T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HİZMET REDDİ SALDIRILARININ DERİN ÖĞRENME İLE TESPİTİ

Ayşegül ÜNAL

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Temmuz-2018
KONYA

TEZ KABUL VE ONAYI

Ayşegül ÜNAL tarafından hazırlanan "HİZMET REDDİ SALDIRILARININ DERİN ÖĞRENME İLE TESPİTİ" adlı tez çalışması 19/07/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri **İmza**

Başkan
Prof. Dr. Harun UĞUZ

Danışman
Dr. Öğr. Üyesi. MEHMET HACIBEYOĞLU

Üye
Dr. Öğr. Üyesi. Onur İNAN

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mehmet KARALI
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ayşegül ÜNAL

Tarih:19.07.2018

ÖZET

YÜKSEK LİSANS TEZİ

HİZMET REDDİ SALDIRILARININ DERİN ÖĞRENME İLE TESPİTİ

Ayşegül ÜNAL

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Dr. Öğrt. Üyesi. MEHMET HACIBEYOĞLU

2018, 82 Sayfa

Jüri
Dr. Öğr. Üyesi. MEHMET HACIBEYOĞLU
Prof. Dr. Harun UĞUZ
Dr. Öğr. Üyesi. Onur İNAN

Siber saldırı tespiti ve sınıflandırılması bilgisayar ağları ve internet teknolojileri için çok önemli bir konudur. İnternet üzerinden hizmet veren kamu kurumları ve özel şirket sayılarının hızla artması ile doğru orantılı olarak kurumsal sistemlere yapılan dijital saldırıların sayısı da artmaktadır. Günümüzde, hizmet reddi (Denial of Service) saldırıları bilgisayar ağları ve internet teknolojileri için en büyük tehdit oluşturan saldırı tiplerinden biridir. Hizmet reddi saldırıları meşru bir kullanıcıyı kullanarak, bir sunucunun hafıza, işlemci ve bant genişliği gibi kaynaklarının tüketilmesini amaçlar. Eğer saldırı doğası gereği dünya üzerindeki farklı coğrafi konumlardan aynı zamanda yapılırsa dağıtık hizmet reddi atağı (Distributed Denial of Service) olarak isimlendirilir ve daha etkili bir saldırı haline gelir. Hizmet reddi saldırılarını tam olarak tespit edebilen ve engelleyebilen bilgisayar ağları güvenlik cihazı veya yazılımı bulunmamaktadır. Literatürde makine öğrenmesi algoritmaları hizmet reddi saldırılarının tespitinde ve saldırı tespit sistemlerinin geliştirilmesinde sıklıkla kullanılmaktadır. Derin öğrenmede makine öğrenmesinin bir alt alanıdır ve hastalık teşhisi, ses tanıma, resim tanıma, sahtekârlık tespiti gibi birçok sınıflandırma probleminin çözümünde kullanılmaktadır. Bu tez çalışmasında hizmet reddi saldırılarının tespit edilebilmesi için derin öğrenme sistemleri geliştirilmiştir. Deneysel çalışmalar NSL-KDD veri seti kullanılarak 10-kat çapraz doğrulama tekniği ile yapılmıştır. Deneysel çalışmalarda NSL-KDD veri kümesi öncelikle direk olarak kullanılmış ve daha sonra problemi zorlaştırmak için sınıf değerlerinde iki farklı etiketleme yapılmıştır. Geliştirilen derin öğrenme modellerinin başarısını anlayabilmek için elde edilen sonuçlar yakın zamanlarda yapılan çalışmalarda sıklıkla kullanılan karar destek sistemleri, yapay sinir ağları ve naive bayes makine öğrenmesi sınıflandırıcıları ile karşılaştırılmıştır. Üç farklı deney çalışmasına göre elde edilen sonuçlara bakıldığı zaman derin öğrenme modelinin diğer makine öğrenmesi sınıflandırıcılarına göre daha başarılı sonuçlar verdiği görülmüştür. Geliştirilen yazılımın ve yapılan değerlendirmelerin bilgi güvenliğinin sağlanmasında katkı yapması beklenmektedir.

Anahtar Kelimeler: Derin Öğrenme, Hizmet Reddi Saldırısı, Saldırı Tespit Sistemleri, Makine Öğrenmesi

ABSTRACT

MS THESIS

DETECTION OF DENIAL OF SERVICE ATTACKS USING DEEP LEARNING

Ayşegül ÜNAL

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN INDUSTRY ENGINEERING

Advisor: Asst. Prof. Dr. MEHMET HACIBEYOĞLU

2018, 82 Pages

Jury
Asst. Prof. Dr. MEHMET HACIBEYOĞLU
Prof. Dr. Harun UĞUZ
Asst. Prof. Dr. Onur İNAN

Detection and classification of cryptic attacks is a very important issue for computer networks and internet technologies. The rapid increase in the number of public institutions and private companies that provide services over the Internet has also increased the number of digital attacks on institutional systems in direct proportion. Today, denial of service attacks are one of the biggest threats to computer networks and Internet technologies. Denial of Service attacks use a legitimate user to exhaust a server's resources such as memory, processor and bandwidth. If the attack is carried out at the same time from the different geographical locations around the world, it is called Distributed Denial of Service and becomes a more effective attack. There is no computer network security appliance or software that can accurately detect and prevent denial of service attacks. In the literature, machine learning algorithms are frequently used in the detection of service rejection attacks and in the development of intrusion detection systems. Deep learning is also a subfield of machine learning and is used to solve many classification problems such as disease diagnosis, voice recognition, image recognition, fraud detection. In this thesis study, deep learning systems have been developed to detect service rejection attacks. Experimental studies were performed using a 10-fold cross-validation technique using the NSL-KDD data set. In experimental studies, the NSL-KDD dataset was used directly first, and then two different labeling were done on the class values to make the problem more difficult. The results obtained in order to understand the success of the developed deep learning models are compared with the decision support systems, artificial neural networks and naive Bayes machine learning classifiers which are frequently used in recent studies. When the results obtained from three different experimental studies are examined, it is seen that the deep learning model gives more successful results than the other machine learning classifiers. It is expected that contribution to the provision of information security of developed software and evaluations made.

Keywords: Deep learning, Denial of Service, Intrusion Detection Systems, Machine learning

ÖNSÖZ

Tez çalışmam boyunca bilgisi ve yardımları ile beni yönlendiren değerli danışmanım Dr. Öğr. Üyesi. Mehmet HACIBEYOĞLU'na, beni daima destekleyen sevgili eşim Serhat Kaan ÜNAL'a, maddi manevi her türlü destekleri ile hep yanımda olan babam Cemil, annem Meral, kardeşlerim Pınar ve Begüm SUNGUR'a teşekkür ederim.

Ayşegül ÜNAL
KONYA-2018

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

ACK: Acknowledgement

CPU: Central Processing Unit

DARPA: Defence Advanced Research Projects Agency

DDoS: Distributed Denial of Service

DoS: Denial of Service

DVM: Destek Vektör Makinesi

FTP: File Transfer Protocol

GPU: Graphics Processing Unit

HTTP: Hyper Text Transfer Protocol

ICMP: Internet Control Message Protocol

IDEVAL: Intrusion Detection Evaluation

IP: Internet Protocol

KDD: Knowledge Discovery and Data Mining

MIT: Massachusetts Institute of Technology

NB: Naive Bayes

OSI: Open Systems Interconnection

R2L: Remote to Local

RELU: Rectified Linear Unit

RMSprob: Root Mean Square Error Probability

SGD: Stochoastic Gradient Descent

SMTP: Simple Mail Transfer Protocol

SNMP: Simple Network Management Protocol

STS: Saldırı Tespit Sistemi

SYN: Synchronize

TCP/IP: Transmission Control Protocol / Internet Protocol

TCP: Transmission Control Protocol

U2R: User to Root

UDP: User Datagram Protocol

YSA: Yapay Sinir Ağları

1. GİRİŞ

Gelişen teknoloji çağı finans, bankacılık, eğitim, devlet işleri gibi günlük hayatımızda kullandığımız birçok işlemi, elektronik ortamlara taşıyan sistemler oluşturarak bilgiye daha kolay ve her zaman erişebilen bir ortam meydana getirmiştir. Elektronik ortamdan gerçekleştirilebilen bu işlemler ile ilgili devlet dairesinden erişilebilecek bir bilginin internet üzerinden erişilebilmesi, bankadan yapılacak bir para yatırma işleminin internet bankacılığı ile gerçekleştirilebilmesi, alışveriş merkezine gitmeden internet sitelerinden alışveriş yapılabilmesi, sınav başvurularının gerçekleştirilebilmesi gibi örnekler verilebilir. Bu işlemler sırasında kişisel ve kurumsal bilgilerin, sürekli bir işleme maruz kaldığı, taşınma veya güncellenme gibi işlemlerle değiştiği görülmüştür.

Günümüzde, elektronik ortamlara aktarılan bu hizmetlerin sayıları ve kullanımları gün geçtikçe artmakta ve elektronik olarak saklanan bilginin boyutu da yükselmektedir. Aynı zamanda, bilginin öneminden dolayı kötü niyetli kişilerin bu bilgilerin tutulduğu sistemlere yapmış oldukları siber saldırıların sayısı da artış göstermektedir.

Geçmişten bugüne bilgi, insanlık için en değerli varlıklardan biri olmuştur. Çünkü bilgi, belirsizlikleri azaltan bir kaynak ve insanın yaşayışını, davranışını, ilerlemesini ifade eden bir veri topluluğudur. Bilgi sürekli değişen ve gelişen bir varlık olarak çoğalmakta ve değerini her zaman korumaktadır.

Bilgi, en basit benzetme ile para gibi varlık olarak düşünülebilir. Kişiler, kurumlar, kuruluşlar ve ülkeler için bilginin elde edilmesi ve saklanması zordur. Fikri mülk olarak tanımlanan bu varlık, bir kurumun bilgi ve özbilgi varlığıdır (Gülmüş, 2011). Elektronik ortamlarda kullandığımız bu bilgilerin önem taşıması, bilginin güvenliğinin sağlanması gerekliliğini meydana getirmiştir. Bilgi güvenliğinin temel amacı; bilginin erişilebilirlik, gizlilik ve bütünlük ilkelerini sağlayarak korunmasıdır. Bilgiye erişilebilirliğin bir an bile kesilmesi birçok işlemde aksaklıklara neden olacaktır. Bilginin gizliliğini yitirmesi bazı kritik kişisel bilgilerin ele geçirilmesine sebep olarak sorunlara neden olacaktır. Bilgiye tam ve doğru bir şekilde erişememek işlemlerde hatalara sebep olacaktır. Bu kritik işlemlerde meydana gelecek güvenlik sorunları sistemlerde maddi ve manevi kayıplara neden olacaktır.

Günümüzde tüm işler artık bilgi ve bilgi araçlarıyla yürütülmektedir. Bilginin bir kısmı herkes tarafından bilinendir (public) ve bu genel olarak da adlandırılabilir. Diğer

bir kısmı ise herkesçe bilinmemesi gereken bilgi olup bu da özeldir (private). İş bu kısım da özel korumayı ve yetkilendirmeyi gerektirir (Gülmüş, 2011).

Kişilerin bilgi güvenliğini direk olarak etkileyen faktörlerden belki de en önemlisi kurumsal bilgi güvenliğidir. Her hangi kişi bilgi sistemlerinden hizmet alırken ya da hizmet sunarken o sistemin kurumsal bilgi varlıklarını doğrudan ya da dolaylı olarak kullanabilir. Bu hizmetler kurumsal hizmetler olabileceği gibi bankacılık işlemleri ya da bir kurum içerisinden alınan her hangi bir bireysel işlem de olabilir. Sonuç olarak, bir kurumun kurumsal bilgi güvenliği sağlanmadıkça, kişisel bilgi güvenliği de sağlanamaz (Vural, Sağıroğlu, 2008).

Elektronik sistemlere karşı yapılabilecek saldırılara karşı güvenliğin sağlanması güvenlik duvarları, antivirüs programları ve saldırı tespit sistemleri (STS) ile sağlanmaktadır. Güvenlik duvarı ve antivirüs sistemlerinin algılayamadığı saldırılar, STS'ler ile algılanabilmektedir. STS'ler ağ trafiğini denetleyerek sistem üzerine meydana gelen saldırıları tespit eden mekanizmalardır.

Günümüzdeki en temel bilgisayar ağları saldırılarından bir tanesi hizmet reddi (Denial of Service, DoS) saldırılarıdır. DoS saldırıları bir sunucunun hafıza, işlemci gibi sistem kaynaklarını veya bant genişliğini tüketerek sunucunun vermiş olduğu servislerin hizmet dışı bırakılmasıdır. Dağıtık hizmet reddi saldırıları (Distributed Denial of Service, DDoS), DoS saldırılarının paralel olarak birden fazla kaynaktan yani daha etkili bir şekilde yapılmasıdır. DoS saldırılarının verebileceği zararı önlemek adına güvenlik sistemleri oluşturulmaktadır. Bu güvenlik sistemleri oluşturulurken makine öğrenmesi algoritmalarından da sıklıkla faydalanılmaktadır. Makine öğrenmesi algoritmalarından olan derin öğrenme algoritması da günümüzde birçok sınıflama probleminin çözümünde yaygın olarak kullanılmaktadır. Bu tez çalışmasında geleneksel makine öğrenmesi algoritmaları dışında başarı oranı daha yüksek olan derin öğrenme algoritması kullanılmıştır. Derin öğrenme algoritması literatürde sıklıkla kullanılan NSL-KDD veri kümesi ile test edilmiş ve elde edilen sonuçlar diğer makine öğrenmesi algoritmaları ile karşılaştırılmıştır.

İkinci bölümde bu çalışma ile ilgili kaynak araştırması yapılmış olup literatürde bulunan çalışmalar incelenmiştir. Üçüncü bölümde materyal ve yöntem çalışması yapılmıştır. Dördüncü bölümde araştırma sonuçları elde edilmiş olup beşinci bölümde sonuçlar ve önerilere yer verilmiştir.

Bu tez çalışmasının önemi, bir saldırı tipi olan dağıtılmış hizmet reddi saldırıların (DDoS) tespit edilmesi için hazırlanacak sistem ile elektronik sistemlere karşı

yapılabilecek saldırıların derin öğrenme algoritması ile tespit edilip oluşabilecek maddi ve manevi zararlara karşı önlem alınmasıdır.

2. KAYNAK ARAŞTIRMASI

Suresh ve Anitha, yaptıkları çalışmada, DDoS saldırılarını etkili bir şekilde tespit etmek için makine öğrenmesi algoritmalarının değerlendirilmesiyle ilgili çalışmışlardır. CAIDA veri seti, saldırı verileri olarak kullanılmış ki-kare ve bilgi kazanma sıralamasına dayalı olarak ilgili özellikler seçilmiştir. Yapılan deneysel çalışmalar, bulanık c-ortalamalar kümelenmesinin daha iyi sınıflandırma sağladığını ve diğer algoritmalara kıyasla hızlı olduğunu göstermiştir (Suresh, Anitha, 2011).

Kaya, yaptığı çalışmada KDD CUP 99 ve NSL-KDD veri setlerini kullanarak makine öğrenmesi tekniklerinden Bayes ağları, destek vektör makinesi (DVM), K en yakın komşu algoritması (KNN), yapay sinir ağları (YSA) ve karar ağaçlarının performanslarını incelemiştir. DoS saldırılarının tespitinde KNN, karar ağaçları ve YSA %100'e yakın bir başarıya ulaşmıştır (Kaya, 2016).

Yuan ve arkadaşları, yaptıkları çalışmada ISCX2012 veri seti kullanarak DeepDefense tekniği ile DDoS saldırılarını tespit etmeye çalışmış ve geleneksel makine öğrenme yöntemlerine kıyasla hata oranının düştüğünü görmüşleridir (Yuan, Li, Li, 2017).

Vijayasarathy, yaptığı çalışmada gerçek zamanlı ve uygulanabilirlik açısından tasarlanmış bir sistemle; TCP ve UDP için NB yöntemi ile DoS ve DDoS saldırılarını tespit etmeye çalışmıştır (Vijayasarathy, 2012).

Uslu, yaptığı çalışmada DoS tespiti için KDD CUP 99 veri setini kullanarak standart ID3 ve yeni önerilen yöntemle iki farklı karar ağacı oluşturmuştur. Bu karar ağaçlarının tespit ettiği saldırıların başarı oranlarını karşılaştırmış ve yeni yöntemlerle oluşturulan karar ağaçları yapısının ID3 algoritmasından %3 daha başarılı olduğunu görmüştür (Uslu, 2009).

Liu ve arkadaşları, hiyerarşik temel bileşen analiz YSA algoritması geliştirmiş ve KDD CUP 99 veri seti ile test ettiğinde DoS saldırılarını %100 başarı ile sınıflandırmışlardır (Liu, Yi, Yang, 2007).

Güven, yaptığı çalışmada KDD CUP 99 veri setini kullanarak DVM, YSA, karar ağaçları, bayes ağları ve K en yakın komşuluk makine öğrenmesi algoritmaları ile duyarlılık, seçicilik, kesinlik ve F-ölçütü yönünden STS'lerin performanslarını incelemiştir (Güven, 2007).

Zhang ve Zhu, en küçük kareler ve DVM kullanarak tasarladıkları STS'yi, greedy algoritması ile geliştirmiş ve KDD CUP 99 veri seti kullanarak test etmişlerdir. DoS saldırı tespitinde %93,81 başarı elde etmişlerdir (Zhang, Zhu, 2010).

Saied ve arkadaşları, TCP'yi algılamak için eğitimli bir yapay sinir ağı algoritması kullanmışlardır. Öğrenme süreci gerçek bir yaşam ortamının ayna görüntüsü olan bir ağ ortamı üreterek başlamıştır. Normal trafik ağa akarken, farklı DDoS saldırıları başlatılmıştır. Veri setleri, önceden işleme tabi tutulmuş ve Java Sinir Ağı Simülatörü kullanılarak algoritmayı eğitmeye hazırlanmıştır. Çalışma imza tabanlı ve diğer ilgili araştırmalara karşı değerlenmiştir. Bu çözüm orijinal paketlerin geçmesine izin verirken, sahte paketlerin kurbana ulaşmasını önleyerek bunu hafifletmek için tasarlanmıştır. Eski ve güncel veri setleriyle eğitilerek tekrar değerlendirmiştir ve eğitim gördüklerine (güncel kalıplara) benzer bilinen ve bilinmeyen DDoS saldırılarını tespit etmeyi başarmıştır (Saied, Overill, Radzik, 2014).

Bhuyan ve arkadaşları bu çalışmada, DDoS saldırıları, algılama şemaları ve son olarak araştırma konuları ve zorlukları hakkında genel bir sunum yapmıştır. Mevcut algılama mekanizmalarının karşılaştırılması ve çoğu şemanın gerçek zamanlı ağ savunması için tüm gereksinimleri karşılayamayacağından bahsedilmiştir. Farklı performans parametrelerinin birbirlerine karşı nazik ve uygun şekilde dengelenmesi gerektiği anlatılmış ve DDoS saldırılarına karşı olası bir çözüm de kısaca özetlenmiştir (Bhuyan, Kashyap, Bhattacharyya, Kalita, 2012).

Çatak ve Mustaçoğlu yaptıkları çalışmada makine öğrenmesi ve derin öğrenme teknolojilerini kullanarak Avustralya Siber Güvenlik Merkezi'nde bulunan Siber Güvenlik Laboratuvarından alınan veri kümesi ile zararlı ağ saldırılarının algılanması ile ilgili bir model önermişlerdir (Çatak, Mustaçoğlu, 2017).

Noureldien ve Izzedin, DoS saldırı tiplerinin sınıflandırılmasında denetimli makine öğrenmesi algoritmalarından PART, BayesNet, IBK, Logistic, J48, Random Committee ve InputMapped algoritmalarını kullanmışlardır. Farklı DoS saldırı tipleri için deneysel çalışmalar yapmışlardır. Deneysel çalışmalarda NSL-KDD veri setini ve WEKA veri madenciliği aracını kullanmışlardır. Sonuç olarak Smurf ve Neptune saldırı tipleri için Random Committee algoritmasının ve diğer saldırı tipleri için ise PART algoritmasının daha başarılı sonuçlar verdiğini belirtmişlerdir. InputMapped algoritmasının ise DoS saldırılarının sınıflandırılmasında en kötü algoritma olduğunu söylemişlerdir (Noureldien, Izzedin, 2016).

Lee ve arkadaşları, ağ saldırı tespitine derin öğrenme yaklaşımlarının karşılaştırmalı bir değerlendirmesini yapmışlardır. DoS saldırılarının tespiti için çoklu derin öğrenme yaklaşımı önermişlerdir. Derin öğrenme modellerinden genel sinir ağı, self-taught learning ve persistence olarak üç tanesini kullanmışlardır. Derin öğrenme modelleri için KDD CUP 99 kullanılmıştır ve bunları NSL-KDD veri setinde yapılan soft-max regresyon (SMR) sonuçlarıyla karşılaştırmışlarıdır. Soft-max regresyon %75.23'lük doğruluk, %63.73'lük duyarlılık ve %75.46'lık bir f ölçüsü sağlamışlardır (Lee, Amaresh, Green, Engels, 2018).

Al-kasassbeh ve arkadaşları, STS'lerin gizliliğini, bütünlüğünü ve kullanılabilirliğini korumak için en son davetsiz misafir saldırısı imzaları ile daima güncel olmasından ve yeni saldırıların öğrenilmesinden bahsetmişlerdir. Bunun yanında STS'lerin hızının da çok önemli olduğuna değinmişleridir. Bu sebeple farklı makine öğrenmesi tekniklerinin test etmişlerdir. Sınıflandırma tekniklerinden J48, Rastgele Orman, Rastgele Ağaç, Karar Tablosu, MLP, Naive Bayes (NB) ve Bayes Ağı algoritmalarını kullanmışlardır. Bu sınıflandırma teknikleri arasında Rastgele Orman sınıflandırıcı tüm KDD veri kümesi saldırı türlerini (DOS, R2L, U2R ve PROBE) tespit etmek ve sınıflandırmak için en yüksek doğruluk oranını elde ettiğini tespit etmişlerdir (Al-kasassbeh, Al-Naymat, Hamadneh, Obeidat, Almseidin, 2018).

Niyaz ve arkadaşları, Dağıtılmış Hizmet Reddi (DDoS), günümüzde kurumsal ağ altyapısının ortaya çıktığı en yaygın saldırılardan biri olduğundan bahsetmiş ve yazılım tabanlı ağ (SDN) ortamında çoklu vektör saldırı tespiti için derin öğrenme tabanlı DDoS algılama sistemi uygulamışlarıdır. Bu sistem bireysel DDoS saldırı sınıfını % 95,65'lik bir doğrulukla tanımlamıştır. Normal ve saldırgan sınıflardaki trafiği diğer çalışmalara göre % 99.82 doğrulukla çok düşük yanlış pozitif olarak sınıflandırmışlardır (Niyaz, Sun, Javaid, 2016).

Ma ve arkadaşları, yaptıkları çalışmada saldırı türlerini tespit etmek için spektral kümeleme (SC) ve derin sinir ağlarından (DNN) yararlanan SCDNN adlı yeni bir yaklaşım öne sürülmüşlerdir. İlk olarak, veri kümesi, SC'deki gibi küme merkezleri kullanılarak örnek benzerliğine dayalı olarak k alt kümelerine bölünmüştür. Daha sonra, bir test setindeki veri noktaları arasındaki mesafe ve eğitim seti benzerlik özelliklerine göre ölçülmüş ve izinsiz giriş tespiti için derin sinir ağı algoritmasına beslenmiştir. Deneysel sonuçlar SCDNN'nin, KDD CUP 99 ve NSL-KDD'den türetilen altı veri kümesi üzerinden en iyi doğruluk oranlarıyla SVM, BPNN, RF ve Bayesian yöntemlerinden daha iyi performans gösterdiğini göstermiştir (Ma, Wang, Cheng, Yu, Chen, 2016).

3. MATERYAL VE YÖNTEM

Günümüzde hızla gelişen ve değişen teknoloji ile bilgisayar ve internet kullanımı yaygın hale gelmiş ve yaşamın her alanında kullanılmaya başlanmıştır. Bankacılık, eğitim, sağlık, finans, ticaret, elektronik devlet alanları gibi önemli bilgilerin bulunduğu ve kritik işlemlerin gerçekleştirildiği bu uygulamalar günlük yaşantımızda sıklıkla kullanılır hale gelmiştir. Bu yüzden bilgisayar sistemlerindeki bu bilgilerin en iyi şekilde korunması, bir bütün olarak saklanması ve bilgiye istenilen her an erişimin sağlanması; bilgi güvenliliğinin gereğini ve önemini üst seviyeye taşımıştır.

3.1. Bilgi Güvenliği

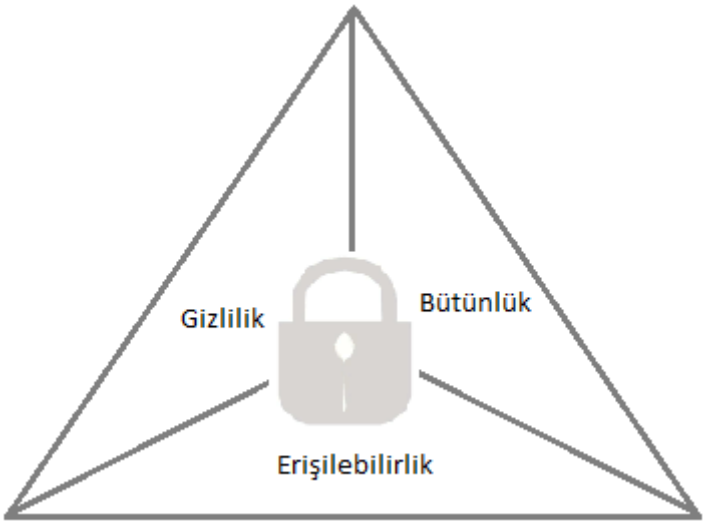
Elektronik ortam üzerinden devlet kurumları ve ticari şirket işlemlerinin gerçekleştirilebilmesi ile bilginin bilgisayar ortamına taşınması kaçınılmaz olmuştur. Bilginin kâğıt üzerindeki varlığının önemi gibi elektronik ortamdaki önemi ile birlikte korunması, saklanması ve taşınması sırasında bilgi güvenliği kavramının gerekliliği meydana gelmiştir.

Bilgi güvenliği, bilgiye sürekli erişilebilen bir ortamda, bilginin göndericisinden başlayarak alıcısına kadar gizlilik içerisinde, bozulmadan, değişikliğe uğramadan ve başkaları tarafından ele geçirilmeden güvenli bir şekilde iletilmesi olarak tanımlanabilir (Vural, 2007). Bilgi güvenliğinin sağlanması için uygun güvenlik politikalarının belirlenmesi ve uygulanması gereklidir. Güvenlik politikaları, gerçekleştirilen faaliyetlerin sorgulanması, yapılan erişimlerin izlenmesi, gerçekleştirilen değişikliklerin kayıtlarının tutulup değerlendirilmesi, veri silme işlemlerinin sınırlandırılması gibi bazı kullanım şekillerine indirgenebilmektedir (Canbek, Sağıroğlu, 2006).

