içindeki örneklerin saldırı tipi ve saldırı türlerine göre sayısal dağılımı Çizelge 7.3'deki gibidir.

Çizelge 7.3. NSL-KDD verisetinde saldırı tipi ve saldırı türlerine göre örneklerin sayısal dağılımı

Saldırı Tipi	Saldırı Türü	Veri Sayısı
Normal		67343
DOS (45927)	back	956
	land	18
	neptune	41214
	pod	201
	smurf	2646
	teardrop	892
PROBE (11656)	ipsweep	3599
	nmap	1493
	portsweep	2931
	satan	3633
U2R (52)	buffer_overflow	30
	loadmodule	9
	perl	3
	rootkit	10
R2L (995)	ftp_write	8
	guess_passwd	53
	phf	4
	multihop	7
	imap	11
	spy	2
	warezclient	890
	warezmaster	20
TOPLAM		125973

Veri setinde bulunan “Protocol_type” ve “service” özellikleri sayısal hale çevrilmiştir. Ayrıca verisetinde bulunan “class” özelliğindeki saldırı isimleri de Normal=0, DOS=1, PROBE=2, U2R=3, R2L=4 olarak sayısallaştırılmıştır.

7.3. Gerçekleştirilen Testler ve Değerlendirme

Makine öğrenmesi tekniklerinin STS’lerdeki performans analizi destek vektör makinesi, yapay sinir ağları, karar ağaçları, Bayes ağları ve k en yakın komşu algoritması kullanılarak yapılmıştır. Elde edilen sonuçlar WEKA’da 10-kat çapraz geçerleme ile elde edilmiştir. Testlerin gerçekleştirildiği platformun özellikleri Çizelge 7.4 ‘de gösterildiği gibidir.

Çizelge 7.4. Testlerin gerçekleştirilme ortamı

Sistem	ASUS VivoBook
İşlemci	Intel Core i5 4200U 1.60 GHz 2.3 Ghz
Birincil Bellek	5,89 GB
İkincil Bellek	670 GB
İşletim Sistemi	Windows 10
Kullanılan Yazılım	WEKA

Bayes ağları, DVM, KNN, YSA ve karar ağaçları ile gerçekleştirilen testlerde sınıflandırma başarısı (Accuracy), duyarlılık (Sensitivity), seçicilik (Specificity), kesinlik (Precision) ve F-Ölçütü (F-Measure) karışıklık matrisi (confusion matrix) kullanılarak Eşitlik (7.1)-(7.5) ile hesaplanmıştır. Test sonucunda ulaşılan sonuçların başarım bilgileri karışıklık matrisi ile ifade edilebilir. Karışıklık (confusion) matrisi, örnek kümesindeki gerçek sınıf etiketi ile tahmin edilen sınıf etiketi sayılarını içerir. İki sınıf için örnek karışıklık Çizelge 7.5'de gösterilmiştir.

Çizelge 7.5. İki sınıf için karışıklık matrisi

		TAHMİN EDİLEN		TOPLAM
		C^+	C^-	
GERÇEK	C^+	DP Doğru Pozitif	YN Yanlış Negatif	Gerçek Pozitif Sayısı (N^+)
	C^-	YP Yanlış Pozitif	DN Doğru Negatif	Gerçek Negatif Sayısı (N^-)
TOPLAM		Tahmin Pozitif Sayısı	Tahmin Negatif Sayısı	Toplam Örnek Sayısı (N)

$$Doğruluk = \frac{DP+DN}{DP+YN+DN+YP} \tag{7.1}$$

Doğruluk (Accuracy), bir sınıflandırıcının doğru sınıflandırdığı örnek sayısının toplam örnek sayısına oranıdır. Doğruluk değerlendirmesi test kümesi kullanılarak hesaplanır. Eğitim sırasında kullanılmayan test verilerinde, doğru sınıflandırdığı örnek sayısı alınarak doğruluk düzeyi hesaplanır.

$$Duyarlılık \ = \frac{DP}{DP+YN} \tag{7.2}$$

Duyarlılık (Sensitivity ya da Recall), gerçek değeri pozitif olup pozitif değere sınıflandırılan örnek sayısının, gerçek değeri pozitif olan tüm örneklerin sayısına oranıdır.

$$Seçicilik \ = \frac{DN}{DN+YP} \tag{7.3}$$

Seçicilik (Specificity), gerçek değeri negatif olup negatif değere sınıflandırılan örnek sayısının, gerçek değeri negatif olan tüm örneklerin sayısına oranıdır.

$$Kesinlik \ = \frac{DP}{DP+YP} \tag{7.4}$$

Kesinlik (Precision), gerçek değeri pozitif olup pozitif değere sınıflandırılan örnek sayısının, pozitif değere sınıflandırılan tüm örneklerin sayısına oranıdır.

$$F\_Ölçütü \ = 2\ x\ \frac{kesinlik\ x\ duyarlılık}{kesinlik+\ duyarlılık} \tag{7.5}$$

Kesinlik ve duyarlılık ölçütleri tek başına anlamlı bir karşılaştırma sonucu çıkarmamıza yeterli olmadığı için her iki ölçütü beraber değerlendirmek daha doğru sonuç verir. Bunun için F-ölçütü tanımlanmıştır. F-ölçütü, kesinlik ve duyarlılığın harmonik ortalamasıdır.

7.3.1. Bayes performans sınaması

Makine öğrenmesi tekniklerinden Bayes ağları ile yapılan testler sonucunda, sınıflandırıcının, saldırı tespit sistemlerindeki performansını değerlendirmemizi sağlayan başarım ölçütlerinin, saldırı tiplerine göre elde edilen değerleri Çizelge 7.6'da gösterildiği gibidir.

Çizelge 7.6. Bayes ağları ile yapılan test sonuçları

Veriseti	Saldırı Tipi	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
KDD CUP99	Normal	0,992	0,969	0,998	0,99	0,98
	DOS	0,991	0,988	1	1	0,994
	R2L	0,995	0,923	0,995	0,366	0,524
	PROBE	0,993	0,98	0,993	0,532	0,69
	U2R	0,997	0,885	0,998	0,282	0,428
NSL-KDD	Normal	0,966	0,963	0,969	0,973	0,968
	DOS	0,978	0,942	0,999	0,999	0,969
	R2L	0,995	0,965	0,995	0,62	0,755
	PROBE	0,981	0,982	0,981	0,844	0,908
	U2R	0,993	0,808	0,993	0,046	0,086

Çizelge 7.6'da görüldüğü üzere, KDD CUP99 veriseti ile yapılan testlerde Bayes ağları, kullanıcı hesabını yönetici hesabına yükseltme saldırılarını tespit etmede diğer saldırı türlerine göre daha başarılıyken, DOS saldırılarına karşı duyarlılık, seçicilik, kesinlik ve F-ölçütü yönünden daha iyi performansa sahiptir. NSL-KDD ile yapılan testlerde ise Bayes ağları yönetici hesabını ele geçirerek yerel ağda oturum açma saldırılarını tespit etmede diğer saldırılara göre daha etkili bir sınıflandırıcıdır. Duyarlılık yönünden bilgi tarama saldırılarına, seçicilik, kesinlik ve F-ölçütü yönünden ise DOS saldırılarına karşı daha başarılıdır.

7.3.2. DVM performans sınaması

Destek vektör makinesi kullanılarak yapılan testler sonucunda, sınıflandırıcının, saldırı tespit sistemlerindeki performansını değerlendirmemizi sağlayan başarım ölçütlerinin, saldırı tiplerine göre elde edilen değerleri Çizelge 7.7'de gösterildiği gibidir.

Çizelge 7.7. Destek vektör makinesi ile yapılan test sonuçları

Veriseti	Saldırı Tipi	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
KDD CUP99	Normal	0,993	0,997	0,993	0,971	0,983
	DOS	0,996	0,995	0,999	1	0,997
	R2L	0,999	0,796	0,999	0,807	0,801
	PROBE	0,992	0,877	1	0,967	0,919
	U2R	0,999	0,462	1	0,96	0,623
NSL-KDD	Normal	0,968	0,98	0,938	0,948	0,964
	DOS	0,993	0,964	0,998	0,996	0,98
	R2L	0,996	0,723	0,998	0,731	0,727
	PROBE	0,979	0,849	0,992	0,913	0,88
	U2R	0,999	0,462	0,999	0,667	0,545

Çizelge 7.7'ye göre KDD CUP99 veriseti ile yapılan testlerde destek vektör makinesi, kullanıcı hesabını yönetici hesabına yükseltme (U2R) ve yönetici hesabını ele geçirerek yerel ağda oturum açma (R2L) saldırına karşı, diğer saldırı türlerine göre daha iyi sınıflandırma başarına sahiptir. Sınıflandırıcı, duyarlılık yönünden normal davranışları algılamada, seçicilik yönünden PROBE ve U2R saldırılarında, kesinlik yönünden DOS saldırılarında ve seçicilik ile kesinliği beraber değerlendirmemizi sağlayan F-Ölçütü değerine göre ise DOS saldırılarında daha iyi performansa sahiptir. NSL-KDD veriseti ile yapılan testlerde ise destek vektör makinesi, kullanıcı hesabını yönetici hesabına yükseltme (U2R) saldırına karşı diğer saldırı türlerine göre daha iyi sınıflandırma başarına sahiptir. Sınıflandırıcı, duyarlılık yönünden normal davranışları algılamada, seçicilik yönünden U2R saldırılarında, kesinlik yönünden DOS saldırılarında ve seçicilik ile kesinliği beraber değerlendirmemizi sağlayan F-Ölçütü değerine göre ise DOS saldırılarında daha iyi performansa sahiptir.

