BİLGİSAYAR AĞLARINDA MAKİNE ÖĞRENİMİ ALGORİTMALARINI KULLANARAK İZİNSİZ GİRİŞ TESPİTİ

Meysam AHAN PANJEH

Yüksek Lisans Tezi
Yönetim Bilişim Sistemleri Anabilim Dalı
Dr. Öğr. Üyesi Serdar AYDIN
2018

T.C.
ATATÜRK ÜNİVERSİTESİ
SOSYAL BİLİMLERİ ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

Meysam AHAN PANJEH

BİLGİSAYAR AĞLARINDA MAKİNE ÖĞRENİMİ ALGORİTMALARINI KULLANARAK İZİNSİZ GİRİŞ TESPİTİ

YÜKSEK LİSANS TEZİ

TEZ YÖNETİCİSİ
Dr. Öğr. Üyesi Serdar AYDIN

ERZURUM-2018

T.C.
ATATÜRK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
TEZ BEYAN FORMU

28/08/2018

SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE

BİLDİRİM

Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Uygulama Esaslarının ilgili maddelerine göre hazırlamış olduğum "**BİLGİSAYAR AĞLARINDA MAKİNE ÖĞRENİMİ ALGORİTMALARINI KULLANARAK İZİNSİZ GİRİŞ TESPİTİ**" adlı tezin/raporun tamamen kendi çalışmam olduğunu ve her alıntıya kaynak gösterdiğimi taahhüt eder, tezimin/raporumun kâğıt ve elektronik kopyalarının Atatürk Üniversitesi Sosyal Bilimler Enstitüsü arşivlerinde aşağıda belirttiğim koşullarda saklanmasına izin verdiğimi onaylarım:

Lisansüstü Eğitim ve Öğretim Uygulama Esaslarının ilgili maddeleri uyarınca gereğinin yapılmasını arz ederim *.

☐ Tezimin/Raporumun tamamı her yerden erişime açılabilir.

☐ Tezimin/Raporumun makale için **altı ay,** patent için **iki yıl** süreyle erişiminin ertelenmesini istiyorum.

28.08.2018

Meysam AHAN PANJEH

---

* **LİSANSÜSTÜ TEZLERİN ELEKTRONİK ORTAMDA TOPLANMASI, DÜZENLENMESİ VE ERİŞİME AÇILMASINA İLİŞKİN YÖNERGE**

.....

ÜÇÜNCÜ BÖLÜM

Çeşitli ve Son Hükümler

Lisansüstü tezlerin erişime açılmasının ertelenmesi MADDE 6– (1) Lisansüstü tezle ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

(2) Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.

Gizlilik dereceli tezler MADDE 7– (1) Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

(2) Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

T.C.
ATATÜRK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ

TEZ KABUL TUTANAĞI

SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Dr. Öğr. Üyesi danışmanlığında, Meysam Ahan Panjeh tarafından hazırlanan bu çalışma 28 / 08 / 2018 tarihinde aşağıda isimleri yazılı jüri tarafından Yönetim Bilişim Sistemleri Anabilim Dalı'nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Abdullah NARALAN İmza: .....

Jüri Üyesi : Dr. Öğr. Üyesi Bilal USANMAZ İmza: .....

Jüri Üyesi : Dr. Öğr. Üyesi Serdar AYDIN İmza: .....

Prof. Dr. Mehmet TÖRENEK
Enstitü Müdürü

İÇİNDEKİLER

ÖZET

YÜKSEK LİSANS TEZİ

BİLGİSAYAR AĞLARINDA MAKİNE ÖĞRENİMİ ALGORİTMALARINI KULLANARAK İZİNSİZ GİRİŞ TESPİTİ

Meysam AHAN PANJEH

Tez Danışmanı: Dr. Öğr. Üyesi Serdar AYDIN

2018, 89 sayfa

Jüri: Dr. Öğr. Üyesi Serdar AYDIN (Danışman)
Prof. Dr. Abdullah NARALAN
Dr. Öğr. Üyesi Bilal USANMAZ

İzinsiz giriş tespit sistemi, bilgisayar ağlarında güvenlik tehditlerini algılamak ve önlemek amacıyla kullanılmaktadır. Saldırıları tespit etmek ve tanımlamak için, hesaplamaların hacmini azaltmak amacıyla tanılama sürecinde yararlı olan uygun özelliklerin seçilmesi gerekmektedir. Uygun özelliklerin seçimi, sınıflandırma algoritmalarının uygulanması için yer ve zaman yükünü azaltmaktadır. Bu tez çalışmasında uygun özelliklerin seçimi için K en yakın komşu algoritması, örneklerin sınıflandırılması için ise destek vektör makinesi kullanılmıştır. Destek vektör makinesi verileri önceden belirlenmiş kategorilere göre yeni bir alana almakta, böylece veriler doğrusal olarak (veya hiper düzlemle (Hyper Plane)) kategorilere ayrılabilmektedir. Bu tez çalışmasında 41 özellik içeren KDD99 veri tabanı kullanılmıştır. Sonuçlar, doğruluk ve DR değerinin sırasıyla 92.61 ve 92.06 olduğunu göstermektedir. Ayrıca, FAR değeri 100 tekrar ile karşılaştırıldığında 200 tekrar durumundan daha az bulunmuştur.

Anahtar Kelimeler: İzinsiz giriş algılama sistemi, özellik seçimi, destek vektör makinesi, K en yakın komşu algoritması

ABSTRACT

MASTER'S THESIS

DETERMINATION OF INTENSIVE ENTRY BY USING MACHINE LEARNING ALGORITHMS ON COMPUTERS

Meysam AHAN PANJEH

Advisor: Assist. Prof. Dr. Serdar AYDIN

2018, Page: 89

Jüri: Assist. Prof. Dr. Serdar AYDIN (Danışman)
Prof. Dr. Abdullah NARALAN
Assist. Prof. Dr. Bilal USANMAZ

The detection system is used to identify and prevent computer network security threats. The selection of proper and useful features in the intrusion detection system is one of the most important issues. Selecting features will reduce the space and time burden for the implementation of classification algorithms. In this thesis, the closest neighbor of the K algorithm is used for property selection. In order to identify and identify attacks, it is necessary to select important features that are useful in the diagnostic process in order to reduce the volume of calculations. A support vector machine is also used to classify instances. The support vector machine takes the data in a new area according to the predetermined categories so that the data can be divided into categories linearly (or hyperplane) (Hyper Plane). In this thesis, KDD99 database containing 41 features is used. The results show that accuracy and DR values are 92.61 and 92.06, respectively. Also, the FAR value is less than 200 cases compared to 100 replicates.

Keywords: Intrusion detection system, Feature selection, support vector machine, K nearest neighbors.

SİMGELER ve KISALTMALAR DİZİNİ

ACO	Ant Colony Optimization
CSVAC	Combining Support Vector Machine and Ant Colony
DDOS	Distributed Denial of Service
DIDS	Distributed Intrusion Detection System
DVM	Destek Vektör Makinesi
FN	Yanlış Negatif
FP	Yanlış Pozitif
HIDS	Host-based Intrusion Detection System
ICMP	İnternet Kontrol Mesajı Protokolü (Internet Control Message 555555Protocol)
IDS	Intrusion Detection System
IPS	Intrusion Prevention Systems
KNN	K-Nearest Neighbor
KNNGA	K-Nearest Neighbor- Genetic Algorithm
LVQ	Learning Vector Quantization
NIDS	Network Intrusion Detection System
NSM	Network System Monitor
R2L	Uzaktan Yerele (Remote to Local)
SOFM	Self-Organizing Map
STS	Saldırı Tespit Sistemleri (Intrusion Detection Systems)
SVM	Support Vector Machines
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TN	Doğru Negatif
TP	Doğru Pozitif
U2L	Kullanıcıdan Yerele (User to Local)
U2R	Kullanıcıdan Yöneticiye (User to Root)
UDP	Kullanıcı Veri Birimi Protokolü (User Datagram Protocol)
YSA	Yapay sinir ağları

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

TEŞEKKÜR

Bu yüksek lisans tez çalışması Atatürk Üniversitesi Sosyal Bilimler Enstitüsü öğretim üyelerinden Sayın Dr. Öğr. Üyesi Serdar Aydın'ın yöneticiliğinde yapılmıştır. Çalışmalarım esnasında yardımlarını esirgemeyen, bilgi ve tecrübelerinden faydalandığım çok değerli hocam Sayın Dr. Öğr. Üyesi Serdar Aydın'a en içten şükranlarımı sunarım.

Bu çalışmanın gerçekleşmesine imkân sağlayan Sosyal Bilimler Enstitüsü öğretim üyeleri ve tüm çalışanlarına teşekkürlerimi sunarım.

Çalışmalarım boyunca bana göstermiş oldukları destek ve teşviklerinden dolayı, her şeyimi borçlu olduğum değerli AİLEME sonsuz teşekkür ederim.

Meysam AHAN PANJEH **Erzurum- 2018**

GİRİŞ

Günümüzde internet günlük hayatın bir parçası olarak önemli bir araç haline gelmiş; iş, eğlence ve eğitim gibi birçok alanda insanlara yardımcı olmaya başlamıştır. İnternet son yıllarda iş modellerinin temel bileşenleri arasında yerini almıştır. Ticari faaliyetlerde hem şirketler hem de müşteriler interneti, iş alanlarındaki gereksinimlerini yerine getirmek amacıyla sıkça kullanmaya başlamıştır. İnternetin günümüzdeki kullanım alanlarının günden güne artmasıyla beraber bilginin güvenliğinin sağlanması bu noktada önem arz etmeye başlamıştır. İnternetin kullanıldığı ortamlarda çok sayıda ağ saldırısı tehdidi bulunduğundan dolayı bu tehditleri önlemek amacıyla çeşitli sistemler tasarlanmıştır. İnternet ortamında bilgi güvenliğinin sağlanabilmesi için saldırı tespit sistemleri hem kurumsal hem de kişisel ağlar için önemli bir araştırma konusu haline gelmiştir. Saldırı tespit sistemleri (IDS), ağın harici saldırılara karşı korunmasına yardımcı olmaktadır. Başka bir ifadeyle, saldırı tespit sistemlerinin tasarlanmasındaki amaç ağa dışarıdan gelen herhangi bir saldırı önleyerek savunma duvarı oluşturmaktır. Bu sistemlerde, çeşitli ağ iletişim türleri ve bilgisayar sistemlerinin herhangi bir saldırıda nasıl davranacağı tanımlanabilir fakat güvenlik duvarı bu işlemleri yapabilme yeteneğine sahip değildir.

Ağa dışarıdan gelebilecek tehditler arasında virüsler, ağa izinsiz girişler gibi kötü amaçlı saldırılara hizmet eden yazılımlar olabilir. Virüsler anti-virüs yazılımı yüklenerek ve düzenli güncelleme yapılarak kontrol edilebilmektedir. Ağa yapılan saldırılar ve bilgisayar kaynaklarına yapılan yetkisiz erişim ağ penetrasyonu olarak adlandırılmaktadır. Ağ üzerinde bulunan tehditlerin tespiti ve savunması için son yıllarda güvenlik duvarı üzerinde yapılan birçok çalışma bulunmaktadır. Bu çalışmalar arasında, ağ penetrasyonunun dinamik ve karmaşık etki davranışlarına karşı gelişmiş savunma yönteminin en olası ve sembolik olarak algılanması dikkat çekmektedir.

İnternet ortamında ağa yönelik saldırı risklerinden dolayı, internet tabanlı saldırıları engellemek amacıyla güvenlik duvarı, anti-virüs ya da saldırı tespit sistemleri gibi yazılım veya donanımsal araçlar tasarlanmıştır. Ağ sistemleri, saldırganlardan verilerini ve sistemlerini korumak için yazılım veya donanımsal güvenlik araçlarından bir ya da birkaçını kullanmaktadır. Ağın korunması için tek olarak güvenlik duvarı kullanmak, kurumsal ya da kişisel ağlara yönelik saldırıları engellemek için yeterli değildir. Ağın korunması için güvenlik duvarının açıklarını kapatmak ve ağda güvenliği

sağlayabilmek adına saldırı tespit sistemi de güvenlik duvarı ile beraber kullanılmaktadır. Saldırı tespit sistemleri (IDS), ağa dışarıdan gelen saldırıları otomatik olarak analiz eden yazılım ve donanımsal sistemlerdir. Makine öğrenme teknikleri saldırı tespit sistemlerinde başarılı bir şekilde kullanılmaya başlanmıştır. Literatürde farklı makine öğrenme teknikleri ile gerçekleştirilmiş pek çok saldırı tespit sistemi vardır. Destek vektör makinesi (DVM), Bayes, Karar ağaçları, Yapay sinir ağları (YSA) en sık kullanılan makine öğrenme teknikleridir. Son yıllarda farklı makine öğrenme tekniklerinin saldırı tespit sistemlerinde kullanılması ve performanslarının karşılaştırılması üzerine çok fazla çalışma bulunmamaktadır (Kaya vd., 2014).

İlk saldırı tespit sistemi tasarımı Anderson tarafından 1980 yılında önerilmiştir. Bace, saldırı tespit sisteminin görevini; bilgisayar ya da ağ sisteminde gerçekleşen tüm olayları denetlemek ve kontrol etmek, güvenlik sorunları ortaya çıktığında ilgili personel ve birimleri uyarmak için alarm göndermek ve olası riskleri azaltmak için gerekli tedbirleri almak şeklinde tanımlamıştır (Base vd., 2001). IDS temelinde bir tetikleme mekanizmasına gereksinim duymaktadır.

Saldırı tespit sistemi, mevcut ağ trafiğini veya istekleri analiz ederek müdahaleci etkinlikleri tespit etmeye çalışan bir yazılımdır. Ağa veya makineye giden trafikte izin verilmeyen bir giriş tespit edilirse ağ yöneticisine uyarı göndermektedir. İzinsiz giriş tespit sistemleri ağ güvenliği altyapısında uygulanmakta olup, sistem yaklaşımı iki kategoriye ayrılmıştır. Birincisi zararlı saldırıların tespiti diğeri ise anormal davranış tespitidir. Penetrasyon algılama sistemleri, uzmanlar tarafından tespit edilen ve bildirilen ağ penetrasyon kalıplarının keşiflerini kullanarak saldırıları tespit etmeye çalışmaktadır. Anormal davranış tespiti yaklaşımı, zararlı saldırıların tespiti yaklaşımının geliştirilmesidir. Bu yaklaşım, ağdaki normal davranış kalıpları çıkarılarak etkinin normal davranışlardan sapması olarak açıklanabilir. İzinsiz giriş tespitinin amacı hem iç hem de dış kullanıcılar tarafından bilgisayar sistemlerinin/ağlarının, yanlış kullanımı ve hasara neden olan yetkisiz kullanımın tespit edilmesidir. Bu tez çalışmasında, makine öğrenme teknikleri için yeni somut bir strateji geliştirilmiştir. Çalışmadaki amaç, kötü niyetli saldırıları tanımlamak ve kullanıcılar için erişilebilir düzeyde güvenlik sağlamaktır.

İzinsiz giriş tespit sistemi, bilgisayar ağları için güvenlik tehditlerini tanımlamak ve önlemek için kullanılır. İzinsiz giriş algılama sistemindeki uygun özelliklerin seçimi

en önemli konulardan biridir. Bu tezde K en yakın komşu algoritması özellikleri seçmek için kullanılmıştır. K en yakın komşu algoritması, normal örneklemin parametrik olmayan tekniklerinden biridir. Bu yöntem, giriş vektörlerinin mesafeleri arasındaki yaklaşık mesafeyi hesaplar ve daha sonra etiketlenmemiş noktayı komşularına en yakın K sınıfına ayırır. Bu yöntemde K' nın sınıflandırılması önemli bir parametredir, eğer K çok büyükse, tahmin etmek için kullanılan komşuların tahmin doğruluğunu sınıflandırmak ve etkilemek için çok fazla zaman harcanmış olur. Saldırıları tanımlamak ve hesaplamaların hacmini azaltmak amacıyla tanılama sürecinde yararlı olan önemli özelliklerin seçilmesi gerekmektedir. Örnekleri sınıflandırmak için bir destek vektör makinesi ve 41 özellik içeren KDD99 veri tabanı kullanılmıştır. Sonuçlar, doğruluk ve DR değerinin sırasıyla 92.61 ve 92.06 olduğunu göstermiştir. Doğruluk ve DR değerlerinin 100 tekrarla sırasıyla 93.76 ve 93.95 olduğu bulgusuna ulaşılmıştır. Ayrıca FAR değeri 300 tekrarda, 100 ve 200 tekrardan daha azdır. Önerilen modelin doğruluk yüzdesi; 100, 200 ve 300 tekrarlama için sırasıyla %93,76, %94,90 ve %98,87'dir. Doğruluk yüzdesi ne kadar yüksek olursa sonuç o kadar olumlu olmaktadır.

BİRİNCİ BÖLÜM

1.1. GENEL ARAŞTIRMA

Bilgisayar ağlarının siber uzay iletişiminde çok önemli bir rolü vardır. Bu nedenle birçok internet saldırısına maruz kalmaktadırlar. Bu saldırıların en önemlisi ağ penetrasyonudur. Yaygın olarak internet, global ağlar ve yeni bilgisayar hack araçlarının kullanılmasıyla birlikte bilgisayar penetrasyonu da artmıştır. Bu nedenle bilgisayar ağlarının güvenliği, ağ yöneticileri için bir gereklilik haline gelmiştir. Çoğu zaman ağ penetrasyonu, verilerin çalınmasına ve ağ içinde oluşturduğu arızaların onarılması için yüksek maliyetlere sebep olmaktadır. Saldırı tespit sistemleri, ağ içindeki ve dışındaki şüpheli etkinlikleri tespit etmek, sistemin infiltrasyonunu ve güvenliğini sağlamak amacıyla kullanılmaktadır (Hamed vd., 2018).

İletişim çağında bilgisayar ağları ve uygulamalarının popüler bir hale gelmesiyle birlikte penetrasyon gibi çeşitli güvenlik tehditleri ortaya çıkmış ve büyük problemler oluşturmuştur. Bir bilgisayar sistemi bir ağa bağlandığında, yüksek güvenlik tehditlerine maruz kalmaktadır. Bilgisayar sistemi için bu tehditler arasında virüsler ve saldırganlar vardır. Saldırı tespit sistemleri, sistem güvenlik yöneticilerine penetrasyonu ve saldırıyı tespit etmede yardımcı olmak için kullanılmaktadır. Amaç saldırıyı tespit etmek ve onu önlemek değil, amaç sistem veya ağdaki güvenlik hatalarını tespit edip sistem yöneticisine bildirmektir (Verma vd., 2018). Ağlar ve bilgi sistemleri elektronik saldırılara tabidir. Saldırı (penetrasyon) tespit sistemi, bir bilgisayar sistemine yetkisiz erişimi önlemektedir.

Saldırı tespit sistemlerinin temeli; penetrasyonun tespit edilmesi, raporlarının kaydedilmesi, arızaların durdurulması ve güvenlik yöneticilerine rapor edilmesidir (Papamartzivanos vd., 2018). İnternet kullanımının artmasıyla birlikte bilgisayar ağlarındaki güvenlik açıklarının saldırganlar tarafından tespit edilmesi ve saldırganların becerilerini günden güne geliştirmesiyle sistem/ağ güvenliği geçmişe kıyasla daha önemli bir hale gelmiştir. Herhangi bir sistem veya bilgisayar ağında güvenliği sağlamak aslında gizlilik, kapsamlılık ve erişilebilirliğin üç temel boyutunu sağlamak anlamına gelmektedir.