Bilgi güvenliği farklı alanlar ve katmanlarda güvenlik sağlamak adına geniş bir alana sahiptir. Ağ güvenliği, veri güvenliği, uygulama güvenliği, kimlik ve erişim güvenliği gibi kategorilere ve bu kategorilerin alt kategorilerine ayrılır. Bütün bu kategoriler bilgi güvenliğinin önemini ve geniş bir kapsama sahip olduğunu vurgulamaktadır.

3.1.1.Bilgi Güvenliği Unsurları

Bilgi güvenliğinin temel amacı her türlü bilginin gizliliğini, bütünlüğünü ve kullanılabilirliğini sürekli olarak sağlamaktır. Şekil 3.1.'de gösterilen bilgi güvenliği unsurlarının, bilgi güvenliğinin sağlanması için zarar görmemesi gerekmektedir.

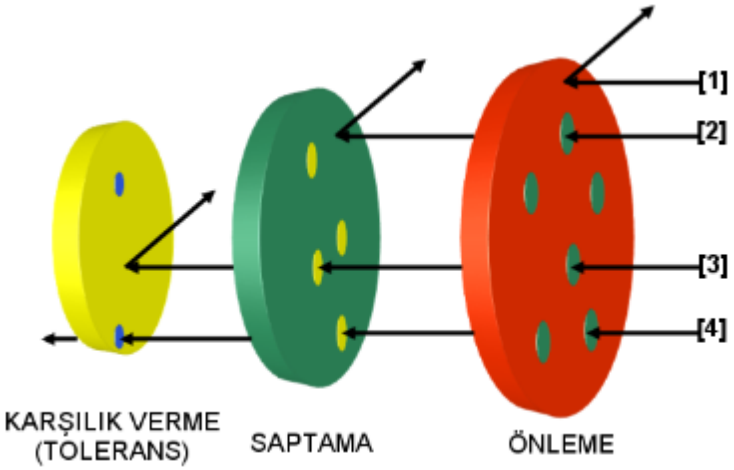


Şekil 3.1. Bilgi güvenliği unsurları

Gizlilik; bilgiye sadece erişim yetkisi olan kişiler tarafından erişilmesi, bütünlük; bilginin gerçek ve değiştirilmemiş olmasının sağlanması, erişilebilirlik; bilgiye herhangi bir yerden herhangi bir zamanda erişiminin sağlanması olarak tanımlanabilir.

3.1.2.Bilgi Güvenliği Süreçleri

Bilgi güvenliğinin sağlanması için sistem üzerinde yapılacak incelemelerde korunmak istenen sistem için gerekli değerlendirmelerin yapılması ve güvenlik yöntemlerinin eksiksiz ve doğru bir şeklide belirlenmesi gerekmektedir. Sistem üzerinde meydana gelebilecek risklerin önceden belirlenmesi, hâlihazırda bulunan risklerin değerlendirilmesi ve risklerin kabul edilebilir bir düzeye getirilmesi risk yönetiminin amaçlarını tanımlar. Risk yönetimi sonucunda kurulacak güvenlik sistemleri ile %100 güvenliğin olmayacağının bilinmesi ile birlikte bilgi güvenliğinin ideal yapılandırılması önleme, saptama, karşılık verme olan üç süreç ile gerçekleştirilir.



Şekil 3.2. Bilgi güvenliği unsurları ve saldırılara tepkileri (Canbek, Sağıroğlu, 2006).

Şekil 3.2.'de sisteme gelen saldırıların güvenlik süreçlerindeki durumları görülmektedir. 1 numaralı saldırı önleme sürecine engellenebilmiş ama 2 numaralı saldırı saptama sürecine geçtikten sonra engellenebilmiştir. 3 ve 4 numaralı saldırılar karşılık ve sürecine kadar ilerlemiş 3 numaralı saldırı bu süreçte engellenebilmesine rağmen 4 numaralı saldırı sisteme zarar vermiştir (Canbek, Sağıroğlu, 2006).

Bilgisayar sistemlerine karşı meydana gelebilecek herhangi bir saldırıya karşı alınacak tedbirler önleme faaliyetidir. Kişisel bilgisayarla anti virüs programlarının kurulması, bilgisayarda şifreli ekran kullanılması, bilgisayar uygulamalarındaki kullanıcı şifrelerinin karmaşık olarak verilip gizli tutulması güvenlik önlemlerine örnek verilebilir.

Kurumsal ortamlarda bilgisayar güvenliğinin sağlanabilmesi için uygulanması gereken önleme adımları çok daha geniş ve karmaşıktır (Canbek, Sağıroğlu, 2006). Bu önleme adımlarını bazıları aşağıdaki gibidir:

- İşletim sistemi ve yazılımların servis paketlerinin ve güncelleme düzenli aralıklarla incelenmesi,
- Kullanıcılara asgari seviyede haklar verilmesi, kullanılmayan protokol, servis, bileşen ve işlemlerin çalıştırılmaması,
- Verilerin şifreli olarak iletilmesi
- Veri iletişiminde korunmasızlık tarayıcıları, sanal Özel Ağ kullanılması,
- Elektronik belge sistemlerinde açık Anahtar Altyapısı (Public Key Infrastructure) ve elektronik imzanın kullanılması
- Kimlik doğrulama için biyometrik tabanlı sistemlerin kullanılması olarak sıralanabilir (Canbek, Sağıroğlu, 2006).

Ancak bu güvenlik önleme yöntemleri her zaman %100 bir güvenlik sağlayamayacağı için saptama ve karşılık verme süreçlerinin de işlemesi gerekmektedir.

Saptama, sistem üzerindeki önleme faaliyetlerinden sonra sistemdeki her türlü olay kayıt altına alınıp olası saldırıda bu kayıtlar incelenerek ileride oluşacak saldırıların önüne geçilmeye çalışılmasıdır. Bu durumda bilgisayar sisteminin bekçisi olarak nitelendirebileceğimiz güvenlik duvarları, virüs programları, STS'ler saptama sürecinde kullanılan bazı yöntemlerdir.

Karşılık verme, önleme süreci ile baş edilemeyen ve saptama süreçleri ile belirlenmiş saldırı girişimlerini, mümkünse anında ya da en kısa zamanda cevap verecek eylemlerin yapılması olarak tanımlanabilir. STS'ler, saptanan tespite cevap verecek bir sistem yöneticisinin veya bir sistemin olması ile anlam kazanır. Aksi halde, hiç kimsenin duyup da önemsemediği bir araba alarmının getireceği yarardan öteye gitmez. Bu açıdan karşılık verme güvenlik sürecini tamamlayan önemli bir halkadır (Canbek, Sağıroğlu, 2006).

3.2.Saldırı Tespit Sistemleri

Kişisel ve kurumsal olarak kullanılan bilgisayar sistemleri kötü niyetli kişiler tarafından bilgi hırsızlığı, bilgi sızdırma, hizmet engelleme gibi çeşitli tehdit ve tehlikelere karşı karşıya gelebilir. Bu durum hem kişiler için hem de kurumlar için maddi ve manevi zararlar açabilir. Bir internet bankacılığı işleminde meydana gelecek açık, bir devlet kuruluşunun veri tabanında bulunan kişisel bilgilere erişimin yetkisiz kişilerce sağlanması veya bir online alışveriş sitesinin uzun süreli hizmet veremez hale gelmesi kişileri ve kurumları olumsuz yönde etkiler. Bütün bu olumsuz senaryolar için güvenlik önlemlerinin alınması gerekliliği meydana gelmiştir.

Bilgi güvenliğini ihlal edebilecek çeşitli tehditler ile güvenliği sağlayacak çeşitli programlar ortaya çıkmış ve güvenlik zafiyetlerini engellemek için güvenlik mekanizmaları oluşturulmuştur.

Bilgisayarların güvenliğini sağlamak, yetkisi olmayan şahısların bilgisayar sistemlere girip önemli bilgileri ele geçirmelerini veya değiştirmelerini önlemek için ilk olarak kimlik doğrulama, yetkilendirme ve erişim kontrolü gibi güvenlik sitemleri geliştirilmiştir. Bu sistemlerin güvenliğin sağlanmasında ilk basamağı oluşturmaktadır. İnternet kullanımının artması ile beraber bilgi sistemlerine yönelik siber tehditlerin sayısında ciddi artışlar ve siber saldırıların tiplerinde farklılıklar olmaktadır. Artan siber

saldırılar nedeniyle, yukarıdaki sistemler dışında yeni sitemlerin varlığına gerek duyulmuştur. Güvenlik sistemlerinin ikinci basamağını güvenlik duvarları (firewall), güvenlik tarayıcıları (vulnerability scanner) ve STS'ler oluştururlar. Bu güvenlik cihazlarının hiçbiri tek başına tam olarak yeterli değildir; çünkü her biri farklı güvenlik görevleri ile farklı güvenlik noktalarına odaklanmışlardır. Güvenli bir bilgisayar sistemi için bu cihazlar birbirlerine destek olacak şekilde beraber kullanılmalıdır (İTÜBİDB, 2013).

Bilgisayar sistemlerine yetkili olmayan kötü niyetli kişiler tarafından erişimin sağlanarak hizmet veremez hale getirilmesi amacıyla gerçekleştirilen faaliyetlere saldırı denir. Bilgi güvenliğinin temel unsurları olan gizlilik, bütünlük ve erişilebilirlik unsurlarının engellenmesi saldırı durumunun meydana geldiğini gösterir.

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 veya öğrenme isteği gibi çok çeşitli konuları kapsamaktadır (Kaya, 2016).

STS, bir bilgisayar sisteminde ya da bilgisayar ağında meydana gelebilecek bütün olayları izleyerek sistemin gizliliği, bütünlüğü veya erişilebilirliğine zarar gelmesi durumunda saldırıyı tespit eden uyarı sistemidir. STS'ler, günlük hayatımızda kullanılan güvenlik alarmlarına benzetilebilir. Bilgisayar ağlarına karşı yapılan saldırılar ağ trafiğinde anormalliklerin tespit edilmesi ile anlaşılabilir. Ağ trafiği izlenerek daha önceden saldırı imzalarının yer aldığı veri tabanı ile karşılaştırılır ve saldırı durumda uyarı verir.

STS'ler bilgisayar sistemlerinin güvenliğinde üç temel yarar sağlamaktadır (Dayıoğlu, Özgit, 2001).

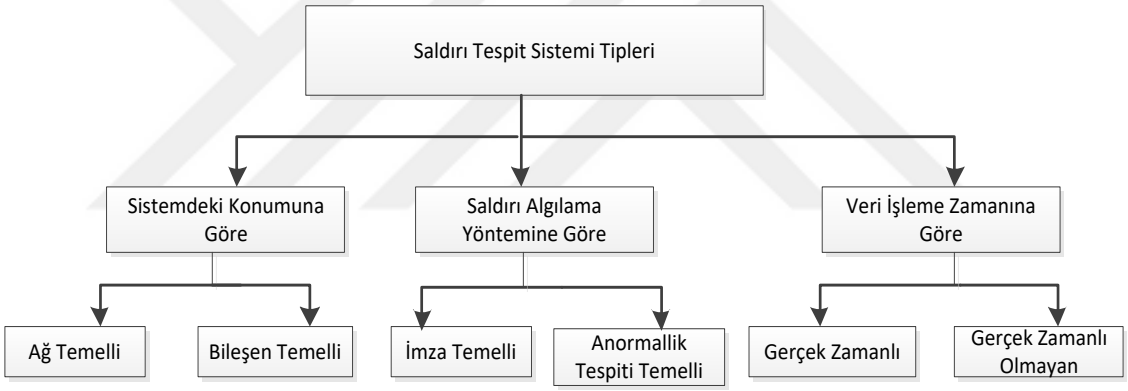
• Erken tespit: STS'ler, başlayan bir ihlali bilgisayar sisteminden sorumlu sistem yöneticisinden çok önce tespit edebilir. Bu olayla ilgili sorumlu personeli kısa mesaj, elektronik posta ya da telefon gibi farklı biçimlerde uyarabilir ve ihlalin etkisinin en kısa sürede azaltılmasını veya ihlalil engellenmesini sağlayarak riskin sınırlanmasına destek olabilir (Dayıoğlu, Özgit, 2001).

• Detaylı bilgi toplanması: STS'lerin kullanılması sayesinde, sistem yöneticileri devam eden veya daha önce gerçekleşmiş saldırı ve ihlallere ilişkin detaylı bilgi elde edebilirler. Elde edilen bu bilgiler saldırıları veya ihlalin kaynağı, büyüklüğü ve sistem üzerindeki etkilerinin incelenmesi noktasında son derece önemlidir (Dayıoğlu, Özgit, 2001).

• Toplanan bilgilerin kanıt niteliği: STS'ler tarafından toplanan bilgiler, hukuki bir durumda kanıt teşkil edebileceği gibi, faklı bir kurumdan kaynaklanan bir ihlalde, ilgili kurumun sistem yöneticileri ile temasa geçildiğinde de görüşmeler için zemin teşkil edebilir (Dayıoğlu, Özgit, 2001).

3.2.1. Saldırı Tespit Sistemi Tipleri

STS'lerin amacı, ağa ait bileşenleri izleyerek anormal ve kötü amaçlı davranışları tespit etmektir. STS'ler, saldırıları önlemez, saldırı anında ya da saldırı gerçekleştikten sonra uyarı veren bir alarm sistemi olarak düşünebiliriz. Teknolojinin gelişmesi ile bilgisayar ve ağ sistemlerinde de gelişmeler olmuş bu da STS'lere de yansımıştır. Günümüze kadar farklı kategorilere göre sınıflandırılarak işlev, zaman ve yer gibi türlere ayrılmıştır.



Şekil 3.3. Saldırı tespit sistemi tipleri

Şekil 3.3.'te STS tipleri gösterilmektedir. STS tipleri sistemdeki konumuna göre, saldırı algılama yöntemine göre ve veri işleme zamanına göre 3 gruba ayrılmıştır.

3.2.1.1.Sistemdeki Konumuna Göre

STS'ler sistemdeki konumuna göre iki gruba ayrılır.

Ağ Temelli, ağ trafiğini dinleyerek ağ paketlerini yakalar ve inceleyerek saldırı olup olmadığını analiz eder. Ağ temelli STS'ler genelde ticari yazılımlarda kullanılır.

Sunucu Temelli, sunucu üzerine kurularak sunucuya yapılan şüpheli işlemleri tespit etmeye çalışır. Sistem içerisindeki log dosyalarına başvurarak sisteme bağlanan kullanıcı erişim kontrolü, sistem dosyalarının durumlarını ve sistem işlemlerinin

davranışlarını inceleyerek saldırıyı yakalarlar. Bu tarz saldırılar genelde R2L ve U2R saldırılarını içerir. Merkezi STS'lerin yanına ek olarak sunucu tabanlı STS'lere destek vermek amacıyla kullanılabilir.

3.2.1.2.Saldırı Algılama Yöntemine Göre

Saldırı Algılama Yöntemine Göre STS anormallik tespiti ve imza tabanlı olarak iki gruba ayrılır.

Anormallik tespiti, normal trafiğin izlenerek belirli kurallar tanımlandıktan sonra gelen trafiğin durumuna göre bu kurallar dışında kalan anormal trafiğin saldırı olarak tespit edilmesidir. Anormallik tespiti yöntemi STS'ler için normal olmayan davranış anlamına gelir. Sistemdeki kullanıcı ve kullanıcı gruplarının davranışları izlenerek bir profil oluşturulur. Eğer kullanıcı davranışında normal seyri dışında bir davranış tespit edilirse saldırı olarak algılar. Bu yöntem ile daha önceden tanımlı olmayan saldırılar tespit edilebilmesi avantajının yanında saldırı olmayan bir durumun saldırı olarak algılanması mümkündür.

Sistemin saldırı alarmı verip vermeyeceği ile ilgili 4 durum meydana gelebilir. Bunlar;

Doğru pozitif, eğer saldırı tahmin edildiyse ve saldıysa oluşan durumdur.

Yanlış pozitif, eğer saldırı tahmin edildiyse ve saldırı değilse oluşan durumdur.

Doğru negatif, eğer saldırı değil olarak tahmin edildiyse ve saldırı değilse oluşan durumdur.

Yanlış negatif, eğer saldırı değil olarak tahmin edildiyse ve saldırı ise oluşan durumdur.

İmza tabanlı, Bilinen saldırıların imza kaydını veri tabanında tutarak gelen bir isteği veri tabanında olup olmadığına göre kontrol eder ve saldırı ya da normal davranış olduğunu tespit eder. Bilinen saldırı tipleri imza veri tabanlarında ağ paketlerinin protokol çeşidi, protokol işaretleri (flags), kaynak ya da hedef IP adresleri, port numaraları ve paketin paket verisi (payload) kısmı gibi karakteristik özelliklerle tutulur. Kayıtlı bu veriler ile yeni gelen istekler karşılaştırılarak saldırı olup olmadığı tespit edilir. İmza tabanlı STS'ler veri tabanlarında bulunan bütün saldırı imzalarını tespit etmeleri açısından avantajlı olmalarına rağmen veri tabanında olmayan bir saldırı geldiğinde saldırıyı algılayamaz. Bu nedenle veri tabanları yeterli büyüklükte ve her zaman güncel olmalıdır.

3.2.1.3.Veri İşleme Zamanına Göre

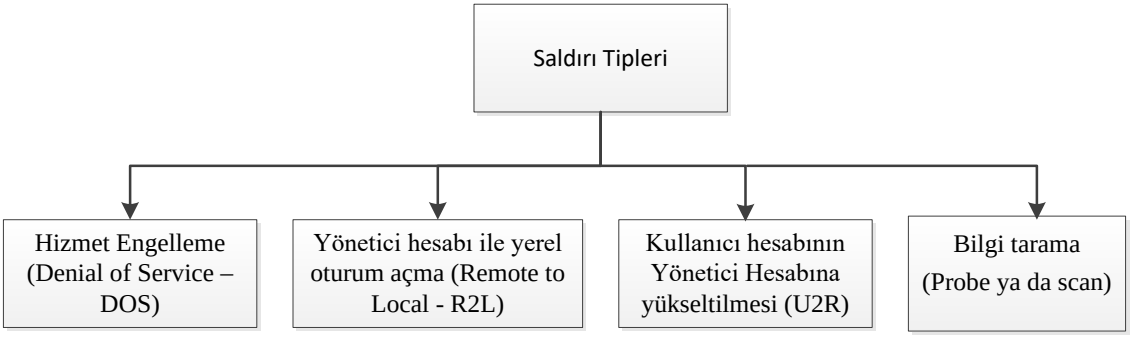
STS'ler için saldırı tespit edildikten sonra sistem yöneticisine haber verilinceye kadar geçen zaman veri işleme zamanıdır. Veri işleme zamanına göre STS gerçek zamanlı ve gerçek zamanlı olmayan olarak iki gruba ayrılır.

Gerçek zamanlı STS'ler saldırı meydana geldiği anda tespit edip anında reaksiyon gösteren sistemlerdir. Bu tip STS'ler saldırı engelleme sistemi (Intrusion Prevention Systems-IPS) olarak bilinen ticari yazılımlardır. Saldırıya anında cevap vermesi avantajının yanında ağ trafiği yoğun olan sistemler için performans açısından düşük ve maliyetlidir.

Gerçek zamanlı olmayan STS'ler saldırı meydana geldiğinde sistem yöneticisine bir bildirim ile haber verirler. Ancak saldırıya anında müdahale etmek yerine tespit edilen saldırı ileri bir tarihte incelenmek üzere saklarlar.

3.2.2. Saldırı Tipleri

Günümüzde bilgisayar ve ağ sistemlerinin gelişimi ile birlikte sistemlere yapılan saldırı türleri artmış ve saldırıların çeşitlilik göstermesine sebep olmuştur. Bu durumda ağ üzerinde yapılan saldırılar en sık karşılaşılan saldırılar olmuştur.



Şekil 3.4. Saldırı tipleri

Şekil 3.4.'te ağ üzerinde yapılan gerçekleştirilen saldırıların gösterilmiştir. Saldırılar 4 temel grupta incelenir.

3.2.2.1.Hizmet Engelleme (DoS)

Sisteme cevap verebileceğinden çok istek göndererek sistemin hizmet veremez hale getirilmesidir. En çok bilinen DoS saldırısıdır. Örnek saldırıları olarak SYN flood, Smurf, UDPstorm, Pingflood, Neptune, Mailbomb saldırıları verilebilir (Mukkamala, Janoski, Sung, 2002).

3.2.2.2.Yönetici Hesabı ile Yerel Oturum Açma (Remote to Local - R2L)

Bilgisayar sistemine girme yetkisi olmayan bir kullanıcının bir ağ üzerinden sisteme paket göndererek kullanıcı gibi bilgisayara erişim yetkisi kazanmasıdır. En çok bilinen R2L örnek saldırıları Dictionary, Guest, Imap,Named, Sendmail, Xlock, Xsnoop 'tur (Mukkamala, Janoski, Sung, 2002).

3.2.2.3.Kullanıcı Hesabının Yönetici Hesabına Yükseltilmesi (User to Root - U2R)

Kullanıcının normal yetkilere sahipken güvenlik açıklarından faydalanarak yönetici yetkilere sahip olmasıdır. En çok bilinen U2R örnek saldırıları Eject, Ffbconfig, Fdformat, Loadmodule, Perl, Ps, Xterm'dir (Mukkamala, Janoski, Sung, 2002).

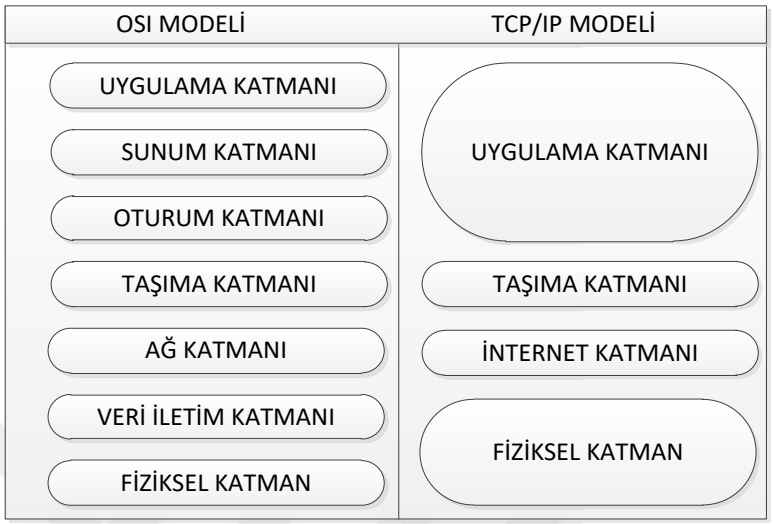
3.2.2.4.Bilgi Tarama (Probe ya da scan)

Bir saldırganın bilgi toplama veya bilinen açıkları bulmak için bir bilgisayar ağı taramasını yaptığı saldırıdır. Bilgisayar sistemindeki geçerli ip, işletim sistemi ve portları öğrenmek için yapılır. Bir ağda bulunan makinelerin ve hizmetlerin bulunduğu bir haritaya sahip bir saldırgan, bu bilgileri kötüye kullanmak için kullanabilir. Ayrıca bu saldırı için kullanılan araçlar güvenlik uzmanları tarafından sistemin güvenlik testinin yapılması için kullanılır. Örnekler Ipsweep, Mscan, Nmap, Saint, Satan (Mukkamala, Janoski, Sung, 2002).

3.3.Ağ Protokolleri

Ağ protokolleri, ağa bağlı bilgisayarların aralarındaki haberleşmenin sağlanması için kullandıkları dillerdir. Bilgisayarlar bir ağ üzerinden birbirleri ile iletişim kurmaları

için aynı protokolleri ya da uyumlu protokolleri kullanmaları gerekmektedir. Ağ protokolleri ve standartları dünyadaki bütün bilgisayarların ortak bir platformda buluşmasını sağlar.

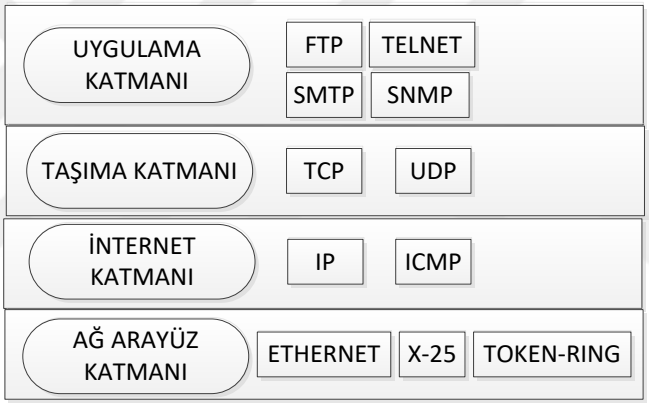


Şekil 3.5. OSI ve TCP/IP modeli katmanları

Bilgisayar ağları ile bilgisayarların haberleşmeye başladığı ilk yıllarda haberleşmenin sağlanması için aynı model ya da aynı marka olması gerekiyordu. Ancak daha sonra üreticilerin geliştirdikleri cihazlar ile farklı marka ve modeldeki bilgisayarlarında haberleşmesi için standart protokoller geliştirilmiştir. Bu geliştirilen protokollerden en yaygınları Açık Sistem bağlantıları olarak bilinen OSI (Open Systems Interconnection) ve Amerikan Savunma Bakanlığı tarafından geliştirilen TCP/IP (Transmission Control Protocol/Internet Protocol)'dir. OSI modeli 7 katmanlı bir ağ sistemi ile haberleşme gerçekleştirirken TCP/IP modeli bunu 4 katmanla gerçekleştirmektedir. OSI modelinin ilk 3 katmanı olan uygulama, sunum ve oturum katmanı TCP/IP modelindeki uygulama katmanına denk gelirken; taşıma katmanı değişmez, ağ katmanı internet katmanı adını almış olup veri iletim katmanı ve fiziksel katman ise TCP/IP modelindeki fiziksel katmana denk gelmektedir. OSI modeli iletim standartlarını belirlemeye yönelik iken TCP/IP daha çok uygulanabilir bir modeldir ve TCP/IP modeli OSI modeline göre daha hızlıdır. Şekil 3.5.'te OSI ve TCP/IP modellerinin katman yapısı gösterilmiştir.

3.3.1.TCP/IP Mimarisi ve Katmanları

İnternet ağ mimarisi katmanlı bir yapıdan oluşmaktadır. TCP/IP mimarisi uygulama programlarının bulunduğu katman sayılmazsa temelde 4 katmandan oluşmuş ve günümüzde temel ağ protokollü olarak yaygın bir şekilde kullanılmaktadır. Bilgisayarlar arası iletişim uygulama, ulaşım, yönlendirme ve fiziksel katman olarak bu 4 katman ile sağlanır. Her katmanda yapılacak görevler, protokoller ile gerçekleştirilmektedir. Şekil 3.6.'da TCP/IP modelinin katmanları ve bu katmanlarda yer alan protokoller gösterilmiştir. TCP ve IP iki ana katmanda yer almaktadır; ancak bu iki protokol birlikte çalıştığı için TCP/IP olarak bilinmektedir. Ağ cihazları, genellikle TCP/IP'nin ilk üç katmanını kullanır bazı durumlarda protokollerin kendi bünyesinde de çalıştırılması gibi bazı durumlarda da dördüncü katmanı da kullanabilirler.