7.3.3. KNN performans sınaması

K en yakın komşu algoritması kullanılarak yapılan testler sonucunda, sınıflandırıcının, saldırı tespit sistemlerindeki performansını değerlendirmemizi sağlayan başarım ölçütlerinin, saldırı tiplerine göre elde edilen değerleri Çizelge 7.8'de gösterildiği gibidir.

Çizelge 7.8. K en yakın komşu algoritması ile yapılan test sonuçları

Veriseti	Saldırı Tipi	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
KDD CUP99	Normal	0,998	0,997	0,998	0,991	0,994
	DOS	0,999	0,999	0,998	1	0,999
	R2L	0,999	0,796	1	0,911	0,85
	PROBE	0,999	0,942	1	0,987	0,964
	U2R	0,999	0,654	1	0,791	0,716
NSL-KDD	Normal	0,996	0,997	0,995	0,996	0,996
	DOS	0,999	0,999	0,999	0,998	0,998
	R2L	0,999	0,908	0,999	0,921	0,914
	PROBE	0,998	0,99	0,999	0,992	0,991
	U2R	0,999	0,404	0,999	0,636	0,494

Çizelge 7.8'e göre KDD CUP99 veriseti ile yapılan testlerde K en yakın komşu algoritması, DOS, U2R, PROBE ve R2L saldırını tespit etmede oldukça başarılı bir sınıflandırıcıdır. Duyarlılık yönünden DOS saldırılarında, seçicilik yönünden R2L, PROBE ve U2R saldırılarında, kesinlik yönünden DOS saldırılarında ve seçicilik ile kesinliği beraber değerlendirmemizi sağlayan F-Ölçütü değerine göre ise DOS saldırılarında daha iyi performansa sahiptir. NSL-KDD veriseti ile yapılan testlerde ise K en yakın komşu algoritmasının DOS, U2R ve R2L saldırılarını tespit etmede diğer saldırılara göre daha etkili olduğu gözlenmiştir. Duyarlılık yönünden DOS saldırılarında, seçicilik yönünden DOS, R2L, PROBE ve U2R saldırılarında, kesinlik yönünden DOS saldırılarında ve seçicilik ile kesinliği beraber değerlendirmemizi sağlayan F-Ölçütü değerine göre ise DOS saldırılarında daha iyi performansa sahiptir.

7.3.4. YSA performans sınaması

Yapay sinir ağları kullanılarak yapılan testler sonucunda, sınıflandırıcının, saldırı tespit sistemlerindeki performansını değerlendirmemizi sağlayan başarım ölçütlerinin, saldırı tiplerine göre elde edilen değerleri Çizelge 7.9'da gösterildiği gibidir.

Çizelge 7.9. Yapay sinir ağları ile yapılan test sonuçları

Veriseti	Saldırı Tipi	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
KDD CUP99	Normal	0,998	0,997	0,999	0,994	0,995
	DOS	0,999	1	0,999	1	1
	R2L	0,999	0,761	1	0,878	0,815
	PROBE	0,999	0,97	1	0,985	0,977
	U2R	0,999	0,654	1	0,723	0,687
NSL-KDD	Normal	0,992	0,996	0,987	0,989	0,992
	DOS	0,997	0,993	0,999	0,998	0,995
	R2L	0,997	0,693	0,999	0,871	0,772
	PROBE	0,998	0,988	0,999	0,99	0,989
	U2R	0,999	0,077	1	1	0,143

Çizelge 7.9'a göre KDD CUP99 veriseti ile yapılan testlerde yapay sinir ağları, DOS, U2R, PROBE ve R2L saldırını yüzde yüze yakın bir sınıflandırma başarısı ile tespit etmiştir. Sınıflandırıcı, duyarlılık yönünden DOS saldırılarında, seçicilik yönünden R2L, PROBE ve U2R saldırılarında, kesinlik yönünden DOS saldırılarında ve seçicilik ile kesinliği beraber değerlendirmemizi sağlayan F-Ölçütü değerine göre ise DOS saldırılarında daha iyi performansa sahiptir. NSL-KDD veriseti ile yapılan testlerde ise yapay sinir ağları, U2R saldırını yüzde yüze yakın bir sınıflandırma başarısı ile tespit etmiştir. Sınıflandırıcı, duyarlılık yönünden DOS saldırılarında, seçicilik yönünden U2R saldırılarında, kesinlik yönünden U2R saldırılarında ve seçicilik ile kesinliği beraber değerlendirmemizi sağlayan F-Ölçütü değerine göre ise DOS saldırılarında daha iyi performansa sahiptir.

7.3.5. Karar ağaçları performans sınaması

Karar ağaçları C4.5 algoritması kullanılarak yapılan testler sonucunda, sınıflandırıcının, saldırı tespit sistemlerindeki performansını değerlendirmemizi sağlayan başarım ölçütlerinin, saldırı tiplerine göre elde edilen değerleri Çizelge 7.10'da gösterildiği gibidir.

Çizelge 7.10. Karar ağaçları ile yapılan test sonuçları

Veriseti	Saldırı Tipi	Doğruluk	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
KDD CUP99	Normal	0,999	0,997	0,999	0,996	0,997
	DOS	0,999	1	1	1	1
	R2L	0,999	0,838	1	0,875	0,856
	PROBE	0,999	0,975	1	0,982	0,979
	U2R	0,999	0,654	1	0,708	0,68
NSL-KDD	Normal	0,998	0,998	0,998	0,998	0,998
	DOS	0,999	0,999	0,999	0,999	0,999
	R2L	0,999	0,956	0,999	0,982	0,969
	PROBE	0,999	0,994	0,999	0,995	0,995
	U2R	0,999	0,558	0,999	0,725	0,63

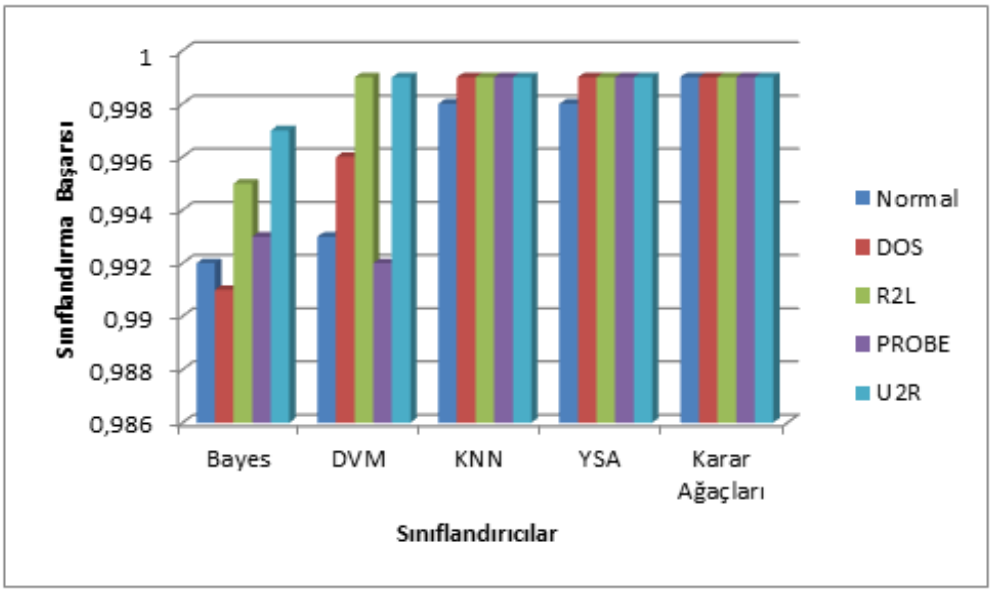
Çizelge 7.10'dan görüldüğü üzere KDD CUP99 veriseti ile yapılan testlerde karar ağaçları, DOS, U2R, PROBE ve R2L saldırını algılamada başarılı bir sınıflandırıcıdır. Sınıflandırıcı, duyarlılık yönünden DOS saldırılarında, seçicilik yönünden DOS, R2L, PROBE ve U2R saldırılarında, kesinlik yönünden DOS saldırılarında ve seçicilik ile kesinliği beraber değerlendirmemizi sağlayan F-Ölçütü değerine göre ise DOS saldırılarında daha iyi performansa sahiptir. NSL-KDD ile yapılan testlere göre ise karar ağaçları DOS, U2R, PROBE ve R2L saldırılarını algılamada aynı başarıya sahiptir. Sınıflandırıcı duyarlılık yönünden DOS saldırılarında, seçicilik yönünden DOS, R2L, PROBE ve U2R saldırılarında, kesinlik yönünden DOS saldırılarında ve F-Ölçütü değerine göre ise DOS saldırılarında daha iyi performansa sahiptir.