Penetrasyon tespit sistemleri genellikle her sistem için özel tasarlanmakta ve sadece o sistem üzerinde çalışmaktadır. Penetrasyon teknikleri iki geniş kategoride

sınıflandırılabilir: yanlış saldırı tespit sistemi ve anomali saldırı tespiti (Barbara vd., 2001).

- **Yanlış Saldırı Tespit Sistemi**

Yanlış algılama tekniği, bir kullanıcının faaliyetlerinin bir sisteme girmeye çalışan saldırganların bilinen davranışlarıyla karşılaştırılmasını içerir (Depren vd., 2005).

Kötüye kullanım tespiti, bu saldırıları yöneten kuralları belirleyerek, iyi bilinen saldırıları ve bunların küçük değişikliklerini tespit etmeyi amaçlar. Doğası gereği kötüye kullanım algılaması düşük yanlış alarmlara sahiptir, ancak bilgisinin ötesinde herhangi bir saldırı tespit edememektedir (Barbara vd., 2001).

- **Anomali Saldırı Tespiti**

Anomali tespiti, normal davranıştan sapmaların kasıtlı veya kasıtsız olarak uyarılan saldırıların veya hataların varlığını gösterdiği, izinsiz giriş tespitinin kilit unsurlarından biridir. Anomali saptama yaklaşımları, normal verinin yapısal modellerini temel alır ve gözlemlenen verilerde normal modelden sapmaları saptar (Depren vd., 2005).

Sistemin normal davranış kurallarına ait herhangi bir sapmayı yakalamak için anormallik tespiti yapılır. Anomali saptama, çok fazla yanlış alarm üretme potansiyeline sahiptir ve gerçek izinsiz girişleri yanlış alarmlardan ayırmak için çok zaman ve emek gerektirir (Barbara vd., 2001).

Bu saldırı tespit sistemi, yeni saldırıları tespit etme özelliğine sahip değildir. Anomalilerin aksine, yeni saldırıları tespit etme yetenekleri vardır ancak yanlış uyarı seviyesi yüksektir. Anomali saldırı tespiti, kullanıcıların normal olmayan davranışlarına dayanarak penetrasyonu tespit etmektedir.

Zayıflıklar ve güvenlik hataları olmaksızın teknik olarak bilgisayar sistemleri oluşturmak neredeyse imkansızdır, bilgisayar sistemleri araştırmasında penetrasyonun tespiti özellikle önemlidir. Bir saldırı tespit sistemi, bilgisayar ağını zararlı faaliyetler için izleyen, yönetim ve güvenlik politikalarının ihlalini tespit eden donanım veya yazılımdır, ağ yönetimi bölümüne rapor verir. Bu sistemlerin amacı saldırıyı önlemek değil, sistem/bilgisayar ağındaki saldırıları ve güvenlik sorunlarını tespit edip bunu sistem yöneticisine bildirmektir (Niksefat vd., 2017). Saldırı tespit sistemleri; izleme (izleme-değerlendirme), penetrasyon ve müdahalenin tespiti olmak üzere üç genel

sorumluluğa sahiptir. Bu sistemlere verilen yanıt genellikle farklı formatlarda uyarıyla sınırlıdır.

Bu tez çalışmasında, makine öğrenme algoritmaları kullanılarak bilgisayar ağlarındaki saldırıların tespitine yönelik bir model sunulmaktadır. Önerilen modelde KDD-1999 verileri (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html), makine öğrenme algoritmaları kullanılarak test edilmiş ve öğrenilmiştir (Sullivan vd., 2011). Bilgisayar ağları ve siber uzay güvenliğini geliştirmede önemli rol oynayan teknoloji alanlarından biri de makine öğrenme algoritmalarıdır. Makine öğrenimi, bilgisayarların verileri kullanmasına ve çeşitli işlevleri geliştirmek için öğrenmelerine olanak tanıyan yöntemleri araştırmaktadır. Bir makine öğrenmesinin hedefi, matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarımlar yapmak ve bu çıkarımlardan yola çıkarak bilinmeyene dair tahminlerde bulunmaktır.

Penetrasyon tespit sistemi, ağdaki zararlı girişleri algılayan koruyucu bir sistemdir. Bu sistemler, port tarama ve yüksek trafik algılama gibi zekâya ihtiyaç duyulan durumları tespit edip, ağ yöneticisine raporlama yapmaktadır. Penetrasyon tespit sistemlerinin öngörülemeyen saldırıları tespit etme ve algılama yeteneklerine sahip olmaları için bir tür zekâya ihtiyaçları vardır. Bu durumda, bu sistemler öğrenme yeteneğine sahiptirler. Ağdaki paketleri analiz edip, normal ve sıra dışı kullanıcıları tespit edebilirler.

Bilgisayar sistemlerinde ki önemli konulardan biri de güvenliktir. Bu sistemlerde karşılaşılan en önemli zorluklardan biri penetrasyon (saldırı) tespitidir. Penetrasyon bir dizi etkinlik olarak tanımlanır, bu faaliyetlerin amacı sistemin bütünlüğünü, güvenilirliğini ve belirli bir kaynağa erişimini tehlikeye sokmaktır. Penetrasyon (saldırı) tespit sistemleri bilgisayar ağlarında güvenliği sağlamak için kullanılan tekniklerden biridir. Bu güvenlik sistemleriyle ilgili sorunlar arasında, sistem saldırı uyarılarının yanlış bildirilmesi gösterilebilir. Bu tez çalışmasında, bilgisayar ağlarında penetrasyonun doğru tespit edilebilmesi amacıyla makine öğrenme algoritmalarını kullanan bir model tasarlanacaktır.

1.2. ÖNCEKİ ARAŞTIRMALAR VE KAVRAMLAR

Bir yandan bilgisayar ağ hizmetlerinin giderek yaygınlaşması diğer taraftan ağlara olan saldırıların artması, bu ağların güvenliği için saldırı tespitini önemli bir araştırma

konusu haline getirmiştir. Saldırı tespit sistemi, bilgisayar ağları için kritik olan kullanıcıların erişimini sınırlamak ve denetlemek için bir dizi önceden tanımlanmış kurallar kullanan etkili ve verimli bir güvenlik aracıdır. Bu kurallar uzman kişilerin bilgisine dayanmaktadır. Saldırı tespit sistemleri, tüm kullanıcı penetrasyonlarını tespit eder ve bu saldırıları durdurmak için gerekli adımları atar. Saldırı tespit problemi, bilgisayar sistem güvenliği alanında geniş çapta çalışılmış, son zamanlarda makine öğrenmesi ve veri madenciliği alanında çok fazla dikkat çekmiştir. Penetrasyon (saldırı) tespit sistemleri, mevcut sistem bileşeni cihazların (sunucu, firewall, backbone, storage, antispam v.b.) log kayıtları üzerinde bilinen saldırı desenleri için tarama yaparken aynı zamanda ağ trafiğini de sürekli takip ederek anomali tespitlerinde bulunur.

Son yıllarda araştırmacılar penetrasyon (saldırı) tespit sistemlerini geliştirmek için büyük çaba sarf etmiş ve bu amaca ulaşmak için çeşitli yöntemler kullanmışlardır. Bilgisayar ağındaki saldırının tespit edilmesi için C4.5, Random Forest (Rastgele Orman) ve Random Tree (Rastgele ağacı) gibi karar ağaçları kullanılmıştır (Nejad vd., 2014).

$$\text{Hassasiyet} = \frac{TP}{TP + FP} \tag{1.1}$$

$$Hatirlama = TPR = \frac{TP}{TP + FN} \tag{1.2}$$

$$\text{F - Ölçüsü} = \frac{2 * \text{Hassasiyet} * \text{Hatilana}}{(\text{Hassasiyet} + \text{Hatirlama}} \tag{1.3}$$

$$\text{Doğruluk} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1.4}$$

$$\text{Hata oranı} = 1 - \text{ Doğruluk} \tag{1.5}$$

TN parametresi (doğru negatif), gerçek kategorisi negatif olan ve kategori algoritmasını eşit-negatif olarak sınıflandıran örnek sayısını temsil etmektedir. TP (doğru pozitif), gerçek kategorisi pozitif olan örneklerin sayısını göstermekte ve kategorizasyon algoritması ayrıca kategorileri doğru olarak tanımlamaktadır. FP (yanlış pozitif), gerçek kategorisi negatif olan ve kategori, kategorizasyon algoritması pozitif olanı tanıyan örnek sayısını temsil etmektedir. FN (yanlış negatif), gerçek kategorisi pozitif olan ve kategori, kategorizasyon algoritması negatif olarak tespit edilen

örneklerin sayısını temsil etmektedir. Tablo (1.1)' de, karar ağaçları arasındaki karşılaştırma gösterilmiştir. IG' nin amacı, bir özelliğin eğitim örneklerini kategoriye göre ayırabildiğini belirlemektir. GR'nin amacı IG üzerinde normalizasyon yaparak oluşabilecek sapmaları(bias) önlemektir.

Tablo 1.1. Saldırı Tespiti İçin Karar Ağaçlarının Arasındaki Karşılaştırma (Nejad vd., 2014)

Modeller	Özellik	Doğruluk	Hassasiyet	Hatırlama	F-Tedbir
C4.5	-	94.69	99.90	89.48	93.28
Rastgele Orman	-	95.06	99.94	90.18	93.79
Rastgele ağaç	-	95.80	99.90	91.70	95.06
IG+ Adaboost+ C4.5	33	96.78	99.98	93.58	96.29
IG+ Adaboost+ Rastgele Orman	37	95.63	99.94	91.32	94.70
IG+ Adaboost+ Rastgele Ağaç	40	97.19	99.94	94.44	96.88
GR+ Adaboost+ C4.5	38	96.53	99.87	93.18	95.96
GR+ Adaboost+ Rastgele Orman	31	95.95	99.91	91.98	95.21
GR+ Adaboost+ Rastgele Ağaç	30	96.33	99.96	92.70	95.75

CSVAC[1] modeli (Subaira vd., 2014), bir karınca kolonisi optimizasyon algoritmasının KDD99 verisi üzerindeki müdahaleyi tespit etmek için test edilmiş ve uygulanmış bir destek vektör makinesinin kombinasyonudur. ACO[2] algoritması; optimizasyon problemlerini, özellik seçimini, kümelemeyi ve sınıflandırmayı çözmek için kullanılan evrim optimizasyon algoritmalarından biridir. Bir destek vektör makinesi, kategorilerin sınırlarını tanımlayan n-boyutlu verilerde destek vektörleri içermektedir. Verilerin sınıflandırılması bunlara dayanmakta, bunlardan biri hareket ettirilerek sınıflandırma çıktısı değiştirilebilmektedir. SVM vektör desteğinde yer alan veriler, makine öğrenmesi ve modelin oluşturulması için yeterlidir. Bu algoritma, diğer

[1]Combining Support Vector Machine and Ant Colony (CSVAC)
[2] Ant Colony Optimization (ACO)

veri noktalarına karşı hassas değildir, amaç tüm kategorilerde (destekleyici vektörler) mümkün olan en uzak mesafeye sahip veriler arasındaki en iyi sınırı bulmaktır. CSVAC modeli örnekleri sınıflandırmak için ACO algoritmasından, SVM' den örnekleri öğretmek için yararlanmaktadır. Sonuçlar, SVM ve ACO modellerinde algılama oranının sırasıyla 72.08 ve 84.10 olduğunu, CSVAC modelinde ise 88.04 olduğunu göstermektedir. Karşılaştırmalar, CSVAC modelinin ACO ve SVM' den daha yüksek bir doğruluğa sahip olduğunu göstermektedir.

Genetik algoritmalar ve saldırı tespit sistemi için bulanık mantığa göre karma model önerilmiştir (Morshedur Hassan vd., 2013). Genetik algoritma, optimizasyon problemlerini çözmek için uygun bir popülasyon tabanlı arama algoritmasıdır. Her hesaplama aşamasında, bir dizi nokta oluşturulmaktadır. Optimize edilmek istenen amaç fonksiyonu için GA tarafından oluşturulan popülasyondaki en iyi seçilim, erişilmek istenen optimum noktaya en yakın olanıdır.

Kombinasyon (karma) modelde genetik algoritmalar; kuralları oluşturmak ve örnekleri sınıflandırmak için bulanık mantığı kullanmaktadır. Kombinasyon modelde, üçgen üyelik fonksiyonu örneklerin fuziasyon ve derece üyeliği için kullanılır. KDD99 verisindeki sonuçlar, kombinasyon modelinin tanısal doğruluğunun 86.10 olduğunu göstermektedir. AS, FAS, ACS ve FACS modelleri ACO algoritması tarafından geliştirilmiştir. Çeşitli ACO modellerinin değerlendirilmesi KDD99 verileri üzerinden gerçekleştirilmiştir.

Tablo 1.2. Farklı ACO Modellerinin KDD99 Veri Seti Üzerinde Değerlendirilmesi (Cai vd., 2013)

Veri Kümeleri	**Örnek Sayısı**	**AS**		**FAS**		**ACS**		**FACS**	
		P	**R**	**P**	**R**	**P**	**R**	**P**	**R**
DS1	48298	99.00	75.00	99.00	75.00	98.00	75.00	98.00	64.00
DS2	97677	99.00	75.00	98.00	71.00	99.00	72.00	99.00	78.00
DS3	147059	99.00	73.00	98.00	78.00	98.00	75.00	98.00	71.00
DS4	195609	98.00	76.00	98.00	76.00	97.00	82.00	99.00	76.00
DS5	244882	98.00	69.00	98.00	65.00	98.00	72.00	98.00	74.00

Araştırmacılar Bayes yöntemine dayanarak saldırı tespit sistemi (IDS) için yeni bir model önermiştir (Altwaijry vd., 2012). Bayes yöntemi, tahmin ve sınıflandırma için olasılıksal bir teoridir. Eğitim ve test aşaması DOS, Probe, U2R ve R2L için geçekleştirilmiştir. Dört sınıf için TP, FP, TN ve FN kriterlerinin doğruluğu %99' un üzerindedir.

Self-Organizing Feature Map (SOFM) ve saldırı tespit sistemi için genetik algoritma önerilmiştir (Obimbo vd., 2012). Self-Organizing Feature Map (SOFM); kontrolsüz öğrenme yoluyla, giriş alanının düşük bir boyutuna sahip uzay mekânı üretmek için eğitilen yapay sinir ağının bir türüdür. Yapay sinir ağları ile Self-Organizing Feature Map (SOFM)' daki eşleştirmeler (iki vektör boşluğu arasında bir işlev), giriş özelliğini korumak için komşu fonksiyonun kullanılması nedeniyle farklıdır. Bu özellik, eşlemelerin büyük boyutlu bir veri kümesinin modelleme için uygun olmasını sağlar. Self-Organizing Feature Map (SOFM), rekabetçi öğrenme yöntemi eğitim için kullanılır ve insan beyninin belirli özelliklerine dayanmaktadır. Bu ağlarda kullanılan rekabetçi öğrenim, öğrenmenin her adımında birimlerin birbiriyle etkileşime girmesi için yarışır. Yarışma aşamasının sonunda, ağırlığı diğer birimlerin ağırlıklarına göre değişen tek bir birim kazanır. SOFM-GA kombinasyon modelinde; GA özellik seçimi için, SOFM[3] örnekleri sınıflandırmak için kullanılır. SOFM modelinde, örnekler üzerinde eğitim ve test gerçekleştirilir. Özellik seçiminin amacı, tespitin doğruluğunu arttırmaktır.

Naïve Bayes, C4.5, J48 karar ağacı, NB karar ağacı ve MLP gibi çeşitli veri madenciliği modelleri, KDD99 verileri üzerindeki saldırıları tespit etmek için test edilmiş ve uygulanmıştır (Choi vd., 2008). Çok katmanlı bir yapay sinir ağı; bir giriş katmanı, bir ara katman ve bir çıkış katmanından oluşmaktadır. Girdi nöronlarının sayısı, KDD99 veri örnek büyüklüğü ile aynıdır ve çıkış nöronlarının sayısı, sınıf sayısı ile aynıdır. Veri madenciliği modelleri, örnek eğitim ve testlere dayalı olarak kategorize edilmiştir. Sonuçlar, NB ağacı algılama doğruluğunun diğer modellerden daha fazla olduğunu göstermektedir.

Yapay sinir ağları, tahmin ve sınıflandırma için önemli modellerden biridir. Yapay sinir ağları, örneklerin sınıflandırılması için tekrarlama ve eğitim kullanır.

[3] Self-Organizing Feature Map (SOFM)

SOFM (Self-Organizing Feature Map) yapay sinir ağı modeli ve genetik algoritma kullanılarak saldırı tespit sistemi önerisinde bulunulmuştur (Depren vd., 2005). Değerlendirme, KDD99 veri örneklerinin %10'u üzerinde gerçekleştirilmiş, SOFM yapay sinir ağı tespit doğruluğu 98.69 bulunmuştur.

Araştırmacılar (Gunupudi vd., 2017), anomaliler ve saldırıları tespit etmek için KNN[4] modelini ve J48 karar ağacını kullanmıştır. Algoritmalar NSL-KDD verileri üzerinde değerlendirilmiştir. KNN modelinin tahmini K için farklı değerlere dayanmaktadır. J48 karar ağacı, kuralları temel alan örnekleri eğitip, sınıflandırma yapmaktadır. Tablo (1.3)'te, KNN modelinin sınıflandırması, hata matrisine dayanan çeşitli saldırılar için gösterilmiştir. Bir hata matrisi, sınıflandırma algoritmasının işlevinin temsil edildiği bir matris olarak adlandırılır. Öğrenme algoritmaları, öğretmenli(supervised), öğretmensiz(unsupervised) ve destekleyici (reinforcement) olarak temelde üçe ayrılır. Öğretmensiz öğrenmede; sisteme sadece girdiler verilir ve örneklerdeki parametreler arası ilişkiyi sistemin kendi öğrenmesi beklenir. Bu daha çok sınıflandırma problemleri için kullanılmaktadır. Bu matrisin uygulanması denetimsiz algoritmalarda karşılaştırma matrisi olarak adlandırılır. Her satırda gerçek örnek varsa, matrisin her bir sütunu tahmin edilen değerin bir örneğini göstermektedir.

Tablo 1.3. Hata Matrisine Dayalı Çeşitli Saldırılar İçin KNN Modelinin Sınıflandırılması (Gunupudi vd., 2017)

Saldırılar	**K=1**					
	Normal	**DOS**	**Probe**	**R2L**	**U2R**	**Doğruluk**
Normal	67151	61	49	65	17	99.70
DOS	44	45880	0	3	0	99.90
Probe	56	1	1	11598	0	99.40
R2L	63	1	928	0	3	94.30
U2R	20	1	6	0	25	55.60
Saldırılar	**K=3**					
	Normal	**DOS**	**Probe**	**R2L**	**U2R**	**Doğruluk**
Normal	67116	78	74	65	10	99.60
DOS	53	45868	0	6	0	99.80
Probe	96	2	1	11557	0	99.40
R2L	86	2	904	0	3	92.10

[4] K-Nearest Neighbor (KNN)

U2R	27	1	3	0	21	61.80
Saldırılar	**K=5**					
	Normal	**DOS**	**Probe**	**R2L**	**U2R**	**Doğruluk**
Normal	67092	96	85	62	8	99.50
DOS	71	45848	0	8	0	99.80
Probe	120	2	0	11534	0	99.40
R2L	97	2	894	0	2	91.20
U2R	28	2	1	1	20	66.70

Tablo (1.4)'te J48 karar ağacı modelinin sınıflandırması, karşılaştırma matrisine dayanan çeşitli saldırılar için gösterilmiştir. Tablo (1.4) 'te, beş sınıf için algılama doğruluğu gösterilmektedir. En doğru teşhis Probe sınıfına aittir.