Şekil 3.6. TCP/IP modelinin katmanları ve protokolleri

Uygulama programlarının bulunduğu katman, kullanıcının kullandığı programlar ve işletim sisteminin arka planda yürüttüğü programlardan oluşmaktadır. Bu katmanın altındaki katmanlar iletişimin sağlanması için kullanılır. Katmanlı yapıda, her katman bir altındaki katmanın işini bitirmesi ile iletişim gerçekleşir.

Uygulama katmanı, bir üst katmanda bulunan kullanıcının kullandığı programlar ve işletim sisteminin kullanıcıya sunduğu program ara yüzlerine hizmet verir. Kullanıcıya hizmet veren programın türüne göre ve kullandığı dosya biçimi bulunarak gönderilen verinin türüne göre uygulama katmanında farklı protokoller çalıştırılır. SMTP (Simple Mail Transfer Protocol-Basit Posta Aktarım Protokolü), TELNET (Telecommunication Network-İletişim Ağı), FTP (File Transfer Protocol-Dosya Aktarım Protokolü), SNMP (Simple Network Management-Basit Ağ Yönetim Protokolü) gibi protokolleri vardır. Uygulama katmanı, taşıma katmanı ile portlar sayesinde haberleşir. Port numaraları

http:80, FTP:21 gibi standart uygulamalardır ve taşıma katmanından gelen paket içeriğinin türünün anlaşılmasında rol oynar (İTÜBİDB, 2013).

Ulaşım katmanında TCP (Transmission Control Protocol-İletişim Kontrol Protokolü) ve UDP (User Datagram Protocol-Kullanıcı Veri Bloğu İletişim Protokolü) protokolleri, bir üst katmandan gelen veriyi paketleyip bir alt katmana gönderir. Bu katman verinin ne şekilde gönderildiğini gösterir. TCP ve UDP iletim sırasında veriye içinde bazı kontrol bilgilerinin yer aldığı başlık (header) ekler. TCP, kayıpsız veri gönderimi sağlayabilmek için kullanılan bir protokoldür. HTTP (Hyper Text Transfer Protocol- Hiper Metin Transfer Protokolü), HTTPS (Secure Hyper Text Transfer Protocol- Güvenli Hiper Metin Transfer Protokolü), POP3 (Post Office Protocol 3 - Postane Protokolü 3), SMTP, FTP ve SFTP (Secure FTP - Güvenli Dosya Taşıma Protokolü) gibi protokoller veri iletimini TCP ile gerçekleştirir. UDP, gönderilen paketin ulaşıp ulaşmadığını kontrol etmez. Bağlantı kurulum işlemleri, veri akış kontrolü ve tekrar iletim işlemleri yapmayarak iletim süresini azaltır ve ağ üzerinde TCP'ye göre daha az bant genişliği kaplar. TFTP (Trivial File Transfer Protocol - Önemsiz Dosya Aktarım Protokolü), SNMP gibi protokoller veri iletimini UDP ile gerçekleştirir (İTÜBİDB, 2013).

Yönlendirme katmanında IP, ICMP (Internet Control Message Protocol- İnternet Kontrol Mesaj Protokolü) protokolleri vardır. Bu katmanın görevi, bir üst katmandan gelen segmentleri alıcıya, uygun yoldan ve hatasız ulaştırmasıdır. Bunun için, IP katmanında gelen segmentlere IP başlık bilgisi ekler ve verinin hangi bilgisayara gönderileceği belirler. Gönderilen bu paket veri bloğu (datagram) halini alır (İTÜBİDB, 2013).

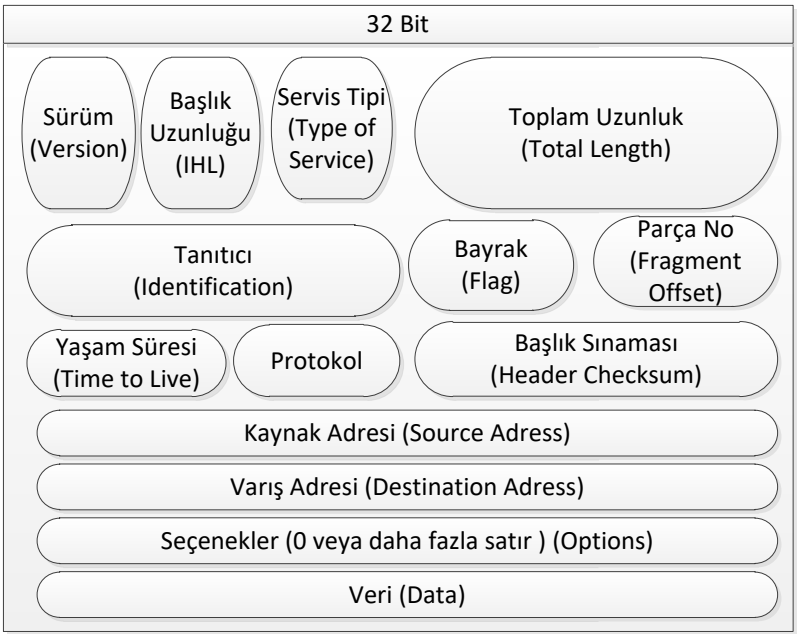
Ağ ara yüz katmanı ise verinin kablo üzerinde alacağı yapıyı tanımlayarak sıfır ve birlerin fiziksel olarak görüntülenmesini sağlar. IP başlığı olan bir bilgi kaynak ve hedef bilgisayarlarının IP bilgilerini içerir. Ayrıca hedef makineye ulaşmak için makinenin ethernet kartının MAC (Media Access Control - Ortama Erişim Adresi) adresinin tespit edilmiş olması da gereklidir. Ethernet, Token-Ring, X25 gibi protokoller bu katmana örnek olarak verilebilir (İTÜBİDB, 2013).

3.3.1.1.TCP

TCP, bir üst katmandan gelen veriyi uygun uzunlukta parçalara ayırır. Ayrılan bütün parçalara bir sıra numarası verir ve alıcı kısım bu sıra numarasına göre veri parçalarını alır. Paketler bir alt katmana gönderilirken birbirlerinden ayrılır ve paketler ulaştıktan sonra yine TCP ile sırasıyla bir bütün haline getirilir. Paketler ulaştıktan sonra TCP bir onay kodu gönderir. Eğer onay kodu gelmezse paketlerde sorun var demektir. Bu durumda paketlerin yeniden gönderilmesi gerekmektedir. Verinin hedefe gidip gitmediğini kontrolünün sağlanması TCP'nin güvenilir bir protokol olduğunu gösterir.

3.3.1.2.IP

İnternete bağlanan bütün bilgisayarların bir IP adresi vardır. Ağa bağlı bir bilgisayardan başka bir bilgisayara ileti göndermek için ağlar arasında yönlendirilerek ulaşması için IP adresi kullanılır. Her bilgisayarın internet üzerinde farklı bir IP adresi vardır. IP adresi 4 gruptan oluşan numaralar ile ifade edilir. Bu numaraların her biri 0-255 arasında bir sayıdır ve aralarına nokta konularak gösterilir. Internet üzerinde her bilgisayarın IP adresi vardır. Bir sitenin IP adresini biliyorsanız o IP adresini web tarayıcısına yazarak da siteye bağlanabilirsiniz. Ancak IP adresini akılda tutmak zor olduğu için IP adreslerine karşılık gelen alan adları verilmiştir.



Şekil 3.7. IP paket yapısı

Uygulamalar arasında veri alışverişi sırasında IP datagramlar kullanılır. 32 bitlik bir IPV4 yapısı Şekil 3.7.'de gösterilmiştir. IPV4 yapısındaki alanların içerikleri aşağıda açıklanmıştır.

- Sürüm (Verison), sisteminin sürümünü gösterir.
- Başlık uzunluğu (IP Header Length), başlık bilgisinin boyutudur.
- Servis tipi (Type of Service), paket iletim boyunca nasıl iletileceği hakkında bilgi verir.
- Toplam uzunluk (Total Length), IP paketinin toplam uzunluğudur.
- Tanıtıcı (Identification), parçalanan verinin hangi parçalardan oluştuğunu tanımlar. Aynı veriyi oluşturan parçalar aynı kimlik numarasını içerir.
- Bayrak (Flag), bilginin parçalanıp parçalanmadığı, onun parçalanma izninin olup olmadığı gibi bilgilere ait kodlar taşır.
- Parça no (Fragment Offset), parçaların hangi sırada birleşip veriyi oluşturacağını gösterir.
- Yaşam süresi, paketin ömrüdür.
- Protokol, üst katman protokolünün (TCP, UDP gibi) ne olduğunu gösterir.
- Başlık sınaması (Header Checksum), pakette hata olup olmadığı kontrol edilir.
- Kaynak adres (Source Adress), bilginin hangi adresten gönderildiğidir.
- Varış adresi (Destination Adress), bilginin hangi adrese gönderildiğidir
- Seçenekler (Options), gerekli zaman kullanılmak üzere güvenlik, kaynak, yönlendirme, yolun kaydedilmesi ve zaman gibi bilgileri tutar.
- Veri (Data), aktarılacak olan veridir.

3.4. DDoS

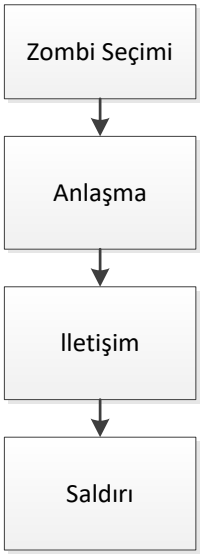
DoS saldırıları sırasında düşman, yakalanmamak ve iz bırakmamak için için zombi denilen aracı bilgisayarlar kullanır. Zombi bilgisayarlar topluklarına botnet (robot network) denir. Bir DDoS saldırısı, tek bir makineye veya birden fazla zombi makinede koordine edilmiş, bir DoS saldırısı başlatmak için çok sayıda bilgisayar kullanan bir saldırı olarak tanımlanabilir. İstemci / sunucu teknolojisini kullanarak suçlu, saldırı platformları görevi gören birden fazla farkında olmayan ortak bilgisayarların kaynaklarını kullanmak suretiyle DoS saldırısının etkililiğini önemli derecede

çoğaltabilir. Bir DDoS saldırganı bir DoS saldırganından daha akıllı sayılır. Silahlarını internet üzerinden "dağıtılmış" şekilde konuşlandırma ve bu güçleri ölümcül trafiğe sürüklemek için toplama yeteneği nedeniyle diğer saldırılardan ayrılır (Bhuyan, Kashyap, Bhattacharyya, Kalita, 2012).

DoS saldırılarının belirtileri:

- Ağ performansının beklenmedik bir şekilde düşmesi,
- İnternet bağlantılarında aşırı derecede bir yavaşlık yaşanması,
- Ağ bağlantılarının kesilmesi
- Spam elektronik postaların sayısının artması,
- Bir web sitesinin belli bölümlerine erişimin imkânsız hale gelmesi,
- Google Analiystic gibi istatistik veriler veren sitelerde, aniden artan istatistik verilerinin olması gibi durumlardır.

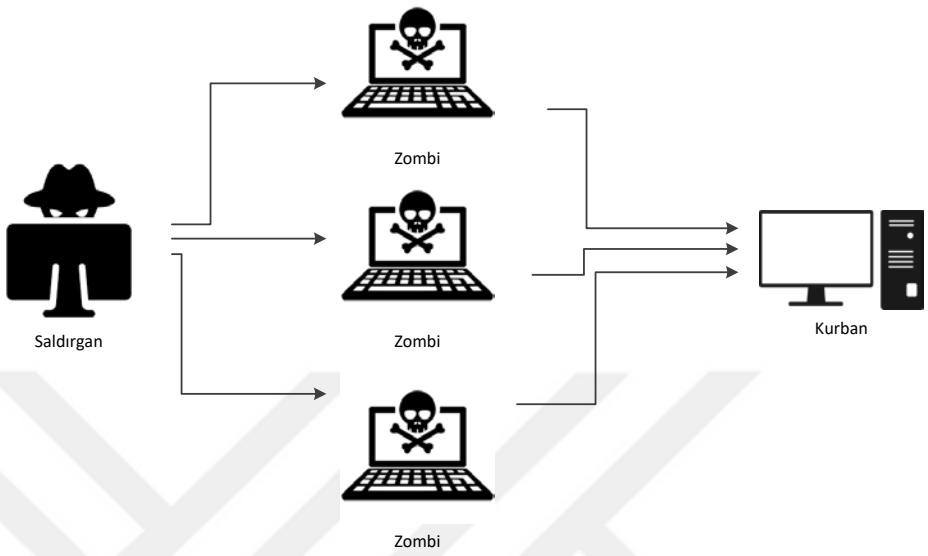
DDoS saldırısı gerçekleştirmek için sıklıkla uygulanan bir yöntem, saldırganın bir kurbana bir paket akışı göndermesidir; bu akış bazı önemli kaynakları tüketir ve bu nedenle mağdurun meşru müşterileri için kullanılamaz hale getirir. Bir diğer yaygın yaklaşım, saldırganın kurban makinedeki bir uygulamayı veya protokolü karıştıran ve onu dondurmaya veya yeniden başlatmaya zorlayan birkaç hatalı biçimdeki paket göndermesidir (Mirkovic, Reiher, 2004).



Şekil 3.8. DDoS saldırısı gerçekleştirme adımları

DDoS saldırısı başlatmada birkaç adım vardır. Bunlar Şekil 3.8.'de gösterilmiştir.

1.Zombi Seçimi: Saldırgan, saldırıyı gerçekleştiren aracıları seçer. İlk yıllarda, saldırganlar bu makinelerin elle kontrolünü ele geçirmeye çalıştı. Bununla birlikte, gelişmiş güvenlik araçlarının geliştirilmesiyle, bu makinelerin otomatik olarak ve anında tanımlanması daha kolay hale geldi (Bhuyan, Kashyap, Bhattacharyya, Kalita, 2012).



Şekil 3.9. DDoS saldırısı

2.Anlaşma: Saldırgan, zombi makinelerinin güvenlik açıklarını ve saldırı kodunu gönderir. Saldırgan ayrıca, yerleştirilen kodu tanımlama ve devre dışı bırakmaya karşı gerekli önlemleri alır. Şekil 3.9.'da gösterilen doğrudan DDoS saldırı stratejisine göre, ele geçirilen düğümler, yani zombiler, saldırgan ve mağdur arasında, İnternet'ten yüksek bant genişliği ile bağlı, çok sayıda korumasız ev sahibinin farkında olmadan ortak ev sahipliği yapmaktadır (Bhuyan, Kashyap, Bhattacharyya, Kalita, 2012).

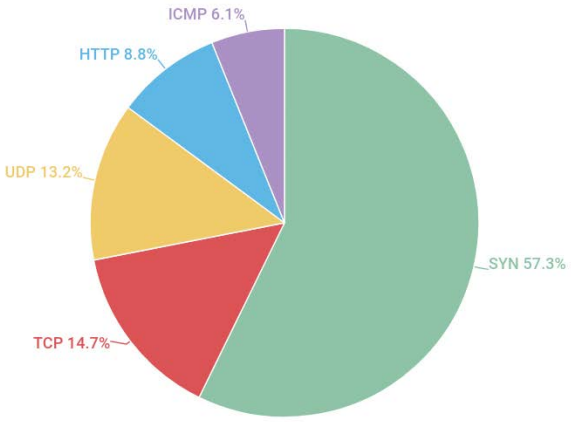
3.İletişim: Saldırgan, hangi aracıların çalışıp çalışmadığını, saldırıların ne zaman planlanacağını veya aracıların ne zaman yükseltileceğini belirlemek için herhangi bir sayıda işleyici ile iletişim kurar. Saldırganlar ve işleyiciler arasındaki bu tür iletişimler ICMP, TCP veya UDP gibi çeşitli protokoller aracılığıyla olabilir. Aracılar, saldırı ağının yapılandırmasına dayanarak, tek bir işleyici veya birden çok işleyici ile iletişim kurabilir (Bhuyan, Kashyap, Bhattacharyya, Kalita, 2012).

4.Saldırı: Saldırı yapan kişi saldırıyı başlatır. Zombi, hücum süresinin yanı sıra saldırıların özel özellikleri, türü, uzunluğu, TTL ve bağlantı noktası numaraları da ayarlanabilir. Saldırı paketlerinin özelliklerinde önemli farklılıklar varsa, saldırgan için algılama işlemi daha da karmaşık olduğundan yararlıdır (Bhuyan, Kashyap, Bhattacharyya, Kalita, 2012).

DDoS saldırılarının gerçekleştirilmesinde birçok sebep vardır. Temel amaç mağdura zarar vermektir. Çoğu zaman kişisel sebepler (muhtemelen intikam amaçlı ev bilgisayarlarına karşı önemli miktarda DDoS saldırısı gerçekleştirilmektedir) veya prestij (popüler Web sunucularına yönelik başarılı saldırılar, bilgisayar korsanlarının topluluğuna saygı duymaktadır). Bununla birlikte, bazı DDoS saldırıları materyal kazancı (bir rakibin kaynaklarına zarar vermek veya şirketlere şantaj yapmak için) veya siyasi sebeplerden dolayı yapılmaktadır (savaştaki bir ülke, düşmanın kritik kaynaklarına saldırı düzenleyebilir ve potansiyel olarak bunun için tüm ülkenin bilgisayar gücünün önemli bir bölümüne üye olabilir) (Mirkovic, Reiher, 2004).

3.4.1. DDoS çeşitleri

DDoS saldırısı yapılan bir sistemde DDoS saldırı çeşitlerini saldırı amaçlarına göre ikiye ayırabiliriz. Bu amaçlar kaynak tüketme ve bant genişliğini doldurma olarak tanımlanır. Kaynak tüketme saldırıları hedef sistemin kaynaklarını tüketmeyi amaçlar. Kaynak tüketme saldırılarında hedef sistem kaynakları kötü ve bozuk paketlerle doldurulur. Bant genişliğini doldurmak isteyen saldırılarda hedef ağ istenmeyen ağ paketleri ile doldurulur (Çelikbilek, 2016).



Şekil 3.10. 2018 yılının ilk çeyreğinde DDoS saldırı tiplerinin dağılımı (Khalimonenko, Kupreev, Badovskaya,2018)

Şekil 3.10.'da DDoS saldırılarının türlerinin 2018 yılının ilk çeyreğindeki dağılımı görülmektedir. SYN-DDoS saldırılarının payı % 55.63'ten % 57.3'e hafifçe

artmıştır. ICMP saldırılarının payı neredeyse % 3,2'den % 6,1'e yükselmiştir. Buna göre; UDP, TCP ve HTTP selleri önceki çeyreğe göre % 1-2 oranında düşmüştür (Khalimonenko, Kupreev, Badovskaya, 2018).

DDoS saldırı çeşitleri yapılış şekillerine göre aşağıda açıklanmıştır.

3.4.1.1.ICMP Flood

Bu saldırı ICMP protokolünü hedef alır. ICMP, genel olarak sistemler arası iletişim ve hata ayıklama amacıyla kullanılan bir protokoldür. Bu saldırının özelliği birçok noktadan yapılmasıdır. Genel olarak Linux üzerinde daha başarılı sonuç vermektedir. Komut sisteminde kullanılan ping komutu ile saldırı gerçekleştirilebilir. Ping komutu genelde belirli bir bilgisayar ya da sunucunun internete bağlı çalışır olup olmadığını anlamak için kullanılır. İstemci sistem, hedef sisteme ICMP Echo Request paketi gönderir. Hedef sistemin ulaşılabilir olduğunu anlamak adına hedef sistem ICMP Echo Reply paketi gönderir. Böylece saldırılan makine çok sayıda gelen ICMP Echo Request paketine karşılık vermeye çalıştığı için sistem yorulur ve erişilemez hale gelir.

3.4.1.2.UDP Flood Saldırısı

UDP, hızlı bir protokol olmasına rağmen gönderilen paketin karşı tarafa ulaşıp ulaşmadığını kontrol etmemesinden dolayı güvenli bir protokol değildir. UDP saldırısı hedef makinenin rastgele portlarına çok sayıda ve hızlı bir şekilde gönderilen UDP paketleri ile gerçekleştirilir. Bu durumda hedef makine portu dinleyen uygulama var mı diye kontrol eder, hiçbir uygulamanın o portu dinlemediğini görünce ICMP ‘Hedefe ulaşılamıyor’ paketi ile cevap verir. Böylece hedef sistem çok fazla UDP paketinden dolayı çok fazla ICMP paketi göndermeye zorlanır. Bu da onun diğer istemciler tarafından erişilemez hale gelmesine neden olur. UDP saldırısında gönderilen paketlerin sahte IP adresleri üzerinden gönderilmesi geri dönen ICMP paketlerinin ona ulaşmamasını sağlar ve saldırganın bağlantısını anonimleştirir. Bu saldırıyı yönetmek için istenmeyen ağ trafiğini filtrelemek üzere ağ içindeki kilit noktalara güvenlik duvarları kurulabilir (Özgenç, 2014)

3.4.1.3.Back

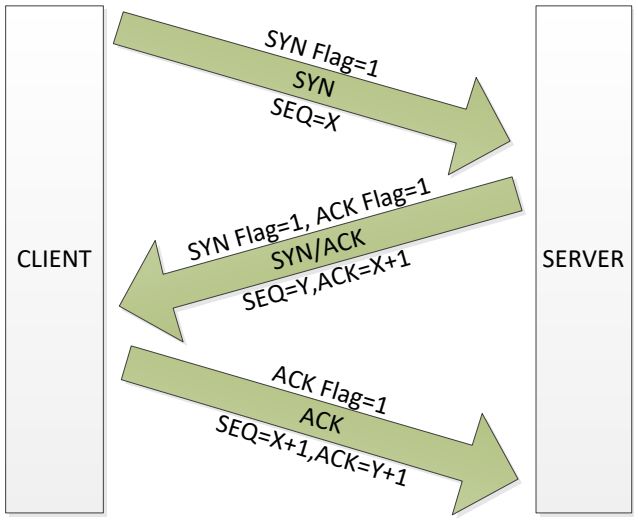
Bu saldırı, URL (Uniform Resource Locator -Birörnek Kaynak Konumlayıcı) açıklamasında çok sayıda ön eğik çizgi (/) karakteri içeren isteklerle sızan bir apache Web sunucusuna karşı başlatılmıştır. Sunucu, tüm bu istekleri işleme koymaya çalıştıkça, diğer meşru istekleri işleme koyamamaktadır ve bu nedenle müşterilerine hizmet reddedilmektedir (Patrikakis, Masikos, Zouraraki, 2004)

3.4.1.4.Smurf

Bu saldırıda, saldırganlar bir internet servisini sömürürler. Bir bilgisayar ya da sunucuya bir ping gönderildiğinde, pingi yollayan tarafa bir cevap paketi yollanır. Bir networke yolladığında networkteki tüm bilgisayarlar cevap verir. Ping edilen network, saldırı hedefi değil saldırı olarak kullanılacak kısımdır. Bir smuf saldırısında, saldırganlar ping isteklerindeki geri dönüş adreslerini kurban bilgisayarın adresleri ile değiştirirler. Böylece hem saldırı gerçekleşir hem de kendini yakalanmaktan korumuş olur.

3.4.1.5.SYN Flood Saldırısı (Neptune)

İnternet üzerindeki bütün bilgisayarların haberleşmesi için en yaygın protokol TCP/IP protokolüdür. TCP protokolü bağlantı temelli bir erişim sağladığı için üç yollu el sıkışma (Threeway Handshake) modeli kullanılır. Bağlantı sağlanacak iki bilgisayar arasında bağlantıyı talep eden taraf karşı tarafa bir senkronize (SYN-Synchronize) paketi gönderir. Karşı taraf ise paketi kabul ettiğini belirten kabul (SYN/ACK) paketi gönderir ve bu sırada bağlantı bilgilerini hafızasında (TCP/IP yığınında) tutar. Daha sonra ilk talebi gönderen tekrar kabul (ACK-Acknowledgement) bilgisini gönderir ve bağlantı kurulumu sağlanmış olur. Bağlantının kurulmasından sonra karşı taraf bu bağlantı bilgisini siler. Şekil 3.11.'de bir TCP haberleşmesi sırasında meydana gelen üçlü el sıkışma işleyişi gösterilmiştir.



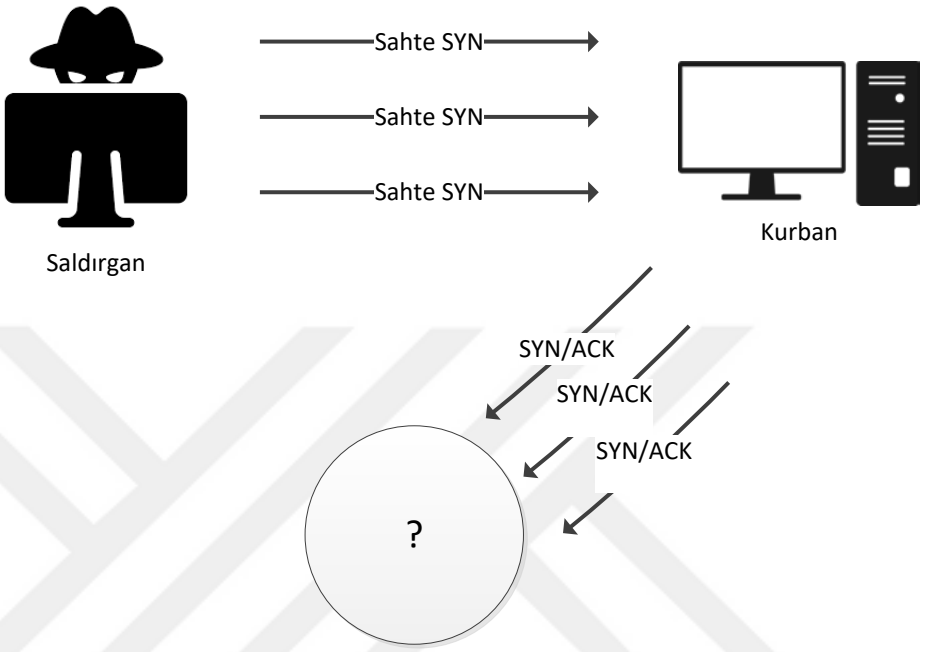
Şekil 3.11. Üçlü el sıkışma işleyişi

İlk adım istemci tarafından gerçekleştirilir. Bağlantıyı başlatacak olan istemci TCP segmenti içindeki SYN bitini 1 olarak ayarlar ve paketleri kontrol etmek için rastgele bir sıra numarası (x) verir. Bu sıra numarası sayesinde, paketler TCP iletişimi meydana gelirken sıralı gelmese bile alıcı tarafın bu paketleri sıraya koymasını sağlar. Özetle istemci sunucuya SYN bayrağı aktif edilmiş ve sıra numarası x olan paketi yollar (Başaranoğlu, 2015).