7.4. Yöntemlerin Kıyaslanması

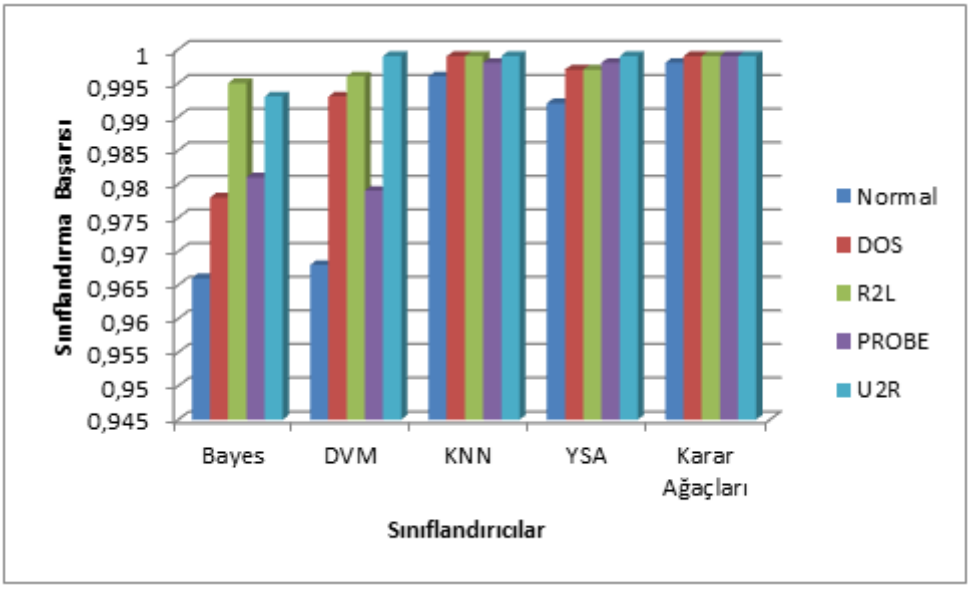
Sınıflandırıcıların saldırıları tespit etmedeki başarısı Çizelge 7.11'de gösterildiği gibidir.

Çizelge 7.11. Sınıflandırıcıların sınıflandırma başarısı (Accuracy)

Veriseti	Saldırı Tipi	Bayes	DVM	KNN	YSA	Karar Ağaçları
KDD CUP99	Normal	0,992	0,993	0,998	0,998	0,999
	DOS	0,991	0,996	0,999	0,999	0,999
	R2L	0,995	0,999	0,999	0,999	0,999
	PROBE	0,993	0,992	0,999	0,999	0,999
	U2R	0,997	0,999	0,999	0,999	0,999
NSL-KDD	Normal	0,966	0,968	0,996	0,992	0,998
	DOS	0,978	0,993	0,999	0,997	0,999
	R2L	0,995	0,996	0,999	0,997	0,999
	PROBE	0,981	0,979	0,998	0,998	0,999
	U2R	0,993	0,999	0,999	0,999	0,999



Şekil 7.3. KDD CUP99 veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırıları sınıflandırma başarısına (accuracy) göre karşılaştırılması

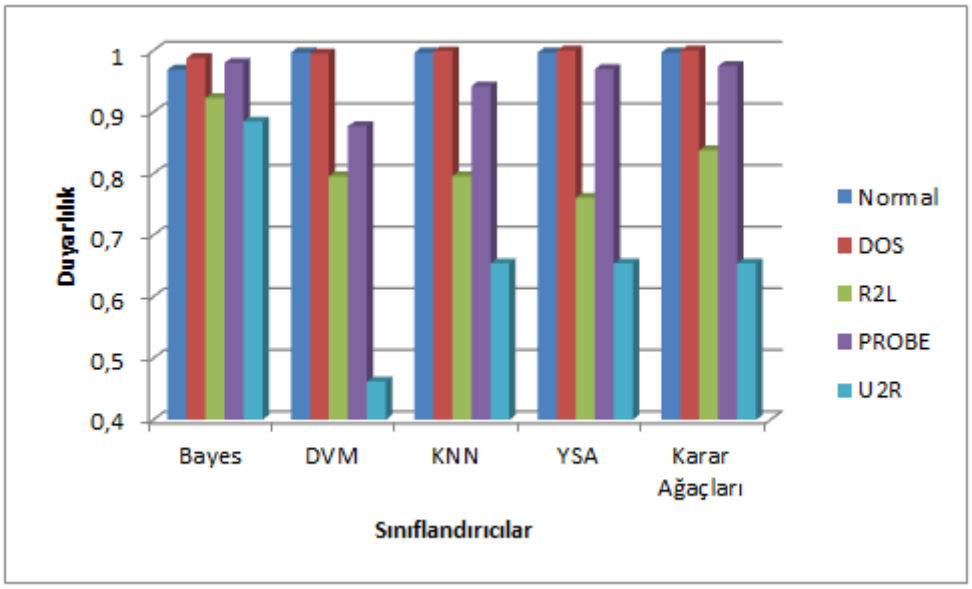


Şekil 7.4. NSL-KDD veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırıları sınıflandırma başarısına (accuracy) göre karşılaştırılması

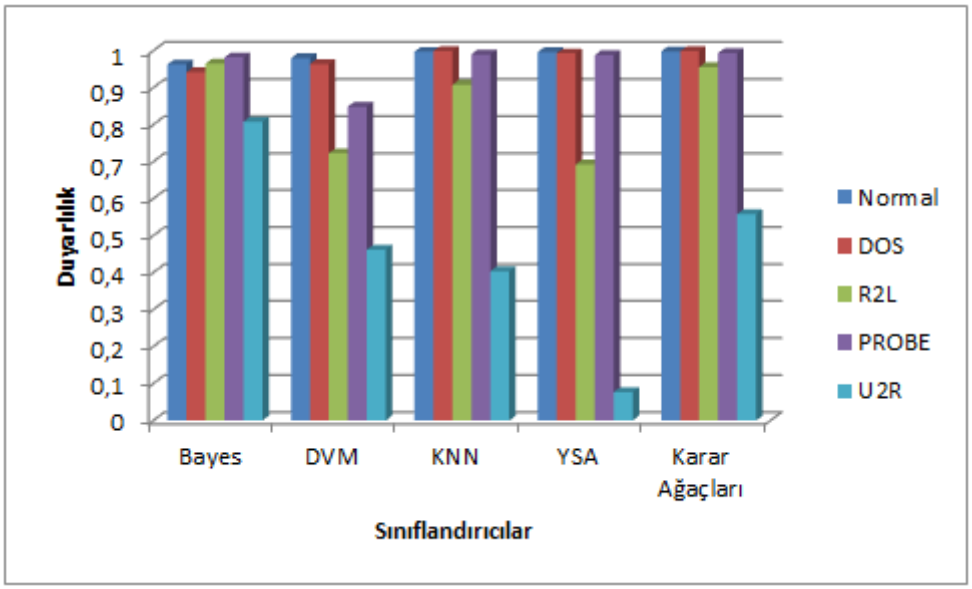
Doğruluk (accuracy), yapılan testler sonucunda, tahmin edilen değerlerin, gerçek değere ne kadar yakın olduğunun ifadesidir. Çizelge 7.11, Şekil 7.3 ve Şekil 7.4'de görüldüğü üzere KDD CUP99 ve NSL-KDD verisetlerine göre yapılan testler sonucunda normal davranışları ayırt etmede, karar ağaçları diğer sınıflandırıcılara göre daha başarılıdır. DOS ve R2L saldırılarının tespitinde KNN ve karar ağaçları %100'e yakın bir başarıya ulaşmıştır. PROBE saldırılarının doğru tespitinde karar ağaçları daha iyi sonuç vermektedir. U2R saldırılarında ise Bayes ağları hariç diğer sınıflandırıcılar daha başarılı sonuçlar vermiştir.