Tablo 1.4. J48 Karar Ağacı Modelinin Hata Matrisine Dayalı Çeşitli Saldırılar İçin Sınıflandırılması (Gunupudi vd., 2017)

Saldırılar	**K=1**					
	Normal	**DOS**	**Probe**	**R2L**	**U2R**	**Doğruluk**
Normal	67244	28	26	41	4	99.80
DOS	17	45903	0	7	0	99.90
Probe	56	8	2	11590	0	99.60
R2L	46	5	942	1	1	96.60
U2R	24	1	2	0	25	83.30

Saldırıları tespit etmek için rastgele orman ve J48 karar ağacı algoritmaları kullanılmıştır (Farnaaz vd., 2016). Rastgele orman ve J48 karar ağacı algoritmaları makine öğrenme tekniklerindendir ve örnekleri eğitime göre sınıflandırmaktadır. Rastgele orman (Random Forest) ve J48 karar ağacı algoritmaları ile yapılan eğitim sonuçlarının değerlendirilmesi için KDD99 veri seti kullanılmıştır. Modelleri karşılaştırmak için doğruluk, DR, FAR ve MCC kullanılmıştır (Farnaaz vd., 2016).

$$DR = \frac{TP}{TP + TN} \tag{1.6}$$

$$FAR = \frac{FP}{FP + TN} \tag{1.7}$$

$$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(FP + TP)(FN + TP)(TN + FP)(TN + FN)}} \tag{1.8}$$

Tablo (1.5)' de, Rastgele Orman (RO) ve J48 algoritmalarının doğruluğu çeşitli saldırılar için gösterilmiştir. Tablo (1.5)' de, dört sınıf için algılama doğruluğu gösterilmektedir. Rastgele ormanın algılama doğruluğu, J48 karar ağacı ile karşılaştırıldığında daha yüksektir.

Tablo 1.5. Rastgele Orman ve J48 Karar Ağacı Algılama Doğruluğu ve Çeşitli Saldırıları (Farnaaz vd., 2016)

Saldırılar	**Rastgele Orman**			
	Doğruluk	**DR**	**FAR**	**MCC**
DOS	99.67	99.84	0.00527	0.99
Probe	99.67	99.82	0.00502	0.99
R2L	99.67	99.82	0.00505	0.99
U2R	99.67	99.84	0.00552	0.99
Saldırılar	**J48 Karar Ağacı**			
	Doğruluk	**DR**	**FAR**	**MCC**
DOS	99.25	99.3	0.00829	0.985
Probe	99.29	99.4	0.0092	0.985
R2L	99.24	99.3	0.008	0.99
U2R	99.28	99.3	0.0072	0.99

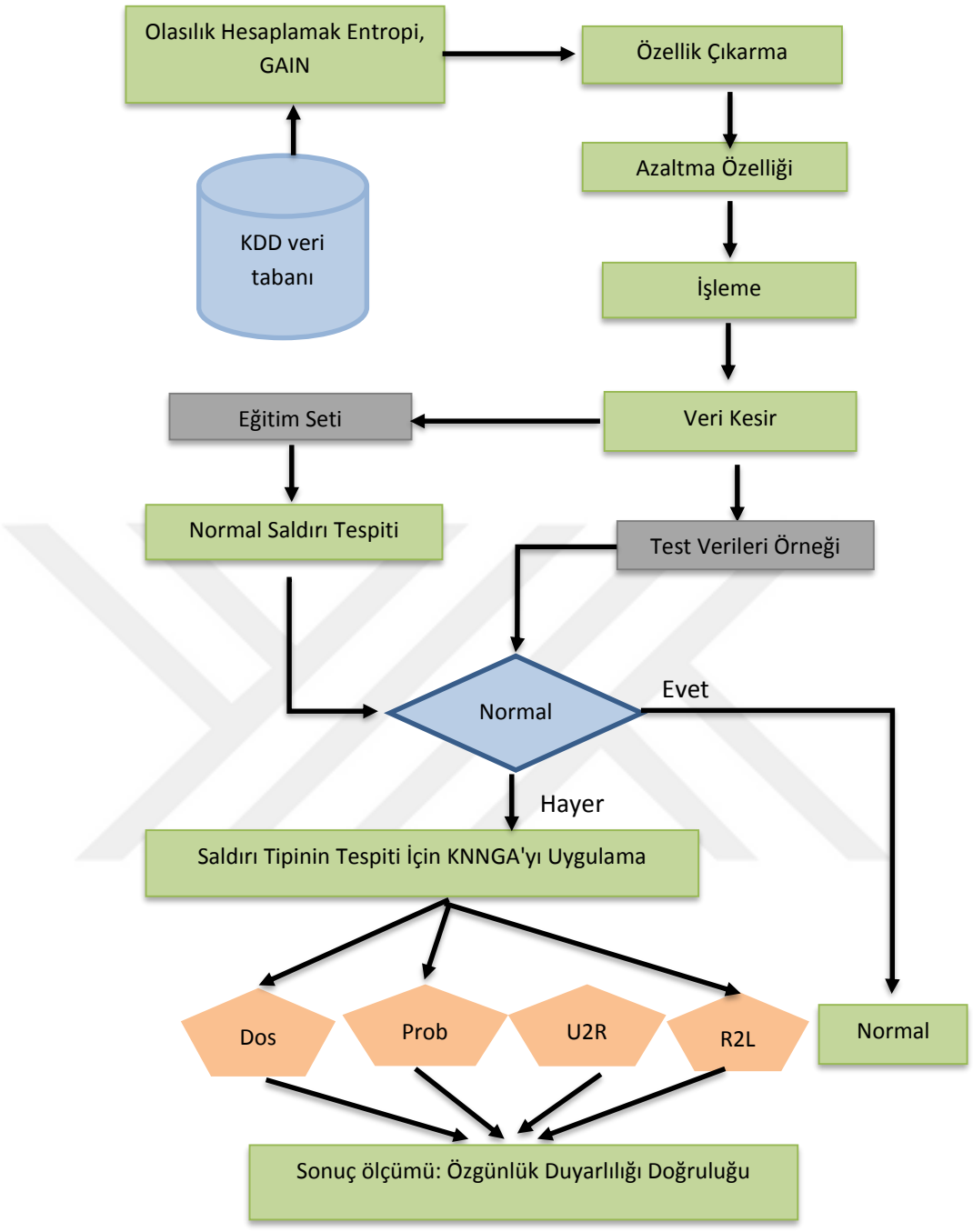
Aziz ve diğerleri (Aziz vd., 2017) bilgisayar ağlarındaki anomalileri tespit etmek için veri madenciliği tekniklerini kullanmışlardır. Anomali tespit sistemlerinde, normal ağ aktivitesinden elde edilen istatistiksel verilere dayanan bir model oluşturulmaktadır. Herhangi bir anda ağın trafik yükü normal ve anormal ağ faaliyetleri arasındaki sınırda dalgalanıyorsa, anomali tespit sistemi sistem yöneticisine bir saldırının başladığını bildirir. Bu şekilde çalışan sistemler yeni saldırıları tespit edebilme yeteneğine sahiptir. Anormal teknikler; zararlı davranışların, sistemin normal davranışlarından ayırt edilebileceği varsayımına dayanmaktadır. Anomalileri tespit etmek için yapay sinir ağı ve karar ağacı yöntemleri önerilmiştir. KDD99 veri seti 13239 numune ile değerlendirilmiş; hassas, geri çağırma ve F ölçümü kriterleri kullanılmıştır. Tablo (1.6), farklı veri madenciliği modellerinin karşılaştırmasını göstermektedir. J48 karar ağacı ve RFT' nin algılama doğruluğu, diğer modellerin algılama doğruluğundan daha büyüktür.

Tablo 1.6. Minkowski Dedektörleri Tarafından Tespit Edilen Anormal Trafiğin Sınıflandırılması İle Elde Edilen Kesinlik Ve Geri Çağırma Değerleri (Aziz vd., 2017).

Sınıflandırıcıyı	**Kriterler**		
Tüm veriler	**Hassasiyet**	**Hatırlama**	**F-Ölçüm**
NB	84.75	73.50	78.73
BF Karar Ağacı	98.24	60.23	74.67
J48 Karar Ağacı	97.68	59.42	73.90
MLP	98.53	61.29	75.57
NB Ağacı	98.44	60.60	75.02
RFT	98.34	58.93	73.69
Sınıflandırıcıyı	**Kriterler**		
Verilerin %20'si	**Hassasiyet**	**Hatırlama**	**F-Ölçüm**
NB	82.86	70.08	75.94
BF Karar Ağacı	98.06	63.48	77.07
J48 Karar Ağacı	98.48	61.06	75.38
MLP	98.19	61.10	75.33
NB Ağacı	98.28	59.70	74.28
RFT	98.51	58.28	73.24

Genetik algoritmayı ve K en yakın komşuyu birleştiren KNNGA[5] (Singh vd., 2015) kombine modeli, saldırı tespitinde kullanılmak üzere önerilmiştir. KNNGA modelinde; KNN sınıflandırma yapmak, GA ise en iyi özellikleri aramak için kullanılmaktadır. GA benzerliğe dayalı en iyi özellikleri bulmakta ve KNN algoritması Öklid mesafesini kullanarak sınıflandırma yapmaktadır.

[5] K-Nearest Neighbor- Genetic Algorithm (KNNGA)



Şekil 1.1. KNNGA'nın Kombine Modelinin Yapısı

Tablo (1.7) 'de; modellerin sonuçları duyarlılık, özgünlük ve doğruluk kriterlerine göre gösterilmiştir. Tablo (1.7) 'de görüleceği üzere, KNNGA modelinin doğruluğu KNN ve SVM modellerinden daha yüksektir. KNNGA modelinin doğruluğu özgünlük kriteri için 100'dür.

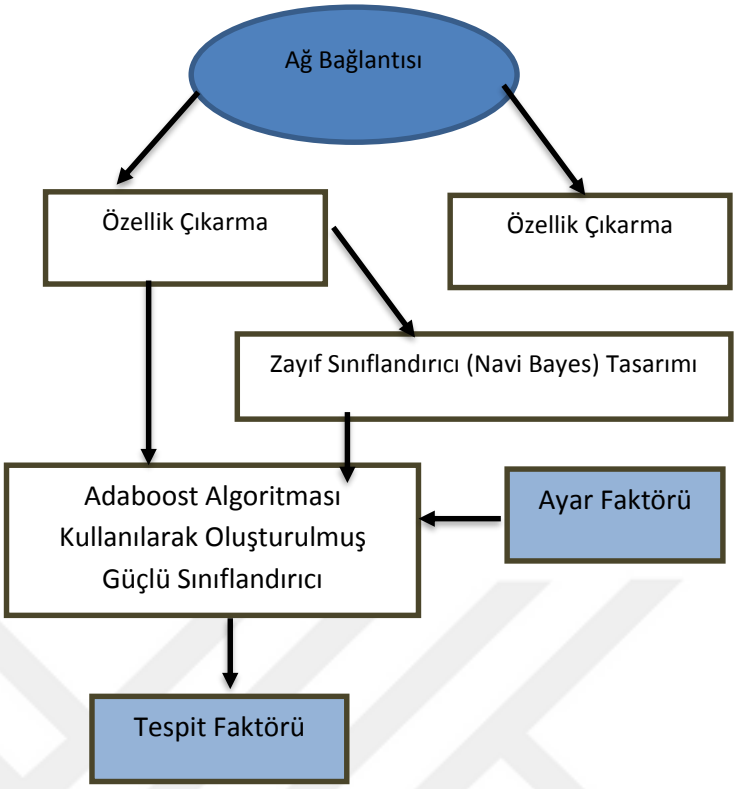
Tablo 1.7. KNNGA Modeli Sonuçlarının KNN ve SVM Modelleri ile Karşılaştırılması

	Modeller	**DOS**	**Probe**	**U2R**	**R2L**
Duyarlılık	**SVM**	99.64	96.64	96.12	75.06
	KNN	98.54	97.69	98.51	98.53
	KNNGA	98.53	98.28	98.47	98.52
Özgünlük	**SVM**	47.81	98.48	91.66	100
	KNN	98.51	98.07	77.11	93.61
	KNNGA	98.53	97.94	98.54	86.52
Doğruluk	**SVM**	74.89	96.72	96.11	75.10
	KNN	98.52	97.70	98.49	98.52
	KNNGA	99.47	98.26	98.47	99.95

KDD99 veri setinin çeşitli yerlerinden random olarak seçilmiş 500 adet veri örneği alınarak LVQ[6] (Linear Vector Quantization) yapay sinir ağı modelinin kullanıldığı bir saldırı tespiti çalışması yapılmıştır (Degang vd., 2007). Bir yapay sinir ağı modeli olan LVQ; her bir çıkış sınıfını temsil eden bir model sınıflandırma yöntemidir ve her biri o sınıfın ağırlık vektörü ile belirlenmektedir. Her sınıfın ağırlık vektörü, eğitim setlerinden biri tarafından başlatılır ve daha sonra öğrenme algoritmalarıyla (denetlenen) optimize edilir. LVQ' yi öğrendikten sonra girdi, vektörün en yakın olduğu sınıfa atılır. Sonuçlar, eğitim ve test seviyelerinde LVQ saptama doğruluğunun sırasıyla %89 ve %80 olduğunu göstermiştir.

Li ve meslektaşı (Li vd., 2010), anomalilerin saptanması için Bayes ve Adaboost 'un hibrit (kombine) modelini önermiştir. Bayes' in yeni hibrit (kombine) modelinde, ilk aşamada örneklerin sınıflandırmaları kullanılmış, ikinci aşamada örneklerin anomalilerini belirlemek için Adaboost kullanılmıştır. Adaboost algoritması, eğitim ve test setlerinde yaygın kullanılan makine öğrenme tekniklerinden biridir ve zayıf öğrenen modelleri güçlü bir model içinde birleştirebilecek yükseltmede bulunur. KDD99 verisindeki sonuçlar, saptama doğruluğunun %96,32 olduğunu göstermiştir.

[6] Learning Vector Quantization (LVQ)



Şekil 1.2. Anomalilerin Saptanması İçin Bayes ve Adaboost 'un Hibrit (Kombine) Modelinin Yapısı (Li vd., 2010)

Tablo 1.8, bilgisayar ağlarında saldırı tespiti için önerilen modellerin karşılaştırmasını göstermektedir.

Tablo 1.8. Bilgisayar Ağlarında Saldırı Tespit İçin Önerilen Modellerin Karşılaştırması

Veri tabanı	Algılama Doğruluğu	Model	Referans
KDD99	94.69	C4.5Karar Ağacı	(Nejad vd., 2014)
KDD99	84.10	Karınca Kolonisinin Optimizasyon Algoritması[7]	(Subaira vd., 2014)
	72.08	Destek Vektör Makinesi[8]	
	88.04	Bir Karınca Kolonisi ve Destek Vektör Makinesi	
KDD99	86.10	Kombine Genetik Model ve Bulanık Mantık[9]	Morshedu vd., 2013)
KDD99	95.29	Karınca Kolonisinin Optimizasyon Algoritması	(Cai vd., 2013)
KDD99	85.35	Bayes Ağlar[10]	(Altwaijry vd., 2012)
KDD99	78.27	SOFM[11]	(Dagli, 2012)
KDD99	Modellerin Hata Oranının Değerlendirilmesi	Naïve Byes Algoritması	(Choi vd., 2014)
		C4.5 Karar Ağacı	
		J48 Karar Ağacı	
		Çok katmanlı Yapay Sinir Ağı[12]	
		NB Tree	
KDD99	98.69	SOFM	(Depren vd., 2005)
KDD99	%96'nın üstünde	KNN	(Gunupudi vd, 2013)
		J48 Karar Ağacı	
	%99'un üstünde	Random Forest	(Farnaaz vd., 2016)
		J48 Karar Ağacı	
KDD99	84.75	NB	(Aziz vd., 2017)
	98.24	BF Karar Ağacı	
	97.68	J48 Karar Ağacı	
	98.53	MLP	
	98.44	NB Karar Ağacı	
	98.34	RFT	
KDD99	%99'un üstünde	SVM	(Singh vd., 2015)
		KNN	
	99%	KNNGA	
KDD99	89%	LVQ	(Degang vd., 2007)
KDD99	96.32%	NB-Adaboost	(Li vd., 2010)

[7] Ant Colony Optimization
[8] Support Vector Machine
[9] Fuzzy Logic
[10] Bayesian Networks
[11] Self-Organizing Map
[12] Multilayer Perceptron

Tablo (1.8)'de görüleceği üzere karınca kolonisi, karar ağacı C4.5 ve Self-Organizing Feature Map (SOFM) için optimizasyon algoritması daha yüksek tespit doğruluğuna sahiptir. Destek vektör makinesi modeli diğer modellere göre daha az tespit doğruluğuna sahiptir. Ayrıca, C4.5 karar ağacındaki tespit doğruluğu yüzdesi ve karınca kolonisi optimizasyon algoritması çok yakın bulunmuştur. Self-Organizing Feature Map (SOFM)' da rekabetçi öğrenme süreci girdi kalıpları halinde düzenlenmektedir. Ağda yapılandırılan birimlerin konumu, ağda anlamlı bir koordinat sisteminin giriş karakteristikleri oluşturulacak şekilde düzenlenmiştir. Bu nedenle, Self-Organizing Feature Map (SOFM) modeli, birimlerin konumlarının girdi deseni özelliklerine karşılık gelen topoğrafik desenli bir giriş haritası oluşturmaktadır. Ayrıca Bayes ağlarının Bayesian kullanma olasılığından dolayı, doğru bir şekilde tespit yapabileceği söylenebilir.