İkinci adım sunucu tarafından gerçekleştirilir. İstemci tarafından gönderilen paketi alan sunucu, istemciye sonraki paketi hazırlar. Bağlantı istediğini onaylamak için SYN+ACK bitini 1 olarak ayarlar. Ayrıca sunucu, istemciden gelen paketin sıra numarasını bir artırarak göndereceği paketin ACK numarasını 1 artırır (x+1). Böylece sunucu taraf hem istemci tarafında gönderdiği paketin sıraya konulmasını sağlamış olur hem de istemci tarafın bir sonraki göndereceği ve sunucu tarafından kabul edilecek olan sıra numarasını belirlemiş olur. Yani bir sonraki adımda (x+1)'i bekler. Ayrıca sunucu da kendi paketlerini kontrol amaçlı bir SEQ numarası (y) alanına yazılacak bir sıra numarası üretir ve paketi tekrar istemciye yollar. Bu sıra numarası gönderdiği paketin cevabını doğru sıraya koymak içindir. Özetle sunucu istemciye SYN+ACK bayrakları aktif edilmiş ve sıra numarası SEQ numarası (y) ve ACK numarası (x+1) olan paketi yollar (Başaranoğlu, 2015).

Son adım istemci tarafından gerçekleştirilir. İstemciye sunucudan paket geldikten sonra sunucuya göndereceği paketi hazırlar. Göndereceği paketin ACK bayrağını 1 olarak ayarlar. Ayrıca sunucudan gelen paketin SEQ numarasına bakar ve 1 artırarak göndereceği paketin ACK numarası alanına (y+1) olarak ayarlar. Bir önceki adımda

sunucu tarafından gönderilen ACK numarasını, istemcinin göndereceği paketin SEQ numarasına (x+1) eşit olacak şekilde ayarlar. Özetle; istemci sunucuya ACK bayrağı aktif edilmiş ve sıra numarası (x+1), ACK numarası y olan bir paket yollar (Başaranoğlu, 2015)



Şekil 3.12. SYN Flood saldırısı

Şekil 3.12.'de bir SYN Flood saldırı şekli gösterilmiştir. TCP'nin bu çalışma şeklinden kötü amaçlı kullanımı ile SYN Flood saldırısı oluşturulmuştur. Bağlantı kurulumu sırasında karşı tarafa gönderilen SYN paketinde yer alan gönderen kişinin bulunduğu IP adresi kısmına, paketi gönderen IP adresi yerine gerçekte var olmayan bir ip adresi yazılır. Karşı taraf bu SYN paketini alır ve gönderen kısma senkronize ve kabul (SYN+ACK) paketini yollar. Bu sırada karşı taraf gönderen taraftan kabul (ACK) cevabını bekler. Ancak var olmayan IP adresinden cevap gelmez ve IP adresi TCP/IP yığınında bekleme devam eder. Bir süre sonra yığın dolar ve diğer isteklere cevap veremez hale gelir. Böylece SYN Flood saldırısı gerçekleşmiş olur.

3.4.1.6.Land Flood Saldırısı

Land flood saldırı tipinde, saldırganlar hedef sistemin IP ve portlarını, kaynak IP adresi olarak kullanarak ağa paketler yollarlar. Bu paket üç yollu el sıkışma sırasında kurban sisteme ACK onay talebi yollar. Hedef ve kaynak adresleri aynı olduğundan

kurban hem dışarıdan paketi alır hem de kendi talebini yanıtlamak zorunda kalır. Böylece sistem bir paket alması gereken zamanda iki paket alır ve saldırı boyutu iki katına çıkmış olur. Bu durumda sistem kendi kendine yanıt vermeye çalışırken kullanılamaz hale gelir (Özgenç, 2014) (Gezgin, Buluş, 2013).

3.4.1.7.Ping of Death

Temel seviyede bir DoS saldırı tipidir. Günümüzde birçok işletim sisteminin bu saldırıya önlem alınmasına karşı daha önce kullanılan etkili bir yöntemdir. Ping uygulaması kullanılarak IP tespiti sırasında izin verilen maksimum değer olan 65535 bayt üzerinde IP paketinin kullanılmasıyla gerçekleşir. Daha sonra bu paketin ağa gönderilmesiyle işletim sistemi çalışamaz hale gelir (Özgenç, 2014) (Gezgin, Buluş, 2013).

3.4.1.8.Teardrop Saldırısı

IP paketlerinin yeniden birleşmesi sırasında oluşabilecek zafiyetlerden yararlanmak için tasarlanmış bir saldırı tipidir. Veri, ağ üzerinde bir yerden başka bir yere gönderilmeden önce parçalara ayrılır ve bu parçalara sıra numarası verilir. Alıcı gelen paketleri sıra numarasına göre birleştirir. Her bir parça orijinal pakete benzer ve paketler parçalanırken pakette bulunan ofsetler kullanılır. Bu offset değerlerinin birbirleriyle çakışmaması yani eşleşmemesi gerekmektedir. Teardrop saldırılarında, pakete üst üste gelecek şekilde offsetler eklenir. Paketler alıcı tarafından bir araya getirilmeye çalışıldığında çakışan offset değerleri yüzünden sistem durabilir, bozulabilir ya da yeniden başlayabilir. Günümüzdeki çoğu işletim sistemlerinde bu saldırıya karşı dayanıklıdır (Özgenç, 2014) (Gezgin, Buluş, 2013).

3.4.2.Ağ İzleme Araçları

Ağ izleme araçları, ağ üzerinde meydana gelen trafiği, analiz edilmesi için toplar. STS'ler için gerekli olan verilerin toplanması için gerekli olan programları kapsamaktadır. DARPA veri setleri TCPdump programı ile toplanmıştır.

3.4.2.1.TCPdump

TCPdump, Linux işletim sistemlerinde komut sisteminde çalışan bir ağ paket analizi yapan bir programıdır. Kullanıcıya bağlı bir ağ üzerinden gönderilen veya alınan paketleri yakalama ve izleme olanağı oluşturur. Adında TCP geçmesine rağmen hem TCP/IP hem de diğer paketleri analiz edebilir. Paket yakalamak için "libcap" kütüphanesini kullanır. Windows işletim sistemi tarafından kullanılabilen hali Windump'tır ve libpcap'in Windows'a port edilmiş hali olan WinPcap kullanır. TCPdump BSD (Berkeley Software Distribution – Berkeley Yazılım Dağıtım) lisansı altında dağıtılan ücretsiz bir programdır (İTÜBİDB, 2013).

3.4.2.2.Wireshark

Wireshark, gelen ve giden paketleri kullanıcının anlayacağı şekle getiren paket süzme programlarından biridir. TCP/IP ağındaki gelen ve giden paketlerin takip edilmesini sağlar. Ethernet kartından gelen ve giden paketler yakalanır ve bu takip sonucu ile port ve IP numaraları, paket içerikleri gibi bilgilere erişilebilmektedir (İTÜBİDB, 2013).

3.5.Veri Kümeleri

STS'nin saldırıyı tespit edebilmesi için sistemin eğitim ve test verileri ile uygulanabilir hale getirilebilmesi gerekmektedir. İçerisinde saldırı verileri içeren bu veri kümelerini bazı geliştiriciler kendileri oluştursa da oldukça zor ve maliyetli bir iştir. Ayrıca STS için teknikler üretmenin yanında bir de veri kümesi oluşturmaya çalışmak oldukça işgücü ve zaman harcamayı gerektirmektedir. Bunun için hem standart geçerli bir veri kümesi oluşturmak hem de çalışmaları hızlandırmak adına veri kümeleri hazırlanmıştır.

Bir ağ trafiği genel olarak bir koklayıcı (sniffer) kullanarak gözlemlenebilir. Ağ paketlerini gözlemlemek ağ trafiğini anlamak için yeterli olmayabilir. Literatürde STS'lerin testleri için geliştirilen birkaç veri seti bulunmaktadır. Bu veri setlerinden bazıları KDD CUP 99, DARPA 1998, DARPA 1999, UNM, SSCNNJU, CUCS, Windows sistem ve network tcpdump data veri setleridir (Çetin, YILDIZ, 2014).

İzinsiz giriş tespit testi ve değerlendirmesi için halka açık birçok veri seti bulunmaktadır. Bununla birlikte, en yaygın olarak kullanılanlar, DARPA (% 24) ve KDD seti (% 28) ki bunlar birlikte % 50'den fazla çalışmada kullanılmıştır. DARPA veri setleri, özellikle test amaçlı 1998, 1999 ve 2000 yıllarında MIT (Massachusetts Institute of Technology) Lincoln Labs'ta üretildi. Setler, simüle edilmiş ana bilgisayar, ağ normal trafiğinden ve manuel olarak oluşturulan ağ tabanlı saldırılardan oluşur. KDD CUP 99 saldırı verileri olarak bilinen KDD seti, DARPA 98 veri setinden türetilmiştir (Tavallaee, Stakhanova, Ghorbani, 2010).

Şekil 3.13. Ana veri kümeleri ve çıkarılan veri kümeleri arasındaki ilişki (Özgür, Erdem, 2016)

Tablo 3.1. DARPA, KDD CUP 99, NSL-KDD veri kümeleri bilgileri (Özgür, Erdem, 2016)

Adı	Öğrenme Boyutu	Test Boyutu	Not
DARPA	6.591.458 kb (6.2 gb)	3.853.522 kb (3.67 gb)	Temel Veri Kümesi. Ham TCP/IP Dökümü Dosyaları
KDDCUP99	4898431	311029	Makine öğrenimi için çıkarılan ve önceden işlenen özellikler
NSL-KDD	125973	22544	Kopyalar kaldırıldı, boyut azaltıldı.

Şekil 3.13. ve Tablo 3.1. bu çalışmada ilgili veri kümeleri için (DARPA, KDD CUP 99 ve NSL-KDD) genel özetini vermektedir. DARPA temel ham veri kümesidir. KDD CUP 99, DARPA veri kümesinin özellik ayıklanmış halidir. NSL-KDD, KDD CUP 99 veri kümesinin kaldırılmış ve boyut küçültülmüş kopyalarıdır (Özgür, Erdem, 2016).

3.5.1.DARPA

IDEVAL (Intrusion Detection Evaluation) STS'lerin değerlendirilmesi ve karşılaştırılması için geliştirilen ilk standart veri kümesi gövdesidir. MIT Lincoln Laboratuvarları, Bilgi Sistemleri Teknolojisi (Information Systems Technology) grubunun, DARPA (Defence Advanced Research Projects Agency) ve AFRL (Air Force Research Projects Agency) desteğiyle yürüttüğü çalışmaların sonucunda oluşturulmuştur. Bu çalışma sonucunda STS'lerin test edilmesi sağlanarak test edilen her bir sistem için

tespit olasılığı ve yanlış alarm olasılığı ölçütleri belirlenmiştir. Bu ölçütler, yeni çalışmalara yön göstermiş ve nesnel bir ölçüm sağlamıştır (Güven, 2007).

1998-2000 yılları arasında Lincoln laboratuvarlarında STS'yi ölçmek için çeşitli çalışmalar yapılmıştır. DARPA, STS çalışmaları kullanılarak gerçek zamanlı olmayan ve gerçek zamanlı olan iki farklı değerlendirme yapabilen 1998 ve 1999 DARPA IDEVAL veri kümeleri oluşturmuştur. 1998 IDEVAL veri kümesinin gerçek zamanlı olmayan kısmı için 1998 DARPA veri kümesi geliştirilmiştir. 1998'de geliştirilen veri kümesinin değiştirilmesi ile DARPA 1999 veri kümesi oluşturulmuştur.

3.5.2.KDD CUP 99

KDD CUP (Knowledge Discovery and Data Mining), veri madencilerinin önde gelen profesyonel organizasyonu olan Bilgi Bulma ve Veri Madenciliği konusundaki ACM (Association for Computing Machinery) Özel İlgi Grubu tarafından düzenlenen yıllık Veri Madenciliği ve Bilgi Buluşması yarışmasıdır.

1998 DARPA Saldırı Tespiti Değerlendirme Programı, MIT Lincoln Labs tarafından hazırlanmış ve yönetilmiştir. Hedef, saldırı tespitinde araştırmayı incelemek ve değerlendirmektir. Askeri bir ağ ortamında taklit edilen çok çeşitli müdahaleleri içeren denetlenecek standart bir veri seti sağlanmıştır. 1999 KDD saldırı tespit yarışması bu veri kümesinin bir sürümünü kullanmıştır.

Lincoln Laboratuvarları, tipik bir ABD Hava Kuvvetleri Yerel Ağ simülasyonunu gerçekleştiren bir yerel alan ağı (LAN) için dokuz hafta boyunca ham TCP dump verileri elde etmek için bir ortam oluşturdular. LAN'ı gerçek bir Hava Kuvvetleri ortamı gibi çalıştırdılar, ancak birden fazla saldırı gönderdiler. Ham eğitim verileri yedi haftalık ağ trafiğinden gelen yaklaşık dört gigabayt sıkıştırılmış ikili TCP dökümü verisi idi. Bu yaklaşık beş milyon bağlantı kaydına dönüştürüldü. Benzer şekilde, iki haftalık test verileri yaklaşık iki milyon bağlantı kaydı sağladı.

Bağlantı, bazı iyi tanımlanmış zamanlarda başlayıp biten bir TCP paket dizisidir; veri paketleri arasında veri, bir kaynak IP adresinden bir hedef IP adresine ve bazı iyi tanımlanmış protokol altında gönderilir. Her bağlantı normal veya bir saldırı olarak tam bir özellikli saldırı türü ile etiketlenir. Her bağlantı kaydı yaklaşık 100 bayttan oluşur. KDD CUP 99 veri setinde 41 öznitelik bulunmaktadır. 42nci öznitelik, ağ bağlantı vektörlerinin çeşitli 5 sınıfı hakkındaki verileri içerir ve bir normal sınıf ve dört saldırı

sınıfı olarak sınıflandırılır. 4 saldırı sınıfı ayrıca DoS, Probe, R2L ve U2R olarak gruplandırılmıştır.

Saldırılar dört ana kategoriye ayrılır:

1. Denial of Service Attack (DoS): Saldırganın meşru istekleri işlemek için çok meşgul veya çok dolu olması veya meşru kullanıcıların bir makineye girmesini engellediği bir saldırıdır (Tavallaee, Bagheri, Lu, Ghorbani, 2009).

2. Kullanıcıdan Kök Saldırıya (U2R): Saldırganın sistemdeki normal bir kullanıcı hesabına (belki de şifreleri koklama, sözlük saldırısı veya sosyal mühendislik tarafından elde edilen) erişimi olan bir sömürü sınıfıdır ve sisteme kök erişimini sağlamak için bazı güvenlik açığından yararlanır (Tavallaee, Bagheri, Lu, Ghorbani, 2009).

3. Uzaktan Yerel Saldırıya (R2L): Bir ağ üzerinden bir makineye paket gönderme olanağına sahip ancak bu makineye bir hesabı olmayan bir saldırgan, bazı güvenlik açığından yararlanarak o makinenin kullanıcısı olarak yerel erişim elde etmesi durumunda ortaya çıkar (Tavallaee, Bagheri, Lu, Ghorbani, 2009).

4. Probing Saldırısı: Güvenlik kontrollerini engellemek için açıkça bir bilgisayar ağı hakkında bilgi toplama girişimidir (Tavallaee, Bagheri, Lu, Ghorbani, 2009).

Test verilerinin eğitim verisi ile aynı olasılık dağılımına sahip olmadığına ve eğitim verilerinde olmayan belirli saldırı tiplerini içerdiğine dikkat etmek önemlidir. Bu, görevi daha gerçekçi yapar. Bazı saldırı uzmanları, yeni saldırıların çoğunun bilinen saldırıların varyantları olduğuna ve bilinen saldırıların "imzasının" yeni değişkenleri yakalamak için yeterli olabileceğine inanmaktadır (KDD, 2018).

3.5.3.NSL-KDD

KDD CUP 99 anomali tespiti için en çok kullanılan veri kümesidir. KDD CUP 99 veri kümesinde karşılaşılan çeşitli dezavantajlar ve çeşitli istatistiksel analizler araştırmacılar tarafından modellenen birçok IDS'in (Intrusion Detection System) tespit doğruluğunu etkiledi. NSL-KDD, tam KDD CUP 99 veri setinin seçilmiş kayıtlarından oluşan yeni bir veri seti oluşturuldu.

Orijinal KDD veri seti üzerinden NSL-KDD'nin avantajları şunlardır:

- Öğrenme setinde gereksiz kayıtlar bulunmamaktadır, bu nedenle sınıflandırıcılar daha sık kayıtlara yöneltilmeyecektir.
- Her bir zorluk seviyesi grubundan seçilen kayıtların sayısı, orijinal KDD veri setindeki kayıt yüzdesi ile ters orantılıdır. Sonuç olarak, farklı makine

öğrenme yöntemlerinin sınıflandırma oranları, farklı öğrenme tekniklerinin doğru bir şekilde değerlendirilmesini daha verimli hale getiren daha geniş bir aralıkta çeşitlilik göstermektedir.

- Öğrenme ve test setlerindeki kayıt sayısı mantıklıdır ve bu da denemeleri küçük bir bölümünü rasgele seçmek zorunda kalmadan komple set üzerinde çalıştırmayı uygun kılar. Sonuç olarak, farklı araştırma çalışmalarının değerlendirme sonuçları tutarlı ve karşılaştırılabilir olacaktır (Datti, Verma, 2010).

Her kayıtta akışın farklı özelliklerini açığa çıkaran 41 özellik ve her birine bir saldırı türü veya normal olarak atanan bir etiket bulunur. Öznitelik adı, açıklamaları ve örnek verileri özniteliklerin ayrıntıları Tablo 3.2, 3.3, 3.4, 3.5'de listelenmiştir. (Dhanabal, Shantharajah, 2015).

1.Temel Özellik: Bu kategori bir TCP / IP bağlantısından çıkartılmış tüm öznitelikleri kapsar. Bu özelliklerin çoğu algılamada gizli bir gecikmeye neden olur.

Tablo 3.2. Her ağ bağlantı vektörünün temel özellikleri

Özellik No	Özellik Adı	Tanım	Örnek Veri
1	Duration	Bağlantı süresinin uzunluğu	0
2	Protocol_type	Bağlantıda kullanılan protokol	TCP
3	Service	Kullanılan hedef ağı hizmeti	ftp_data
4	Flag	Bağlantı durumu - Normal veya Hata	SF
5	Src_bytes	Tek bağlantıda kaynaktan hedefe aktarılan veri baytı sayısı	491
6	Dst_bytes	Tek bağlantıda hedeften kaynağa aktarılan veri baytı sayısı	0
7	Land	Kaynak ve hedef IP adresleri ve bağlantı noktası numaraları eşitse, bu değişken 1 değilse 0 değer alır	0
8	Wrong_fragment	Toplam yanlış parça sayısı	0
9	Urgent	Acil paketlerin sayısı. Acil paketler, acil bit etkinleştirilmiş paketlerdir.	0

2.İçerik Özellikleri: DoS ve Probing saldırılarının çoğunun aksine, R2L ve U2R saldırılarının sık karşılaşılan sıralı kalıplara uygun herhangi bir saldırıları bulunmamaktadır. Bunun nedeni, DoS ve Probing saldırıları, çok kısa bir süre içinde bazı host veya hostlar ile birçok bağlantı içerir; ancak R2L ve U2R saldırıları, paketlerin veri bölümlerine gömülür ve normal olarak yalnızca tek bir bağlantı içerir. Bu tür saldırıları tespit etmek için, veri bölümündeki şüpheli davranışları (Örneğin Başarısız oturum açma denemeleri) aramak için bazı özelliklere ihtiyaç vardır. Bu özelliklere içerik özellikleri denir (Tavallaee, Bagheri, Lu, Ghorbani, 2009).

Tablo 3.3. Her ağ bağlantı vektörünün içeriğe ilişkili özellikleri

Özellik No	Özellik Adı	Tanım	Örnek veri
10	Hot	İçerikte "hot" göstergeler sayısı: bir sistem dizini girişi, program oluşturma ve programları yürütme	0
11	Num_failed_logins	Başarısız giriş denemelerinin sayısı	0
12	Logged_in	Oturum Açma Durumu: Başarılı bir şekilde oturum açılmışsa 1; değilse 0	0
13	Num_compromised	Tehlikeli durumların sayısı	0
14	Root_shell	Eğer root shell elde edilirse 1; değilse 0	0
15	Su_attempted	Eğer "su root" komutu denenmiş ya da kullanılmışsa 1; Aksi halde 0	0
16	Num_root	Bağlantıda root olarak gerçekleştirilen "root" erişim sayısı veya işlem sayısı	0
17	Num_file_creations	Bağlantıda dosya oluşturma işlemleri sayısı	0
18	Num_shells	Shell komut istemlerinin sayısı	0
19	Num_access_files	Erişim kontrol dosyalarındaki işlem sayısı	0
20	Num_putbound_cmds	FTP oturumunda giden komutların sayısı	0
21	Is_hot_login	Eğer giriş "hot" listesine yani root veya admin'e aitse 1; yoksa 0	0
22	Is_guest_login	Giriş bir "guest" girişiyse 1;değilse 0	0

3. Trafik özellikleri: Bu kategori bir pencere aralığına göre hesaplanan özellikleri içerir ve iki gruba ayrılır:

a) "Aynı Ana Bilgisayar" özellikleri: Geçerli bağlantıyla aynı hedef ana bilgisayara sahip yalnızca son 2 saniyedeki bağlantıları inceleyen özelliklerdir ve protokol davranışı, hizmet vb. ile ilgili istatistikleri hesaplar.

b) "Aynı Servis" özellikleri: Geçerli bağlantıyla aynı servise sahip yalnızca son 2 saniyedeki bağlantıları inceler.

Yukarıda bahsedilen iki "trafik" türüne zamana dayalı denir. Bununla birlikte, ana makineyi (veya bağlantı noktalarını) 2 saniyeden daha büyük bir zaman aralığı, örneğin dakikada bir olmak üzere tarayan birkaç yavaş tarama saldırısı vardır. Sonuç olarak, bu saldırılar, 2 saniyelik bir zaman penceresi ile saldırı desenleri oluşturmaz. Bu sorunu çözmek için "aynı ana bilgisayar" ve "aynı servis" özellikleri yeniden hesaplanır. Ancak 2 saniyelik bir zaman aralığı yerine 100 bağlantıların bağlantı penceresine dayanır. Bu özelliklere bağlantı tabanlı trafik özellikleri denir (Datti, Verma, 2010).

Tablo 3.4. Her ağ bağlantı vektörünün zamanı ile ilgili trafik özellikleri

Özellik No	Özellik Adı	Tanım	Örnek veri
23	Count	Son iki saniye içinde geçerli bağlantı ile aynı hedef bilgisayara olan bağlantıların sayısı	2
24	Srv_count	Son iki saniyede geçerli bağlantı ile aynı hizmete (bağlantı noktası numarası) olan bağlantı sayısı	2
25	Serror_rate	Count (23) içinde toplanan bağlantılar arasında flag (4) s0, s1, s2 veya s3 etkinleştiren bağlantı yüzdesi,	0
26	Srv_serror_rate	Srv_count (24) içinde toplanan bağlantılar arasında flag (4) s0, s1, s2 veya s3 etkinleştiren bağlantı yüzdesi	0
27	Rerror_rate	Count (23) içinde toplanan bağlantılar arasında bayrağı (4) REJ etkinleştiren bağlantı yüzdesi	0
28	Srv_rerror_rate	Srv_count (24) içinde toplanan bağlantılar arasında bayrağı (4) REJ etkinleştiren bağlantı yüzdesi	0
29	Same_srv_rate	Count (23) içinde toplanan bağlantılar arasında aynı servise olan bağlantı yüzdesi	1
30	Diff_srv_rate	Count (23) içinde toplanan bağlantılar arasında farklı servise olan bağlantı yüzdesi	0
31	Srv_diff_host_ rate	Srv_count (24) içinde toplanan bağlantılar arasında farklı hedef makineleri olan bağlantı yüzdesi	0

Tablo 3.5.Ağ bağlantı vektöründe host temelli trafik özellikleri

Özellik No	Özellik Adı	Tanım	Örnek Veri
32	Dst_host_count	Aynı hedef host IP adresine sahip bağlantıların sayısı	150
33	Dst_host_srv_count	Aynı port numarasına sahip bağlantıların sayısı	25
34	Dst_host_same_srv_rate	Dst_host_count (32) içinde toplanan bağlantılar arasında aynı servise olan bağlantı yüzdesi	0.17
35	Dst_host_diff_srv_rate	Dst_host_count (32) içinde toplanan bağlantılar arasında farklı hizmetlere olan bağlantı yüzdesi	0.03
36	Dst_host_same_src_port_rat e	Dst_host_srv_count (33) içinde toplanan bağlantılar arasında aynı kaynak bağlantı noktasına olan bağlantı yüzdesi	0.17
37	Dst_host_srv_diff_host_rate	Dst_host_srv_count (33) içinde toplanan bağlantılar arasında farklı hedef makinelere olan bağlantı yüzdesi	0
38	Dst_host_serror_rate	Dst_host_count (32) içinde toplanan bağlantılar arasında bayrağı (4) s0, s1, s2 veya s3 etkinleştiren bağlantı yüzdesi	0
39	Dst_host_srv_serror_rate	Dst_host_srv_count (33) biriminde toplanan bağlantılar arasında bayrağı (4) s0, s1, s2 veya s3 etkinleştiren bağlantı yüzdesi	0
40	Dst_host_rerror_rate	Dst_host_count (32) biriminde toplanan bağlantılar arasında bayrağı (4) REJ etkinleştiren bağlantı yüzdesi	0.05
41	Dst_host_srv_rerror_rate	Dst_host_srv_count (33) biriminde toplanan bağlantılar arasında bayrağı (4) REJ etkinleştiren bağlantı yüzdesi	0

KDD CUP 99 ve NSL-KDD 4 çeşit saldırı tipi içerir. Bunlar DoS, U2R, R2L ve Probe'dir. Tablo 3.6'da KDD CUP 99 ve NSL-KDD'deki saldırı kategorileri ve saldırı tipleri eğitim ve test verilerindeki sayıları gösterilmiştir.