Çizelge 7.12. Sınıflandırıcıların duyarlılık (sensitivity/recall) sonuçları

Veriseti	Saldırı Tipi	Bayes	DVM	KNN	YSA	Karar Ağaçları
KDD CUP99	Normal	0,969	0,997	0,997	0,997	0,997
	DOS	0,988	0,995	0,999	1	1
	R2L	0,923	0,796	0,796	0,761	0,838
	PROBE	0,98	0,877	0,942	0,97	0,975
	U2R	0,885	0,462	0,654	0,654	0,654
NSL-KDD	Normal	0,963	0,98	0,997	0,996	0,998
	DOS	0,942	0,964	0,999	0,993	0,999
	R2L	0,965	0,723	0,908	0,693	0,956
	PROBE	0,982	0,849	0,99	0,988	0,994
	U2R	0,808	0,462	0,404	0,077	0,558



Şekil 7.5. KDD CUP99 veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırılara karşı duyarlılıklarının karşılaştırılması

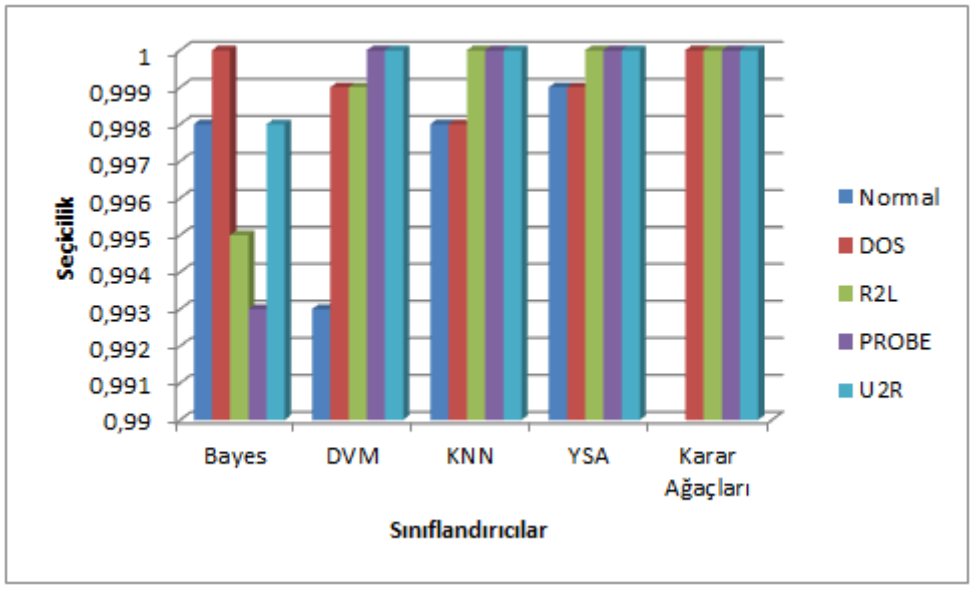


Şekil 7.6. NSL-KDD veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırılara karşı duyarlılıklarının karşılaştırılması

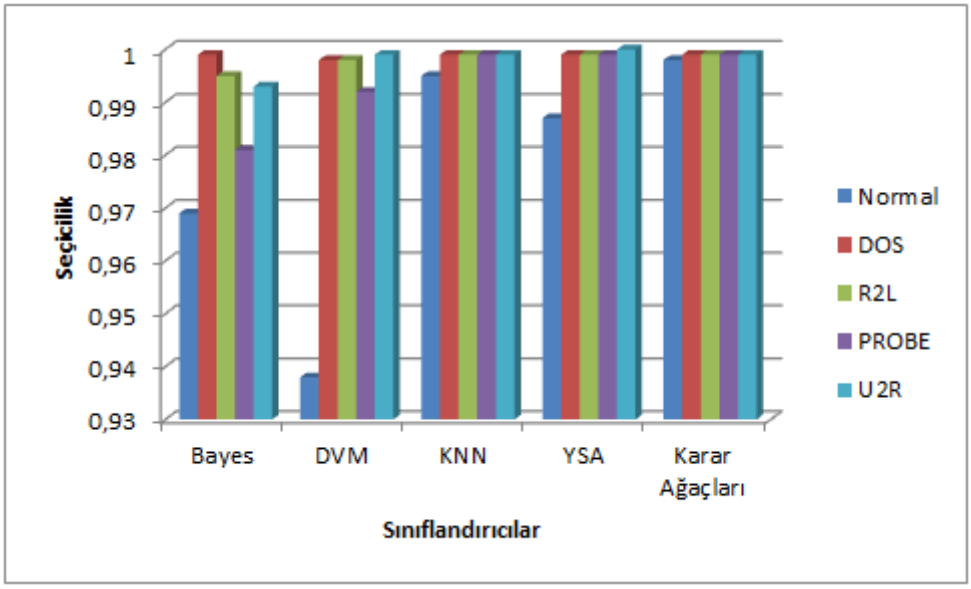
Çizelge 7.12, Şekil 7.5 ve Şekil 7.6'da sınıflandırıcıların her bir saldırı türü için duyarlılık değerleri gösterilmiştir. Duyarlılık (precision), yapılan test sonuçlarının birbirine ne ölçüde yakın olduğunun ifadesidir. İyi bir duyarlılık çoğunlukla iyi bir doğruluk derecesinin göstergesidir. Ancak bu her zaman geçerli değildir. Çok iyi bir duyarlılığı olduğu halde zayıf doğruluk dereceli ölçümler de olabilir. KDD CUP99 ve NSL-KDD verisetlerine göre yapılan testler sonucunda duyarlılık değerlerine baktığımızda, Normal davranışları algılamada karar ağaçları daha iyi bir sınıflandırıcıdır. DOS saldırılarında KNN ve karar ağaçları, R2L ve U2R saldırılarında Bayes, PROBE saldırılarında ise karar ağaçları daha iyi sınıflandırma işlemi yapmaktadırlar.

Çizelge 7.13. Sınıflandırıcıların seçicilik (specificity) sonuçları

Veriseti	Saldırı Tipi	Bayes	DVM	KNN	YSA	Karar Ağaçları
KDD CUP99	Normal	0,998	0,993	0,998	0,999	0,999
	DOS	1	0,999	0,998	0,999	1
	R2L	0,995	0,999	1	1	1
	PROBE	0,993	1	1	1	1
	U2R	0,998	1	1	1	1
NSL-KDD	Normal	0,969	0,938	0,995	0,987	0,998
	DOS	0,999	0,998	0,999	0,999	0,999
	R2L	0,995	0,998	0,999	0,999	0,999
	PROBE	0,981	0,992	0,999	0,999	0,999
	U2R	0,993	0,999	0,999	1	0,999



Şekil 7.7. KDD CUP99 veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırıları seçiciliğinin karşılaştırılması

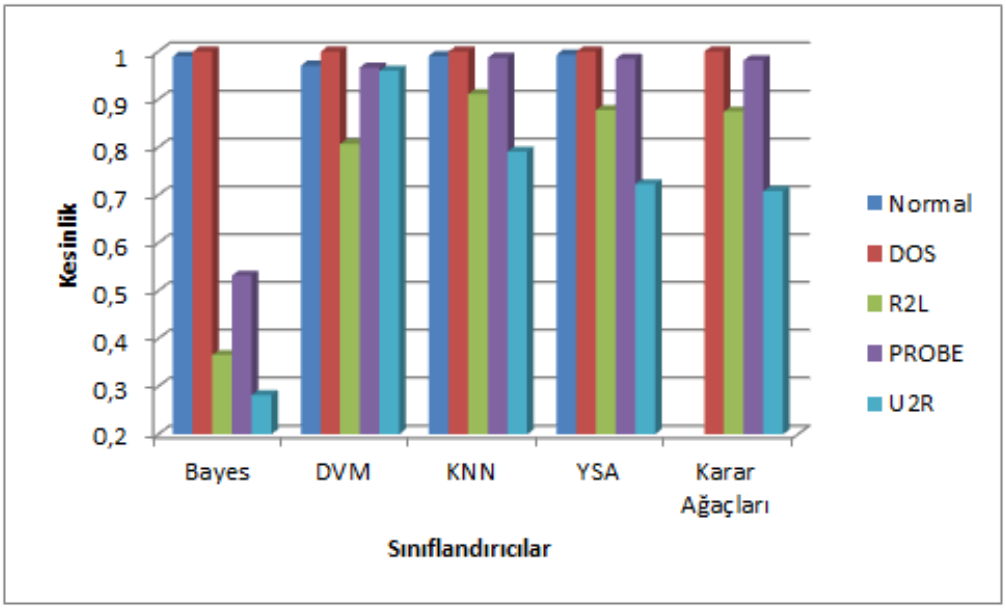


Şekil 7.8. NSL-KDD veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırıları seçiciliğinin karşılaştırılması