İKİNCİ BÖLÜM

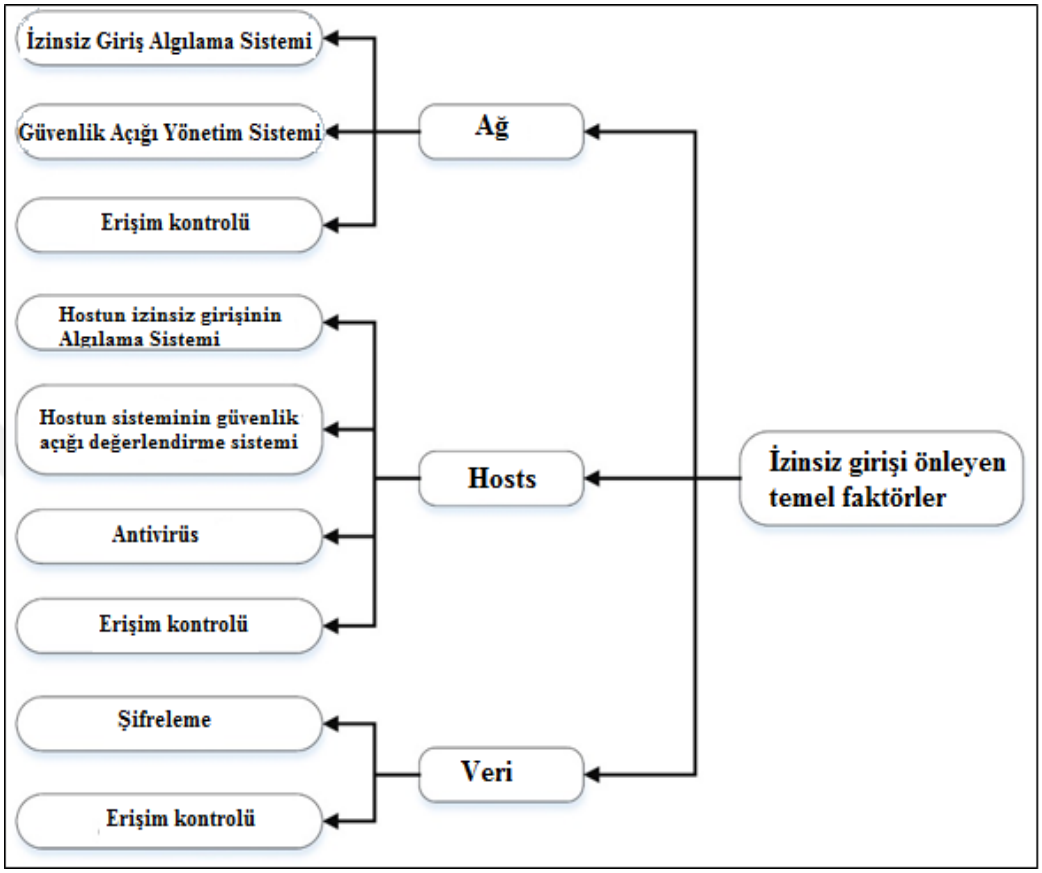
2.1. İNTERNETTE GÜVENLİK

Günümüzde dünyayı çevreleyen ağ (World Wide Web)- çeşitli hizmetler, çevrimiçi iletişim ve e-ticaret için yeni bir araç olarak bilinmektedir. Birçok organizasyon-veri merkezi, iletişim dünyasına ve geniş internet hizmetinin kullanımına kapsamlı bir girişimde bulunmuştur. İnternet'in kullanıcılarına sunduğu çeşitli hizmetler göz önüne alındığında, kullanıcı sayısı gün geçtikçe artmaktadır. İnternetteki milyonlarca bilgisayar artık birbirine bağlı bir şekilde kullanıcılarına farklı hizmetler sunmaktadır. Dünyayı çevreleyen ağın hızla yayılımı ve kullanıcılara merkezi kontrol uygulamanın imkânsızlığı göz önünde bulundurulduğunda, internet üzerinde bilgi güvenliğini sağlamak en önemli konulardan biri olmuştur. Dünyada, hacking de dahil olmak üzere çeşitli nedenlerle internet sitelerinde bulunan bilgileri etkileme eğilimi olan insanlar bulunmaktadır (Wang vd., 2004).

E-ticaretin gelişmesi ve internetin dünya ticareti için yeni bir araç olarak kullanılması, internette bilgi güvenliği konusunu önemli bir hale getirmiştir. İnternete bağlanan kuruluşlar, bilgilerini korumak için belirli güvenlik sistemlerini kullanmak zorunda kalmıştır. Bu bağlamda, ilk adım uygun güvenlik politikası ve güvenlik yöntemi belirlemektir. Bilgisayar ağlarında bilgi koruma yönetimi, bilginin yetkisiz erişime karşı korunması, bilginin doğruluğunu korumak ve bilgilerin zamanında erişilebilirliğini sağlamak olmak üzere üç ana bölümden oluşmaktadır (Debar vd., 2005).

Güvenlik önlemlerinden biri de internete bağlı bilgisayar ağlarının etkisinin azaltılması ve belirli şebeke hizmetlerinin iç ve dış erişiminin kontrolüdür (Debar vd., 2005). Bu yöntemde, güvenlik duvarı adı verilen özel bir sistem kullanılarak, internete bağlı bir kuruluşun koruma politikası uygulanmaktadır. Bir güvenlik duvarı kullanarak ağ yöneticisi, yetkisiz kullanıcıların ve saldırıların ağ dışından iç ağ hizmetlerine erişimini kısıtlar (Yang vd., 2012). Güvenlik duvarı, ağ trafiği giriş-çıkışını izleme ve kontrol etmekten sorumlu bir tür ağ güvenlik sistemi olarak adlandırılmaktadır (Kim vd., 2002). Güvenlik duvarının birçok çeşidi bulunmaktadır. Paket filtreleri veya ağ katmanı güvenlik duvarları olarak bilinen bazı güvenlik duvarları, adres ve bağlantı noktaları gibi bilgileri denetleyerek hangi paketlerin geçeceğini veya engelleneceğini

belirler. Güvenlik duvarının diğer çeşitleri arasında uygulama katmanında ve Proxy sunucusu üzerinde çalışanlar bulunmaktadır (Kim vd., 2002). Şekil (2.1)'de, esas olarak penetrasyonu önlemek için üç temel faktör gösterilmektedir.



Şekil 2.1. Penetrasyonu Önlemek İçin Temel Faktör

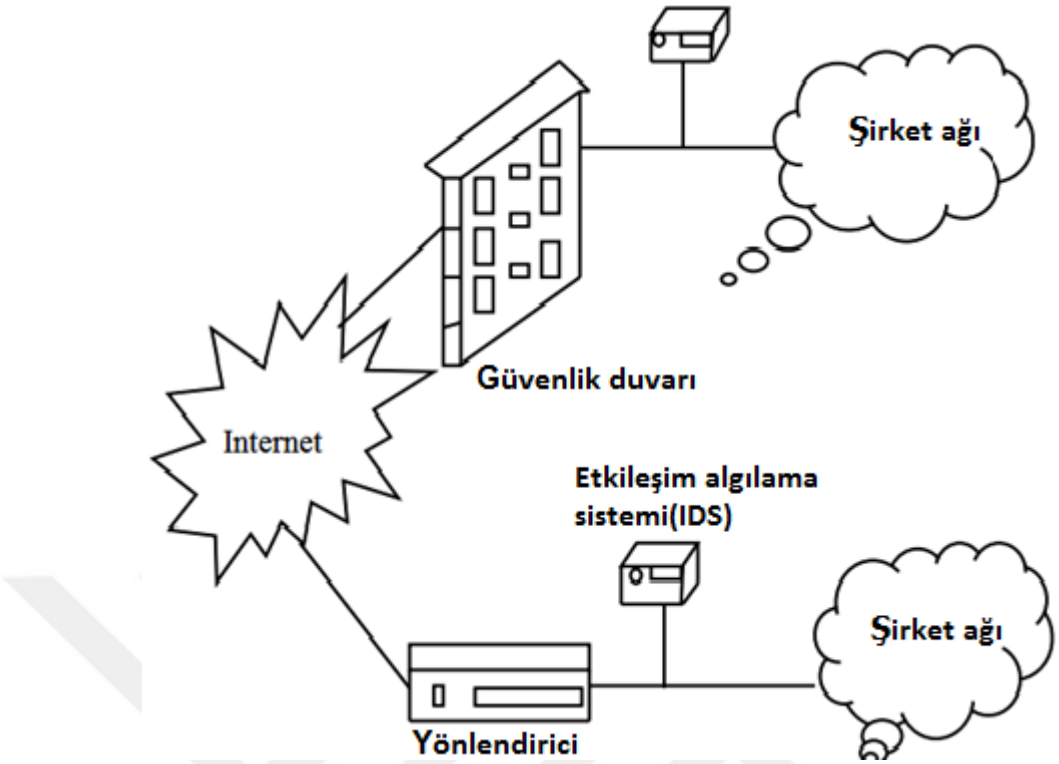
Ağ seviyesi, merkezin geniş alan ağına (WAN) ve yerel alan ağına (LAN) bağlıdır. Yerel alan ağı çeşitli ölçeklerde olabilir. Bir şirket veya okul yerel ağında uç sistemler, aktif cihazlar, sunucular, güvenlik cihazları, vpn bağlantıları bulunabilir. Geniş alan ağı; sunucular, kişisel bilgisayarlar, anahtarlar, yönlendiriciler vb. araçlarla ilgilidir. Her takımın bir dizi ayarlanabilir parametresi vardır ve yanlış yapılandırıldıklarında, güvenlik açıkları oluşabilir (Wang, 2014). Bu parametreler kayıt defteri ayarları, hizmetler (servisler), operasyonel işlevler ya da işletim sistemi API'leri veya önemli yazılım yardımcı programlarını içermektedir.

2.2. SALDIRI TESPİT SİSTEMİ

Saldırganlar veya bilgisayar hackerlarının amacı bilgisayar ağlarına saldırmaktır. Sistem ayarlarının şifrelenmesi ve emniyet altına alınması, bilgisayar korsanlarının ağa büyük ölçüde erişmesini önler. Penetrasyon tespit sistemleri ve güvenlik duvarları, ağa yapılan saldırıları büyük ölçüde önleyebilecek teknolojilerdir. Bir saldırı tespit sistemi, ağdaki mevcut etkinlikleri analiz eder ve gerçekleştirilen faaliyetin müsaade edilebilir olup olmadığını, olağandışı veya yetkisiz bir girişi saptamak için veri tabanındaki bilgilerini kullanarak belirler (Kizza, 2017). Ayrıca bu etkinliğin ağa zarar verip vermeyeceğini belirler ve kullanıcılarını bu tür aktiviteler hakkında uyarır.

Genellikle imza tabanlı tanı olarak bilinen Misuse Intrusion Detection System (Mohamadi vd., 2008), şablon olarak saklanan önceden tanımlanmış etki desenlerini (imza) içerir. Böylece her şablon farklı bir özel etki yaratır ve penetrasyon sisteminde saklanan modellere uygun bir model olması durumunda ilan edilir. Bu yöntemlerde, algılayıcı genellikle bir imza kümesine veya saldırı modeline sahiptir ve ağ trafiğini kontrol ederek verilerinde saklananlara benzer desenler bulmaya çalışır. Bu tip yöntemler sadece bilinen saldırıları tespit edebilmektedir ve yeni ağ saldırıları durumunda bunları algılayamamaktadır. Bu nedenle ağ yöneticisi, saldırı girişine her zaman yeni saldırıların modelini eklemelidir (Tylman vd., 2008). Bu yöntemin avantajı, sisteme aynı desen verilen saldırıları tespit etmede hassas olmasıdır.

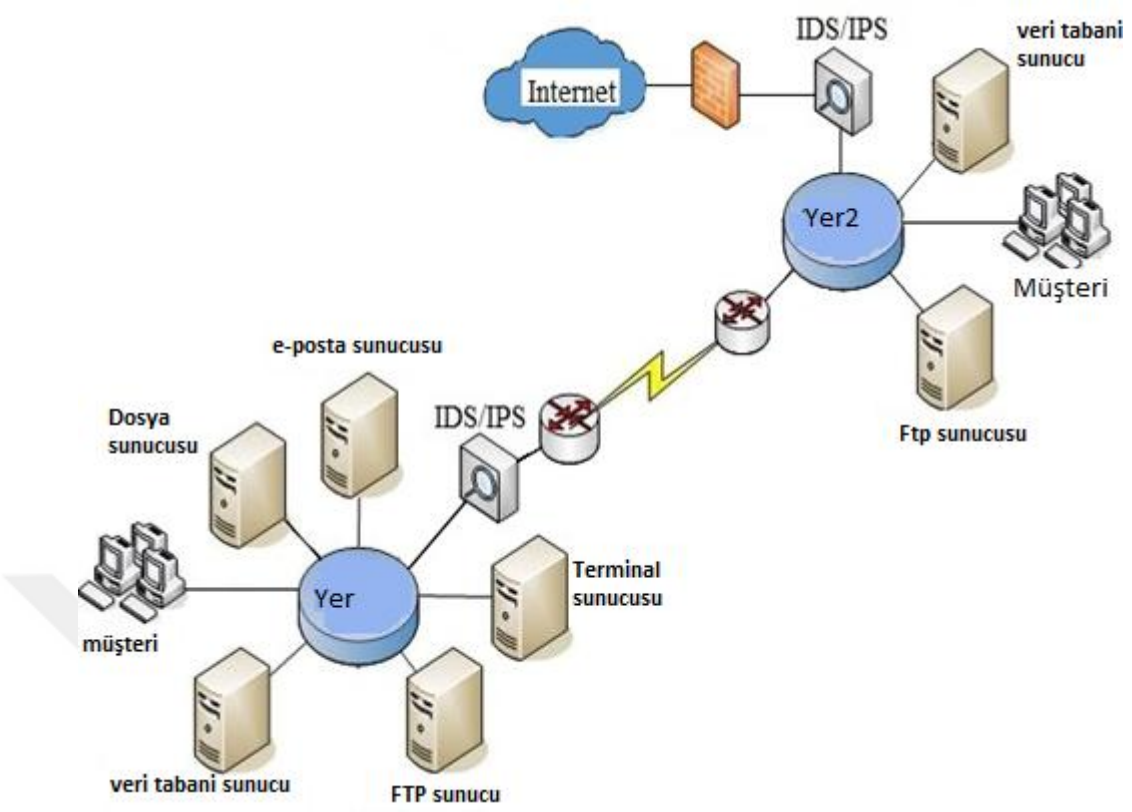
(Shah,2018) göre; farklı donanımlar üzerine opensource snort ve suricata yazılımları kurularak IDS açısından performans farkları gözlenebilir. Bilgisayar ağını kötü amaçlı etkinlikler, yönetim ve güvenlik ilkelerindeki hatalar için izler, ağ saldırılarını bildirmek için ağ yönetimi bölümüne raporlar gönderir. Saldırı tespit sistemi, herhangi bir dahili ve harici kullanıcı tarafından sömürü veya hasar ile sistem kullanımının tespit edilmesinden görevlidir. Bu sistemlerin amacı saldırıları önlemek değil, sistemde veya bilgisayar ağında saldırıları tespit etmek, güvenlik hatalarını algılamak ve sistem yöneticisine bildirmektir. Genel olarak güvenlik duvarlarının yanında saldırı tespit sistemleri kullanılır ve bunlara tamamlayıcıdır.



Şekil 2.2. Saldırı Tespit Sisteminin Yapısı (www.coherentnews.com/cloud-ids-and-ips-are-methodology-used-to-escalate-the-security-level-of-network.com)

Saldırı tespit sistemi teknolojileri, ağdaki trafiği güvenlik duvarından daha ayrıntılı olarak analiz etmektedir. IDS araçları, saldırıyı ağ yöneticisine bildirmek için kötü amaçlı trafiği tespit etmekte ancak IPS[13] cihazları bir adım daha ileri giderek hassas trafiği otomatik olarak engellemektedir (Yousufi vd., 2017).

[13] Intrusion Prevention Systems (IPS)



Şekil 2.3. IDS ve IPS Sisteminin Yapısı (www.slideshare.net/DerisStiawan/network-attack-and-intrusion-prevention-system.com)

IDS bir paket-sniffer tipidir, ağda alınan ve gönderilen tüm paketleri almakta ve analiz etmektedir (Qadeer vd., 2010). Saldırı tespit sistemi, ağdaki çeşitli iletişim protokolleri ile çalışabilme özelliğine sahiptir. Ağ denetleyicisi veya ağ analizcisi olarak da adlandırılan bir paket algılayıcısı, ağ yöneticileri tarafından ağda bulunan trafiği görüntülemek ve hata ayıklamak için kullanılır. Paket algılayıcısı, hata paketlerinin ve kritik darboğazların yardımı ile ağ içerisinde veri transferi için tanımlanacak ve gerekli altyapı sağlanacaktır (Chen vd., 2015). Basit bir şekilde ifade etmek gerekirse bir paket dinleyicisi, incelemek ve analiz edilmelerine izin vermek için ağa belirli bir ara yüzden gönderilen tüm paketleri toplayacaktır. Paket dinleyicisi (Paket Sniffer) uygulamaları şunlardır:

- Ağ problemlerini analiz etme
- İzinsiz saldırıların tespiti
- Dahili ve harici kullanıcılar tarafından ağın kötüye kullanımının algılaması
- Bir ağa saldırmada bilgi edinme

- Ağda harici ve dahili kullanıcılarının kullanımını izleme
- Ağın veri akışındaki verileri izleme
- Ağ istatistiklerinin derlenmesi ve raporlanması
- Şüpheli bilgileri ağ trafiğinden filtreleme
- Ağdaki istemci/sunucu iletişimi ile ilgili hata ayıklama
- Ağ protokollerinde hata ayıklama

Saldırganlar, ağda bir paket dinleyicisi (packet sniffer) kurarak tüm ağ trafiğini toplayıp analiz edebilmektedirler. İsim, şifre bilgisi genellikle düz metin halinde ve şifrelenmiş biçimde iletilir, yani paketlerin analiz edilmesiyle bu hassas bilgilerin görüntülenmesi mümkün olacaktır. Paket dinleyicisi, sadece belirli bir alt ağdaki paketler hakkında bilgi toplayabilmektedir. Bu nedenle, bir saldırgan ağda bir paket dinleyicisini kuramaz ve ağdaki diğer makineleri kullanabilmek için kullanıcı ağına erişemez, isim ve şifre toplayamaz. Saldırganların amacına ulaşması için ağdaki bilgisayarda bir paket dinleyicisi tasarlamaları ve hedeflerine ulaşmaları gerekmektedir.

2.3. SALDIRI TESPİT SİSTEMLERİNİN TÜRLERİ

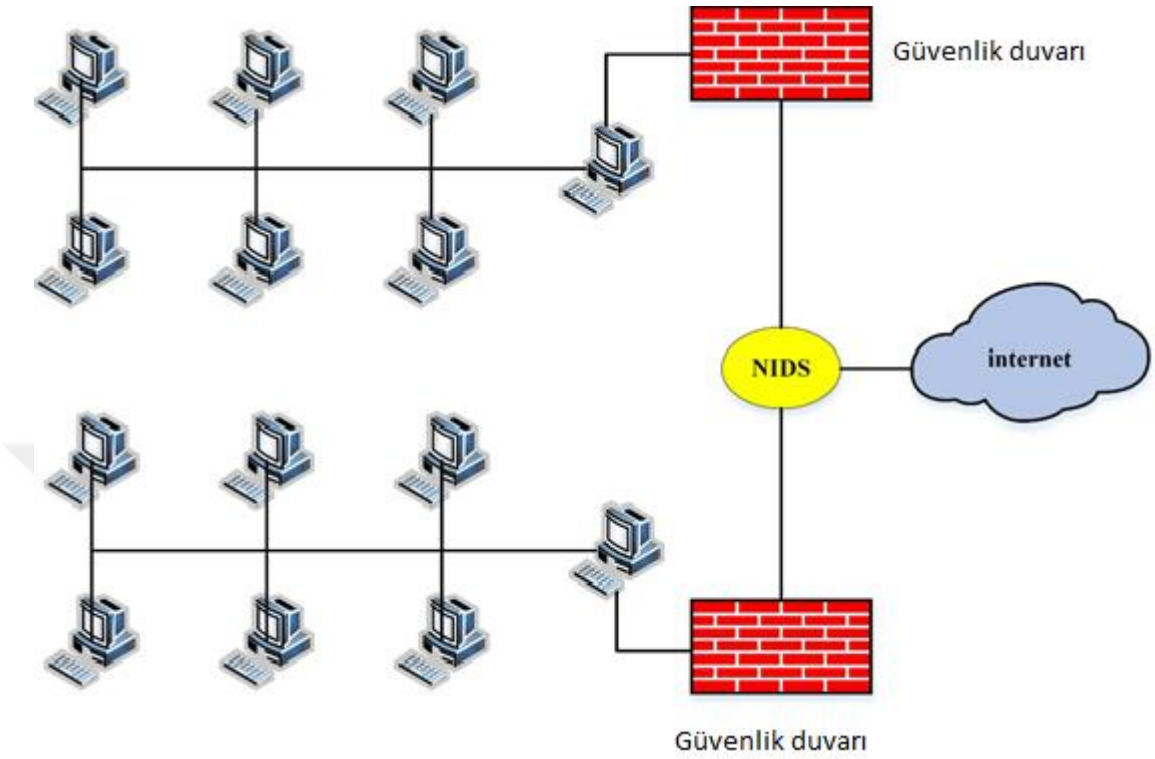
Saldırı tespit sistemleri, saldırıyı tanımlamak için arama türü açısından üç kategoriye ayrılmaktadır.