Tablo 3.6.KDD CUP 99 ve NSL-KDD'deki saldırı tiplerinin dağılımı (Aghdam, Kabiri, 2016)

Saldırı Kategori	Saldırı Tipi	KDDCUP 99		NSL-KDD	
		Öğrenme Seti kddcup.data10percent	Test Seti corrected.gz	Öğrenme seti KDDTrain+20Percent	Test Seti KDDTest+
Denial of Service (DoS)	Neptune	107201	58001	8282	4657
	Smurf	164091	280790	529	665
	Pod	264	87	38	41
	Teardrop	979	12	188	12
	Land	21	9	1	7
	Back	2203	1098	196	359
	Apache2	–	794	–	737
	Udpstorm	–	2	–	2
	Process-table	–	759	–	685
	Mail-bomb	–	5000	–	293
User to Root (U2R)	Buffer-overflow	30	22	6	20
	Load-module	9	2	1	2
	Perl	3	2	0	2
	Rootkit	10	13	4	13
	Spy	2	–	1	–
	Xterm	–	13	–	13
	Ps	–	16	–	17
	Http-tunnel	–	158	–	133
	Sql-attack	–	2	–	2
	Worm	–	2	–	2
	Snmp-guess	–	2406	–	331
Remote to Local (R2L)	Guess-password	53	4367	10	1231
	Ftp-write	8	3	1	3
	Imap	12	1	5	1
	Phf	4	2	2	2
	Multihop	7	18	2	18
	Warezmaster	20	1602	7	944
	Warezclient	1020	–	181	–
	Snmpgetattack	–	7741	–	178
	Named	–	17	–	17
	Xlock	–	9	–	9
	Xsnoop	–	4	–	4
	Send-mail	–	17	–	14
Probe	Port-sweep	1040	354	587	157
	IP-sweep	1247	306	710	141
	Nmap	231	84	301	73
	Satan	1589	1633	691	735
	Saint	–	736	–	319
	Mscan	–	1053	–	996

3.6.Makine Öğrenmesi

Makine öğrenmesi, yapay zekâ ve derin öğrenme günümüzde oldukça popüler olan ve aslında birbirleri ile iç içe geçmiş kavramlar olarak düşünülebilir. Yapay zekâ makine öğrenmesinden daha eski bir kavramdır. Makine öğrenmesi, yapay zekânın bir alt kolu olarak düşünülebilir. 1950 yılında Alan Turning, makinelerin düşünüp düşünememesi konusunu ortaya atmıştır. Eğer bir insan karşılaşmış olduğu bir etkileşimin insan tarafından mı yoksa makine tarafından mı meydana geldiğini anlamıyorsa bu makinenin zeki olduğunu açıklar. Daha sonra 1959 yılında John McCarthy düzenlediği bir konferansta yapay zekâ kavramından bahsetmiştir (Şener, 2017). Yapay zeka, insanlar gibi düşünen veya davranan makineleri içerir. Yapay zekâ, makine öğrenmesi dışında doğal dil işleme, robotik, bilgi tabanları gibi çalışmaları kapsamaktadır. 1959 yılında Arthur Samuel makine öğrenmesini, makinelerin programlamadığı sonuçları bile öğrenebilme yeteneği olarak tanımlamıştır. Arthur Samuel bilgisayar ortamında çalışabilen, oynanan oyunlardaki verileri analiz edip karşılaştırarak iyi ve kötü pozisyonları öğrenen bir dama oyunu geliştirdi.

Makine öğrenmesi, bir algoritma kullanarak verileri ayrıştırma daha sonra bu ayrıştırılan verileri öğrenme ve dünyadaki herhangi bir şey için belirleme ve tahmin yapma uygulamasıdır. Makine öğrenmesi, belirli bir görevi yerine getirmek için belirli talimatları kodla yazmak yerine, görevi yerine getirmek için öğrenme yeteneği kazandıran büyük sayıda veri ve algoritma kullanan eğitilmiş makineler meydana getirir (Hasırcıoğlu, 2017). Makine öğrenmesinin popülerliğini sağlayan en önemli uygulamalardan bir tanesi resim tanımadır. İlk önce makine çeşitli eğitim verileri ile eğitiliyor. Bir resmi tanıması için binlerce kez benzer resimler gösteriliyor. Böylece makine benzer resimlerin ortak özelliklerini tespit ederek resim tanımlanmasını gerçekleştirebiliyor (Şener,2017).

Günümüzde pek çok alanda makine öğrenmesi uygulamalarına rastlamaktayız. Bu konuda en çok bilinen Google Search uygulamasıdır. Google arama motorunda arama işlemi gerçekleştirirken yanlış bir sorgu yazdıysanız size yazdığınız sorgu ile alakalı bir tavsiye uyarısı çıkaracaktır. Buradaki tavsiyeye tıklarsanız algoritma bunu öğrenip bir dahaki yazım hatalarında sizi düzeltmeyi öğrenir. Ayrıca tavsiye motorları (alışveriş, sosyal medya), sıralamalar (reklamlar, arama) ve kişisel asistanlar (Siri) gibi birçok alanda karşımıza çıkmaktadır (Saunders A. A., 2017).

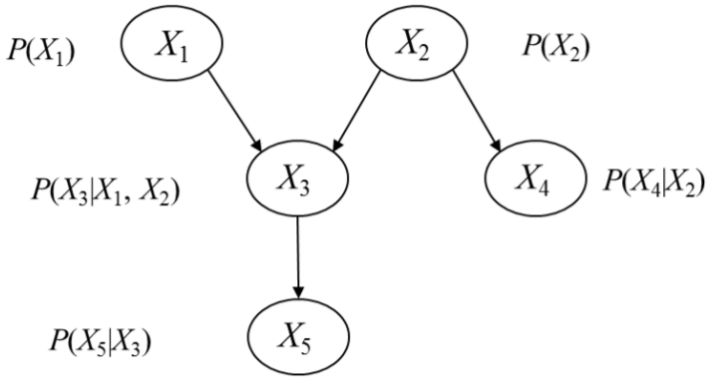
STS'lerde saldırıyı önceden tahmin etmek adına veri kümeleri kullanılarak makine öğrenmesi teknikleri kullanılmaktadır. Literatürde, STS'lerde sıklıkla kullanılan makine öğrenme teknikleri aşağıdaki gibidir.

- Naive Bayes sınıflama
- Destek vektör makinesi
- Yapay sinir ağları

3.6.1.Naive Bayes Sınıflandırma

NB sınıflandırma algoritması İngiliz matematikçi Thomas tarafından geliştirilmiştir ve adını ondan alan bir sınıflandırma algoritmasıdır. Olasılık ilkelerine dayanarak verilerin sınıfını yani kategorisini tespit etmeyi amaçlayan bir sınıflandırıcıdır.

Bu sınıflandırma algoritmasında belirli bir öğretilmiş veri sunulur. Bu verilerin bir sınıfı bulunmalıdır. Öğretilmiş veriler ile yapılan olasılık işlemlerindeki değerlere göre sınıflandırma için sisteme sunulan yeni veriler işletilir ve hangi kategoride olduğu tespit edilmeye çalışılır. Öğretilmiş veri sayısı ne kadar çok ise doğruluk oranı o kadar artar (Usta, 2014).



Şekil 3.14. Beş değişkenli örnek NB yapısı (Çinicioğlu, Atalay, Yorulmaz, 2013).

NB ağlarında grafiksel kısım ağın yapısını oluşturmaktadır. Ağdadaki her hangi iki düğüm birbirine ok ile bağlanmaktadır. Okun başında bulunan düğüm ebeveyn düğüm, okun bitişinde bulunan düğüm ise çocuk düğüm olarak adlandırılmaktadır. Şekil 3.14.'te X_1, X_2, X_3, X_4 ve X_5 değişkenlerinden oluşan örnek bir NB ağının gösterimi yapılmaktadır. Bu ağda X_1 ve X_2 değişkenleri X_3 değişkeninin ebeveyni, X_3 değişkeni ise X_5 değişkeninin ebeveynidir. Ayrıca X_4 değişkeni X_2 değişkeninin çocuk değişkenidir.

$P(X_1)$, $P(X_2)$, $P(X_3| X_1, X_2)$, $P(X_4 | X_2)$ ve $P(X_5 | X_3)$ değişkenlerin sahip oldukları koşullu olasılık dağılımlarıdır. Ağdaki herhangi değişkenin, diğer bir değişkenle arasında herhangi bir ok olmaması o değişkenin ağda yer alan diğer değişkenlerle arasında olasılıksal bir bağ bulunmadığını yani ağda uç olasılık dağılımı bulunduğunu gösterir (Çinicioğlu, Atalay, Yorulmaz, 2013).

$$P(X_4|X_2) = \frac{P(X_2|X_4)P(X_4)}{P(X_2)} \quad (3.1)$$

NB sınıflandırma algoritması teoremi denklem 3.1'de gösterilmektedir. Bu denkleme göre;

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

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

$P(X_4)$ = X_4 eğitim verisini olasılığı

$P(X_2)$ = X_2 eğitim verisini olasılığı olarak ifade edilmektedir.

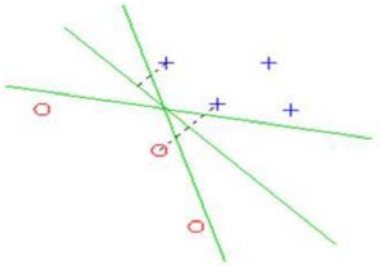
NB sınıflandırıcıları, saldırıları tespit etmede sıklıkla kullanılan yöntemlerden biridir (Altwaijry, Algarny,2012) (Mukherjee, Sharma, 2012).

3.6.2.Destek Vektör Makinesi

DVM, Vladimir Vapnik tarafından önerilen bir etkili ve basit öğrenme algoritmasıdır. Temel istatiksel yöntemlere dayanır. DVM, bir düzlemde yer alan iki sınıfa ayrılmış örnekleri en iyi ayıran destek vektörleridir. İki sınıfı ayıran sonsuz sayıda doğru olabilir ama en iyi ayıran doğru iki sınıfa da belirli bir uzaklıkta olan doğrudur. DVM yüz tanıma, ses analizi gibi birçok alanda kullanılmaktadır. STS alanında sınıflandırıcı olarak kullanılmaktadır (Tan, Tan, Li, 2016) (Enache, Patriciu, 2014)

DVM'de sınıflandırma için bir düzlemde bulunan iki grup arasına bir sınır çizilerek ayrılmaktadır. Bu iki grup arasına çizilecek çizgi iki grubun üyelerine en uzak olan yer olmalıdır. Sınıflandırmayı belirleyen en uygun çizgi DVM sayesinde belirlenmektedir.

DVM'de etiketli veriler girdi olarak kullanılmaktadır. İki sınıflı veri setlerinde çıkış verileri için iki sınıf oluşturulmaktadır. Her bir veri iki gruptan birisinde olacak şekilde gruplandırılmaktadır. Gruplandırılan örnekler bir düzlem üzerinde ayrılmakta ve yeni örneklerin hangi gruba dahil olacağı tahmin edilmektedir.



Şekil 3.15.Sınıflandırılacak verileri ayıran farklı düzlemler (Ayhan, Erdoğmuş, 2014).

Şekil 3.15'te giriş verilerini ayıran düzlemlerle ilgili bir geometrik çizim gösterilmektedir. Şekilde görüldüğü üzere farklı sınıflara ait verileri birbirinden ayıran birçok düzlem olabilmektedir. Bu durumda destek vektör makineleri, farklı sınıflara sahip destek vektörleri arasındaki uzaklığı maksimize eden ayırma hiper düzleminin bulunmasını amaçlamaktadır (Ayhan, Erdoğmuş, 2014).

Doğrusal olarak ayrılan iki sınıflı bir sınıflandırma probleminde destek vektör makinelerinin eğitimi için k sayıda X eğitim verisi için $\{x_i,y_i\}$,i=1,….,k olduğu kabul edilirse, optimum hiper düzlem için aşağıdaki eşitsizlikler kullanılmaktadır.

$$w.xi + b \geq +1 \text{ her } y = +1 \text{ için} \quad (3.2)$$

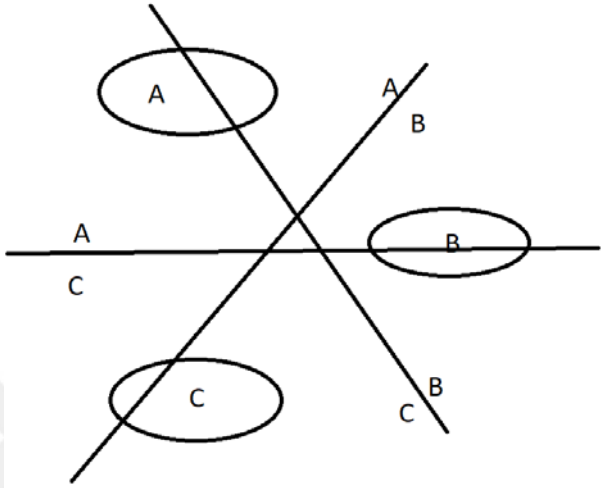
$$w.xi + b \leq +1 \text{ her } y = +1 \text{ için} \quad (3.3)$$

Denklem 3.1 ve 3.2.'de x $\in R^n$ olup N boyutlu bir uzayı, y$\in \{-1,+1\}$ sınıf etiketlerini, w ağırlık vektörünü ve b eğilim değerini göstermektedir. Optimum hiper düzlemin belirlenebilmesi için sınırları oluşturan ve bu düzleme paralel iki hiper düzlem belirlenmektedir. Bu hiper düzlemi oluşturan noktalar destek vektörleri olarak adlandırılır ve bu düzlemler $w.x_i+b=\pm1$ denklemi ile gösterilmektedir (Kavzoğlu, Çölkesen, 2010).

Destek vektör makineleri iki sınıflı ve çok sınıflı sınıflandırma problemlerini çözmek için kullanılabilmektedir. Çok sınıflı DVM, sınıflandırılacak grupların ikiden fazla olmasıdır. Bu durumda aşağıdaki üç yaklaşım tercih edilebilir (Şeker,2008):

- Problemin ikili gruplara indirilmesi (one-to-one, bire bir yaklaşım)
- Problemin tek gruptan bütün gruplara modellenmesi (one-to-many, bire çok yaklaşım)
- Çok sınıf sıralama DVM'leri (multiclass ranking SVM)

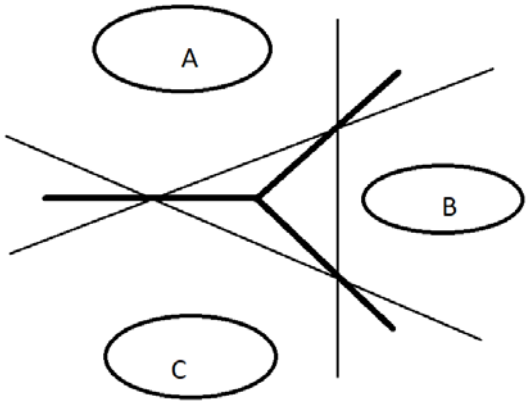
Bu yaklaşımlardan en çok kullanılan ilk yaklaşım olan bire bir yaklaşımda sınıflar kendi içinde ikili gruplara ayrılarak eğitilir. Örneğin bir girdinin 3 sınıftan hangisine gireceğini hesaplamak istediğimiz bir sınıflandırmada sınıfları A,B,C olarak ayrılır. Daha sonra ikili gruplar halinde DVM yöntemi ile eğitilerek sınıfların birbirlerine göre DVM çarpanları çıkarılır (Şeker,2008).



Şekil 3.16. Bire bir yaklaşımlı çoklu DVM

Şekil 3.16. 'da ikili grupların çarpanları gösterilmektedir. Burada her ikili grup ayrı DVM ile eğitilmiştir. Yeni bir verinin sınıflandırılmasında ikili olarak AB, BC ve AC ikili grupları sorgulanır. Sorgulama sonucunda hangisinden cevap alınırsa o sınıfa ait olarak sınıflandırılır. Örneğin yeni gelen örnek sonucunda AB->A, BC->B, AC->A olarak sonuçlar verdiğinde 3 sorgudan ikisi A sonucu verdiği için yeni örnek A sınıfına ait olduğu söylenebilir (Şeker,2008).

Çoklu sınıflandırma yaklaşımlarından ikincisi olan bire çok yaklaşıma göre sınıflandırma mantığı "kazanan hepsini alır" olarak düşünülebilir. Bu yaklaşımda yeni gelen bir verinin hangi sınıfta olacağı diğerlerine göre daha baskın sınıf olarak seçilebilir. Bu yaklaşım bire bir yaklaşıma göre daha az tercih edilmektedir (Şeker,2008).



Şekil 3.17. Bire çok yaklaşımlı çoklu DVM

Şekil 3.17'de 3 sınıf arasında DVM sınıflandırması gerçekleştirilmektedir. Önce 3 adet doğru denklemi bulunmuş (ince çizgiler) sonra bu doğru denklemlerinin birleştirilmesi ile 3 sınıfı ayırmaya yarayan genel bir DVM (kalın çizgiler) elde edilmiştir (Şeker,2008).

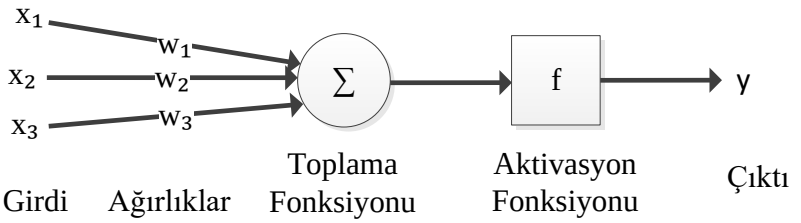
Üçüncü yaklaşım olan çoklu sınıflı sıralama yaklaşımında sınıflar üzerinde bir DVM çalışarak bu sınıfların bütün verilerini içeren bir sınıflandırma yapılmaya çalışılmaktadır. Ancak bu yaklaşım diğer iki yaklaşıma göre çok az tercih edilmektedir. Çünkü bu yaklaşımda eğitim süresi çok fazla ve sonucu bulamama gibi ihtimaller bulunmaktadır (Şeker,2008).

3.6.3.Yapay Sinir Ağları

YSA, basit bir şekilde insan beynini taklit eder. YSA, biyolojik sinir hücrelerinin çalışmasını örnek alan bir makine öğrenmesi tekniğidir. Ağı oluşturan her bir elemana yapay sinir denilmektedir. Yapay sinirler çeşitli ağırlıklandırmalar ile birbirine bağlanarak yapay sinir ağını oluştururlar. Beynin öğrenme, hatırlama, düşünme, genelleme yapma gibi işlevlerini gerçekleştiren yazılımlardır. Genelleme, eğitim ya da öğrenme sırasında karşılaşılmayan veriler için YSA'nın uygun tepkiler vermesidir. Günümüzde birçok bilim alanında YSA'nın yer almasına neden olan birçok özelliği vardır. Doğrusal olmama, paralel çalışma, öğrenme, genelleme, uyarlanabilme ve hata toleransı bu gibi özellikleridir (Ergezer, Dikmen, Özdemir, 2003).

Biyolojik sistemlerde öğrenme nöronlar arasındaki sinaptik bağlantıların ayarlanması ile olur. İnsanlar hayatları boyunca öğrenme süreci içerisindedir ve beyin

sürekli bir gelişme gösterir. Tecrübe arttıkça sinaptik bağlantılar ayarlanır veya yeni bağlantılar oluşur. YSA'larda öğrenme eğitilerek yani girdi ve çıktı verilerinin işlenmesi ile olur. Eğitme algoritması bu verileri kullanarak bağlantı ağırlıkları bir yakınsama sağlayıncaya kadar tekrar tekrar ayarlanmasıyla olur.



Şekil 3.18. YSA yapısı

YSA'nın temel elemanı düğümlerdir. Düğümler işlem elemanı olarak tanımlanır. Şekil 3.18'de bir düğüm yapısı ve meydana gelen işlemler gösterilmiştir. Temel YSA hücresinde girişler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıkışlar bulunur. Düğümler sinirleri temsil eder ve oklar ile birbirlerine bağlantıları gösterilirler. Harici bir kaynaktan ya da diğer birimlerden istediği sayıda giriş ve tek bir çıkış bağlantısı alır. Her bir girdi ilişkili bir ağırlığa sahiptir. Girişler ağırlık değerleri ile çarpıldıktan sonra sonuç elde edilir

Denklem 3.4.'te x_i girdi değerlerini, w_i ağırlıkları, n ise bir hücreye gelen toplam girdi sayıları olarak ifade edilmektedir. Toplama fonksiyonu probleme göre değişkenlik göstermektedir.

$$NET = \sum_i^n w_i x_i \tag{3.4}$$

Giriş: YSA'ya dış ortamlardan veya diğer bir hücreden gelen veridir. Bunlar ağın öğrenmesi istenen örnekler tarafından belirlenir.

Ağırlık: Hücreye gelen bilgilerin etkisini gösterir. Girdiler ağırlıkları ile çarpıldıktan sonra hücre çekirdeğe iletilir. Böylelikle girdilerin üretilecek çıktı üzerindeki etkisi ayarlanabilir. Ağırlıklar pozitif değerde, negatif değerde veya sıfır olabilir. Ağırlığı sıfır olan girdiler üretilecek çıktı üzerinde etkili olmaz. Ağırlıkların büyük ya da küçük olmasından ağırlığın önemli veya önemsiz olduğu anlamı çıkarılmamalıdır.

Toplama Fonksiyonu: Hücreye gelen bilgilerle, bu hücrelerin ağırlıklarının çarpımını toplar ve o hücrenin net girişini hesaplar. Bunun için değişik fonksiyonlar vardır. En çok kullanılan ağırlıklı toplamıdır.

Çıkış: Aktivasyon fonksiyonlarının oluşturduğu çıkış bilgisidir. Bu bilgi diğer hücreye ya da kendisine giriş bilgisi olarak aktarılabilir.

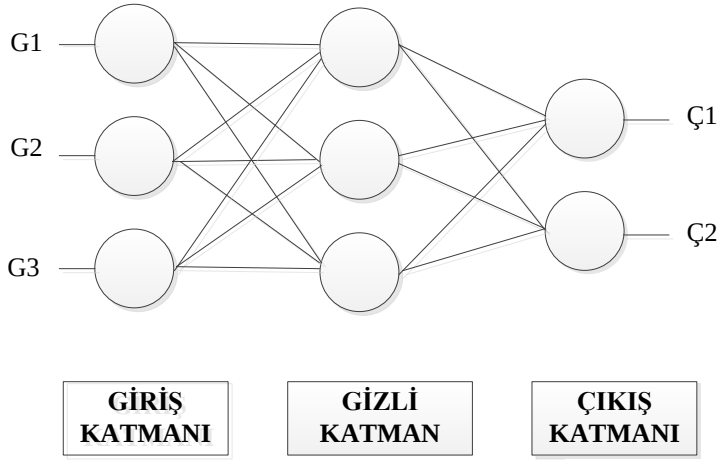
YSA'ların 3 temel bileşeni vardır. Bunlar mimarı yapı, öğrenme algoritması ve aktivasyon fonksiyonudur.

1.Mimari Yapı: YSA'lar mimari açıdan ileri beslemeli ve geri beslemeli olmak üzere ikiye ayrılır. İleri beslemeli ağlarda giriş katmanından çıkış katmanına doğru tek yönde ilerlenir. Bu ağlara örnek olarak Multi Layer Perceptron (MLP), Radial Basis Function Network (RBFN) ve Learning Vektor Quantization (LQV) verilebilir. Geri beslemeli ağlarda çıkış ve ara katman çıkışları, kendinden önceki katmanlara ya da girişe geri beslenir. Bu ağlara örnek olarak Adaptif Rezonans Teori (ART), Self-Organizing Maps (SOM) ve Elman ve Jordan örnek verilebilir.

2.Öğrenme Algoritması: YSA'larda bilgi ağdaki sinirlerin bağlantıları arasındaki ağırlıklarda tutulur. Öğrenme işlemi, ağdaki ağırlıkların en iyi değerlere ulaşması ile gerçekleşir.

3.Aktivasyon Fonksiyonu: Aktivasyon fonksiyonları girdi ve çıktı verileri arasındaki eğrisel eşleşmeyi sağlar. Aktivasyon fonksiyonlarının doğru seçilmesi ağın performansını etkiler. Genelde tek kutuplu (0 1) veya çift kutuplu (-1 +1) ve doğrusal olarak seçilebilir. Sigmoid, Gaussian, Doğrusal, Hiperbolik Tanjant aktivasyon fonksiyonları bu fonksiyonlara verilebilir.

YSA'lar yapay sinir hücrelerinden oluşur. Bu hücreler birbirleriyle katmanlar halinde ve katmanlarda birbirleriyle paralel olarak bağlanırlar. Şekil 3.19'da bir yapay sinir ağı katmanları, bir giriş, ara katman ve bir çıkış katmanından oluştuğu gösterilmektedir. Giriş katmanında bilgiler alınarak ara katman iletilir. Ara katmanda giriş katmandan gelen bilgiler işlemden geçirilen bilgiler çıkış katmanına aktarılır. Çıktı katmanında ara katmandan gelen bilgiler işlenip ağın girdilerine uygun olarak çıkış değerleri üretilirler.

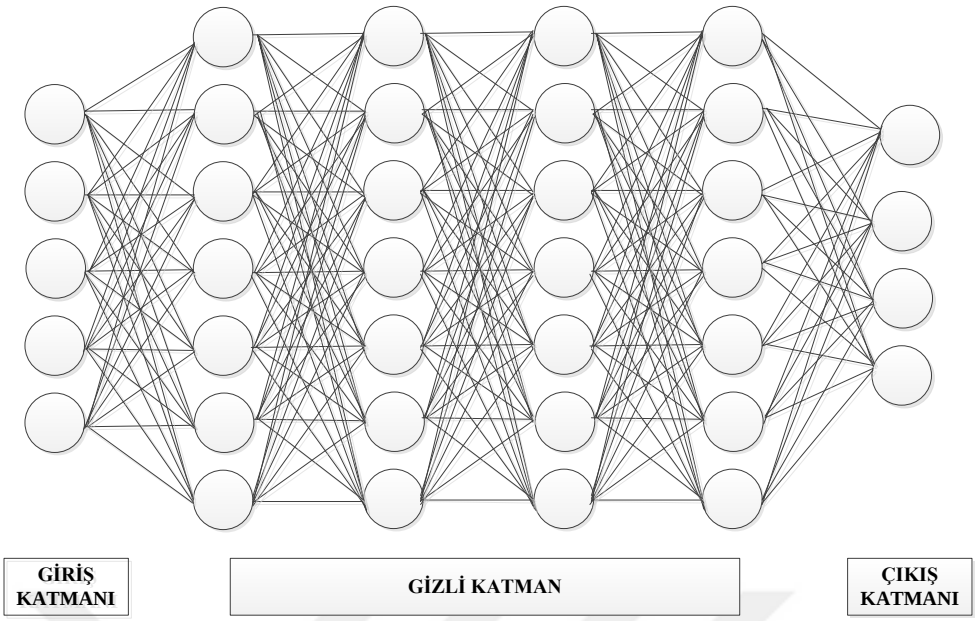


Şekil 3.19. Yapay sinir ağı katmanları

YSA geliştirme sürecinde eğitim ve test verileri yer alır. Eğitim verileri, ağın eğitilmesi için kullanılır. Eğitim ve test verilerindeki önemli nokta veri miktarıdır. YSA mümkün olduğunca çok veri ile eğitilmelidir. Eğitim verisinin yeterli olup olmadığını anlamak için daha fazla eğitim verisi verildiği zaman ağın performans durumu gözlenmelidir. Eğitim setindeki veri miktarı YSA modeline ve problemin karmaşıklığına göre değişiklik gösterebilir. Test verileri, eğitilmiş ağın doğruluğunun test edilmesi için kullanılır. YSA modeline verilen test verilerinin çıktıları ile istenilen çıktı değerleri karşılaştırılır. Yeterli genelleme gerçekleşirse o zaman YSA modeli kullanılabilir (Tüzen, 2017).