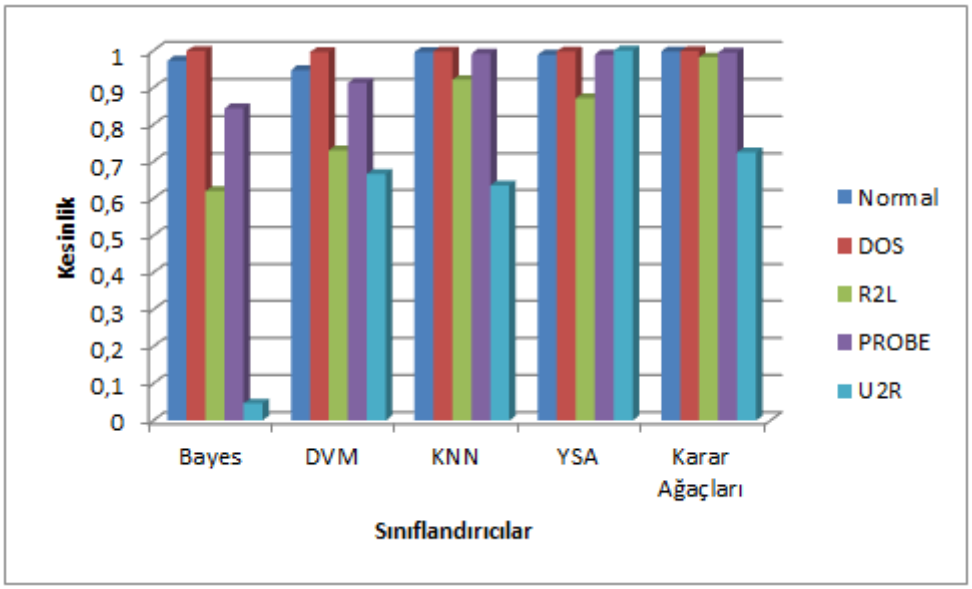
Seçicilik, yapılan test sonuçlarında gerçekteki değerleri tahmin edebilme yeteneğidir. KDD CUP99 ve NSL-KDD verisetlerine göre yapılan testler sonucunda Çizelge 7.13, Şekil 7.7 ve Şekil 7.8 ile gösterilen seçicilik değerlerine baktığımızda normal davranışlarda karar ağaçları, DOS saldırılarında DVM hariç diğer sınıflandırıcılar, R2L ve PROBE saldırılarında KNN, YSA ve karar ağaçları, U2R saldırılarında ise YSA seçicilik yönünden daha iyi sınıflandırıcılardır.

Çizelge 7.14. Sınıflandırıcıların kesinlik (precision) sonuçları

Veriseti	Saldırı Tipi	Bayes	DVM	KNN	YSA	Karar Ağaçları
KDD CUP99	Normal	0,99	0,971	0,991	0,994	0,996
	DOS	1	1	1	1	1
	R2L	0,366	0,807	0,911	0,878	0,875
	PROBE	0,532	0,967	0,987	0,985	0,982
	U2R	0,282	0,96	0,791	0,723	0,708
NSL-KDD	Normal	0,973	0,948	0,996	0,989	0,998
	DOS	0,999	0,996	0,998	0,998	0,999
	R2L	0,62	0,731	0,921	0,871	0,982
	PROBE	0,844	0,913	0,992	0,99	0,995
	U2R	0,046	0,667	0,636	1	0,725



Şekil 7.9. KDD CUP99 veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırılara göre kesinlik değerlerinin karşılaştırılması

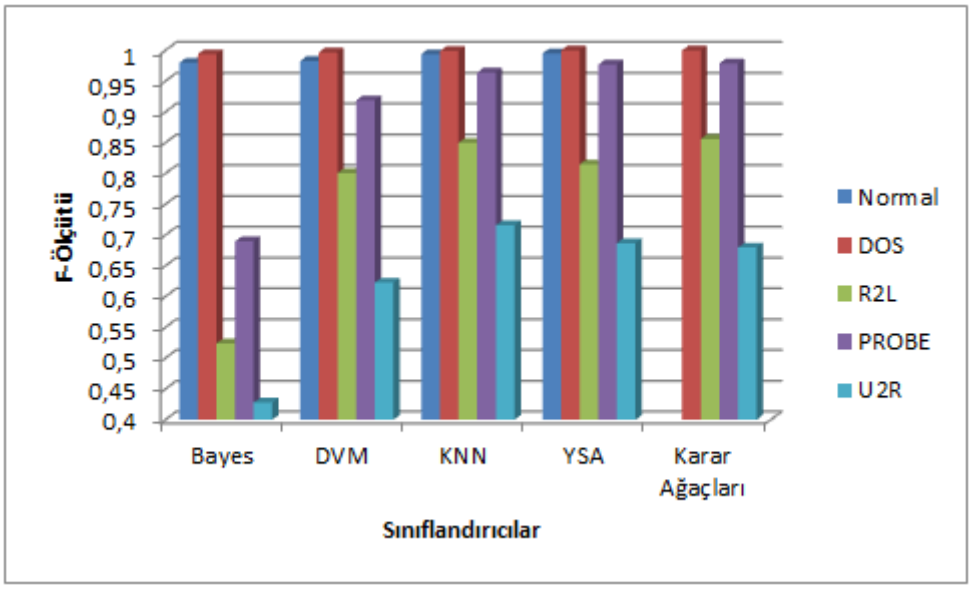


Şekil 7.10. NSL-KDD veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırılara göre kesinlik değerlerinin karşılaştırılması

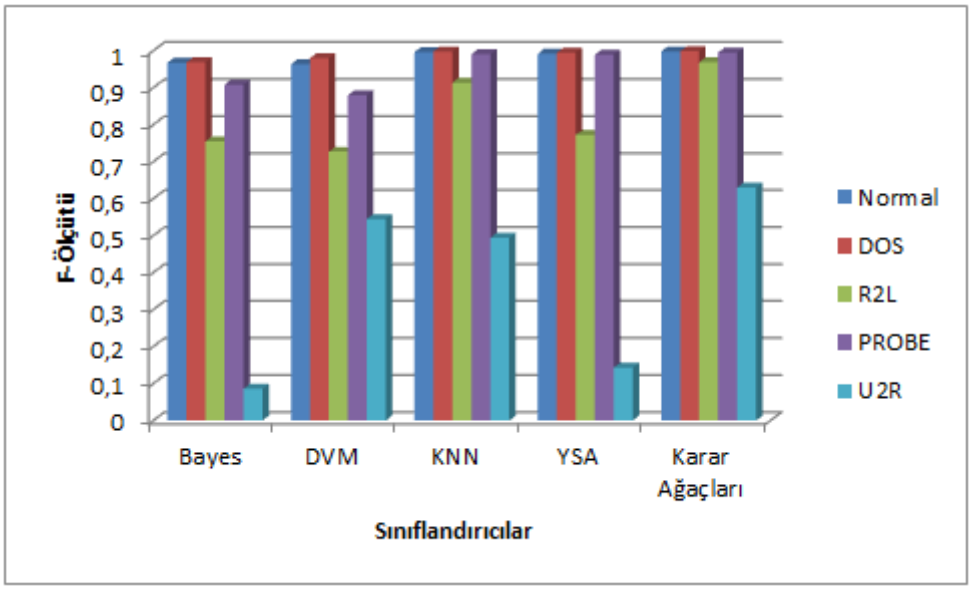
Kesinlik (Precision), getirilen bilginin ne kadarının, istenilen bilgiyle ilgili olduğunu gösterir ve getirilen bilgideki doğru sonuçların, getirilen bilginin tamamına oranı olarak hesaplanır. Çizelge 7.14, Şekil 7.9 ve Şekil 7.10'daki kesinlik değerlerine baktığımızda KDD CUP99 ve NSL-KDD verisetlerine göre yapılan testler sonucunda normal davranışlarda karar ağaçları, DOS saldırılarında Bayes ve karar ağaçları, R2L saldırılarında karar ağaçları, PROBE ve U2R saldırılarında ise YSA kesinlik yönünden daha iyi sınıflandırıcılardır.

Çizelge 7.15. Sınıflandırıcıların F-ölçütü (F-Measure) sonuçları

Veriseti	Saldırı Tipi	Bayes	DVM	KNN	YSA	Karar Ağaçları
KDD CUP99	Normal	0,98	0,983	0,994	0,995	0,997
	DOS	0,994	0,997	0,999	1	1
	R2L	0,524	0,801	0,85	0,815	0,856
	PROBE	0,69	0,919	0,964	0,977	0,979
	U2R	0,428	0,623	0,716	0,687	0,68
NSL-KDD	Normal	0,968	0,964	0,996	0,992	0,998
	DOS	0,969	0,98	0,998	0,995	0,999
	R2L	0,755	0,727	0,914	0,772	0,969
	PROBE	0,908	0,88	0,991	0,989	0,995
	U2R	0,086	0,545	0,494	0,143	0,63



Şekil 7.11. KDD CUP99 veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırılara göre F-Ölçütü değerlerinin karşılaştırılması



Şekil 7.12. NSL-KDD veriseti ile yapılan testler sonucunda sınıflandırıcıların saldırılara göre F-Ölçütü değerlerinin karşılaştırılması

Kesinlik ve duyarlılık ölçütlerini tek başına değerlendirmek yanlış sonuçlara götürebileceğinden bu iki ölçütü beraber değerlendirmek için, her iki değerin harmonik ortalaması olan F-Ölçütüne bakmak gerekir. Çizelge 7.15, Şekil 7.11 ve Şekil 7.12'den KDD CUP99 ve NSL-KDD verisetlerine göre yapılan testler sonucunda elde edilen F-ölçütü değerlerine baktığımızda normal davranışları algılamada ve DOS, PROBE, U2R, R2L saldırılarında karar ağaçları diğer sınıflandırıcılara göre daha iyi performans göstermektedir.