2.3.1. Ağ Saldırı Tespit Sistemi

Ağ saldırı tespit sistemi kısaca NIDS[14] olarak da adlandırılmaktadır (Jadhav vd., 2013). Aslında, kendisini ağa bağlayan, ağ trafiğini izleyen ve raporlarını sunan bir IDS modelidir. Bu sistemler genellikle ağ güvenlik duvarının önüne yerleştirilir. NIDS'i bir ağ güvenlik duvarı önüne yerleştirmek, IDS' nin tüm trafik akışını tam olarak izlemesini sağlar. Bu çözüm, ağdaki büyük miktarda girdi ve çıktı verilerini analiz eder, ağda iletilen tüm verileri görmesine izin verir. NIDS' in güvenlik duvarının arkasına yerleştirilmesi durumunda sadece güvenlik duvarına iletilen ağ trafiğinin izlenmesi anlamına gelmektedir. Bu durumda güvenlik duvarına iletilmeyen diğer ağ trafiği NIDS tarafından izlenemez (Yuan vd., 2017). Bu çözüm daha az veriyi analiz ederken, ağ

[14] Network Intrusion Detection System (NIDS)

içinde meydana gelen çok sayıda saldırının tespit edilememesine ve algılanamamasına neden olur.



Şekil 2.4. Ağ Altyapısı Tespit Sistem Yapısı

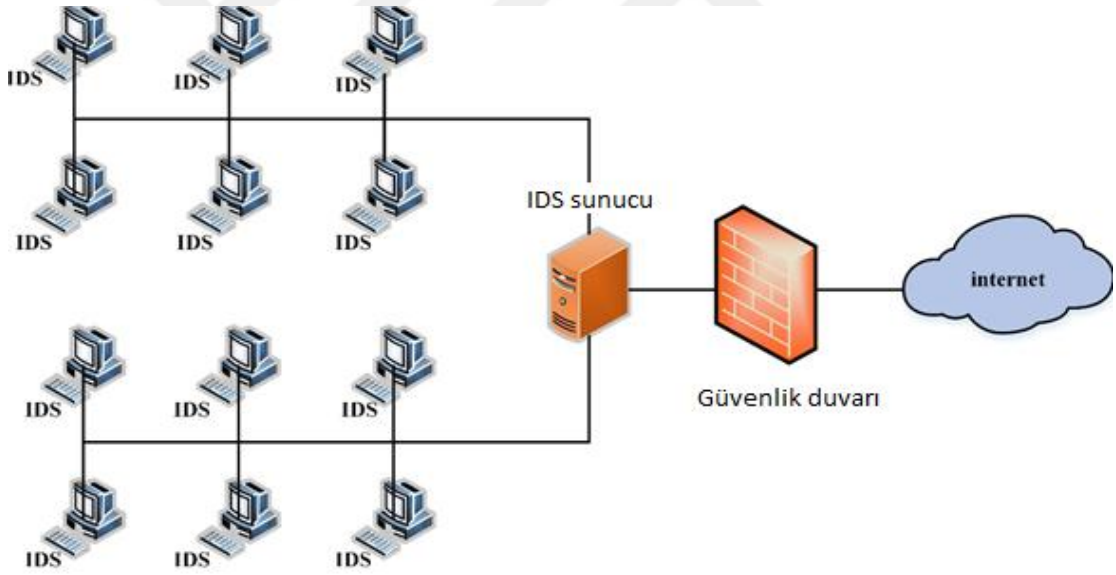
Ağ tabanlı saldırı tespit sistemleri gözlemci ve etken olmak üzere iki bölümden oluşmaktadır. Bir gözlemci, şüpheli paketleri bulmak için parçaları veya ağ trafiğini kontrol etmektedir. Etken bir yazılımdır ve genellikle gerekli bilgisayarların her biri için bağımsız bir şekilde uygulanır ve bilgiyi gözlemciye geri bildirim olarak gönderme rolünden sorumludur. Yönetici konsoluna (süpervizöre) güvenli bir şekilde (doğrulama ve şifreleme ile) bağlanan ve raporu alan başka bir bölüm de olabilir, ayrıca sistemin yapılandırması hakkında bilgi alışverişinde bulunmaktadır. Bu tip saldırı tespit sistemleri, ağ üzerinden akış ve aktarım yapan ağ bilgi paketlerinden veri ayıklamak için paket dinleme gibi teknikler kullanmaktadır. Ağ tabanlı saldırı tespit sistemlerinin birtakım dezavantajları bulunmaktadır. Örneğin, ağ paketlerinin birikimi ve analizi nedeniyle, yüksek hızlı ağlarda sorun yaşanabilir ve verimli çalışmayabilir. Başka bir örnek ise, ağ trafiği şifrelenirse kimlik sistemi bu saldırıyı anlayamayabilir.

Güvenlik duvarı genellikle intranet veya diğer iç ve internet ağları arasında çit olarak kullanılmaktadır. Güvenlik duvarının işlevi, içeriden dışarıya veya dışarıdan ağa

hangi trafiğin gireceğine ve hangisinin engelleneceğine karar vermektir (Yunfeng vd., 2012). Bir ağın birkaç güvenlik duvarı olabilir, ancak yine de tüm kötü amaçlı saldırıları durdurmada etkili olmayabilir. Örneğin, her virüsün ona ait bir imzası bulunmaktadır. Güncellenen güvenlik duvarları ve anti-virüsler, virüslerin imzalarını biliyor olacaklarından dolayı onları engelleyebilir. Ancak, güvenlik duvarı yeni bir virüsü tespit edemezse virüs duvardan geçebilir. Örneğin, daha yeni piyasaya sürülen ve imzası henüz tespit edilmemiş bir virüs ağa girebilir. Bu nedenle, güvenlik duvarı ve anti-virüs yazılımlarının sürekli güncellenmesi gerekmektedir.

2.3.2. Host Tabanlı Saldırı Tespit Sistemi (HIDS)

Saldırı tespit sistemlerinden birisi olan HIDS[15] (Host-Based IDS) host üzerindeki şüpheli durumları takip ettiği için ilgili hostun işletim sistemi üzerinde yazılım olarak yer almaktadır (Bukac vd., 2012).



Şekil 2.5. Host Tabanlı Saldırı Tespit Sistemi

Host tabanlı saldırı tespit sisteminde (HIDS) kolayca çözülemeyen iki temel problem bulunmaktadır. İlk sorun, sistemin yapısını değiştirmektir. İşletim sistemi ayarlarında ani bir değişiklik olursa, IDS' nin raporladığı günlük dosyalar sorunlu olur ve geçerliliğini yitirir. Bu IDS sisteminin bozulmasına ve aldığı kararların doğru

[15] Host-based Intrusion Detection System (HIDS)

olmamasına neden olmaktadır. HIDS ile ilgili bir başka sorunda, bu sisteme ihtiyaç duyan her sistem için kurulması ve çalıştırması gerekliliğidir. Bu konu, sistem yöneticileri için yönetim sorunları ve özel bakım sorunlarına neden olmaktadır.

Host tabanlı saldırı tespit sisteminin uygulaması ve yönetilmesi zaman alıcı olabilir. Bu sistemler sürekli görüntüleme ve güncelleme gerektirdiğinden dolayı genellikle doğru yönetim için fazla zaman gerekir. Bu sistemleri kurmak çoğu zaman zordur, ayarlamak için büyük çaba sarf edilmesi gerekmektedir. Ayrıca işletim sistemi ağda ne kadar fazla olursa, host tabanlı sistem yaklaşımı o kadar masraflı olur ve bu aracın yönetimi daha da zorlaşmaktadır. Ayrıca, bir ağda bulunan çok sayıda güvenlik aracıyla, uyarıların ve hataların sayısı artabilir.

2.3.3. Dağıtılmış Saldırı Tespit Sistemleri

Dağıtılmış saldırı tespit sistemleri[16] (Colom vd., 2018), birkaç merkezi yönetim sistemi ile birkaç HIDS, NIDS veya bunların kombinasyonundan oluşmaktadır. Bu şekilde, ağdaki herhangi bir dağıtılmış saldırı tespit sistemi raporlarını merkezi yönetime gönderir, merkezi sistem raporları kontrol etmek, gerekirse uyarmak ve gerekli önlemleri almakla yükümlüdür. Sistemin görevlerinden birisi ise saldırı veri tabanı kurallarının güncel tutulmasıdır. 1990 yılına kadar birçok dağıtılmış saldırı tespit sistemi, host tabanlı saldırı tespit sistemi olarak hareket etmiş, işletim sistemi veya uygulama seviyesinden bilgi toplayıp incelemiştir. NSM'nin[17] genişletilmesiyle, bu sınırlama çözülmüş ve dağıtılmış saldırı tespit sistemleri, ağ trafiğinden toplanan veriler temelinde çalışmalarına başlamıştır. İnternetin gelişmesi ve güvenlikle ilgili sorunların ortaya çıkmasıyla hem host tabanlı hem de ağ tabanlı modelleri toplayan bir sistem oluşturma gerekliliği ortaya çıkmıştır. Bu şekilde tanıtılan ilk sistem; Amerika Birleşik Devletleri Hava Kuvvetleri, Kaliforniya Üniversitesi, Haystack Laboratuvarları ve Lawrence Livermore Labs tarafından uygulanan DIDS'dir.

[16] Distributed Intrusion Detection System (DIDS)
[17] Network System Monitor

2.4. SALDIRI VE VERİLERE YETKİSİZ ERİŞİM

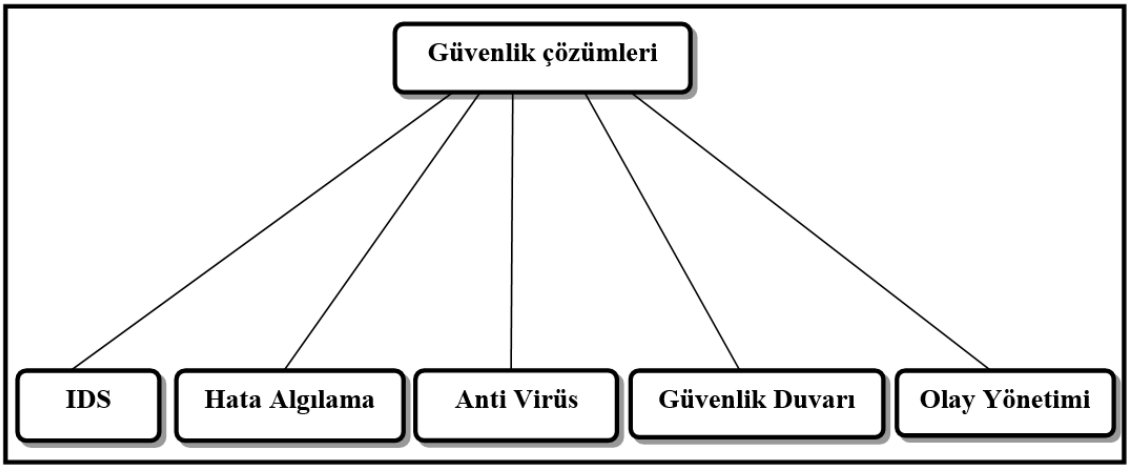
Ağ penetrasyonu için bilgisayar hackerleri öncelikle her host bilgisayarın güvenlik açıklarını incelemiş, bu amaçla özel cihazlar, yazılımlar kullanmış ve sonra sisteme giriş ve saldırıda başarılı olmuşlardır. Saldırganların iki tür yetkisiz erişimi bulunmaktadır.

- **Veri Erişimi**

Bu erişim türünde saldırgan, yetkisiz ağ erişim bileşenlerindeki verileri bulmaktadır. Saldırgan dahili bir kullanıcı veya dışardan biri olabilir. Ayrıcalıklı ve önemli veriler genellikle bazı ağ kullanıcıları tarafından erişilebilir ve diğerlerinin bu verilere erişim hakkı bulunmamaktadır. Aslında, diğerleri gizli bilgilere erişmek için yeterli ayrıcalığa sahip değildir, ancak ayrıcalık artırılarak gizli bilgilere erişim sağlanabilir.

- **Sisteme Erişim**

Bu saldırı türü daha tehlikeli ve daha kötüdür. Saldırı sırasında, saldırgan sistem kaynaklarına ve cihazlarına erişir. Bu erişim, sistemde çalışan programları içerebilir ve saldıran komutları yürütmek için sistem kaynaklarını kullanabilir. Ayrıca saldırgan, ağ ekipmanlarına ve depolama cihazlarına da erişebilir. Saldırganın saldırısını veya sistemde yüklü bir yazılımın zayıf yanlarını tespit etmek için, cihaz (alet) kullanımı bu tür saldırıların en göze çarpan örneklerinden biridir. Verilere yetkisiz erişimi önlemek için bazı güvenlik çözümleri Şekil 2.6'de gösterilmektedir.

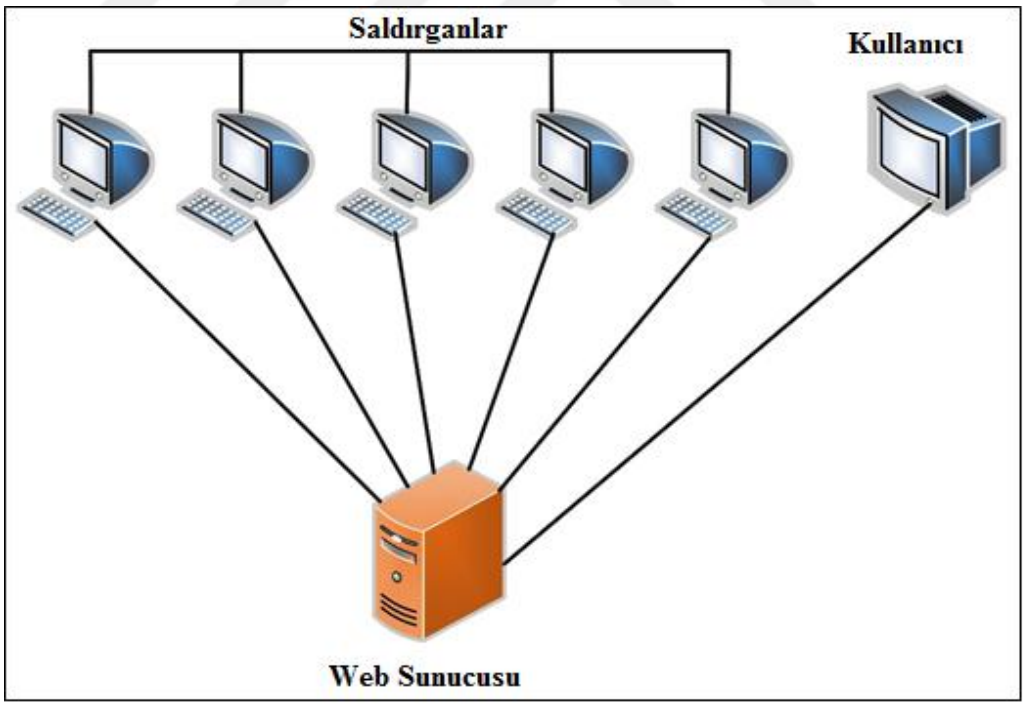


Şekil 2.6. Saldırıyı Önlenmesi İçin Güvenlik Çözümleri

Çeşitli işletim sistemleri ve uygulamalar bilgi şifreleme için olanaklar sunmaktadır. Olay geçmişlerini incelemek ve analiz etmek, yöneticiye sistemin performansı ve sorunları hakkında bilgi sağlayacaktır.

2.5. DDOS SALDIRISI

DDOS[18] saldırısında bir sunucu (server) (bilgisayar kurbanı veya hedef) çok sayıda saldırıya maruz kalmakta ve aşırı kaynak kullanımı (işlemci, veritabanı, bant genişliği, bellek vb.), kullanıcıların ulaşmak istedikleri servisin kesintiye uğramasına veya ulaşılamamasına neden olmaktadır (Chen vd., 2007). Bu tür saldırılarda sürekli olarak istenilen ve istenmeyen bilgisayarlardan (belirtilen IP adresi ile) sunucuya istek gönderilir ve işlem gücü nedeniyle servis kesintiye uğrar. DDOS saldırısı genellikle yoğun bir şekilde yapılır ve bu durumda, ağ tamamen kullanılamaz hale gelir ve zararlı bağlantılardan etkilenmiş olur.

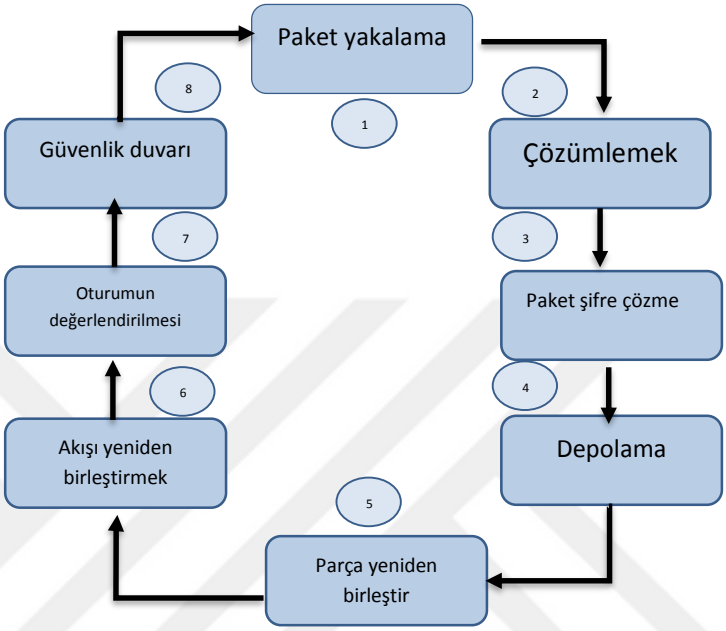


Şekil 2.7. DDOS'a Saldırı ve Ağ Kesintisi

[18] A Distributed Denial of Service (DDOS)

2.6. SALDIRI TESPİT SİSTEMİNDE ÇALIŞMA AŞAMALARI

Saldırı tespit sisteminin en önemli görevi, teşhis prosedürlerini doğru bir şekilde değerlendirip, şüpheli ve eksik paketleri tespit etmektir. Şekil 2.8'de saldırı tespit sisteminin akış şeması gösterilmektedir.



Şekil 2.8. Saldırı Tespit Sisteminin Akış Şeması

Saldırı tespit sistemindeki adımlar aşağıdaki gibidir:

- **Tamamlanmamış Paketlerin Toplanması**

Saldırı tespit sistemindeki bilgi akışının ilk adımı, tamamlanmamış paketlerin toplanması ve verilerin sistemin bir sonraki bileşene gönderilmesidir. Ağ kartı güvenilir olmayan bir durumda çalıştığı zaman, paketin belirli bir ağ kartına gönderilip gönderilmediğine bakılmaksızın tüm paketler kabul edilir (normal olarak, her ağ kartı sadece kendi adresleriyle aynı katman adresine sahip olan paketleri (ağ kartı adresi) kabul eder) ve diğer paketleri ayırır (uzaklaştırılır)). Bir saldırı tespit sisteminin eksik paketleri tanımlaması gerekiyorsa, iki ağ kartının kullanılması gerekmektedir. Saldırı tespit sistemi, analiz edilecek ve işlenecek eksik paketleri saklamaktadır. Tipik olarak paketler bellekte depolanır, böylece ilk işlem onlar üzerinde gerçekleştirilir, sonraki paketler belleği boşaltmak için bir dosyada veya veri yapısında saklanır.