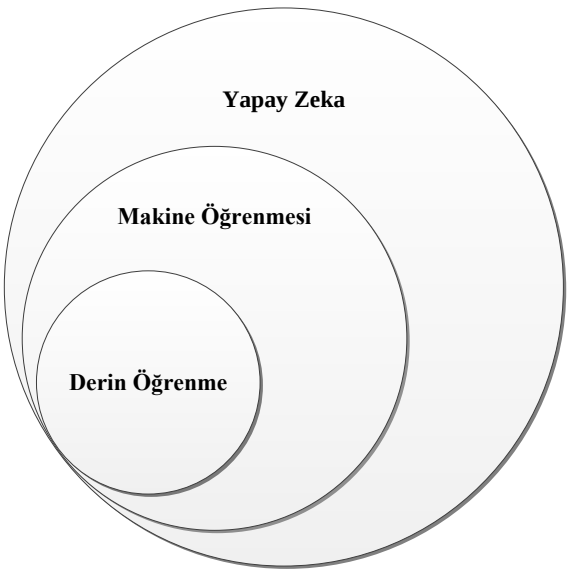
3.7.Derin Öğrenme

Derin öğrenme, makine öğrenmesi tekniklerinden biridir ve makine öğrenmesinin tek katmanda yaptığı işi birçok katmanda aynı anda yapmasıdır. Bir grup makine öğrenmesi algoritmalarının aynı anda kullanılmasıyla tek işlemde sonuç oluşturulmaya çalışmasıdır. Büyük sayıda etiketlenmemiş eğitim verileri ile özellik tespiti yapabilen sistemler oluşturmak için ileri teknolojiye sahip çok seviyeli “derin” sinir ağlarının kullanılmasına derin öğrenme denir. Şekil 3.20’de giriş katman, birden fazla gizli katman ve çıkış katmanları gösterilmektedir.



Şekil 3.20. Derin öğrenme sinir ağı

Bu zamana kadar makine öğrenmesi algoritmaları edinmiş olduğu tecrübeleri parametreler ile makineye öğretmeye çalışmaktadır. Örneğin bir balık resminden hangi türe ait olup olmadığını anlamak için en, boy, parlaklık gibi parametrelerin tanıtılması gerekmektedir. Derin öğrenme algoritmalarında parametreler verilmeden kendiliğinden öğrenebilmektedir. Yani sadece portakal ve muz resimleri derin öğrenme sistemlerine gösterilir ve sistem öğrenme parametrelerini kendi kendine tespit eder. Böylece dışarıdan verilen herhangi bir öğrenme parametresine ihtiyaç duymadan kendi kurallarını oluşturarak işlemlerini gerçekleştirir.



Şekil 3.21. Yapay zeka, makine öğrenmesi ve derin öğrenme ilişkisi

Şekil 3.21'de derin öğrenme makine öğrenmesinin bir alt kümesini oluşturuyor ve her ikisi de yapay zekanın alt kümelerini oluşturuyor. Bu durumu söyle özetleyebiliriz;

- Yapay zeka insan davranışlarını, makinelere ve bilgisayar sistemlerine taklit eder.
- Makine öğrenimi, yapay zekanın bir alt bölümüdür. Makinelerin tecrübesiyle görevlerini geliştirmesini sağlayan soyut istatistiksel teknikler içerir. Bilgisayarlar, veri modellerini tanımlar ve bunlara göre hareket eder ve zaman içinde açık programlama olmaksızın doğruluğunu geliştirmeyi öğrenirler.
- Derin öğrenme, makine öğrenmesinin bir alt bölümüdür. Gelişmiş sinir ağları olarak düşünülebilir. İnsan beyninin algılama şeklinin taklit eder.

Derin öğrenmenin hayatımıza girmesinde birçok etken vardır. Bunlar;

1.Veri miktarının artması: Gün geçtikçe internetin yaygın hale gelmesi çok büyük boyutlarda verinin dijital ortamda üretilmesi ve saklamasına neden olmuştur. Bu büyük verilerin kullanılması derin öğrenme sistemleri ile gerçekleştirilmiştir (Genç, 2016).

2.GPU'lar ve işlem gücünün artması: GPU (Graphics Processing Unit – Grafik İşleme Ünitesi) hesaplama ile elde edilen, güçlü ve verimli paralel hesaplamalar yapılabilmektedir. GPU'lar, çok daha büyük eğitim setleri kullanarak derin öğrenme algoritmalarını çok daha kısa sürelerde ve çok daha az veri merkezi altyapısı kullanarak eğitmek için kullanılmaktadır (Genç, 2016).

3.Derinliğin artması: İşlem gücü artması ile derin modellerin pratikte kullanılabilmesi sağlanmış oldu. Derin öğrenme modelleri de çok katmanlı yapıya sahip modellerdir. Derinliği anlamak adına insan beyninin görme sistemi ile bağlantı kurabiliriz. Gözlerdeki sinirler vasıtasıyla beyne gelen sinyaller, birkaç katmanlı bir yapıda hiyerarşik değerlendirilerek işlenir. Sinyalin gözden sonra uğradığı ilk katmanda, kenarlar köşeler gibi görüntünün daha yerel ve temel özellikleri tanınır. Bir sonraki katmanda bu kenar ve köşeler bir araya getirilerek ağız burun gibi şekilleri, daha sonraki katmanda yüzleri, daha sonraki katmanda ise kişi ve nesnelerin yerleşim gibi görüntünün bütününe ait özellikleri tanınabilir. Birçok derin öğrenme sistemi bu prensipte çalışır (Genç, 2016).

Derin öğrenmenin popülerleşmesinde, modellerin daha derin ve karmaşık hale gelmesi ve bu ağların büyük verilerle eğitilmesini sağlaması ile birlikte bunları masaüstü bir bilgisayarda gerçekleştirebilmesi etkili olmuştur.

Derin öğrenme genelde zorlu ses ve görüntü tanımlama işlemlerinde kullanılmaktadır. Ayrıca;

- Plaka tanıma sistemleri
- Yüz tanıma sistemleri
- Parmak izi okuyucular
- İris okuyucular
- Ses tanıma sistemleri
- Sürücüsüz arabalar
- Spam (istenmeyen) e-posta tespitinde de kullanılmaktadır.

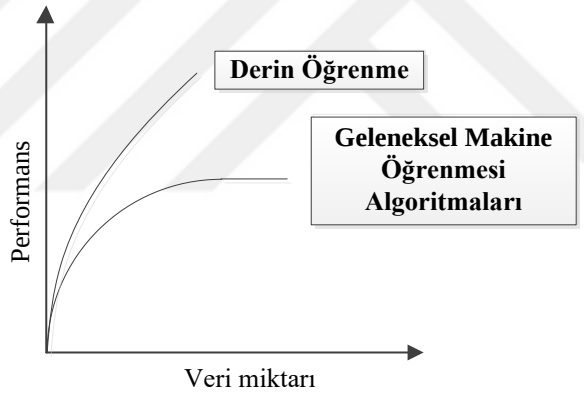
Makine öğrenmesi sayesinde ses tanıma uygulamaları alanında büyük bir gelişme sağlanmış olup büyük oranda doğruluk sağlanmaktadır. Derin öğrenme sayesinde milyarlarca ses parçası daha az sürede işlenerek yüzde 100'e kadar doğruluğun sağlanması beklenmektedir.

GO, Çin'de icat edilmiş iki oyuncu ile oynanan bir strateji oyunudur ve amacı rakibini yenmektir. GO dünyanın en karmaşık oyunlarından biridir ve sezgiye dayalı, çok az kuralı olan, zeka, yürek, sezgi, deneyim gerektiren bir oyundur. AlphaGo isminde bir yapay zeka uygulaması, insanları GO oyununda yenmeyi başardı. Google Mind tarafından geliştirilen bu uygulama Mart 2016'da dünyanın en iyi GO oyuncusunu yendi. Tarihte ilk kez bir bilgisayar, GO oyununu oynayanların alabileceği en yüksek puanı almış oldu. Bu olay yapay zeka için en önemli kilometre taşlarından biri oldu. Geleneksel makine öğrenmesi uygulamalarında kullanılan teknikler GO oyununun bir bilgisayar tarafından oynanması için yeterli değildi. Bu sebeple AlphaGo uygulaması yeni bir teknik olan derin öğrenme yöntemiyle eğitilmiştir.

3.7.1.Derin Öğrenme Nasıl Çalışıyor?

Artan katman sayısı daha derin bir karar mekanizması meydana getirmektedir. Derin öğrenme, geleneksel YSA'lardan farklı olarak daha çok sayıda katman ile çalışabilmektedir. Klasik YSA yapısında her hücre (node) bir önceki ve bir sonraki katmanlardaki bütün hücrelere bağlıdır. Her bağlantı için hesaplama yapılması gereken matematiksel işlemler bulunmaktadır. Katman sayısı ve hücre sayısı arttıkça bu işlemler yüksek miktarda CPU (Central Processing Unit – Merkezi İşlem Birimi) gücü gerektirmektedir. Derin bir ağ yapısı oluşturmak için kişisel bilgisayarlarda bulunan CPU

gücü yetersiz kalmaktadır. GPU'nun geliştirilmesi ile birlikte derin öğrenmenin geliştirilmesi de hızlanmıştır. GPU ve CPU arasındaki fark bir process'in işlenme biçimidir. CPU'lar birkaç tane çekirdek barındırır. Her çekirdek yüksek işlem kapasitesine sahiptir ve işlemler bu çekirdeklere dağıtılarak seri bir şekilde işlenir. GPU'larda ise yüzlerce çekirdek bulunur. Her çekirdeğin işlem kapasitesi CPU'ya kıyasla daha azdır. Ancak yüksek paralel işlem gücüne sahip olup eş zamanlı birçok işlem yapabilmektedir. GPU'nun bu işlem kapasitesi derin öğrenmenin uygulanabilmesi için önemli bir noktadır. Bu sayede çok büyük miktarda veriler ile yapılan eğitimde performans artmıştır. Bu noktada derin öğrenmeyi geleneksel makine öğrenmesi algoritmalarından ayıran bir noktadır. Çünkü geleneksel makine öğrenmesi algoritmalarında yüksek miktarda veri, başarının bir miktar artırmasına ve sonra başarının sabit kalmasına neden olmaktadır. Bu başarı Şekil 3.22.'deki grafikte gösterilmektedir (Büber,2017).



Şekil 3.22. Derin öğrenme ve makine öğrenmesi algoritmalarının performans grafiği

Derin öğrenme yöntemleri çok sayıda veri girişine göre ayırt edici özellikleri kendisi öğrenir. Bu özellik öğrenmenin başarılı bir şekilde yapılabilmesi için yeterince eğitilmesi gerekmektedir. Özellik öğrenme, katmanlardan oluşur. Alt katmanlar daha az ayırt edici özelliğe sahipken üst seviyedeki katmanlar daha çok ayırt edici özelliğe sahiptir. Alt seviyedeki özellikler üst seviyedeki özelliklerin oluşturulması için temel oluşturur. Bu tarz öğrenme şekli makine öğrenmesinden farklıdır. Çünkü makine öğrenmesinde eğitim aşamasında insanlar tarafından belirlenen özellikler ile eğitilir. Yani makine öğrenmesi algoritmaları insan bağımlı iken derin öğrenme insan bağımsız çalışmaktadır. Bu yaklaşım derin öğrenmenin başarısında önemli bir etkendir.

3.7.2.Derin Öğrenme Kütüphaneleri

En yararlı ve en popüler olan 5 derin öğrenme kütüphanesinden bahsedilmektedir.

Caffe, BVLC (Berkeley Vision and Learning Center – Berkeley Görüntü ve Öğrenme Merkezi) kullanıcı topluluğu tarafından geliştirilmiştir. Model ve optimizasyonlar kodlama yapılmaksızın ayar dosyası üzerinden yapılabilmektedir (Caffe, 2018). C++ dilinde geliştirilmiş olup hızlı ve modüler olarak tasarlanmıştır.

Torch, FAIR (Facebook Artificial Intelligence Researchers- Facebook Yapay Zekâ Araştırmacılar) tarafından 2015 yılının başında açık kaynak kodlu hale getirilmiştir. Sinyal işleme, makine öğrenmesi ve görüntü işleme konularında hazır algoritmalar sunması yanında derin öğrenme ve katmanlı sinir ağlarının modellenmesi konusunda önemli bir sistem sunar. Torch, Lua yazılım dilinde geliştirilmiştir ve LuaJIT olarak adlandırılan başka bir betik dili ile modellenebilmektedir (Arıkuşu,2017). Torch, kompleks sinir ağları modellemesinde esnek ve kolay kullanım sağlayan kütüphanelere sahiptir. CPU ve GPU'lar arasında etkili bir şekilde paralel işlem yapabilir.

Theano, Montreal Üniversitesi tarafından Python programlama dili ile geliştirilmiştir. 2006'dan sonra hızlanan derin öğrenme çalışmalarında ihtiyaç duyulan yazılım çözümü olarak ortaya çıkmıştır. Halen birçok akademik çalışmada kullanılan Theano, daha sonra ortaya çıkan derin öğrenme kütüphanelerine de referans olmuştur (Arıkuşu,2017). Keras ve Lasagne, Theano üzerine kurulmuştur. Çok boyutlu diziler kullanarak matematiksel işlemler yapmak için kullanılan bir kütüphanedir. Hızlıdır ve GPU kullanarak optimize edilmiştir. Sinir ağları için temel yapı taşları olarak işlev görür (Tüzen,2017).

Tensorflow, makine öğrenmesi ve derin öğrenme sinir ağlarını araştırmasını yürütmek amacıyla Google'ın Yapay Zeka araştırma grubu içinde yer alan Google Beyin Takımı'nda çalışan araştırmacılar ve mühendisler tarafından geliştirildi. Ancak sistem daha sonra diğer alanlarda da kullanılacak kadar genelleşti. Veri akış grafikleri kullanarak sayısal hesaplama için açık kaynak kodlu bir yazılım kütüphanesidir. Grafik köşeleri, aralarında iletilen çok boyutlu veri dizilerini temsil ederken grafikteki düğümler matematiksel işlemleri temsil eder. Esnek mimarisi sayesinde, hesaplamayı tek bir API ile bir bilgisayar sistemlerindeki bir veya daha fazla CPU'ya ya da GPU'ya dağıtmaya imkan sağlar. Theano'ya göre daha yavaştır ve desteklediği yelpaze daha dardır.

Doğal dil işleme, yazı ya da görsel içerikleri anlamlandırma, öneri sistemleri gibi Google tarafından kullanılan yapay zeka uygulamalarıdır. Bu uygulamaların çalışmasını

sağlayan derin öğrenme alt yapısı olan Tensorflow 2015 yılında açık kaynak kodlu olarak dünyadaki tüm araştırmacı ve geliştiricilerin erişimine açıldı. C++ dilinde yazılan Tensorflow, Python ve Java gibi birçok programlama diline arayüz sağlar. Apple iOS telefon ve tabletlerde çalışabilmesinin sağlanması ile önemi daha da artmıştır (Arıkuşu,2017).

Lasange, Theano'nun optimize edilmiş hesaplama işlemini kullanarak sinir ağları kurmaya ve eğitmeye olanak tanır. Çok boyutlu diziler yerine sinir ağ yapıları ile programlanabilir. Keras basitliğine ve Theano esnekliğine sahip bir kütüphane olarak düşünebiliriz (Tüzen, 2017).

Keras, Tensorflow ya da Theano kütüphaneleri üzerinde çalışan, derin öğrenme uygulamaları için yazılmış bir Python kütüphanesidir. GPU ya da CPU üzerinde çalışmasını bu temel kütüphaneler üzerinden sağlar. Daha üst seviye bir kütüphane olduğundan Tensorflow ve Theano kütüphanelerine göre daha kolay uygulama geliştirmeyi sağlar.

Kolay ve hızlı prototipleme (kullanıcı dostu, modülerlik ve genişletilebilirlik) sağlar. Hem konveksiyonel ağları hem de yinelenen ağları ve ikisinin kombinasyonlarını destekler. CPU ve GPU üzerinde kusursuz çalışır.

Keras, makineler için değil insanlar için tasarlanmış bir API'dir. Kullanıcı deneyimini öne ve ortama koyar. Keras, bilişsel yükü azaltmak için en iyi uygulamaları izleyip tutarlı ve basit API'ler sunar, ortak kullanım vakaları için gereken kullanıcı eylemlerinin sayısını en aza indirir ve kullanıcı hatası üzerine açık ve uygulanabilir geri bildirim sağlar. Yeni modüller (yeni sınıflar ve işlevler gibi) eklemek kolaydır ve mevcut modüller bol örnekler sağlar. Kolayca yeni modüller yaratmak, Keras'ı ileri araştırmalara uygun hale getirmek için toplam ifade gücü sağlar (Keras,2018).

3.7.3.Derin Öğrenme Uygulamalarında Hiper Parametreler

Derin öğrenme ile problem çözmek, çok katmanlı ağ yapısını en iyi ve en uygun şekilde tasarlamak ile eşdeğer bir haldedir. Giriş verileri ile öğrenen makine öğrenmesi modelleri tasarlanırken modelde kullanılacak algoritmalar ve teknikler için tasarımcıların karar vermesi gereken bazı parametreler bulunmaktadır. Aynı şekilde derin öğrenme modellerinde de seyreltme (dropout) değerine, katman sayısına, nöron sayısına tasarımcı karar vermektedir. Genelde bu parametreler başlangıçta kesin olmamakla birlikte probleme ve verisetine göre değişiklik göstermektedir. Bu da tasarımcıların kararına

bırakılmıştır. Probleme ve verisetine göre değişen parametrelere hiper parametre denilmektedir. Hiper parametrelerin belirlenmesi genel olarak önceki tecrübelere, farklı uygulamaların kendi problemlerimize yansıtılmasına ve tasarımcı sezgilerine göre değişir (Çarkacı, 2018).

3.7.3.1.Veri seti Boyutu

Veri setinin büyüklüğü ve çeşitliliği derin öğrenme algoritmalarındaki en önemli faktörlerden biridir. Veri seti ne kadar büyük ve çeşitli olursa öğrenme oranı da o kadar arttığı gibi öğrenme için harcanan zamanda o kadar artmaktadır. Depolama alanının yeterli olmadığı durumda veri seti boyutunun iyi ayarlanması gerekmektedir. Veri seti arttıkça öğrenme oranı sürekli artmaz, belirli bir noktadan sonra başarı küçük oranda artar (Çarkacı, 2018).

3.7.3.2.Mini Batch Boyutu

Derin öğrenme uygulamalarında veri boyutunun büyük olması ile tüm verilerin aynı anda işlenmesi zaman ve bellek harcamaktadır. Çünkü öğrenmenin her iterasyonunda geri yayılım işlemi ile ağ üzerinde gradyan (gradient descent) hesaplaması yapılmakta ve ağırlık değerleri güncellenmektedir. Veri sayısı ne kadar fazla olursa bu işlemde o kadar uzun sürecektir. Bu sebeple veriler parçalar halinde işlenmektedir. Birden fazla girdinin parçalar halinde işlenmesine mini batch denir. Mini batch parametresi aynı anda kaç verinin işleneceğini göstermektedir. Mini batch durumunda loss değerleri artsa da veri grupları işlendikçe bu değerler düzelecektir. Ayrıca mini batch uygulandığı durumda her iterasyonda farklı veri grupları için bazı parametreler tam uygun olurken bazılarında uygun olmayabilir. Bu sebeple hata oranlarında zigzaglar oluşur. Bu durum küçük öğrenme değeri ile azaltılabilir.

Mini batch değerleri 1 ile tüm eğitim veri boyutu arasında herhangi bir değer olabilir. Eğer 1 olursa SGD (Stochoastic Gradient Descent) ile aynı iş yapılmaktadır. Bu durumda model gürültüyü öğrenebilir ve grafikteki zigzaglar çoğalır. Çünkü her iterasyonda farklı veri kullanılmaktadır. Local optimum'da takılıp global optimuma hiç ulaşmayabilir. Global minimuna ulaşmadan onun etrafında dolanabilir. Momentum ile SGD salınımı azaltılarak daha hızlı ve tutarlı bir optimizasyon algoritması oluşturulabilir. Salınımı azaltmanın bir diğer yolu da RMSprob (Root Mean Square Error Probability)

kullanmaktır. Mini batch değeri tüm eğitim veri seti boyutu seçilirse BGD (Batch Gradient Descent) ile aynı olacaktır. Bu durumda model daha az gürültülü öğrenecektir. Aynı anda bütün veriyi işlediği için daha uzun sürecek ve optimizasyon işleminde büyük adımlarla ilerleme olacaktır. Mini batch değerinin seçiminde en uygun değer 1 ile tüm eğitim veri seti boyutu arasında ne çok küçük ne de çok büyük değer olacak şekilde belirtilmemelidir. Bu hızlı bir öğrenmeyi sağlamaktadır. Eğer veri seti kendini tekrar ediyorsa ya da gereğinden fazla çalışırsa ezberleme (overfitting) olma riski olmaktadır (Çarkacı, 2018).

3.7.3.3.Öğrenme hızı (Learning Rate) ve Momentum Katsayısı

Derin öğrenme algoritmalarında parametrelerin güncellenmesi geri yayılım işlemi sırasında yapılmaktadır. Geri yayılım sırasında "chain rule" olarak adlandırılan geriye doğru türev alınarak farkın bulunması ve bulunan fark değerinin "learning rate" parametresi ile çarpılması, çıkan sonuç ağırlık değerlerinden çıkarılarak yeni ağırlık hesaplanmaktadır. Bu işlem sırasında öğrenme hızı sabit bir değer, adım adım artan bir değer, momentum değerine bağlı bir değer olarak belirlenebilir ya da adaptif algoritmalar ile öğrenme sırasında öğrenilebilir. Öğrenme hızının yüksek olması veriden çok etkilendiği anlamına gelmektedir. Öğrenme hızının yüksek olması salınımı artırır. Düşük olması öğrenme süresinin uzatır. En uygun çözüm yüksek değerden başlayıp azaltmaktır. Genelde değeri 0.01 ile başlatıp belirli bir epoch'tan sonra 0.001'e düşürmektir (Çarkacı, 2018).

3.7.3.4.Optimizasyon Uygulamaları

Derin öğrenme uygulamaları temelde optimizasyon problemidir. Doğrusal olmayan problemlerde optimum değeri bulmak için optimizasyon yöntemleri kullanılmaktadır. Derin öğrenme uygulamalarında sık kullanılan optimizasyon algoritmaları stochastic gradient descent, adagrad, adadelta, adam, adamax gibidir. Bu algoritmalar arasında başarım ve hız farkı bulunmaktadır. Adaptif algoritmalar öğrenme hızını kendisi öğrenmektedir ve dinamiktir. Ağırlık katsayılarının güncellenmesi için uygulanacak yöntem ayarlanır. Optimizasyon fonksiyonu metotları ismini string olarak vererek varsayılan parametreler ile işleme dahil edilebilir ya da kendi istediğimiz değerleri verilebilir (Çarkacı, 2018).

SGD

Derin öğrenme algoritmalarında genel olarak kullanılan optimizasyon algoritmasıdır. Resim tanıma gibi bazı problemlerde çok kötü sonuçlar vermektedir (Çarkacı, 2018).

RMSprop

Genellikle kendini tekrar eden sinir ağları için tercih edilmesi doğrudur. Parametre değerleri varsayılan değerlere bırakılması uygun görülmektedir (Keras, 2018).

AdaGrad

Seyrek parametreler için büyük güncellemeler yaparken sık parametreler için daha küçük güncellemeler yapmaktadır. Bu sebeple NLP ve resim tanıma gibi seyrek veriler için uygundur. Adagrad algoritmasında her parametrenin kendi öğrenme oranı vardır ve algoritmanın özelliklerine göre giderek azalmaktadır. Bu neden öğrenme oranı sürekli düşer ve bir süre sonra öğrenme durur. Bu en büyük dezavantajıdır. Bu dezavantaj RMSprop ve benzeri olan AdaDelta ile çözülmektedir (Çarkacı, 2018).

Adam

Adam parametrelerin her birinin öğrenme oranlarının yanında momentumda meydana gelen değişiklikleri de önbellekte (cache) saklar; yani RMSprop ve momentumu birleştirir (Çarkacı, 2018).

3.7.3.5.Eğitim Tur Sayısı (Epoch)

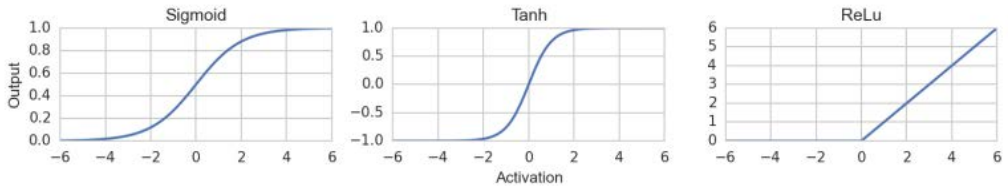
Model eğitilirken batch sayısı kadar veri eğitilir ve geri yayılım ile ağırlıklar güncellenir. Daha sonra diğer eğitim veri setleri için aynı işlem uygulanır. Böylece her bir eğitim adımında en uygun ağırlık değerleri hesaplanmaya çalışılmaktadır. Bu eğitim adımlarının sayısına epoch denir. Derin öğrenmede ağırlık değerleri adım adım hesaplandığı için ilk epochlarda başarı oranı yüksek olmasa da ağırlıklar güncellendikçe başarım oranı artmaktadır. Belirli bir adımdan sonra öğrenme durmaktadır. Epoch'un çok olması daha iyi verim sağlamak yerine eğitim verisini ezberlemeye başlamasına neden olabilir. Epoch'un az olması da istenilen performansa ulaşamamaya neden olabilir (Çarkacı, 2018).

3.7.3.6.Ağırlık Değeri

Girdilerin üretilecek çıktı üzerindeki etkisi belirleyen değerlerdir. Ağrılıkların belirlenmesi modelin öğrenmesini ve hızını etkilemektedir. Ağırlığı sıfır olan girdilerin üretilecek çıktı üzerinde herhangi bir etkisi olmaz (Çarkacı, 2018).

3.7.3.7.Aktivasyon Fonksiyonu

Aktivasyon fonksiyonları çok katmanlı YSA'larda ağırlık hesaplaması yapıldıktan sonra çıktı değerlerini doğrusal olmayan değerlere dönüştürürler. Derin öğrenme yöntemlerinin bir özelliği olan doğrusal olmama, aktivasyon fonksiyonlarının doğrusal olmamasından kaynaklanır ve doğrusal olmayan problemlerin çözümünde kullanılmaktadır. Gizli katmanlarda geri türev alınabilmesi için çıktısı bazı aktivasyon fonksiyonları ile normalize edilmektedir. Geri beslemeli ağlarda aktivasyon fonksiyonunun türevi de kullanıldığı için hesaplamanın yavaşlamaması için türevi de kolay hesaplanabilir olmalıdır. Şekil 3.23'te aktivasyon fonksiyonlarından bazıları olan sigmoid, tanjant ve relu grafiksel gösterilmektedir (Çarkacı, 2018).