Çizelge 7.16. Sınıflandırıcılar için CPU zamanı

Veriseti		Bayes	DVM	KNN	YSA	Karar Ağaçları
KDD CUP99	Eğitim	1,44 Sn	20,87 Sn	0,03 Sn	8,47 Dk	3,44 Sn
	Eğitim+Test	16,56 Sn	2,53 Dk	19,09 Dk	1,28 Saat	29,16 Sn
	Toplam	18 Sn	3,01 Dk	19,10 Dk	1,37 Saat	33 Sn
NSL-KDD	Eğitim	8,26 Sn	11,36 Dk	0,03 Sn	21,44 Dk	45,21 Sn
	Eğitim+Test	1.13 Dk	2,45 Saat	43,35 Dk	3,43 Saat	5,36 Dk
	Toplam	1,22 Dk	2,56 Saat	43,36 Dk	4,04 Saat	6,21 Dk

Sınıflandırıcıları işlem zamanına göre değerlendirdiğimizde Çizelge 7.16'dan görüldüğü gibi en hızlı sınıflandırıcı Bayes ağları iken onu sırasıyla karar ağaçları, KNN, DVM ve YSA takip etmektedir. YSA ile sınıflandırma işlemi diğer sınıflandırıcılara göre oldukça fazla işlem zamanı gerektirmektedir.

8. SONUÇ VE ÖNERİLER

Saldırı tespit sistemleri, halen üzerinde araştırma yapılması gereken önemli bir çalışma alanıdır. Daha etkin saldırı tespit sistemi tasarlamak için makine öğrenme teknikleri sıklıkla kullanılmaktadır. Yapılan çalışmada KDD CUP99 ve NSL-KDD verisetleri kullanılarak, makine öğrenmesi tekniklerinden Bayes ağları, destek vektör makinesi, K en yakın komşu algoritması, yapay sinir ağları ve karar ağaçlarının, işlem zamanı, sınıflandırma başarısı, duyarlılık, seçicilik, kesinlik ve F-ölçütü yönünden saldırı tespit sistemlerindeki performansı incelenmiştir. Yapılan testler sonucunda, sınıflandırıcıları, doğruluk, duyarlılık, seçicilik, kesinlik ve F-ölçütü sonuçlarına göre değerlendirdiğimizde şu sonuçlar çıkmıştır.

Sınıflandırıcıları, sınıflandırma başarısına göre değerlendirdiğimizde, normal davranışları ayırt etmede, karar ağaçları diğer sınıflandırıcılara göre daha başarılıdır. DOS saldırılarının tespitinde KNN, karar ağaçları ve YSA %100'e yakın bir başarıya ulaşmıştır. PROBE saldırılarının doğru tespitinde KNN, YSA ve karar ağaçları daha iyi sonuç vermektedir. R2L ve U2R saldırılarında ise Bayes ağları hariç diğer sınıflandırıcılar daha başarılı sonuçlar vermiştir.

Duyarlılık değerlerine baktığımızda, normal davranışları algılamada DVM, KNN, YSA ve karar ağaçları daha iyi bir sınıflandırıcıdır. DOS saldırılarında YSA ve karar ağaçları, R2L, PROBE ve U2R saldırılarında ise Bayes ağları daha iyi sınıflandırma işlemi yapmaktadırlar.

Seçicilik değerlerine göre sınıflandırıcıları karşılaştırdığımızda, normal davranışları doğru tespit etmede karar ağaçları, DOS saldırılarında Bayes ve karar ağaçları, R2L saldırılarında KNN, YSA ve karar ağaçları, PROBE ve U2R saldırılarında ise DVM, KNN, YSA ve karar ağaçları seçicilik yönünden daha iyi sınıflandırıcılardır.

Kesinlik ölçütüne göre, normal davranışlarda Karar ağaçları, DOS saldırılarında incelenen tüm sınıflandırıcılar, PROBE ve R2L saldırılarında KNN, U2R saldırılarında ise DVM daha iyi sınıflandırıcıdır.

F-ölçütü, yanlış pozitif ve yanlış negatif değerlerini de içerdiğinden, başarım ölçütü olarak doğruluk ölçütüne göre daha güvenilir sonuç verir ve algoritmaların birbiriyle karşılaştırılma analizinde kullanılmaktadır. F-ölçütü değerlerine baktığımızda normal davranışları doğru algılamada karar ağaçları, DOS saldırılarında YSA ve karar ağaçları, PROBE ve R2L saldırılarında karar ağaçları, U2R saldırılarında ise KNN daha iyi sınıflandırıcılardır.

STS'lerde saldırıları kısa sürede tespit edip müdahale edebilmek çok kritik bir durumdur. Bu nedenle sınıflandırıcılar saldırıları sınıflandırırken harcadığı işlem zamanına göre de değerlendirilmiştir. Sınıflandırıcıları, işlem zamanına göre değerlendirdiğimizde Bayes ağları ve karar ağaçlarının diğer sınıflandırıcılara göre oldukça hızlı olduğu görülmüştür.

Aşağıda vurgulanan bulgular, gelecekte makine öğrenme teknikleri ile daha etkin saldırı tespit sistemi tasarlamak isteyen araştırmacılara faydalı olabilir.

- Tercih edilen sınıflandırıcı, geliştirilen STS'nin başarısında çok önemli rol oynar. YSA en sık kullanılan ve aynı zamanda en yüksek başarı oranını elde eden sınıflandırıcıdır. Ancak DVM da bazı saldırı tiplerinde etkin çözümler üretebilmektedir.
- Sistem eğitiminde ve test aşamasında kullanılan veriseti önemlidir. KDD CUP99 sıklıkla tercih edilen açık (public) veriseti olmuştur. Elbette araştırmacılar kendi verilerini toplayabilir. Ancak bu hem maliyetli hem de ortaya konulan çalışmanın kıyaslanması açısından bazı sorunları gündeme getirecektir.
- Kullanılan veriseti kadar, sistem eğitimi için ayrılan eğitim verisi de önemlidir. Yeteri kadar eğitim verisi sistem eğitimi aşamasında kullanılmalıdır.

Oluşturulan STS'nin başarısını ölçmek için yeteri kadar test verisine ihtiyaç vardır. Yeterli miktarda test verisi, sistem başarısının doğru ölçülmesinde önemli rol oynar. Ancak yine de önerilen STS ile elde edilmiş test sonuçları, gerçek ortamda aynı başarıyı vereceği anlamına gelmemektedir.

KAYNAKLAR

1. İnternet: Akamai's [state of the internet]/security Q4 2015 report, URL: **http://www.webcitation.org/query?url=https%3A%2F%2Fwww.akamai.com%2Fus%2Fen%2Fmultimedia%2Fdocuments%2Freport%2Fq4-2015-state-of-the-internet-security-report.pdf&date=2016-07-14** Son Erişim Tarihi: 14.07.2016.

2. Jemili, F., Zaghdoud, M., and Ben Ahmed, M. (2007). *A Framework for an Adaptive Intrusion Detection System using Bayesian Network*. IEEE Intelligent and Security Informatics, 66-70.

3. Farid, D., and Rahman, M. (2010). Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm. ***Journal of computers***, 5(1), 23-31.

4. Zhang, Y., and Zhu, Y. (2010). *Application of Improved Support Vector Machines in Intrusion Detection*. 2nd International Conference on e-Business and Information System Security, 1-4.

5. Wang, J., Li, T., and Ren, R. (2010). *A Real Time IDSs Based on Artificial Bee Colony Support Vector Machine Algorithm*. Third International Workshop on Advanced Computational Intelligence, 91-96.

6. Mu, Q., Chen, Y., and Zhang, Y. (2012). *Incremental SVM Algorithm to Intrusion Detection Base on Boundary Areas*. International Conference on Systems and Informatics, 198-201.

7. Bahrololum, M., Salahi, E., and Khalegni, M. (2009). *Machine Learning Techniques for feature Reduction in Intrusion Detection Systems: A Comparison*. Fourth International Conference on Computer Sciences and Convergence Information Technology, 1091-1095.

8. Alazab, A., Hobbs, M., Abawajy, J., and Alazab, M. (2012). *Using Feature Selection for Intrusion Detection System*. International Symposium on Communications and Information Technologies (ISCIT), 296-301.

9. Sharma, V., and Nema, A. (2013). *Innovative Genetic approach For Intrusion Detection by Using Decision Tree*. International Conference on Communication Systems and Network Technologies (CSNT), 418-422.

10. Liu, G., Yi, Z., and Yang, S. (2007). A Hierarchical Intrusion Detection Model Based on the PCA Neural Networks. ***Neurocomputing***, 70, 1561-1568.

11. Gong, X., and Guan, X. (2012). *Intrusion Detection Model Based on the Improved Neural Network and Expert System*. IEEE Symposium on Electrical & Electronics Engineering (EEESYM), 191-193.

12. Lin, W.-C., Ke, S.-W., and Tsai, C.-F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. ***Knowledge-Based Systems***, 78, 13-21.

13. Tsai, C.-F., and Lin, C.-Y. (2010). A triangle area based nearest neighbors approach to intrusion detection. ***Pattern Recognition***, 43, 222-229.