- **Analiz**

Paketlerin toplanması ve kontrol edilmesi zaman alıcı bir süreç olduğundan dolayı saldırı tespit sisteminin mevcut tüm paketleri toplaması gerekmez. Birçok durumda paketin türüne, göndericinin ağ adresine vb. bilgilere bağlı olarak paketin analizi istenebilir. Örneğin, veri iletim hızının yüksek olduğu bir ağda, bir paket toplama işlemi verimliliği azaltacaktır. Bu nedenle, toplama için gerekli paketlerin sayısını sınırlandırmak gerekecektir. Ayrıca organizasyon, sadece belirli türdeki paketleri, örneğin, TCP[19] tabanlı trafiği toplamak isteyebilir. Çünkü araştırmalar birçok TCP tabanlı güvenlik saldırısının gerçekleştiğini göstermektedir.

- **Paket Şifre Çözümü**

Analiz işleminden sonra, ağın farklı katmanları ve paket yapısının kontrol edilmesi için paketler bir dekodere gönderilir. Örneğin, ikinci katmanda bir paketin ethernet veya döngü protokolü kullandığı belirlenir. Üçüncü katmanda söz konusu paketin IP protokol versiyonunun dört, beş (isteğe bağlı değerleri olmayan dördüncü versiyon) veya altı kullanmasının amaçlandığı tespit edilir. Ayrıca üçüncü katman protokolünün (IP gibi) kaynak ve hedef adresleri, dördüncü katman protokolünün kaynak ve hedef portu (TCP ve UDP[20] gibi) incelenir. Şifre (kod) çözme işlemi tutarsızlıklarını tespit etmek için paketler, ilgili standartlarla incelenir. Örneğin, TCP veri boyutu IP veri boyutuna eşit değilse, paket atılır. Genel olarak, düzgün bir şekilde çözülemeyen paketler atılmaktadır, çünkü saldırı tespit sistemi ile onların işlenmesi mümkün değildir (Kshirsagar vd., 2017, Mkuzangwe vd., 2017).

- **Depolama**

Paketleri çözdükten sonra, elde edilen veriler bir dosyada veya bir veri yapısında depolanır, böylece bir sonraki paketlerin belleği boş olur. Örneğin, verileri ikili dosyada saklamak, veri alımını daha da kolaylaştırmaktadır. Veriler mevcut bir dosyanın sonuna eklenebilir veya yeni bir dosyada saklanabilir. Saldırı tespit sisteminde verilerin dosyada depolanmasının bazı dezavantajları bulunmaktadır. Örneğin, bir veya daha fazla dosyada bulunan büyük miktardaki veri arasından, veriyi sınıflandırmak ve

[19] Transmission Control Protocol (TCP)
[20] User Datagram Protocol

kategorize etmek, belirli verileri bulmak veya veriler arasında bir korelasyon oluşturmak zordur ve zaman alıcıdır. Ayrıca depolama yönetimi de gerektirmektedir.

- **Parça restorasyonu**

Ağ trafiğinin büyük bir bölümünü anahtarlar, yönlendiriciler, güvenlik duvarları ve saldırı tespit sistemleri tarafından kontrol edilmesi gereken parçalanmış paketler oluşturmaktadır. Yanlış bölümleme, bir sisteme saldırmak veya tespiti önlemek için kullanılabilir. Yanlış bölümleme aşağıdaki yollarla yapılmaktadır:

1. Paketler birbirini engelleyecek şekilde parçalara ayrılabilir. (Başka bir ifadeyle, sonraki parça bir önceki parçanın bir kısmında yer almaktadır).
2. Dağınık parçalar, birçok durumda DOS saldırısı ve algılamayı önlemek için kullanılır.
3. Paketler uygun boyutta olmayabilir. Parçanın boyutu çok yüksekse (65535 bayttan fazla), alıcının sisteminde işlemcinin aşırı tüketimi gibi anormal durumlara neden olabilir. Bu nedenle, DOS saldırısı için çok büyük paketler kullanılır. Parça boyutu çok küçükse (64 bayttan az), paket başlığı birden çok parçaya sahip olabilir. Sonuç olarak, ağ aygıtları paketleri tamamen işleyemediğinden dolayı tamamı işlenmemiş paketler hedefe gönderilir. Ağ cihazlarında kullanılan mekanizma, gönderilmeden önce paketleri yeniden birleştirmek için gereklidir, bu yüzden cihazın çok fazla belleğe ihtiyacı olacaktır ve böylece bir DOS saldırısı olasılığı artacaktır.
4. Paketler, ikinci parça birinci parçaya tamamen yerleştirilecek şekilde kesilebilir, böylece onarım zor olacaktır.

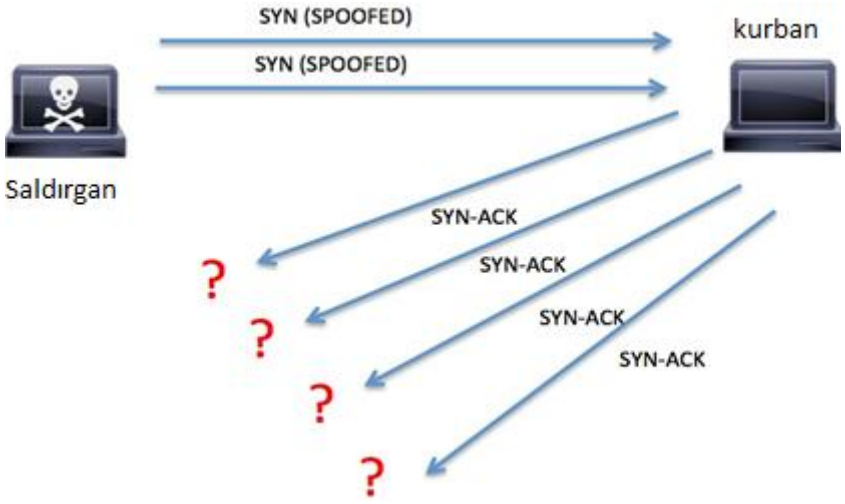
Birçok parçalanma atağı olmasına rağmen, bunların çoğu bilinmekte ve işletim sistemlerinde, ağ cihazlarında bir çözüm üretilmeye çalışılmaktadır. Örneğin, ağ cihazları işleyemeyen paketleri engellemektedir.

- **Akış Restorasyonu**

Akış restorasyonu, bir ağ trafiğindeki paketleri numaralarına göre sıralar, böylece alınan paketler gönderildikleri gibi sıralanmış olunur.

Bu gönderilen veriler bir akışın başlangıç ve bitişinin belirlenebilmesi için gereklidir. Örneğin, her bir TCP bağlantısı bir SYN paketi gönderilerek başlatılır ve bir RST veya FIN / ACK paketi gönderilerek sonlandırılır. Bir bağlantıyı başlatan, TCP paketi herhangi bir verisi olmadan SYN = 1 ve ACK = 0 bitleri ayarlanarak yeni bir bağlantı talep eder. Eğer karşı tarafın iletişim kurma eğilimi var ise; SYN ve ACK bayrakları 1 olarak kurulu paketler ile bu isteğini iletir. İzinsiz giriş tespit sistemi, sırasız (düzensiz) verileri analiz etme yeteneğine sahip değildir. Akış restorasyonunda, aşağıdakiler dikkate alınmalıdır:

1. TCP standardına göre FIN paketi sadece bağlantı kapatıldığında gönderilir. Bu paket zaten kapatılmış bir bağlantı için gönderilmişse, sistem bir RST paketi göndererek yanıt vermelidir. Bu protokolde RST paketleri tek yönlü bağlantıyı kesmek için kullanılır. Yani ilişkinin taraflarından biri mevcut bağlantıyı sonlandırmak zorundadır. Ayrıca, RST biti bir iletişim durumunun işareti olarak kullanılabilir. Henüz kurulmamış bir bağlantı için ACK paketinin alındığını bilmek çok önemlidir. Her iki durumda da iletişimin iki tarafının belirlenmesi sonraki analizler için önemlidir. Veri alıcısı bir TCP paketini, başlığı 1 olan FIN alanına sahip göndericiye gönderir. Diğer taraf talebi alır ve tek yönlü bir şekilde sonlandırmayı kabul eder (Tran vd., 2012, Zaman vd., 2009).



Şekil 2.9. İletişim Olmadan Bir ACK Paketi Alınması
(Https://www.cisco.com/c/en/us/about/security-center/guide-ddos-defense.html)

2. Akışın yeniden yapılandırılması, taraflar arasındaki iletişimin yanı sıra kayıp paketlere de neden olmaktadır. Bir dosyada veya veri yapısında, akışın yeniden yapılandırılmasından elde edilen veriler kaydedilir.

3. İzinsiz saldırı tespit sistemi, ağ trafiğinde oluşabilecek paket çakışmalarının (collision) farkında olmalıdır (bu konudaki çakışan paketlerin tümünü veya bir kısmını saklamak hususunda bir izleme politikası olmalıdır). TCP protokolünde paketlerin yeniden iletilmesi önemli bir konudur, ana paketler kaydedilmeli veya yeniden paketlenerek gönderilmelidir.

4. FIN, SYN ve ACK paketlerinde verinin mevcut olup olmadığını belirlemek ve sistemin bu sorunu gerçekten araştırdığını bilmek önemlidir. Sistem bu verileri kabul ederse ve veriler düzgün kontrol edilmezse saldırıyı tespit etmek mümkün olmayabilir.
5. UDP ve ICMP protokollerinin akışını, bu iki protokolün birbiriyle bağlantılı olmamasına (TCP protokolünden farklı olarak) rağmen yeniden yapılandırmak mümkündür. Bu iki protokol bazı saldırı durumlarında birlikte değerlendirilebilir. Alınan paketler bu minvalde sıralanabilir.

- **Oturum Değerlendirmesi**

Oturum değerlendirmesi, daha önce oluşturulan oturumlardan birine ait paket üyeliğini değerlendirir ve paketin bu oturuma ait olup olmadığı incelenir. Burada sorun şudur ki, saldırı tespit sistemi önceki paket oturumuna ait paketi kontrol etmek zorunda kalırsa, bir saldırgan çok sayıda paket göndererek bir DOS saldırısını yürütebilir. Paketler ayrıca önceden oluşturulmuş bir oturumu temsil edecek şekilde de gönderilmiş olabilir, oysaki oturum aslında oluşturulmamıştır. Bu problemleri çözmek için bir oturum değerlendirmesi kullanılır. Bu şekilde, üretilen oturumlardan veri içeren bir tablo üretilir ve alınan paketler bu tablo ile kontrol edilir ve eğer aynı oturum mevcut değilse, paket atılır. Bu yöntem, bir oturumda saldırının imzasını kontrol etmek için imza eşleştirmesi kullanarak algılama yönteminde kullanılmaktadır. Ayrıca, performans taramalarını tespit etmek için de yararlı olabilir.

- **Güvenlik Duvarının Görevleri**

Saldırı tespit sistemindeki bilgi döngüsünün bir kısmı paketlerin analizinde görevlidir, basit bir şekilde de olsa güvenlik duvarının görevini üstlenir. Trafiğin incelemesinden sonra daha karmaşık yapıya sahip bir güvenlik duvarı uygulanabilir. Veri analizinin temel amacı saldırı tespit sisteminin iş yükünü azaltmaktır. Trafiğin incelenmesi sonrasında güvenlik duvarının uygulanma amacı, saldırı tespit sistemini korumaktır. Saldırgan, saldırı tespit sisteminin düzgün çalışmasını engelleyebilir veya tamamen devre dışı bırakmak için bir saldırı düzenleyebilir. Güvenlik duvarının amacı, bu tür saldırıları ortadan kaldırmaktır.

2.7. KDD99 VERİ TABANI

KDD99 veri tabanı 494021 örnek ve 42 özellik içermektedir (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html). MIT Lincoln Lab' den IST grubu, DARPA ve AFRL / SNHS gözetiminde, saldırı tespit sistemlerinin değerlendirilmesi için ilk standart verileri toplamışlardır. Bu bilgiler DARPA saldırı tespit sistemini test etmek için bir simülasyonda birkaç hafta kullanılmıştır. Bu veri seti, (1998-1998) veri koleksiyonuna göre sınıflandırılmıştır. KDD veri toplaması üçüncü uluslararası bilgi ve veri arama yarışmasında kullanılmıştır. Bu veri tabanı standart bağlantı kayıtlarını içeren ve askeri bir ağda simüle edilmiş saldırılardan oluşan bir koleksiyondur (seridir).

Tablo 2.1. KDD Veri Tabanının Türüne Göre Özellikleri

Sıra	Özellik	Tür	Etiket	Sıra	Özellik	Tür	Etiket
1	duration	Numeric	B	23	count	Numeric	T
2	protocol_type	Symbolic	B	24	srv_count	Symbolic	T
3	service	Symbolic	B	25	serror_rate	Numeric	T
4	flag	Symbolic	B	26	srv_serror_rate	Numeric	T
5	src_bytes	Numeric	B	27	rerror_rate	Numeric	T
6	dst_bytes	Numeric	B	28	srv_rerror_rate	Numeric	T
7	land	Nominal	B	29	same_srv_rate	Numeric	T

8	wrong_fragment	Numeric	B	30	diff_srv_rate	Numeric	T
9	urgent	Numeric	B	31	srv_diff_host_rate	Numeric	T
10	hot	Numeric	C	32	dst_host_count	Numeric	H
11	num_failed_logins	Numeric	C	33	dst_host_srv_count	Numeric	H
12	logged_in	Nominal	C	34	dst_host_same_srv_rate	Numeric	H
13	lnum_compromised	Numeric	C	35	dst_host_diff_srv_rate	Numeric	H
14	lroot_shell	Numeric	C	36	dst_host_same_src_port_rate	Numeric	H
15	lsu_attempted	Numeric	C	37	dst_host_srv_diff_host_rate	Numeric	H
16	lnum_root	Numeric	C	38	dst_host_serror_rate	Numeric	H
17	lnum_file_creations	Numeric	C	39	dst_host_srv_serror_rate	Numeric	H
18	lnum_shells	Numeric	C	40	dst_host_rerror_rate	Numeric	H
19	lnum_access_files	Numeric	C	41	dst_host_srv_rerror_rate	Numeric	H
20	lnum_outbound_cmds	Numeric	C	42	class	-	-
21	is_host_login	Nominal	C	B=Basic, C=Content, T=Traffic, H=Host			
22	is_guest_login	Nominal	C				

Tablo (2.2), KDD veri tabanının aralığını göstermektedir. Her bir örneğin değeri, saldırı türünü belirtmektedir. Bu nedenle her bir özelliğin kapsamı, saldırı türünün belirlemesinde çok etkilidir.

Tablo 2.2. KDD Veri Tabanının Etki Alan Özellikleri

Sıra	Etki alanı	Sıra	Etki alanı	Sıra	Etki alanı
1	[0.0, 58329.0]	15	{0, 1, 2}	29	[0.0, 1.0]
2	{ICMP, TCP, UDP}	16	[0.0, 993.0]	30	[0.0, 1.0]
3	{HTTP, …, Z39_50}	17	[0.0, 28.0]	31	[0.0, 255.0]
4	{OTH, … , SH}	18	{0, 1, 2}	32	[0.0, 255.0]
5	[0.0, 6.9337564E8]	19	{0, 1, 2, 3, 4, 6, 8}	33	[0.0, 1.0]
6	[0.0, 5155468.0]	20	{0}	34	[0.0, 1.0]
7	{0, 1}	21	{0}	35	[0.0, 1.0]

8	{0, 1, 3}	22	{0, 1}	36	[0.0, 1.0]
9	{0, 1, 2, 3}	23	[0.0, 511.0]	37	[0.0, 1.0]
10	[0.0, 30.0]	24	[0.0, 511.0]	38	[0.0, 1.0]
11	{0, 1, 2, 3, 4, 5}	25	[0.0, 1.0]	39	[0.0, 1.0]
12	{0, 1}	26	[0.0, 1.0]	40	[0.0, 1.0]
13	[0.0, 884.0]	27	[0.0, 1.0]	41	[0.0, 1.0]
14	{0, 1}	28	[0.0, 1.0]	42	[0.0, 1.0]

(Http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html)

Tablo (2.2) 'de, saldırının adı protokol türüne bağlı olarak gösterilmektedir. Her protokol için çeşitli saldırılar vardır, bu da her protokolün dezavantajları olduğunu göstermektedir. KDD veri tabanı Normal, DOS, U2R, R2L ve Probe sınıflarını içermektedir:

- **Normal sınıf**

Bu sınıfta, örnekler normaldir ve saldırganlar tarafından herhangi bir saldırı yapılmamıştır.

- **DOS sınıfı**

Bu tür saldırılarda, saldırganlar host veya ağ kaynaklarına müdahale etmektedir. Saldırgan sunucuya (servere) servis için çok sayıda istek gönderir, böylece sunucudan servis alınır. Sunucu sayısız saldırıya yanıt vermekte olduğu için gerçek kullanıcılara yanıt verememektedir. Kurban makinesi bir web sunucusu, e-posta sunucusu veya benzeri bir sunucu olabilir. Bir saldırganın ağ üzerinde bilgi akışı oluşturmaya çalışması, en yaygın ve en önemli DOS saldırı türüdür. Kullanıcılar belirli bir web sitesi URL'sini tarayıcılarına yazdıklarında, kullanıcı istekleri sunucuya gönderilmektedir. Sunucu, herhangi bir anda sınırlı sayıdaki isteklere yanıt verebilir. Bu nedenle, bir saldırganın birden fazla istek göndererek hizmet işlem hacmini artıracağı durumlarda sunucunun isteklere cevap vermesi mümkün olmamaktadır.

- **Probe Sınıfı**

Bu tür saldırılarda, saldırgan IP adreslerini tespit etmek için bir uygulama kullanır. Sistemlerdeki güvenlik açıklarına erişebilir ve sistemlerin güvenlik açıklarını keşfettikten sonra kurban sisteminden bilgi toplamaya başlayabilir.

- **U2R Sınıfı**

Bu tür saldırılarda, saldırgan sistemin gerçek kullanıcısı gibi davranmaktadır. Sistemin güvenlik açıklarını izinsiz bir şekilde kullanarak sistem kaynaklarına erişir.

- **R2L Sınıfı**

Bu tür saldırılarda, saldırgan ağa yerel bir makine kullanıcısı olarak uzak ağlar üzerinden erişir ve daha sonra makine açıklarını suiistimal ederek kullanır.

Tablo 2.3. Protokol Türüne Göre Saldırı Grupları

Protokol türü	Saldırı İsmi
UDP	Normal, teardrop (DOS), satan (Probe), nmap (Probe), rootkit (U2R)
TCP	Normal, 40eptune (DOS), guess_passwd (R2L), land (DOS), portsweep (Probe), buffer_overflow (U2R), phf (R2L), warezmaster (R2L), ipsweep (Probe), multihop (R2L), warezclient (R2L), perl (U2R), back (DOS), ftp_write (R2L), loadmodule (U2R), satan (Probe), spy (R2L), imap (R2L), rootkit (U2R)
ICMP	Normal, portsweep (Probe), ipsweep (Probe), smurf (DOS), satan (Probe), pod (DOS), nmap (Probe)

(Http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html)

Tablo (2.4) 'deki bilgilerde gösterildiği gibi örnek verilerinin farklı sınıflardaki dağılımı önemli ölçüde değişmektedir. Örneğin, DOS saldırı sınıfı yaklaşık 400.000 örnek içeriyor olsa da U2R sınıfında 52'den fazla örnek bulunmamaktadır.