Şekil 3.23. Aktivasyon fonksiyonları grafikleri (Çarkacı, 2018)

- **Sigmoid Aktivasyon Fonksiyonu**

Sürekli ve türev alınabilir bir fonksiyondur. Doğrusal olmadığından dolayı YSA'larda sık kullanılan aktivasyon fonksiyonlarındandır. Girdi değerlerinin her bir değeri için 0 ile 1 arasında bir değer üretir.

- **Tanjant Hiperbolik Aktivasyon Fonksiyonu**

Sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonda çıkış değerleri 0 ile 1 arasında değişirken tanjant hiperbolik fonksiyonda -1 ile 1 arasında değişir.

- **Relu Aktivasyon Fonksiyonu (Rectified Linear Unit - Düzeltilmiş Lineer Ünite)**

En çok kullanılan aktivasyon fonksiyonlarından biridir. Genellikle ileri beslemeli ağlarda kullanılmaktadır. Doğrusal bir ünite, eğer girdi 0'dan az ise çıkış 0, aksi halde işlenmemiş çıkış vardır. Yani, eğer girdi 0'dan büyükse, çıktı girişe eşittir.

ReLU Aktivasyon Fonksiyonu, kullanabileceğiniz en basit doğrusal olmayan etkinleştirme işlevidir. Girdi pozitif çıktığında, türev sadece 1'dir, bu nedenle sigmoid işlevinden gelen geri iletilen hatalarla karşılaştığınız sıkma etkisi yoktur. Araştırma, ReLU'ların büyük ağlar için daha hızlı eğitimle sonuçlandığını göstermiştir.

- **Softmax**

Bu fonksiyona verilen her bir girdinin bir sınıfa ait olma olasılığını göstermektedir. [0,1] arası çıktı değerleri üretmektedir. Çoklu sınıflandırma problemleri için kullanılmaktadır (Karakuş,2018).

3.7.3.8.Seyreltme Değeri (Dropout)

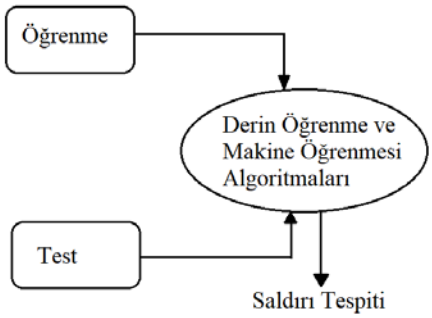
Tam bağlı katmanlı ağlarda belli bir eşik değeri altındaki düğümlerin seyreltilmesi başarıyı arttırmaktadır. Seyreltme değeri eşik değeri olarak kullanıldığında [0,1] arasında değer almaktadır (Çarkacı, 2018).

3.7.3.9.Katman Sayısı ve Gizli Katman

Derin öğrenme uygulamalarını diğer YSA'dan ayıran en önemli özellik katman sayısıdır. Katmanlar ve gizli katmanlar bir derinlik oluşturur ve derinlik arttıkça öğrenme hızı ve oranı artmaktadır. Katman sayısı modelin tasarlanmasına göre değişiklik göstermektedir. Katman sayısı attıkça geri yayılım ilk katmanlara daha az ulaşabilecektir. Bu durumda da katman sayısının çok olması her zaman olumlu etki oluşturmamaktadır. Derin öğrenmede başarı oranı hem katman sayısına hem de hiper parametrelere bağlıdır. Nöron sayısı ağda tutulan bilgiyi ifade etmektedir. Nöron sayısı fazla olursa bellek ihtiyacı ve hesaplama artmaktadır. Nöron sayısının az olması yetersiz uyuma (underfitting) sebep olmaktadır. Nöron sayısının ilk katmanlarda çok olup sonradan azalarak gitmesi "regularization" (başarım iyileştirme) etkisi meydana getirmektedir (Çarkacı, 2018).

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

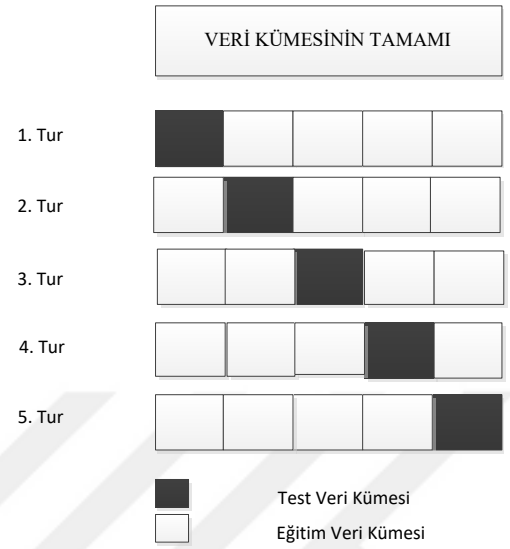
Derin öğrenmenin DoS saldırılarındaki performansını değerlendirmek için 3 farklı deney gerçekleştirilmiştir. Deney sonuçları aynı zamanda bilinen makine öğrenmesi sınıflandırma algoritmaları DVM, NB ve YSA ile karşılaştırılmıştır. DoS saldırılarını tespit etmek için tasarlanan derin öğrenme sistemleri keras ve sklearn kütüphaneleri kullanılarak Python programlama dilinde gerçekleştirilmiştir. DVM, NB ve YSA makine öğrenmesi sınıflandırma algoritmaları Weka veri madenciliği aracı kullanılarak test edilmiştir. Şekil 4.1'de DoS saldırılarının tespit edilmesinde kullanılan mimari gösterilmektedir. Derin öğrenme ve makine öğrenmesi algoritmaları öğrenme veri setleri ile 10-kat çapraz doğrulama kullanılarak eğitilmiş ve test edilmiştir.



Şekil 4.1. DoS saldırı tespiti için kullanılan mimari

K-kat çapraz doğrulama, sınıflandırma modellerinde kullanılan modelin eğitilmesi ve değerlendirilmesi için veri setini parçalara ayırma yöntemidir. Model eğitilirken veri setinin bir kısmı öğrenim bir kısmı da test verisi olarak ayrılmaktadır. Bu ayrım yapılırken hatalar ve sapmalar meydana gelmektedir. K-kat çapraz doğrulama belirli oranda veriyi belirlenen bir k sayısına eşit olarak bölmektedir. Bu parçalar hem eğitim hem de test verisi olarak kullanılmaktadır. Böylece her veri eğitim ve test aşamalarında kullanılmış olduğu için sapmaların ve hataların en az seviyeye inmesi sağlanmaktadır. Örneğin 1000 kaydı olan bir veri seti için k değeri 5 seçilirse ilk 200 kayıt test verisi geriye kalan 800 kayıt eğitim verisi olarak kullanılmaktadır. İkinci turda, ikinci 200'luk kayıt yani 201-400 arası kayıtlar test verisi diğer kayıtlarda eğitim verisi olarak kullanılmaktadır. En son tur da tamamlandıktan sonra elde edilen sonuçlar k değerine yani 5'e bölünmekte ve modelin performansı elde edilmektedir (Şirin,2017).

Şekil 4.2'de K-kat çapraz doğrulama sisteminin k değerinin 5 seçildiği örnek gösterilmektedir. Koyu renk kutucuklar test veri kümesini açık renk kutucuklar eğitim veri kümesini göstermektedir. 5 tur boyunca eğitim ve test verilerinin farklı veri grupları olarak seçildiği gösterilmektedir.



Şekil 4.1. K-kat çapraz doğrulama sistemi çalışması

Denklem 4.1'de SF sınıflandırma fonksiyonu, VK veri kümesi, k kaç kat çapraz doğrulama olacağı ve t veri kümesi üzerinden seçilen her bir test verisi olarak gösterilmektedir. Formül sonucu; bütün sınıflandırma fonksiyonlarının başarısının, k sayısına bölünüp ortalaması alınarak hesaplanmasıdır (Şeker,2013).

$$t_i \in VK\ olmak\ \text{üzere}, Sonuç = \frac{\sum_{i=0}^{k} SF(t_i, VK - t_i)}{k} \quad (4.1)$$

Yapılan çalışmada K-kat çapraz doğrulama için kullanılan k değeri 10 olarak belirlenmiştir. 10-kat çapraz doğrulama için kullanılan bu değer yapılan denemeler sayesinde düşük hata ve sapma oranına sahip modellerin elde edilmesi ile bulunmuştur (Brownlee,2018). En iyi sonucu vermesinden dolayı 10-kat çapraz doğrulama kullanılmıştır.

Testlerin gerçekleştirildiği platformun özellikleri Tablo 4.1'de gösterildiği gibidir.

Tablo 4.1. Testlerin gerçekleştirildiği ortam

İşlemci	Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 Ghz (2 işlemci)
Birinci hafıza	24 GB
İşletim Sistemi	Windows 10 64 bit

Yapılan bütün deneylerde sınıflandırma başarısı (accuracy), seçicilik (specifity), kesinlik (precision), duyarlılık (sensitivity) ve F-ölçütü (F-Measure) değerlendirme kriterleri karışıklık matrisi (confision matrix) kullanılarak elde edilmiştir. Karışıklık matrisi bir sınıflandırıcı tarafından tahmin edilen sınıflandırmalar hakkında bilgi içerir. Satırlar gerçek değerleri ve sütunlar ise örnek değerleri temsil eder. İki sınıfa sahip bir veri kümesi için örnek karışıklık matrisi Şekil 4.3'de gösterilmiştir.

		TAHMİN EDİLEN SINIF DEĞERİ	
		C^+ (Pozitif Sınıf)	C^- (Negatif Sınıf)
GERÇEK SINIF DEĞERİ	C^+ (Pozitif Sınıf)	DP (Doğru Pozitif)	YN (Yanlış Negatif)
	C^- (Negatif Sınıf)	YP (Yanlış Pozitif)	DN (Doğru Negatif)

Şekil 4.3. İki sınıf için karışıklık matrisi

Sınıflandırma başarısı, sınıflandırıcının örnekleri ne kadar iyi sınıflandırdığını gösterir. Doğru sınıflandırılan örnek sayısının toplam örnek sayısına oranını verir. Sınıflandırma başarısı denklem 4.2'ye göre hesaplanır.

$$Sınıflandırma\ Başarısı = (DP + DN)/(DP + DN + YP + YN) \quad (4.2)$$

Duyarlılık, doğru sınıflandırılan pozitif örneklerin oranıdır ve denklem 4.3'e göre hesaplanır.

$$Duyarlılık\ = DP/(DP + YN) \quad (4.3)$$

Seçicilik, sınıflandırıcının negatif örnekleri nasıl sınıflandırdığını işaret eder ve denklem 4.4'e göre hesaplanır.

$$Seçicilik = DN/(DN + YP) \quad (4.4)$$

Kesinlik, sınıflandırıcının pozitif örnekleri nasıl sınıflandırdığını gösterir ve denklem 4.5'e göre hesaplanır.

$$Kesinlik = DP/(DP + YP) \quad (4.5)$$

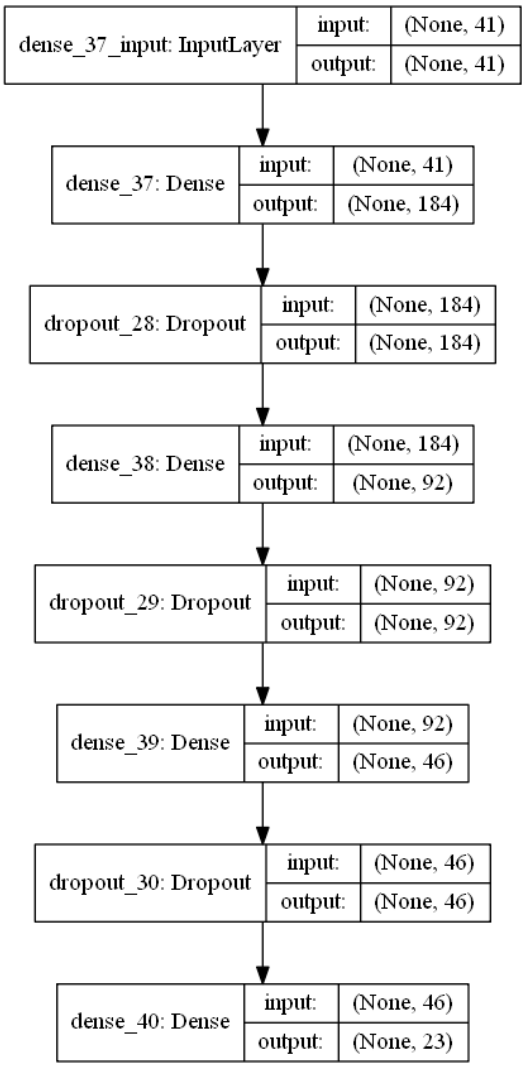
F-ölçütü, kesinlik ve duyarlılık ölçütlerinin harmonik ortalamasıdır. Her iki ölçütü beraber değerlendirip daha doğru sonuç almamızı sağlar ve denklem 4.6'ya göre hesaplanır.

$$F - Measure = 2 * (Kesinlik * Duyarlılık)/(Kesinlik + Duyarlılık) \quad (4.6)$$

Bu çalışmada DoS saldırılarının sınıflandırılmasında üç farklı deney yapılmıştır. Birinci deneyde DoS saldırıları saldırı türlerine göre, ikinci deneyde DoS saldırı tiplerine göre ve üçüncü deneyde de DoS saldırısı olup olmadığına göre sınıflandırılmıştır.

4.1. Saldırı Türlerine Göre DoS Saldırılarının Sınıflandırılması

Yapılan birinci deneyde NSL-KDD veri seti orijinal olarak kullanılmıştır. Orijinal veri kümesinde 22 adet saldırı türü ve normal trafik olmak üzere toplamda 23 adet farklı sınıf bulunmaktadır. Giriş değerleri sayısı 41 tanedir. Veri kümesinde herhangi bir ön işlem yapılmamış, sadece kategorik değerlere sahip protocol_type, service, flag ve class özellikleri sayısal olarak kodlanmıştır. Bu deney için oluşturulan derin öğrenme modelinde ara katman sayısı 3 olarak belirlenmiştir ve her katmandan sonra aşırı öğrenmeyi engellemek için 0,2 seyreltme değerinde seyreltme katmanı eklenmiştir. Aktivasyon fonksiyonu ara katmanlar için sigmoid fonksiyonu ve çıkış katmanı için ise softmax fonksiyonu olarak seçilmiştir. Optimizasyon fonksiyonu olarak Adam ve hata fonksiyonu olarak ise sparse_categorical_crossentropy kullanılmıştır. Derin öğrenme algoritması 100 iterasyon çalıştırılmış ve batch_size değeri ise 100 olarak belirlenmiştir. Birinci deney için tasarlanan derin öğrenme modeli Şekil 4.4'te ve derin öğrenme deney sonuçları Tablo 4.2 gösterilmektedir.



Şekil 4.4. Saldırı türlerine göre tespit için oluşturulan derin öğrenme modeli

Oluşturulan derin öğrenme modeli bir giriş, üç adet dense, üç adet dropout ve bir çıkış olmak üzere toplam sekiz katmandan oluşmaktadır. Ara katmanlardaki node sayıları çıkış katmanından giriş katmanına doğru iki kat arttırılarak hesaplanmıştır.

Tablo 4.2. Saldırı türleri deneyi için derin öğrenme 10-kat çapraz doğrulama sonuçları

Saldırı Türü	**Sınıflandırma Başarısı**	**Duyarlılık**	**Seçicilik**	**Kesinlik**	**F-Ölçütü**
Back	0,99859	0,95921	0,99889	0,86837	0,91153
Land	0,99991	0,77778	0,99994	0,66667	0,71795
Neptune	0,9999	0,99988	0,99992	0,99983	0,99985
Pod	0,99966	0,81095	0,99996	0,97024	0,88347
Smurf	0,99989	0,99509	0,99999	0,99962	0,99735
Teardrop	1	1	1	1	1
Ağırlıklı Ortalama	0,99987	0,99785	0,99990	0,99682	0,99725

Tablo 4.2'de görüldüğü üzere NSL-KDD veri setinin orijinal hali ile yapılan deneylerde tasarlanan derin öğrenme modeli bütün DoS saldırı türleri için yaklaşık %100 oranında sınıflandırma başarısıyla sınıflandırmıştır. Land ve Pod saldırı türlerinde örnek sayıları diğer saldırı türlerine göre biraz daha az olduğu için pozitif örneklerin sınıflandırılmasında yanlış tahminde bulunulmuş ve duyarlılık değerleri daha düşük çıkmıştır. Land saldırı türü içinse saldırı olmayan saldırı türlerini de saldırı olarak algıladığı için kesinlik değeri biraz küçük çıkmıştır. Ağırlıklı ortalama değerlerine bakıldığında derin öğrenme modelinin gayet başarılı olduğu söylenebilir.

Derin öğrenme algoritmasının birinci deneye göre performansını daha iyi karşılaştırabilmek için birinci deney kümesi DVM, NB ve YSA ile de sınıflandırılmıştır. Yapay sinir ağında ara katman sayısı üç olarak belirlenmiş ve katmanlardaki node sayıları derin öğrenme modelinde olduğu gibi sırasıyla 184, 92 ve 46 olarak ayarlanmıştır. YSA 100 iterasyon çalıştırılmış ve öğrenme oranı 0,3 olarak belirlenmiştir. DVM'de kernel fonksiyonu olarak Polykernel seçilmiş, epsilon değeri $1 * E^{-12}$, tolerans parametresi 0,001 ve c değeri 0,1 olarak ayarlanmıştır. Bu üç sınıflandırma algoritmalarına ait deney sonuçları sırasıyla Tablo 4.3, 4.4 ve 4.5 'te gösterilmiştir.

Tablo 4.3 Saldırı türleri deneyi için destek vektör makineleri 10-kat çapraz doğrulama sonuçları

Saldırı Türü	**Sınıflandırma Başarısı**	**Duyarlılık**	**Seçicilik**	**Kesinlik**	**F-Ölçütü**
Back	0,99674	0,60146	0,99976	0,95041	0,7367
Land	0,99994	1	0,99994	0,72	0,83721
Neptune	0,99992	0,9999	0,99993	0,99985	0,99987
Pod	0,99998	0,98507	1	1	0,99248
Smurf	0,99961	0,99849	0,99964	0,98325	0,99081
Teardrop	0,99994	0,99103	1	1	0,99549
Ağırlıklı Ortalama	0,99984	0,99129	0,99991	0,99776	0,99369

Destek vektör makineleri Back saldırı türünde pozitif örnekleri iyi sınıflandıramadığı için duyarlılık değeri düşük çıkmıştır. Land saldırı türü içinse saldırı pozitif örneklerin tamamını sınıflandırmış fakat saldırı olmayan negatif örneklerinde bir kısmını saldırı olarak tespit ettiği için kesinlik değeri biraz düşük çıkmıştır. Diğer saldırı türleri ve ağırlıklı ortalama değerleri için DVM'nin performansı gayet iyidir.

Tablo 4.4. Saldırı türleri deneyi için NB 10-kat çapraz doğrulama sonuçları

Saldırı Türü	Sınıflandırma Başarısı	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
Back	0.91202	0.96234	0.91164	0.07688	0.14239
Land	0.99946	0.88889	0.99948	0.19512	0.32
Neptune	0.94908	0.84656	0.99894	0.99743	0.91582
Pod	0.82994	0.97512	0.82971	0.00907	0.01797
Smurf	0.90953	0.99849	0.90762	0.18824	0.31676
Teardrop	0.99996	0.99439	1	1	0.99719
Ağırlıklı Ortalama	0.94652	0.86117	0.99114	0.92706	0.86262

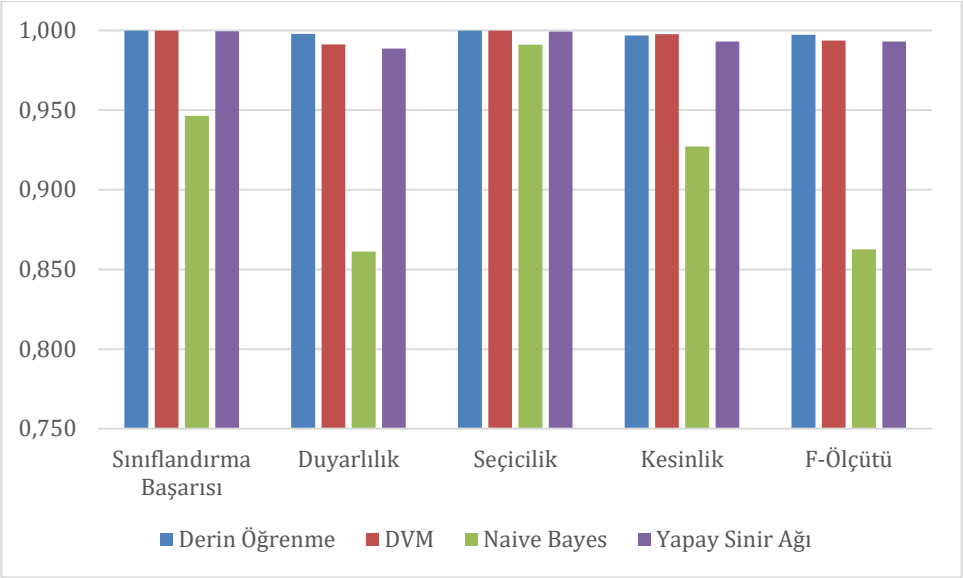
NB algoritması Back, Land, Pod ve Smurf saldırı türlerinde farklı bir saldırıyı veya normal trafiği DoS atağı olarak tahmin ettiği için kesinlik değeri çok düşük çıkmıştır. NB algoritması en iyi Neptune ve Teardrop saldırılarını diğer saldırı türlerine göre daha iyi sınıflandırmıştır.

Tablo 4.5. Saldırı türleri deneyi için YSA 10-kat çapraz doğrulama sonuçları

Saldırı Türü	Sınıflandırma Başarısı	Duyarlılık	Seçicilik	Kesinlik	F-Ölçütü
Back	0.99534	0.50105	0.99912	0.81324	0.62007
Land	0.99986	0	1	0	0
Neptune	0.99951	0.9999	0.99932	0.99859	0.99924
Pod	0.99979	0.92537	0.99991	0.94416	0.93467
Smurf	0.99956	0.99811	0.99959	0.98142	0.98969
Teardrop	0.99994	0.99215	1	1	0.99606
Ağırlıklı Ortalama	0.99944	0.98854	0.99935	0.99314	0.99006

YSA en az örnek sayısına sahip olan saldırı türü Land'in tüm pozitif örneklerini ve Back saldırı türünün de büyük oranda ki pozitif örneklerini yanlış sınıflandırmıştır. Ayrıca Back saldırı türünde farklı bir saldırıyı veya normal trafiği Back saldırı türü olarak tespit ettiği için kesinlik değeri 0.61351 çıkmıştır. Ağırlıklı değerler de NB ve DVM'den iyi derin öğrenmeden daha düşük değerler elde etmiştir.

Bütün yöntemlerin veri kümesinin tamamı için ağırlıklı ortalama değerlerinin grafik olarak karşılaştırılması Şekil 4.5'te verilmiştir. Bu şekil incelendiğinde algoritmalar arasında bariz bir fark olmadığı görülmekle birlikte, derin öğrenmenin diğer algoritmalara göre biraz daha iyi olduğu görülmektedir. Bu deneyde en kötü sınıflandırma sonucunu NB algoritması elde etmiştir.



Şekil 4.5. Saldırı türleri deneyine göre derin öğrenme ve kullanılan makine öğrenmesi algoritmalarının ağırlıklı ortalama değerlerinin karşılaştırılması

Sınıflandırıcıların birinci deney için çalışma zamanları Tablo 4.6'da gösterilmektedir.

Tablo 4.6. Sınıflandırıcıların çalışma zamanları

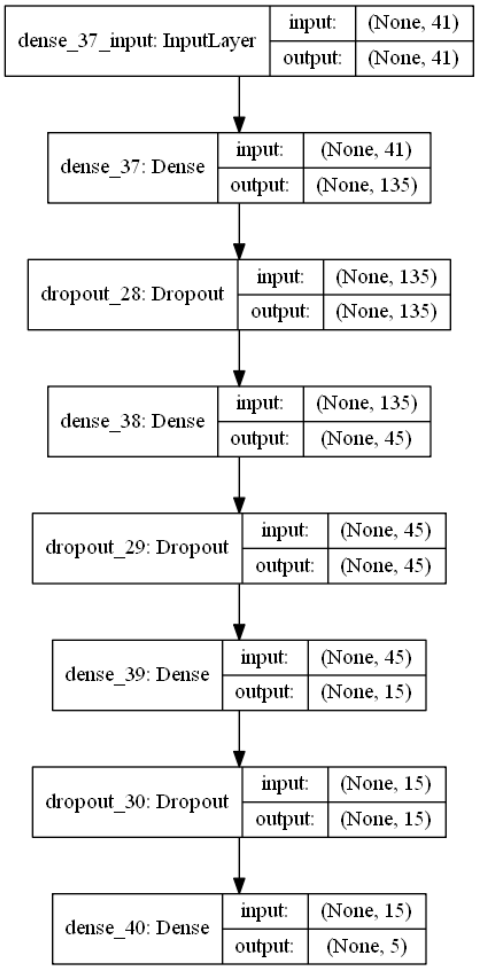
Sınıflandırıcı	**Zaman (s)**
Derin Öğrenme	9549
DVM	590
NB	19
YSA	125992

Sınıflandırıcıların çalışma zamanları analiz edildiğinde NB ve DVM'nin diğer iki sınıflandırıcıya göre çok kısa bir sürede çalışmayı bitirdiği görülmektedir. Derin öğrenme yönteminin ise yapay sinir ağına göre çok daha hızlı çalıştığı görülmektedir.

4.2. Saldırı Tiplerine Göre DoS Saldırılarının Sınıflandırılması

Yapılan birinci deneyde NSL-KDD veri setinde bulunan 22 saldırı türü dört saldırı tipi olarak etiketlenmiştir, böylece normal trafik ile birlikte veri kümesi beş farklı sınıfa ayrılmıştır. Veri kümesinde herhangi bir ön işlem yapılmamış, sadece kategorik değerlere sahip protocol_type, service, flag ve class özellikleri sayısal olarak kodlanmıştır. Bu deney için oluşturulan derin öğrenme modelinde ara katman sayısı 3 olarak belirlenmiştir ve her katmandan sonra aşırı öğrenmeyi engellemek için 0,2 seyreltme değerinde

seyreltme katmanı eklenmiştir. Aktivasyon fonksiyonu ara katmanlar için sigmoid fonksiyonu ve çıkış katmanı için ise softmax fonksiyonu olarak seçilmiştir. Optimizasyon fonksiyonu olarak Adam ve hata fonksiyonu olarak ise sparse_categorical_crossentropy kullanılmıştır. Derin öğrenme algoritması 100 iterasyon çalıştırılmış ve batch_size değeri ise 100 olarak belirlenmiştir. İkinci deney için tasarlanan derin öğrenme modeli Şekil 4.6'te gösterilmektedir.