14. Xiang, C., Yong, P., and Meng, L. (2008). Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. ***Pattern Recognition Letters***, 29, 918-924.

15. Jemili, F., Zaghdoud, M., and Ben Ahmed, M. (2009). *Intrusion Detection based on Hybrid Propagation in Bayesian Networks*. IEEE International Conference on Intelligence and Security Informatics, 137-142.

16. Muda, Z., Yassin, W., Sulaiman, M., and Udzir, N. (2011). *Intrusion Detection based on K-Means Clustering and Naïve Bayes Classification*. IAS 2011, 192-197.

17. Fan, W., Bouguila, N., and Ziou, D. (2011). *Unsupervised Anomaly Intrusion Detection via Localized Bayesian Feature Selection*. 11th IEEE International Conference on Data Mining, 1032-1037.

18. Altwaijry, H., and Algarny, S. (2012). Bayesian Based Intrusion Detection System. ***Journal of King Saud University - Computer and Information Sciences***, 24(1), 1-6.

19. Mukherjee, S., and Sharma, N. (2012). Intrusion Detection Using Naive Bayes Classifier with Feature Reduction. ***Procedia Technology***, 4, 119-128.

20. Koc, L., Mazzuchi, T., and Sarkani, S. (2013). A Network Intrusion Detection System Based on a Hidden Naïve Bayes Multiclass Classifier. ***Expert Systems with Applications***, 39, 13492-13500.

21. Pei-li, Q., and Shi-feng, C. (2009). *Intrusion Detection System Technique Based on BP SVM*. International Conference on Management and Service Science, 1-3.

22. Shon, T., and Moon, J. (2007). A Hybrid Machine Learning Approach to Network Anomaly Detection. ***Information Sciences***, 17, 3799-3821.

23. Zhou, H., Meng, X., and Zhang, L. (2007). *Application of Support Vector Machine and Genetic Algorithm to Network Intrusion Detection*. International Conference on Wireless Communications, Networking and Mobile Computing, 2267-2269.

24. Li, Y., and Wang, Z. (2007). *An Intrusion Detection Method Based on SVM and KPCA*. International Conference on Wavelet Analysis and Pattern Recognition, 1462-1466.

25. Li, H., and Wang, J. (2007). *Intrusion Detection System by Integrating PCNN and Online Robust SVM*. International Conference on Network and Parallel Computing, 250-254.

26. Ding, X., Zhang, G., Ke, Y., Ma, B., and Li, Z. (2008). *High Efficient Intrusion Detection Methodology with Twin Support Vector Machines*. International Symposium on Information Science and Engineering, 560-564.

27. Li, Y., Wang, Z., and Ma, Y. (2008). *An Intrusion Detection Method Based on KICA and SVM*. 7th World Congress on Intelligent Control and Automation, 2141-2144.

28. Ma, J., Liu, X., and Liu, S. (2008). *A New Intrusion Detection Method Based on BPSO-SVM*. International Symposium on Computational Intelligence and Design, 473-477.

29. Chen, Z., and Zhang, G. (2009). *Support Vector Machines Improved by Artificial Immunisation Algorithm for Intrusion Detection*. International Conference on Information Engineering and Computer Science, 1-4.

30. Du, H., Teng, S., Fu, X., Zhang, W., and Pu, Y. (2009). *A Cooperative Intrusion Detection System Based on Improved Parallel SVM*. Joint Conferences on Pervasive Computing, 515-518.

31. Liu, H., Jian, Y., and Liu, S. (2010). *A New Intelligent Intrusion Detection Method Based on Attribute Reduction and Parameters Optimization of SVM*. Second International Workshop on Education Technology and Computer Science, 1, 202-205.

32. Xiaoqing, G., Hebin, G., and Luyi, C. (2010). *Network Intrusion Detection Method Based on Agent and SVM*. The 2nd IEEE International Conference on Information Management and Engineering, 399-402.

33. Horng, S., Su, M., Chen, Y., Kao, T., Chen, R., Lai, J., and Perkasa, C. (2011). A Novel Intrusion Detection System Based on Hierarchical Clustering and Support Vector Machines. ***Expert Syst. Appl***, 38(1), 306-313.

34. Somwang, P., and Lilakiatsakun, W. (2011). *Computer Network Security Based On Support Vector Machine Approach*. 11th International Conference on Control, Automation and Systems, 155-160.

35. Song, G., Guo, J., and Nie, Y. (2011). *An Intrusion Detection Method based on Multiple Kernel Support Vector Machine*. International Conference on Network Computing and Information Security, 119-123.

36. Ning, L., and Jianhua, Z. (2012). *Intrusion Detection Research Based on Improved PSO and SVM*. International Conference on Automatic Control and Artificial Intelligence, 1263-1266.

37. Yang, X., and Yilai, Z. (2012). *An Intelligent Anomaly Analysis for Intrusion Detection based on SVM*. International Conference on Computer Science and Information Processing, 739-742.

38. Chandrasekhar, A., and Raghuveer, K. (2013). *Intrusion Detection Technique by Using k Means, Fuzzy Neural Network and SVM Classifiers*. International Conference on Computer Communication and Informatics, 1-7.

39. Peddabachigari, S., Abrahamb, A., Grosanc, C., and Thomas, J. (2007). Modeling Intrusion Detection System Using Hybrid Intelligent Systems. ***Journal of Network and Computer Applications***, 30, 114-132.

40. Leet, J., Leet, J. H., Sohn, S. G., and Ryu, J. H. (2008). *Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System*. 10th International Conference on Advanced Communication Technology, 1170-1175.

41. Sheen, S., and Rajesh, R. (2008). *Network Intrusion Detection Using Feature Selection and Decision Tree Classifier*. IEEE Region 10 Conference, TENCON, 1-4.

42. Wu, S. Y., and Yen, E. (2009). Data mining-based intrusion detectors. ***Expert Systems with Applications***, 36(3), 5605-5612.

43. Hong, D., and Haibo, L. (2009). *A Lightweight Network Intrusion Detection Model Based on Feature Selection*. 15th IEEE Pacific Rim International Symposium on Dependable Computing, 165-168.

44. Liu, Y., Li, N., Shi, L., and Li, F. (2010). *An Intrusion Detection Method Based on Decision Tree*. International Conference on E-Health Networking, Digital Ecosystems and Technologies. 1, 232-235.

45. Sangkatsanee, P., Wattanapongsakorn, N., and Charnsripinyo, C. (2011). Practical Real-Time Intrusion Detection Using Machine Learning Approaches. ***Computer Communications***, 34, 2227-2235.

46. Kumar, M., Hanumanthappa, M., and Kumar, T. V. (2012). *Intrusion Detection System Using Decision Tree Algorithm*. 14th International Conference on Communication Technology (ICCT), 629-634.

47. Deng, H. R., and Wang, Y. H. (2007). *An Artificial-Neural Network-Based Multiple Classifiers Intrusion Detection System*. Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, 2, 683-686.

48. Yu, L., Chen, B., and Xiao, J. (2007). *An Integrated System of Intrusion Detection Based on Rough Set and Wavelet Neural Network*. Third International Conference on Natural Computation, 194-199.

49. Kumar, P. G., and Devaraj, D. (2007). *Network Intrusion Detection Using Hybrid Neural Networks*. *I*nternational Conference on Signal Processing, Communications and Networking, 563-569.

50. Beghdad, R. (2008). Critical Study of Neural Networks in Detecting Intrusions. ***Computers & Security***, 27, 168-175.

51. Powers, S. T., and He, J. (2008). A Hybrid Artificial Immune System and Self Organising Map for Network Intrusion Detection. ***Information Sciences***, 178, 3024-3042.

52. Zhou, T. J., and Yang, L. (2008). *The Research of Intrusion Detection Based on Genetic Neural Network*. International Conference on Wavelet Analysis and Pattern Recognition, 276-281.

53. Karimi, H., Montazeri, M. A., and Jazi, M. D. (2008). *A New Approach for Detecting Intrusions Using Jordan/Elman Neural Networks*. First International Conference on Complexity and Intelligence of the Artificial and Natural Complex Systems. Medical Applications of the Complex Systems. Biomedical Computing, CANS'08, 62-68.

54. Han, X. (2009). An Improved Intrusion Detection System Based on Neural Network. ***Intelligent Computing and Intelligent Systems***, 1, 887-890.

55. Tong, X., Wang, Z., and Yu, H. (2009). A Research Using Hybrid RBF/Elman Neural Networks for Intrusion Detection System Secure Model. ***Computer Physics Communications***, 180, 1795-1801.

56. Poojitha, G., Kumar, K. N., and Reddy, P. J. (2010). *Intrusion Detection Using Artificial Neural Network*. Second International conference on Computing, Communication and Networking Technologies, 1-7.

57. Wang, G., Hao, J., Mab, J., and Huang, L. (2010). A New Approach to Intrusion Detection Using Artificial Neural Networks and Fuzzy Clustering. ***Expert Systems with Applications***, 37, 6225-6232.