Tablo 2.4. KDD'99 Verilerindeki Örneklerin Yüzdesi

Sınıf	Örnek sayısı	Yüzde
Normal	97287	19.96%
Probe	4107	0.83%
DOS	391458	79.24%
U2R	52	0.01%
R2L	1126	0.23%
Toplam	494021	100%

KDD veri tabanında görülen saldırılar beş ana kategoriye ayrılmaktadır. Bunlar: Normal, Probe, DOS, U2R ve R2L'dir. Her saldırıda özellik ve protokol türü değişmektedir.

2.8. SONUÇ

Bu bölümde, saldırı tespit sistemlerine yönelik önceki çalışmalar incelenmiştir. Araştırmacılar; karar ağacı, destek vektör makinesi, karınca kolonisi optimizasyon algoritması, genetik algoritma, bulanık mantık ve yapay sinir ağları gibi modelleri saldırı tespiti için kullanmıştır. Bilgisayar sistemlerine uygulanan en büyük saldırı DDOS saldırısıdır. Saldırıda, kullanıcı sistemine çok sayıda istek gönderilir. Çoğu saldırı, TCP protokolünü kullanan bilgisayar kullanıcılarına yöneliktir.

ÜÇÜNCÜ BÖLÜM

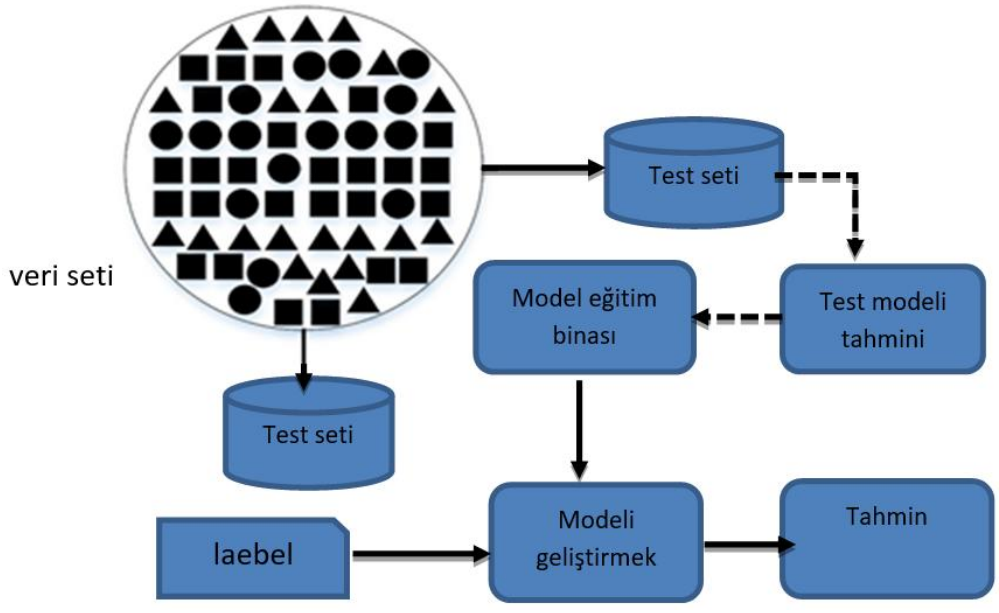
ÖNERİLEN MODEL

3.1. MAKİNE ÖĞRENİM ALGORİTMALARI

Makine öğrenim algoritmalarının iki aşaması vardır (Gastaldo vd., 2017). İlk adımı eğitimdir. Öğrenim gerçekleşir, eğitim örnekleri incelenir ve buna dayalı bir sınıflandırma modeli oluşturulur. İkinci aşama testtir ve eğitim aşamasında yapılan bir model yeni gözlemlenen örnekleri sınıflandırmak için kullanılır. Makine öğrenme algoritmaları dört ana türe ayrılmaktadır.

3.1.1. Denetimli Öğrenme

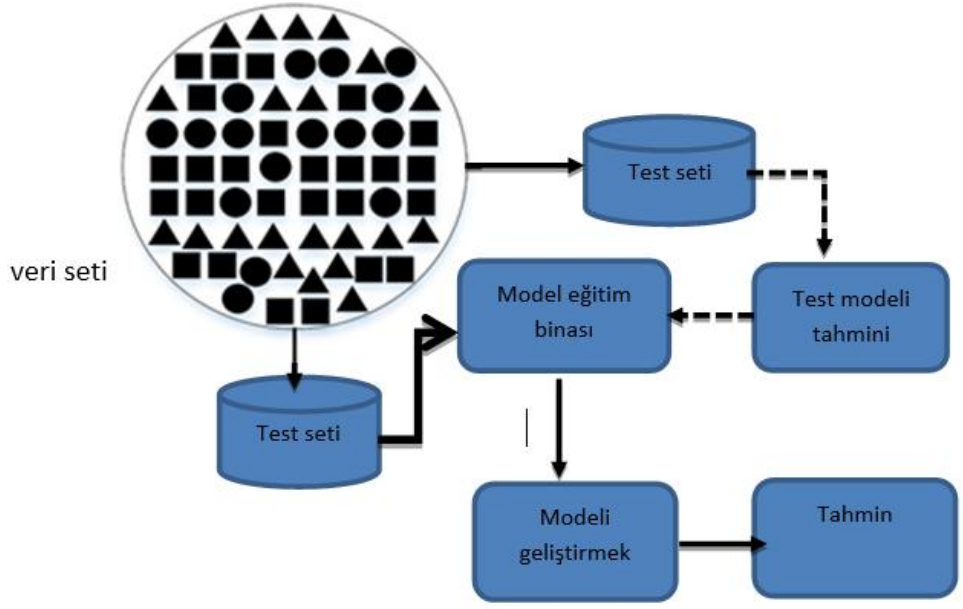
Denetimli öğrenme yönteminde, algoritmayı öğretmek için örnekler etiketlenerek kullanılır (Silva, 2017). Etiketli örnekler, verilerin sonuçları ve yanıtları ile birlikte mevcut olduğu anlamına gelmektedir. Örneğin, hasta veya sağlıklı sınıflandırması yapması için geliştirilen öğrenme algoritması eğitilmeden önce eğitim setindeki veriler etiketlenir. Girdilere karşın etiketlenmiş çıktıları olan eğitim seti algoritmaya gösterilerek eğitim yapılır. Eğitimden sonra algoritma, yeni bir numunenin hastayla ilgili veya sağlıklı olup olmadığını belirlemek için etiketlenmemiş yeni verileri sınıflandırabilir. Karmaşık konuları denetimli öğrenmek, performans açısından daha iyidir.



Şekil 3.1. İzlenmeyle Öğrenme Yapısı

3.1.2. Denetimsiz Öğrenme

Denetimsiz öğrenmede, hedef değişken olarak bilinmemektedir ve algoritmanın çıktısı belirsizdir (Mazzeo vd., 2017). Örneğin, müşterilerini kümelendirmek için bir mağaza, kişisel bilgiler ve müşteri satın alımları temelinde otomatik olarak eşit gruplara bölünür. Denetimsiz öğrenmenin avantajı; istatistiklere dayanan tahminlerin, etiketlenmemiş örnekler arasındaki kalıpları, korelasyonları ve ilişkileri tanımlamak için kullanılmasıdır. Kalıplar tanımlandığında, algoritma veri kümesindeki sınırları tanımlamak için istatistikleri kullanmaktadır. Aynı modeldeki örnekler bir grupta kümelenmektedir. Veri kümeleme işlemine devam ederek, algoritma veri kümesi modelini tanır ve yeni veriler için tahmin yapar. Makine öğrenmesinde kümeleme analizi denetimsiz öğrenme dallarından biri olarak gösterilebilir. Birbirine benzer örnekler gruplara ayrılır, bu gruplar küme olarak adlandırılır (Mazzeo vd., 2017). Bu nedenle kümeleme, örneklerin benzer olduğu bir koleksiyondur ve diğer kümelerdeki örneklerle uyumlu değildir. Aslında kümeleme, önceden tanımlanmış sınıfların bulunmadığı denetimsiz bir sınıftır. Başka bir deyişle, kümelenme bir tür denetimsiz öğrenme türüdür.



Şekil 3.2. Denetimsiz Öğrenmenin Yapısı

3.1.3. Takviyeli Öğrenme

Takviyeli öğrenmede, öğrenen sistem bir kontrolörün gözetimindedir, sistem bir karar verdiğinde bu kararın doğru olduğu durumlar için ödüllendirilir, yanlış olduğu durumlar için cezalandırılır. Makine mevcut pozisyona (mevcut değişkenler kümesi) ve yetkili işlemlere (örneğin ileriye veya geriye doğru hareket ederek) dayalı bir karar alır, ilk seferde bu karar tamamen rastgele olabilir ve ortaya çıkan her eylem veya davranış için sistem ona geri bildirim verir. Bu geri bildirimde, makine doğru kararı vermiş olup olmadığını öğrenir. Takviyeli (güçlendirilmiş) öğrenme interaktif bir yöntemdir ve denetimli öğrenme yönteminden farklıdır. Denetimli öğrenmede destek vektör makinesi veya karar ağacı gibi sınıf etiketleri eğitim aşamasında belirlenir fakat takviyeli (güçlendirilmiş) öğrenmede, sistem kendi öğrenmelidir. Bu yönüyle ne kümeleme ne de sınıflandırma yöntemi olarak değerlendirilmemelidir. (Sengupta vd., 2012).

Tüm denetimli öğrenme algoritmaları, bir eğitim aşamasına sahiptir ve modeli, eğitim prototipleri (eğitici şablonlarla) yardımıyla oluşturulur. Öğrenilene göre bir test aşaması yapılır ve test verileri kategorize edilir. Takviyeli öğrenme, karar vermek için önemli olan tüm konular ve oyunlar için (satranç ve yönlendirme gibi) kullanılır. Böylelikle, etkenler başladıkları noktadan nasıl hareket edebileceklerini, hedeflerine nasıl ulaşacaklarını ve her durumda en iyi uygulamanın hangisi olduğunu öğrenir. Bu metodoloji interaktif bir yöntemdir; öğrenim kazanımları deneme ve yanılma ile elde edilir. İnsanlarda olduğu gibi kişinin çevre ile etkileşimi neticesinde yaptığı deneme ve yanılmalar yine kişinin bilgi kazanımı olarak kendisine döner. Takviyeli öğrenme olasılık içerir. Algoritma, doğru durumlar için ödül alır. Algoritmanın başarısı ne kadar yüksek olur ise ödül de aynı nispette yüksek olur. Algoritma tarafından en yüksek ödülün elde edilmesi sürekli hedeflenir.

3.1.4. Yarı Denetimli Öğrenme

Yarı denetimli öğrenim aynı zamanda bir öğrenme türüdür ve öğrenmenin doğruluğunu artırmak için, hem sınıflandırılmış (etiketli) hem de sınıflandırılmamış (etiketlenmemiş) verileri eşzamanlı olarak kullanmaktadır (Ding vd., 2018). Yarı-denetleme öğrenme, iki aşamalı sınıflandırma ve kümeleme içermektedir. Kümeleme

algoritması, etiketli numunelerden ve etiketlenmemiş örneklerden oluşan eğitim örneklerini benzer kümelere ayırmaktadır. Eğitim örneklerinin kümelenmesinden sonra küme örnekleri, numuneleri eşlemek için (numuneleri izole etmek) etiketli örneklerden küme örneklerini tanımlama için kullanmaktadır. Test aşamasında, eğitim dışı ve etiketlenmemiş örnekler kullanılmaktadır.

3.2. MAKİNE ÖĞRENME ALGORİTMALARINDA DEĞERLENDİRME KRİTERLERİ

Birçok makine öğrenimi algoritmasının işlevi, girdi modellerinin benzerliği veya uyumsuzluğunu ölçmek için kullanılan ölçütlere bağlıdır. Mesafe fonksiyonları, çeşitli uygulamalarda kullanılan en önemli kriterdir. Örneğin öğrenme yöntemlerinden biri olan K-en yakın komşu algoritması ve K-Means kümeleme algoritması (MacQueen, 1967), denetimsiz öğrenme yöntemlerindendir. Her iki yöntemin performansı, seçilen mesafe kriterine önemli ölçüde bağlıdır. Yaygın olarak kullanılan en önemli mesafe ölçümlerinden biri Öklid mesafesidir. Tablo (3.1) 'de, çeşitli mesafe fonksiyonları örnekleri gösterilmektedir.

Tablo 3.1. Benzer Verileri Bulmak İçin Mesafe Fonksiyonlarının Hesaplaması

Mesafe fonksiyonu	**Formül**
Öklid Mesafesi (Euclidean distance)	$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$
Chebyshev Mesafesi (Chebyshev)	$\lim_{p\to\infty}\left(\sum_{i=1}^{n}\lvert x_i-y_i\rvert^p\right)^{1/p}\Rightarrow d(x,y)=\max_k\lvert x_{ik}-y_{ik}\rvert$
Minkowski Mesafesi (minkowski)	$d(x,y)=\sqrt[p]{\sum_{i=1}^{n}\lvert x_i-y_i\rvert^p},\ p>0\Rightarrow d(x,y)=\left(\sum_{i=1}^{n}\lvert x_i-y_i\rvert^p\right)^{1/p}$
Canberra Mesafesi (Canberra)	$d(x,y)=\sum_{i=1}^{n}\frac{\lvert x_i-y_i\rvert}{x_i+y_i},\ x_i and y_i are\_positive$

(www.nl.mathworks.com/help/stats/pdist.html) ve (Gueorguieva vd., 2017, Kaur vd., 2014)

Benzerlik kriteri mesafeye göre sınıflandırmaya dayanmaktadır. İki örnek arasındaki mesafe ne kadar az olursa, iki örnek daha çok benzer ve buna göre bir grupta toplanır. Bu nedenle, iki veri arasındaki mesafenin hesaplanması, sınıflandırmada çok önemlidir.

- **Öklid Mesafesi**

En popüler mesafe oranlarının biri Öklid mesafesidir. Öklid mesafesi, Pisagor teoremi ile elde edilen iki noktanın normal mesafedir. İki nokta arasındaki mesafeyi bulmak için Öklid mesafesi kullanılır (iki nokta, iki şehir gibi iki farklı nesne olabilir) ve en kısa mesafe bulunur. İki nokta, x ve y arasındaki mesafe onları birbirine bağlayan paralaks boyutudur. Örneğin, X = (x_1, x_2, ..., x_n) ve Y = (y_1, y_2, ..., y_n) ise, aralarındaki mesafe aşağıdaki gibi tanımlanır.

$$d(x,y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+...+(x_n-y_n)^2}=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2} \quad (3.1)$$

Örneğin, A (0,3,4,5) ve B (7,6,3,-1) noktaları arasındaki Öklid mesafesi aşağıdaki gibi hesaplanır.

$$d_{BA}=\sqrt{(0-7)^2+(3-6)^2+(4-3)^2+(5+1)^2}=9.74 \quad (3.2)$$

- **Chebyshev Mesafesi**

Chebyshev'in mesafesi, rastgele örneklerde veya olasılık dağılımında değerlerin ortalamaya yakın olmasını sağlar. Örnekler orijinal örneğe yakın ve örnekler arasında benzerlik yüksek olmalıdır. Örneğin, A (0,3,4,5) ve B (7,6,3,-1) noktaları arasındaki Chebyshev mesafesi aşağıdaki gibi hesaplanır.

$$d_{BA}=\max\{|0-7|,|3-6|,|4-3|,|5+1|\}=\max\{7,3,1,6\}=7 \quad (3.3)$$

- **Minkowski Mesafesi**

$$d(x,y)=\sqrt[p]{\sum_{i=1}^{n}|x_i-y_i|^p},\ p>0 \Rightarrow d(x,y)=\left(\sum_{i=1}^{n}|x_i-y_i|^p\right)^{1/p} \quad (3.4)$$

Minkowski mesafesinde p>=1 verilebilir. Eğer p=1 verilir ise formül Manhattan mesafesi gibi, p=2 verilir ise öklit mesafesi gibi davranacaktır. Bu mesafe çoğu

durumda, Öklid mesafesi ile aynı sonuçlara sahiptir. Örneğin, A (0,3,4,5) ve B (7,6,3,-1) noktaları ve p = 3 ise arasındaki mesafe aşağıdaki gibi hesaplanır.

$$d_{BA} = (|0-7|^3 + |3-6|^3 + |4-3|^3 + |5+1|^3)^{1/3} = \sqrt[3]{343+27+1+216} = 8.37 \quad (3.5)$$

- **Canberra Mesafesi**

Canberra; 1967 yılında Lance, Williams ve Adkins tarafından tanıtılmıştır. Bu mesafe genellikle dağınık veriler için kullanılır ve verilerin doğruluğuna duyarlıdır. Canbera mesafesinde x ve y örnekleri gerçektir, yani x = (x_1, x_2, ..., x_n) ve y = (y_1, y_2, ... y_n) ve bu mesafenin amacı en az fark olan örnekler arasındaki mesafeyi bulmaktır. Örneğin, A (0,3,4,5) ve B (7,6,3,-1) noktaları arasındaki Canberra mesafesi aşağıdaki gibi hesaplanır.

$$d_{BA} = \frac{|0-7|}{0+7} + \frac{|3-6|}{3+6} + \frac{|4-3|}{4+3} + \frac{|5+1|}{5+1} = 1 + \frac{1}{3} + \frac{1}{7} + 1 = 2.47 \quad (3.6)$$

Denetimli algoritmalarda, örneklerin benzerlik derecesi için bir dizi prototip seçilmiştir. Sınıflandırma, örneklerinin değerlendirilmesine göre yapılmaktadır ve bu adımdan sonra, hangi kategoriye ait olacağını belirlemek için her bir kategorinin yeni örnekler ile arasındaki mesafesi hesaplanır, bu işlem kategoriler değişmeyene kadar tekrarlanır.

3.3. ÖZELLİK SEÇİMİ

Özelliklerin seçimi, sınıflandırma algoritmalarının uygulanması için yer ve zaman yükünü azaltmaktadır (Dhyaram vd., 2018). Uygun bir sınıflandırma modeli oluşturmak için uygun eğitim verisine ihtiyaç vardır. Sınıflandırma kalitesi, eğitim setindeki veri sayısına ve özelliklerine bağlıdır. Ön işleme yöntemlerinden biri olarak özelliklerin seçimi, katmanlı bir model oluşturmak için eğitim verilerinin kalitesini arttırabilir (Dhyaram vd., 2018). Özellik seçiminin birçok avantajı bulunmaktadır. İlk olarak, verileri anlamak kolaydır. İkinci olarak, model öğrenme süresini azaltır. Üçüncü olarak özellikleri seçerek, özelliklerin değerlerini ölçmeye ve depolamaya daha az ihtiyaç duyulur. Özellik seçimi, sınıflandırma için daha iyi bir doğrulama modeli oluşturmayı mümkün kılar ve öte yandan, büyük miktarda validasyon verisi azaltılarak, örneklerin özelliklerinin toplanması-değerlendirilmesi için maliyet ve zamanı azaltmaktadır.