Şekil 4.6. Saldırı tiplerine deneyi için oluşturulan derin öğrenme modeli

Bu derin öğrenme modeline bir giriş, üç adet dense, üç adet dropout ve bir çıkış olmak üzere toplam sekiz katmandan oluşmaktadır. Ara katmanlardaki node sayıları çıkış katmanından giriş katmanına doğru üç kat arttırılarak hesaplanmıştır.

Yapay sinir ağının ara katman sayısı üç olarak belirlenmiş ve katmanlardaki node sayıları derin öğrenme modelinde olduğu gibi sırasıyla 135, 45 ve 15 olarak ayarlanmıştır. YSA 100 iterasyon çalıştırılmış ve öğrenme oranı 0,3 olarak belirlenmiştir.

DVM'de kernel fonksiyonu olarak Polykernel seçilmiş, epsilon değeri $1 * E^{-12}$, tolerans parametresi 0,001 ve c değeri 0,1 olarak ayarlanmıştır.

Saldırı tiplerine göre yapılan deneysel çalışmalarda DoS saldırılarının sınıflandırma sonuçları Tablo 4.7'de gösterilmektedir.

Tablo 4.7. Saldırı türleri deneyi için derin öğrenme ve kullanılan makine öğrenmesi algoritmalarının 10-kat çapraz doğrulama sonuçları

	Sınıflandırma Başarısı	**Duyarlılık**	**Seçicilik**	**Kesinlik**	**F-Ölçütü**
Derin Öğrenme	**0.99844**	**0.99652**	**0.99954**	**0.99919**	**0.99785**
DVM	0.99175	0.98293	0.99681	0.99438	0.98862
NB	0.9677	0.94861	0.97865	0.96225	0.95538
Yapay Sinir Ağı	0.99267	0.98654	0.99618	0.99329	0.9899

Saldırı tiplerine göre yapılan sınıflandırma işlemleri sonucunda derin öğrenme algoritmasının bütün değerlendirme kriterleri için diğer sınıflandırıcılardan daha iyi sonuçlar verdiği görülmektedir. Sınıflandırıcıların çalışma zamanları Tablo 4.8'de gösterilmektedir. DVM ile YSA sınıflandırıcılarının benzer sonuçlar verdiği gözlemlenirken, NB sınıflandırıcısının bir önceki deneyde olduğu gibi tekrar en kötü sonucu vermiştir.

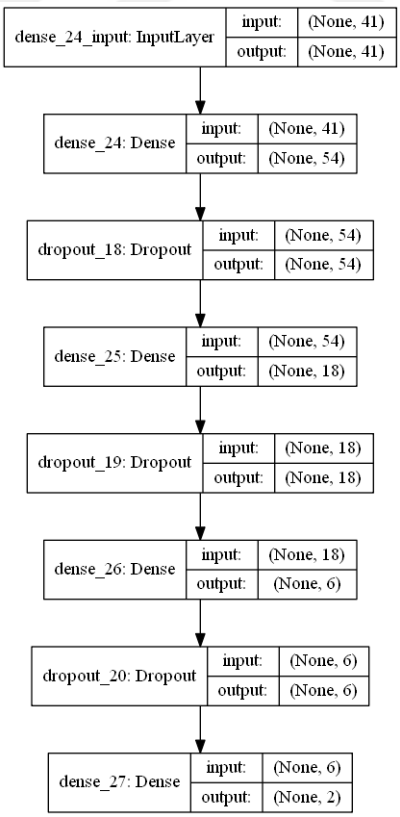
Tablo 4.8. Saldırı tipleri deneyi için sınıflandırıcı çalışma zamanları

Sınıflandırıcı	**Zaman (s)**
Derin Öğrenme	6010
DVM	3555
Bayes Ağları	28
YSA	72604

Sınıflandırıcıların çalışma zamanları analiz edildiğinde NB'nin diğer sınıflandırıcılara göre çok daha kısa sürede çalıştığı görülmektedir. DVM ikinci en kısa çalışan sınıflandırıcıdır. Derin öğrenme ve YSA ise ara katmanlarda ve çıkış katmanında node sayılarının saldırı türü deneyine göre daha az olmasından dolayı daha kısa çalışmışlardır. YSA saldırı türü deneyinde olduğu gibi en uzun süre çalışan algoritma olmuştur.

4.3. DoS Saldırısı Olup Olmadığına Göre Sınıflandırma

Yapılan üçüncü deneyde veri kümesi sınıflandırma işlemi iki farklı sınıfa ayrışmıştır. Back, Land, Neptune, Pod, Smurf ve Teardrop DoS saldırı tipleri DoS olarak ve bunlar dışındaki saldırılar ve normal trafik DoS değil olarak etiketlenmiştir. Sonuç olarak 45927 tane DoS atağı olarak etiketlenmiş örnek ve 80046 tane DoS atağı olarak etiketlenmemiş örnek bulunmaktadır. Oluşturulan bu veri kümesi sınıflandırma açısından en zor veri kümesidir. Çünkü her iki etiket içinde farklı özelliklere sahip birden fazla örnek türü bulunmaktadır. Üçüncü deney çalışması için oluşturulan derin öğrenme modelinde ara katman sayısı 3 olarak belirlenmiştir ve her katmandan sonra aşırı öğrenmeyi engellemek için 0,2 seyreltme değerinde seyreltme katmanı eklenmiştir. Aktivasyon fonksiyonu ara katmanlar için sigmoid fonksiyonu ve çıkış katmanı için ise softmax fonksiyonu olarak seçilmiştir. Optimizasyon fonksiyonu olarak Adam ve hata fonksiyonu olarak ise sparse_categorical_crossentropy kullanılmıştır. Derin öğrenme algoritması 100 iterasyon çalıştırılmış ve batch_size değeri ise 100 olarak belirlenmiştir. İkinci deney için tasarlanan derin öğrenme modeli Şekil 4.7'da gösterilmektedir.



Şekil 4.7. DoS saldırısı olup olmadığına göre sınıflandırma deneyi için oluşturulan derin öğrenme modeli

Oluşturulan derin öğrenme modeli bir giriş, üç adet dense, üç adet dropout ve bir çıkış olmak üzere toplam sekiz katmandan oluşmaktadır. Ara katmanlardaki node sayıları çıkış katmanından giriş katmanına doğru üç kat arttırılarak hesaplanmıştır.

Yapay sinir ağının ara katman sayısı üç olarak belirlenmiş ve katmanlardaki node sayıları derin öğrenme modelinde olduğu gibi sırasıyla 54, 18 ve 6 olarak ayarlanmıştır. YSA 100 iterasyon çalıştırılmış ve öğrenme oranı 0,3 olarak belirlenmiştir.

DVM'de kernel fonksiyonu olarak Polykernel seçilmiş, epsilon değeri $1 * E^{-12}$, tolerans parametresi 0,001 ve c değeri 0,1 olarak ayarlanmıştır.

DoS saldırısı olup olmamasına göre yapılan deneysel çalışmanın sonuçları Tablo 4.9'da gösterilmektedir.

Tablo 4.9. DoS saldırısı olup olmadığına göre sınıflandırma deneyi için derin öğrenme ve kullanılan makine öğrenmesi algoritmalarının 10-kat çapraz doğrulama sonuçları

	Sınıflandırma Başarısı	**Duyarlılık**	**Seçicilik**	**Kesinlik**	**F-Ölçütü**
Derin Öğrenme	**0.99586**	0.99044	**0.99898**	**0.9982**	**0.9943**
DVM	0.98974	0.9813	0.99458	0.99046	0.98586
NB	0.95224	0.95876	0.94849	0.91438	0.93604
YSA	0.97465	**0.99785**	0.9342	0.96354	0.98039

Saldırı tiplerine göre yapılan sınıflandırma işlemleri sonucunda derin öğrenme diğer algoritmalara göre sınıflandırma başarısı, seçicilik, kesinlik ve F-ölçütü değerlerinde daha iyi sonuçlar elde etmiştir. Fakat duyarlılık kriterinde YSA'nın derin öğrenmeye göre biraz daha iyi bir sonuç elde ettiği görülmektedir. Sınıflandırma başarısı ve F-ölçütü değerleri üzerinden bir sıralama yapılırsa derin öğrenmenin birinci, DVM'nin ikinci, YSA'nın üçüncü ve daha önceki deneylerde olduğu gibi NB'nin sonuncu olduğu görülmektedir. Sınıflandırıcıların çalışma zamanları Tablo 4.10'da gösterilmektedir.

Tablo 4.10. DoS saldırısı olup olmadığına göre sınıflandırma deneyi için sınıflandırıcı çalışma zamanları

Sınıflandırıcı	**Zaman (s)**
Derin Öğrenme	6010
DVM	3555
Bayes Ağları	105
YSA	26405

Sınıflandırıcıların çalışma zamanları analiz edildiğinde NB sınıflandırıcısını diğer sınıflandırıcılara göre çok daha kısa sürede çalıştığı görülmektedir. DVM ikinci en kısa

çalışan sınıflandırıcıdır. Derin öğrenme ve YSA ara katmanlarda ve çıkış katmanında node sayılarının saldırı türü deneyine göre daha az olmasından dolayı daha kısa çalışmışlardır. YSA daha önceki bütün deneylerde olduğu gibi en uzun süre çalışan algoritma olmuştur.

Tablo 4.11'de DoS saldırı tipi için literatürdeki NSL_KDD veri kümesini kullanan bazı çalışmalara ait derin öğrenme ve çeşitli makine öğrenmesi algoritmalarının sonuçları gösterilmektedir.

Tablo 4.11. İncelenen çalışmalarda kullanılan veri kümesi ve başarı oranları

Referans	**Algoritma**	**Sınıflandırma Başarısı**
Kaya,2016	NB	0,978
Kaya,2016	DVM	0,993
Kaya,2016	YSA	0,997
Kaya,2016	Karar Ağaçları	0,999
Kaya,2016	K-En yakın Komşu	0,999
Lee ve ark.,2018	Derin öğrenme	0,71
Revathi, Malathi, 2013	Random Forest	0,987
Revathi, Malathi, 2013	CART	0,827
Revathi, Malathi, 2013	J48	0,824

Literatürde yapılan çalışmalar ile bu tez çalışmasında yapılan çalışmalar kıyaslandığında önerilen derin öğrenme modelinin DoS saldırılarının tespitinde kullanılabileceği görülmektedir.

5. SONUÇLAR VE ÖNERİLER

Meşru kullanıcıların kaynakları kullanılarak gerçekleştirilen DoS saldırılarının tespiti oldukça zor bir problemdir. DoS saldırılarının dağıtılmış bir versiyonun da çok sayıda zombilerin oluşturduğu bir ordu bilişim tabanlı kurumsal sistemin hizmet dışı kalmasına sebep olur. Günümüzde hizmet reddi saldırılarını %100 oranında tespit eden herhangi bir güvenlik yazılımı veya güvenlik cihazı bulunmamaktadır. Bu konuda makine öğrenmesi algoritmaları ile literatürde çeşitli çalışmalar yapılmıştır. Bu çalışmalarda kullanılmak üzere farklı veri kümeleri oluşturulmuştur. Makine öğrenmesinin bir alt konusu olan derin öğrenme son yıllarda görüntü tanıma, ses tanıma ve hastalık teşhisi gibi birçok sınıflama problemlerinin çözümünde kullanılmaktadır. Bu tez çalışmasında derin öğrenme metodunu kullanarak NSL-KDD veri kümesindeki DoS saldırılarının sınıflandırması gerçekleştirilmiştir. Derin öğrenme algoritmasının performansını değerlendirmek için deneysel çalışmalar literatürde sıklıkla kullanılan NB, YSA ve DVM sınıflandırıcıları ile karşılaştırılmıştır.

Deneysel çalışmalar aşamasında NSL-KDD veri kümesi üç farklı şekilde kullanılmıştır. Birinci deneyde orijinal olarak kullanılmış ve 23 farklı saldırı tipi içerisinden DoS saldırıları derin öğrenme ve makine öğrenmesi sınıflandırıcıları ile karşılaştırılmıştır. İkinci deneyde NSL-KDD örneklerinin sınıfları dört farklı saldırı tipine ve normal trafik olmak üzere beş farklı tip olarak etiketlenmiştir. Üçüncü deneyde NSL-KDD veri kümesi DoS atağı ve DoS atağı değil olarak etiketlenmiştir. Yapılan her üç deneyde de derin öğrenme algoritması diğer makine öğrenmesi algoritmaları DVM, YSA ve NB'ye göre biraz daha iyi sınıflandırma başarısı elde etmiştir.

İlerleyen çalışmalarda tezden önerilen derin öğrenme modelinin gerçek zamanlı bir yazılım haline getirilecek ve bilgisayar ağlarına yapılan saldırılarda anlık olarak DoS saldırılarını tespit edip kullanıcıyı uyarması sağlanacaktır.

KAYNAKLAR

Aghdam, M. H., & Kabiri, P. 2016, Feature Selection for Intrusion Detection System Using Ant Colony Optimization. IJ Network Security, 18(3), 420-432.

Al-kasassbeh, M., Al-Naymat, G., Hamadneh, N., Obeidat, I., & Almseidin, M. (2018). Intensive Preprocessing of KDD Cup 99 for Network Intrusion Classification Using Machine Learning Techniques. *arXiv preprint arXiv:1805.10458*.

Altwaijry, H., & Algarny, S. 2012, Bayesian based intrusion detection system. Journal of King Saud University-Computer and Information Sciences, 24(1), 1-6.

Arıkuşu, S., 2017, AÇIK KAYNAK KODLU YAPAY ZEKÂ KÜTÜPHANELERİ [online], Türkiye Yapay Zeka İnisiyatifi, https://turkiye.ai/acik-kaynak-kodlu-yapay-zeka-kutuphaneleri/, [Ziyaret Tarihi: 06 Şubat 2018]

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.

Başaranoğlu, E., 2015, TCP Üçlü El Sıkışma – TCP Three Way Handshake [online], Siber Portal, http://www.siberportal.org/green-team/constructing-network-environment/tcp-three-way-handshake/, [Ziyaret Tarihi :21 Ocak 2018]

Bhuyan, M. H., Kashyap, H. J., Bhattacharyya, D. K., & Kalita, J. K. 2013, Detecting distributed denial of service attacks: methods, tools and future directions. The Computer Journal, 57(4), 537-556.

Brownlee, J., 2018, A Gentle Introduction to k-fold Cross-Validation, https://machinelearningmastery.com/k-fold-cross-validation/ [Ziyaret Tarihi: 20 Temmuz 2018]

Büber, E.,2017, Derin Öğrenme Nedir? [online], Cyber Security with Machine Learning, https://cybrml.com/2017/06/06/derin-ogrenme-uygulamalari/ [Ziyaret Tarihi: 06 Şubat 2018]

Caffe, 2018, Caffe, http://www.derinogrenme.com/caffe/, [Ziyaret Tarihi:28 Ocak 2018]

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

Çarkacı, N., 2018, Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler, https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4, [Ziyaret Tarihi: 24 Ocak 2018]

Çatak, F. Ö., Mustaçoğlu, A. F., 2017, Derin Öğrenme Teknolojileri Kullanarak Dağıtık Hizmet Dışı Bırakma Saldırılarının Tespit Edilmesi,2017. The 5th High Performance Computing Conference.

Çelikbilek, İ. 2016, TCP SYN seli saldırısının etkilerini azaltmak için yeni SYN çerezleri gerçeklemesi (Doctoral dissertation).

Çetin, K. A. Y. A., & YILDIZ, O. 2014, Makine öğrenmesi teknikleriyle saldırı tespiti: Karşılaştırmalı analiz. Marmara Fen Bilimleri Dergisi, 26(3), 89-104.

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

Datti, R., & Verma, B. 2010, B.: Feature Reduction for Intrusion Detection Using Linear Discriminant Analysis. In International Journal on Engineering Science and Technology.

DAYIOĞLU, B.,ÖZGİT, A. 2001, İnternet'de Saldırı Tespiti Teknolojileri. İletişim Teknolojileri 1. Ulusal Sempozyumu ve Fuarı, 17-21 Ekim 2001 1, Ankara/Türkiye

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

Enache, A. C., & Patriciu, V. V. 2014, May. Intrusions detection based on support vector machine optimized with swarm intelligence. In Applied Computational Intelligence and Informatics (SACI), 2014 IEEE 9th International Symposium on(pp. 153-158). IEEE.

Ergezer, H., Dikmen, M., & Özdemir, E. 2003, Yapay sinir ağları ve tanıma sistemleri. PiVOLKA, 2(6), 14-17.

Genç, Ö., 2016, Keras ile Derin Öğrenmeye Giriş, https://medium.com/turkce/keras-ile-derin-%C3%B6%C4%9Frenmeye-giri%C5%9F-40e13c249ea8, [Ziyaret Tarihi: 06 Şubat 2018]

GEZGİN, D. M., & BULUŞ, E. 2013, Kablosuz Ağlar İçin Bir DoS Saldırısı Tasarımı. INTERNATIONAL JOURNAL OF INFORMATICS TECHNOLOGIES, 6(3), 17-23.

Gülmüş, M. 2011, Kurumsal bilgi güvenliği yönetim sistemleri ve güvenliği (Doctoral dissertation).

Güven, E. N. 2007, Zeki Saldırı Tespit Sistemi Tasarımı ve Gerçekleştirilmesi.

Hasırcıoğlu, E. S., 2017, Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme Arasındaki Fark Nedir? [online], https://yapayzeka.ai/yapay-zeka-makine-ogrenmesi-ve-derin-ogrenme-arasindaki-fark-nedir/, [Ziyaret Tarihi: 01 Şubat 2018]

İTÜBİDB, 2013, Saldırı Tespit Sistemleri [online], İstanbul, İTÜ Bilgi İşlem Daire Başkanlığı, https://bidb.itu.edu.tr/seyirdefteri/blog/2013/09/07/sald%C4%B1r%C4%B1-tespit-sistemleri [Ziyaret Tarihi: 02 Ocak 2018]

İTÜBİDB, 2013, TCP/IP Protokolü [online], İstanbul, İTÜ Bilgi İşlem Daire Başkanlığı, https://bidb.itu.edu.tr/seyirdefteri/blog/2013/09/07/tcp-ip-protokol%C3%BC [Ziyaret Tarihi: 18 Ocak 2018]

İTÜBİDB, 2013, Tcpdump Kullanımı [online], İstanbul, İTÜ Bilgi İşlem Daire Başkanlığı, http://bidb.itu.edu.tr/seyirdefteri/blog/2013/09/06/tcpdump-kullan%C4%B1m%C4%B1 [Ziyaret Tarihi: 30 Ocak 2018]

İTÜBİDB, 2013, Wireshark'da Paket Süzme (Packet Sniffing) İşlemi, http://bidb.itu.edu.tr/eskiler/seyirdefteri/blog/2013/09/07/wireshark'da-paket-s%C3%BCzme-(packet-sniffing)-i%C5%9Flemi [Ziyaret Tarihi: 15 Mart 2018]

Karakuş, S. B., 2018, Derin Sinir Ağları için Aktivasyon Fonksiyonları, http://buyukveri.firat.edu.tr/2018/04/17/derin-sinir-aglari-icin-aktivasyon-fonksiyonlari/, [Ziyaret Tarihi:01 Temmuz 2018]

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

Kaya, Ç. 2016, Saldırı Tespitinde Makine Öğrenmesi Tekniklerinin Performans Analizi.

KDD, 2018, KDD CUP 1999: Computer network intrusion detection [online], http://www.kdd.org/kdd-cup/view/kdd-cup-1999/Tasks, [Ziyaret Tarihi:28 Ocak 2018]

Keras, 2018, Keras: The Python Deep Learning library, https://keras.io/, [Ziyaret Tarihi: 20 Şubat 2018]

Khalimonenko, A., Kupreev, O., Badovskaya E., 2018, DDoS attacks in Q1 2018 [online], https://securelist.com/ddos-report-in-q1-2018/85373/, [Ziyaret Tarihi: 24 Mayıs 2018]

Lee, B., Amaresh, S., Green, C., & Engels, D. (2018). Comparative Study of Deep Learning Models for Network Intrusion Detection. *SMU Data Science Review*, *1*(1), 8.

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

Ma, T., Wang, F., Cheng, J., Yu, Y., & Chen, X. (2016). A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors*, *16*(10), 1701.

Mirkovic, J., & Reiher, P. 2004, A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review, 34(2), 39-53.

Mukherjee, S., & Sharma, N. 2012, Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technology, 4, 119-128.

Mukkamala, S., Janoski, G., & Sung, A. 2002, Intrusion detection using neural networks and support vector machines. In Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on (Vol. 2, pp. 1702-1707). IEEE.

Niyaz, Q., Sun, W., & Javaid, A. Y. 2016, A deep learning based DDoS detection system in software-defined networking (SDN). *arXiv preprint arXiv:1611.07400*.

Noureldien, N. A., & Yousif, I. M. 2016. Accuracy of machine learning algorithms in detecting DoS attacks types. *Science and Technology*, *6*(4), 89-92.

Özgenç, B., 2014, DDoS Türleri [online], İstanbul, BTRisk Bilgi Güvenliği ve BT Yönetişim Hizmetleri Blog Sayfası, http://blog.btrisk.com/2014/05/dos-turleri.html [Ziyaret Tarihi: 20 Ocak 2018]

Özgür, A., & Erdem, H. 2016, A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ PrePrints, 4, e1954v1.

Patrikakis, C., Michalis M., and Zouraraki O., 2004, Distributed Denial of Service Attacks," The Internet Protocol Journal, vol. 7, number 4., National Technical University of Athens[online], https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-30/dos-attacks.html [Ziyaret Tarihi: 29 Ocak 2018]

Revathi, S., & Malathi, A. 2013, A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research and Technology. ESRSA Publications.

Saied, A., Overill, R. E., & Radzik, T. 2014, June. Artificial Neural Networks in the detection of known and unknown DDoS attacks: Proof-of-Concept. In International Conference on Practical Applications of Agents and Multi-Agent Systems (pp. 309-320). Springer, Cham.

Saunders A. A., 2017, Yapay Zeka ve Makine Öğrenimi Arasındaki Fark Nedir? [online] ,İstanbul, https://www.exastax.com.tr/makine-ogrenimi/yapay-zeka-ve-makine-ogrenimi-arasindaki-fark-nedir/ [Ziyaret Tarihi: 01 Şubat 2018]

Suresh, M., & Anitha, R. 2011, Evaluating machine learning algorithms for detecting DDoS attacks. Advances in Network Security and Applications, 441-452.

ŞEKER, Ş., E., 2008, Çok sınıflı DVM (Multiclass SVM), http://bilgisayarkavramlari.sadievrenseker.com/2008/12/01/cok-sinifli-dvm-multiclass-svm/ [Ziyaret Tarihi: 20.07.2018]

ŞEKER, Ş., E., 2013, K Fold Cross Validation (K Katlamalı Çarpraz Doğrulama) http://bilgisayarkavramlari.sadievrenseker.com/2013/03/31/k-fold-cross-validation-k-katlamali-carpraz-dogrulama/, [Ziyaret Tarihi: 01.07.2018]

ŞENER, S., 2017, Yapay Zeka, Makine Öğrenimi ve Derin Öğrenme Arasındaki Farklar[online], İstanbul, http://www.endustri40.com/yapay-zeka-makine-ogrenimi-ve-derin-ogrenme-arasindaki-farklar/ [Ziyaret Tarihi: 01 Şubat 2018]

ŞİRİN, E., 2017, Bir Bakışta K-Fold Cross Validation ,http://www.datascience.istanbul/2017/08/29/bir-bakista-k-fold-cross-validation/ [Ziyaret Tarihi: 01.07.2018]

Tan, B., Tan, Y., & Li, Y. 2016, Research on Intrusion Detection System Based on Improved PSO-SVM Algorithm. Chemical Engineering Transaction, 51, 583-588.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. 2009, July. A detailed analysis of the KDD CUP 99 data set. In Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on (pp. 1-6). IEEE.

Tavallaee, M., Stakhanova, N., & Ghorbani, A. A. 2010, Toward credible evaluation of anomaly-based intrusion-detection methods. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40(5), 516-524.

Tüzen, E., 2017, Python için 5 Muhteşem Derin Öğrenme Kütüphanesi [online], https://yapayzeka.ai/python-icin-5-muhtesem-derin-ogrenme-kutuphanesi/, [Ziyaret Tarihi: 06 Şubat 2018]

Tüzen, E., 2017, Yapay Sinir Ağlarının Çalışma Prensibi [online], https://yapayzeka.ai/yapay-sinir-aglarinin-calisma-prensibi/, [Ziyaret Tarihi: 11 Şubat 2018]

Uslu, N. Celal. 2009, Veri Madenciliği ile Bilgisayar Ağlarında Yeni Bir Saldırı Tespit Algoritması

Usta, R., 2014, Naïve Bayes Sınıflandırma Algoritması, https://kodedu.com/2014/05/naive-bayes-siniflandirma-algoritmasi/, [Ziyaret Tarihi: 11 Şubat 2018]

Vijayasarathy, R. 2012, A system approach to network modeling for DDoS detection using a Naìve Bayesian classifier.

Vural, Y. 2007, Kurumsal bilgi güvenliği ve sızma testleri. Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, 40.

Vural, Y., & SAĞIROĞLU, Ş. 2008, Kurumsal Bilgi Güvenliği ve Standartlari Üzerine Bir Inceleme. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 23(2).

Yuan, X., Li, C., & Li, X. 2017, May, DeepDefense: Identifying DDoS Attack via Deep Learning. In Smart Computing (SMARTCOMP), 2017 IEEE International Conference on (pp. 1-8). IEEE.

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.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı:	Ayşegül (SUNGUR) ÜNAL
Uyruğu:	Türkiye Cumhuriyeti
Doğum Yeri ve Tarihi:	Ereğli/1990
Telefon:	
Faks:	
e-mail:	aysegulsngr@gmail.com

EĞİTİM

Derece	**Adı, İlçe, İl**	**Bitirme Yılı**
Lise:	Karatay Süleyman Demirel Milli Piyango Anadolu Lisesi Karatay/KONYA	2008
Üniversite:	Selçuk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Selçuklu/KONYA	2013
Yüksek Lisans:	Necmettin Erbakan Üniversitesi Mühendislik ve Mimarlık Fakültesi Endüstri Mühendisliği/Bilgisayar Meram/KONYA	Devam ediyor

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2014-2016	Atiker Yazılım	Yazılım Uzmanı
2016-Devam ediyor	PTT Genel Müdürlüğü	Bilgisayar Mühendisi

UZMANLIK ALANI

Yazılım

YABANCI DİLLER

İngilizce

YAYINLAR

Sungur, C., Babaoglu, I., & Sungur, A. (2015, April). Smart Bus Station-Passenger Information System. In *Information Science and Control Engineering (ICISCE), 2015 2nd International Conference on* (pp. 921-925). IEEE.

Unal, S. A., Hacıbeyoğlu, M., Detection of DDOS Attacks In Network Traffıc Using Deep Learning. International Conference on Advanced Technologies, Computer Engineering and Science (2018, May).