58. Xia, D. X., Yang, S. H., and Li, C. G. (2010). *Intrusion Detection System based on Principal Component Analysis and Grey Neural Networks*. Second International Conference on Networks Security, Wireless Communications and Trusted Computing, 142-145.

59. Huang, W., and Ju, L. (2010). *Intrusion Detection Method Based On Sparse Neural Network*. International Conference on Multimedia Technology (ICMT), 1-3.

60. Govindarajan, M., and Chandrasekaran, R. M. (2011). Intrusion Detection Using Neural Based Hybrid Classification Methods. ***Computer Networks***, 55, 1662-1671.

61. Xiangmei, L., and Zhi, Q. (2011). *The Application of Hybrid Neural Network Algorithms in Intrusion Detection System*. International Conference on E -Business and E -Government (ICEE), 1 – 4.

62. Zhang, B. (2011). *A Heuristic Genetic Neural Network for Intrusion Detection*. International Conference on Internet Computing and Information Services (ICICIS), 510-513.

63. Devaraju, S., and Ramakrishnan, S. (2011). *Performance Analysis of Intrusion Detection System Using Various Neural Network Classifiers*. International Conference on Recent Trends in Information Technology (ICRTIT), 1033-1038.

64. Ippoliti, D., and Zhou, X. (2012). A-GHSOM: An Adaptive Growing Hierarchical Self Organizing Map for Network Anomaly Detection. ***Parallel Distribution. Computation.***, 72, 1576-1590.

65. Jianga, X., Liub, K., Yana, J., and Chen, W. (2012). Application of Improved SOM Neural Network in Anomaly Detection. ***Physics Procedia***, 33, 1093-1099.

66. Srivastav, N., and Challa, R. K. (2013). *Novel Intrusion Detection System integrating Layered Framework with Neural Network*. 3rd International Advance Computing Conference (IACC), 682-689.

67. Ning, L. (2013). *Network Intrusion Classification Based on Probabilistic Neural Network*. International Conference on Computational and Information Sciences, 57-59.

68. Zhang, B., and Saeed, X. J. (2009). A Joint Evolutionary Neural Network for Intrusion Detection. ***Information Engineering and Computer Science***, 1-4.

69. Canbek, G., ve Sağıroğlu, Ş. (2006). Bilgi, bilgi güvenliği ve süreçleri üzerine bir inceleme. ***Politeknik Dergisi***, 9(9), 165-174.

70. McHugh, J. (2001). Intrusion and intrusion detection. ***International Journal of Information Security***, 1(1), 14-35.

71. Co, J. (1980). *Computer Security Threat Monitoring and Surveillance.* Pennsylvania: James P. Anderson Company, Fort Washington.

72. Kendall, K. (1999). *Database of Computer Attacks for the Evaluation of Intrusion Detection Systems*. MIT Department of Electrical Engineering and Computer Science.

73. Carl, G., Kesidis, G., Brooks, R., and Suresh, R. (2006). Denial of Service Attack Detection Techniques. ***Internet Computing,*** 10(1), 82-89**.**

74. Specht, S., and Lee, R. (2004). *Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures*. ISCA PDCS, 543-550.

75. Gezgin, D., ve Buluş, E. (2013). Kablosuz Ağlar için bir DoS Saldırısı Tasarımı. ***Internatıonal Journal of Informatıcs Technologıes***, 6(3), 17-23.

76. Ashoor, A., and Gore, S. (2011). *Intrusion Detection System (IDS): Case Study*. International Conference on Advanced Materials Engineering. Singapore.

77. Stenico, J., and Ling, L. (2014). *Network Traffic Monitoring and Analysis*. A.-S. Pathan (Dü.) içinde, The State of the Art in Intrusion Prevention and Detection, 23-46. CRC Press.

78. Şahin, O. (2008). Sınır Güvenliği. *Tübitak-UEKAE*, Ankara.

79. Axelsson, S. (2000). Intrusion Detection Systems: A Survey and Taxonomy. Technical Report.

80. Michie, D., Spiegelhalter, D., and Taylor, C. (1994). *Machine Learning Neural and Statistical Classification.* New York: Ellis Horwood Limited.

81. Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. ***ACM Computing Surveys (CSUR)***, 34(1), 1-47.

82. Anderson, J., Michalski, R., and Mitchell, T. (1983). *Machine learning: An artificial intelligence approach.* M. Kaufmann.

83. Nguyen, T., and Armitage, G. (2008). A Survey of Techniques for Internet Traffic Classification Using Machine Learning. ***IEEE Communications Surveys and Tutorials***, 10(4), 56-76.

84. Domingos, P., and Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. ***Springer*** ,29, 103-130.

85. Çinicioğlu, E. N., Atalay, M., ve Yorulmaz, H. (2013). Trafik Kazaları Analizi için Bayes Ağları Modeli. ***Bilişim Teknolojileri Dergisi***, 6(2).

86. Vapnik, V. (1998). *Statistical Learning Theory.* New York: John Wiley.

87. Ayhan, S., ve Erdoğmuş, Ş. (2014). Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü İçin Çekirdek Fonksiyonu Seçimi. ***Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi***, 9(1), 175-198.

88. Kavzoğlu, T., ve Çölkesen, İ. (2010). Destek Vektör Makineleri ile Uydu Görüntülerinin Sınıflandırılmasında Kernel Fonksiyonlarının Etkilerinin İncelenmesi. ***Harita Dergisi***, 144(7), 73-82.

89. Cheng, X., Liu, B.-X., Li, K., .and Yan, J. (2009). *Intrusion Detection System Based on KNN-MARS*. WRI World Congress on Software Engineering, 1, 392-396.

90. Liao, Y., and Vemuri, V. R. (2002). Use of K-Nearest Neighbor classifier for intrusion detection. ***Computers & Security***, 21(5), 439-448.

91. Adetunmbi, A. O., Falaki, S. O., Adewale, O. S., and Alese, B. K. (2008). Network intrusion detection based on rough set and k-nearest neighbour. ***International Journal of Computing and ICT Research***, *2*(1), 60-66.

92. Bitter, C., Elizondo, D. A., and Watson, T. (2010). *Application of artificial neural networks and related techniques to intrusion detection*. The 2010 International Joint Conference on Neural Networks (IJCNN), 1-8.

93. Haykin, S., and Network, N. (2004). *A comprehensive foundation.* Neural Networks.

94. Lippmann, R. P. (1987). An Intoduction to Computing with Neural Nets. ***IEEE Acoustic Speech and Signal Processing***, 4(2), 4-22.

95. Wu, S., and Banzhaf, W. (2010). The Use of Computational Intelligence in Intrusion Detection Systems:A Review. ***Applied Soft Computing***, 10(1), 1-35.

96. Witten, I., and Frank, E. (2011). *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition).* Yer: Amerika, Morgan Kaufmann Publication.

97. Dunham, M. (2003). *Data Mining Introductory and Advanced Topics* . Yer: Amerika, Prentice Hall Pearson Education Inc.

98. Emel, G. G., ve Taşkın, Ç. (2005). Veri Madenciliğinde Karar Ağaçları ve Bir Satış Analizi Uygulaması. ***Eskişehir Osmangazi Üniversitesi Sosyal Bilimler Dergisi***, 6(2).

99. Kaya, Ç., ve Yıldız, O. (2014). Makine öğrenmesi teknikleriyle saldırı tespiti: Karşılaştırmalı analiz. ***Marmara Fen Bilimleri Dergisi***, 3, 89-104.

100. Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. (2009). *A detailed analysis of the KDD CUP 99 data set*. 2009 IEEE International Conference Computation Intelligence Security Defense Application., 53-58.

101. Can, E. (2007). *Gerçek Zamanlı Veriler Yardımıyla Karar Veren Bir Bilgisayar Ağı Saldırı Tespit Sisteminin Tasarlanması ve Gerçeklenmesi*. Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : Kaya, Çetin
Uyruğu : T.C.
Doğum tarihi ve yeri : 06.04.1989, Safranbolu
Medeni hali : Bekâr
Telefon : 0 (545) 497 77 87
e-mail : ckaya@kho.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Müh.	2016
Lisans	Kocaeli Üniversitesi / Bilgisayar Müh.	2012
Lise	Fatih Gelenbevi A.L.	2006

İş Deneyimi

Yıl	Yer	Görev
2012-Halen	Kara Harp Okulu	Öğretim Görevlisi

Yabancı Dil

İngilizce

Yayınlar

Kaya, Ç., Yıldız O., ve Ay, S. (2016). *Saldırı Tespitinde Makine Öğrenmesi Tekniklerinin Performans Analizi,* 24. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU2016), 1473- 1476, Zonguldak.

Kaya, Ç. ve Yıldız O. (2014). Makine Öğrenmesi Teknikleriyle Saldırı Tespiti: Karşılaştırmalı Analiz, ***Marmara Üniversitesi Fen Bilimleri Enstitüsü Dergisi***, 3, 89- 104.

Hobiler

Futbol, Basketbol

GAZİ GELECEKTİR...