Sınıflandırmada özellik seçimi önemlidir, çünkü birçok işe yaramayan özellik bulunmaktadır ve bunların birçoğu da az bilgiye sahiptir. Bu özelliklerin kaldırılmaması bilgi açısından bir sorun yaratmaz, ancak istenen uygulama için hesaplama yükünü artırır. Özellik seçiminin amacı, denetimli veya denetimsiz yöntemlerle sınıflandırma performansını artırarak bir özellik kümesini aramaktır. Aslında özelliğin boyutları azaltılmalıdır, burada özelliklerin en iyi ve en küçük alt kümesini elde etmek amaçlanmaktadır. Genel olarak özellik alanının boyutunun azaltılması için özellik seçimi ve özellik çıkarması (ekstraksiyonu) olmak üzere iki yol bulunmaktadır. Özellik seçim yönteminde; kategorilemede yararlı olan özellikler, eksiksiz özellikler kümesinden seçilmektedir. Özellik çıkarma yönteminde, orijinal özellikler kullanılarak yeni özellikler oluşturulur.

Özelliklerin seçimi için filtreleme yöntemleri[21] ve kaplama yöntemleri[22] olmak üzere iki yöntem bulunmaktadır. Özellik filtreleme yöntemi, bir algoritmanın olası özellik alanı boyunca arama yaptıklarına işaret eder, özelliklerin herhangi bir alt türünü seçtikten ve bir filtreleme işlevini yürüttükten sonra bu özellikler sıralanır. Bu yaklaşım, özelliklerin en iyi alt kümesini seçerken sınıflandırma modelini dikkate almaz. Aslında filtreleme yöntemi, karakteristik vektörünün genel özelliklerine ve sınıflandırma modeliyle birleştirmeden alt grupların seçilmesine dayanır. Filtreleme yöntemleri yüksek hıza sahiptir, bu nedenle bu yöntem verilerin çok sayıda özellik içerdiği problemleri çözmek için çok uygundur. Ayrıca, bu yöntemler farklı sınıflandırma modellerine kolayca uygulanabilir.

Kaplama yöntemi, bir optimizasyon algoritması kullanmaktadır ve farklı alt kümeler oluşturmak için çeşitli özellikleri ekler veya azaltır. Sonrasında bu yerleşik alt kümeleri değerlendirmek için sıralama modeli kullanılır. Sınıflandırıcı model sıkça arandığı için filtreleme yöntemlerinden daha düşük bir hıza sahiptir ancak daha iyi bir algılama doğruluğu vardır.

Filtreleme yönteminin sınırlamaları için, kullanıcıların seçmek istedikleri özellik sayısını veya durdurma ölçütlerini belirtmeleri gereklidir. Ancak kaplama yönteminde; seçilen özelliklerin sayısı, sınıflandırma modelinin başardığı doğruluk hassasiyetine dayalıdır ve bir alt özellik kümesi seçilir, en yüksek doğruluk verilir. Kaplama metotları

[21] Filtering
[22] Wrapper

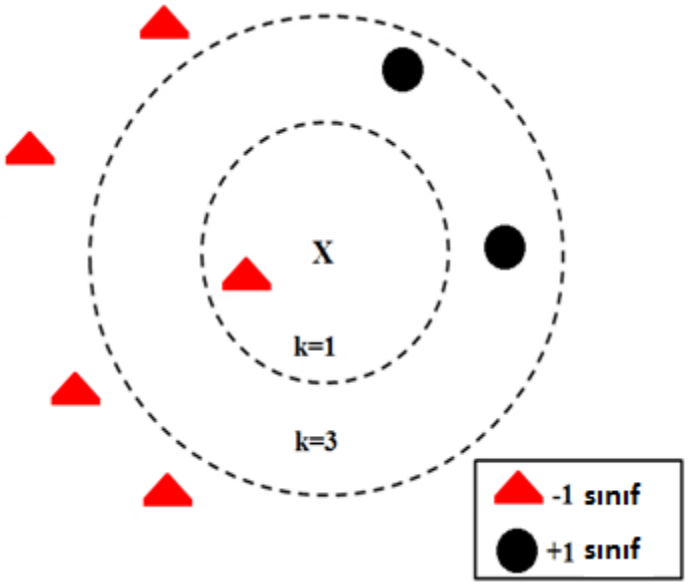
için uygun arama parametrelerinin belirlenmesi özellik sayısının yüksek olduğu durumlarda büyük önem taşımaktadır. Bir parametre doğru şekilde ayarlanmamışsa, büyük alt grupların oluşumu ve bunların doğruluğu nedeniyle parametrelerin hesaplanması zaman almaktadır.

3.4. K EN YAKIN KOMŞU ALGORİTMASI

KNN modeli (Martin, 1995), tahmin için kullanılmasına rağmen genellikle sınıflandırma için kullanılmaktadır. KNN modelinde sınıflandırılmamış bir örnek, eğitim setindeki en benzer örneklerle karşılaştırılarak kolayca bulunabilir. Bu nedenle, örnekler arasındaki mesafeyi belirlemek için bir kriter belirtmek gerekir. Eğer q ve p iki vektörü varsa, iki özellik q_i ve p_i arasındaki mesafeyi elde etmek için denklem (3.7) 'ye göre Öklid mesafesi kullanılır (Martin, 1995).

$$d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \Rightarrow d(q,p) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{3.7}$$

KNN yönteminde bir grup, en yakın test örneğine ulaşmak için komut setinden K örneğini içermelidir ve yeni bir test örneği durumunda ilgili kategorinin veya etiketin üstünlüğüne dayanmalıdır. Diğer bir deyişle, seçilmiş komşuda en yüksek sayıda örneği olan kategori seçilmelidir. Sonuç olarak, sınıflandırılmak istenen yeni bireyin daha önce sınıflandırılan K adet komşusu arasındaki Öklid mesafesine bakılır. Bunlardan en yakın olanlar hangi sınıfa dahil ise yeni bireyi de etiketleyerek o sınıfa dahil eder.



Şekil 3.3. İki Örnek Verilere Dayanarak KNN Algoritmasının Şeması

Tablo (3.2), 10 eğitim örneği ve bir örnek testi içermektedir. Amaç, KNN algoritmasını kullanarak Öklid (Euclidean) mesafesine dayalı sınıf örneğini test etmektir. İlk adımda, örnekler arasındaki mesafe hesaplanmalı ve sıralama mesafeye göre gerçekleştirilmelidir. Daha sonra, test örneği K sayısına bağlı olarak, M veya F sınıfı atanmaktadır. Eğer k = 3 ise, bu durumda [F = 1], [M = 2] ve [F = 3] olacaktır. Bu nedenle, test örnek sınıfı F'ye eşittir çünkü M, F'den daha büyüktür.

Tablo 3.2. Eğitim ve Test Örneklerini Görüntülemesi

Veriler	Sıra	x	y	Sınıf	Mesafe	Sıralama
Eğitim verileri	1	28	32	M	SQRT(28-28)^2+(34-32)^2=2	2
	2	33	35	M	SQRT(28-33)^2+(34-35)^2=5.09	9
	3	27	33	F	SQRT(28-27)^2+(34-33)^2=1.41	1
	4	31	36	F	SQRT(28-31)^2+(34-36)^2=3.60	6
	5	26	32	F	SQRT(28-26)^2+(34-32)^2=2.82	3
	6	27	38	M	SQRT(28-27)^2+(34-38)^2=4.12	7

	7	25	30	M	SQRT(28-25)^2+(34-30)^2=5	8
	8	26	29	F	SQRT(28-26)^2+(34-29)^2=5.38	10
	9	28	31	F	SQRT(28-28)^2+(34-31)^2=3	5
	10	27	36	M	SQRT(28-27)^2+(34-36)^2=2.23	4
Test verisi	**11**	**28**	**34**	**F**	-	-

Tablo (3.3), 16 eğitim örneği ve hasta verilerinden oluşan bir test örneğini içermektedir. Amaç, test örneği için KNN algoritmasını kullanarak Öklid mesafesine dayalı bir sınıfı (kötü veya iyi) belirlemektir. İlk adımda, örnekler arasındaki mesafe hesaplanmalı ve sıralama, mesafeye göre gerçekleştirilmelidir. Daha sonra K sayıya dayanarak, test örneği Kötü (bad) veya İyi (good) sınıfa atanmalıdır. Eğer k = 3 ise, bu durumda [Kötü =1], [Kötü =2] ve [Kötü=3], yani örnek test sınıfı kötü olmaktadır. Çünkü tüm komşuların değeri kötü' ye (Bad'e) eşittir.

Tablo 3.3. Eğitim ve Test Örneklerini Görüntülemesi

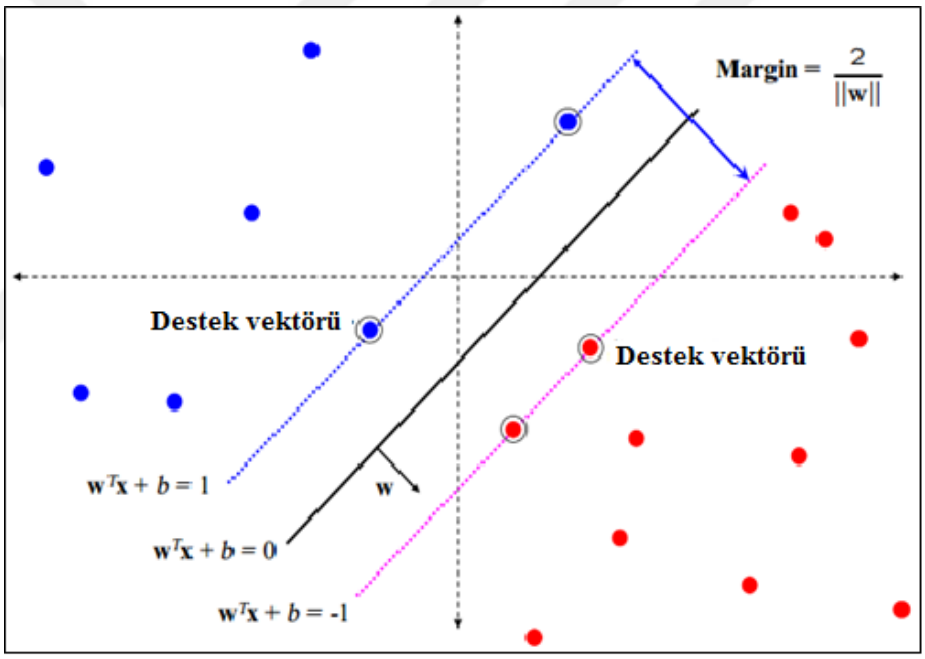
Veriler	Sıra	x	y	Sınıf	Mesafe	Sıralama
Eğitim verileri	1	15	52	İyi	SQRT(18-15)^2+(56-52)^2=5	8
	2	13	55	İyi	SQRT(18-13)^2+(56-55)^2=5.09	9
	3	19	53	Kötü	SQRT(18-19)^2+(56-53)^2=3.16	4
	4	20	56	Kötü	SQRT(18-20)^2+(56-56)^2=2	2
	5	12	60	İyi	SQRT(18-12)^2+(56-60)^2=7.21	11
	6	20	58	Kötü	SQRT(18-20)^2+(56-58)^2=2.82	3
	7	21	60	Kötü	SQRT(18-21)^2+(56-60)^2=5	8
	8	19	59	Kötü	SQRT(18-19)^2+(56-59)^2=3.16	4
	9	15	58	İyi	SQRT(18-15)^2+(56-58)^2=3.60	5
	10	16	52	Kötü	SQRT(18-16)^2+(56-52)^2=4.47	7
	11	12	60	İyi	SQRT(18-12)^2+(56-60)^2=7.21	11
	12	15	59	İyi	SQRT(18-15)^2+(56-59)^2=4.24	6
	13	16	56	Kötü	SQRT(18-16)^2+(56-56)^2=2	2
	14	19	57	Kötü	SQRT(18-19)^2+(56-57)^2=1.41	1
	15	13	55	İyi	SQRT(18-13)^2+(56-55)^2=5.09	9
	16	12	59	İyi	SQRT(18-12)^2+(56-59)^2=6.70	10
Test verisi	17	**18**	**56**	Kötü	-	-

3.5. DESTEK VEKTÖR MAKİNESİ

Destek vektör makinesi (M.Fıschettı 2016) bir grup bağlayıcıdır ve destek vektörleri tarafından belirlenmekte, veriler arasında en iyi sıralama ve ayırımı yapmaktadır. Destek vektörler, kategorilerin sınırlarını belirten n boyutlu verilerdeki nokta kümeleridir. Verilerin bölümlendirilmesi ve sınıflandırılması bunlara dayanmaktadır ve bunlardan birisini hareket ettirerek sınıflandırma çıktısı değiştirilebilir. Destek vektör makinesi, sadece destek vektörlerindeki verileri öğrenme ve modelin temelini oluşturmaktadır. Bu algoritma diğer veri noktalarına duyarlı değildir. Amaç, veriler arasında en iyi sınırı bulmaktır. Böylece tüm kategorilerde

(destek vektörler) mümkün olan en uzak mesafeye sahiptir. Ayırıcı hiper düzlemi (Hyper Plane) ayıran en yakın eğitim örneklerine, vektör yedeklemeleri denilmektedir.

Destek vektör makinesi önceden hazırlanmış kategorilerine göre verileri yeni bir alana taşır, böylece bu veriler doğrusal olarak kategorilere ayrılabilir. Daha sonra yedek hattı bularak, iki grup arasında en büyük mesafeyi oluşturan doğrusal denklemi bulmaya çalışır. Şekil (3.4)'te, veriler iki mavi ve kırmızı çizgi halinde görüntülenmektedir. Noktalı çizgiler, her bir kategori için iki çizgi daireyle işaretlenmiş destek vektörlerini temsil eder ve sürekli siyah çizgi, iki destek vektörü arasındaki sınır olmaktadır. Destek vektörlerinin her biri tanımlanmış bir denkleme sahiptir ve her bir kategorinin sınırını açıklamaktadır.



Şekil 3.4. Sınıfları Ayırmak İçin Destek Vektörlerinin Görüntülemesi

{x_1, x_2, ..., x_i, x_N} olarak bazı özellik vektörleri veya öğrenme kalıpları olduğunu varsayalım. Bunların her biri, y_i etiketli bir d boyutlu vektördür ve $y_i \in$ {-1, + 1} denkleminde amaç sınıflandırılmış iki sınıflı bir problemin en iyi ve uygun şekilde çözümünü bulmaktır. Bu iki sınıfın farklılaşma fonksiyonu f(x) ve bir hiper düzlem H (3-2) ile ayrıldığı varsayılmıştır.

$$H : w.x + b = 0 \tag{3.8}$$

$$f(x) = sign(w.x + b) \tag{3.9}$$

Optimal sınırda, sınıfın tüm +1 örnekleri sınırın bir tarafında olacak şekilde ve sınıfın tüm -1 örnekleri sınıfın diğer tarafında yer alacak şeklinde hesaplanır. Karar sınırı, her iki sınıfın en yakın eğitim seti örnekleri arasındaki karar sınırının dikey yönünde maksimum olacak şekildedir. X parametresi, karar sınırında bir noktadır. W, karar sınırına dik olan n boyutlu bir vektördür. B, bias değeridir ve w. x'in anlamı skaler çarpmadır. İki sınıf verilerinin ayrılabilir olduğunu ve H^+ hiper düzlem üzerindeki birinci dereceden sınır-özellik vektörünü ve H^-hiper düzlem üzerindeki ikinci derece sınır-özellik vektörünü varsayalım. H^+ ve H^- hiper düzlemleri bu şekilde tanımlanmaktadır.

$$H^+ : w.x + b = +1 \tag{3.10}$$

$$H^- : w.x + b = -1 \tag{3.11}$$

Devrik matris[23] olan w, w^T sembolü ile gösterilmektedir. Devrik veya transpoz bir matris, satırların ve sütunların orijinal matrisle değiştirildiği bir matristir. Yani, ilk satırın öğeleri yerine ilk sütunun öğeleri ve ilk sütunun öğeleri yerine ilk satırın öğeleri yerleştirilmiştir. Diğer satır ve sütun elemanlarının yerleri de aynı şekilde değiştirilmiştir. H^+ ve H^- hiper düzlemlerine yerleştirilen kalıplara destek vektörü denilmektedir. H^+ ve H^- iki hiper düzlem arasındaki bölge, margin veya sınır bölgesi olarak adlandırılmaktadır. H^+ ve H^- iki hiper düzlem arasındaki mesafe 2 / (|| w ||) değerine eşit olmaktadır.

Optimal karar sınırını hesaplamanın ilk adımı, iki sınıfın en yakın örneklerini bulmaktır. Bir sonraki adımda bu noktalar arasındaki mesafe, iki sınıfı tamamen ayıran sınırlara dik bir çizgide hesaplanır. Optimal karar sınırı, maksimum kenar boşluğu olan sınırdır. Optimal karar sınırı, denklem (3.12) 'e göre optimizasyon problemi çözülerek hesaplanmaktadır.

$$\max_{w,b} \min_{i=1,\ldots,L} \left[\frac{y_i (w.x_i + b)}{|w|} \right] \tag{3.12}$$

Verilerin iki sınıftan oluştuğu varsayılırsa, sınıflar toplamda i = 1, ..., L'dir ve x_i bir vektördür. Bu iki sınıf $y_i = \pm 1$ ile etiketlenmiştir. Denklem (3-12)'de y denklemin

[23] Transpose

çıktısı ve y_i, x_i'in deneysel örnek sınıf değeridir. X = (x_1, x_2, ..., x_n) vektörü bir giriş verisini temsil etmektedir ve x_i, i = 1,2, ..., n bir destekleme vektörüdür.

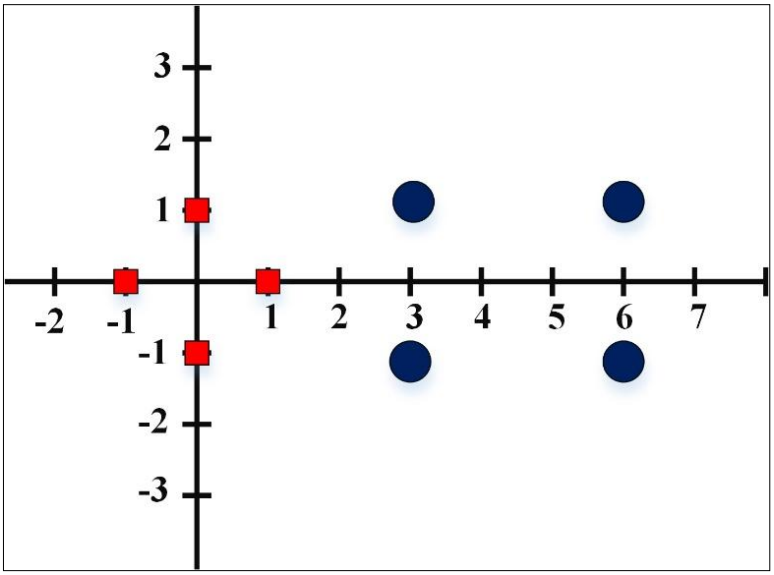
$$y = sign\left(\sum_{i=1}^{N} y_i \alpha_i K(X.X_i) + b\right) \qquad (3.13)$$

K (X, Xi) fonksiyonu, veri alanında farklı doğrusal olmayan yüzeylere sahip makineler oluşturmak için skaler çarpma üreten Kernel fonksiyonudur. Genel olarak, radyal çekirdek kernel[24] fonksiyonu daha iyi bir performans tahminine sahiptir. Radyal kernel fonksiyonu denklem (3.14)'e göre tanımlanmıştır. $|X-X_i||$ eğitim veri kümesi ve γ çekirdek genişliği, kullanıcı tanımlı bir parametredir.

$$K(X, X_i) = \exp(-\gamma \|X - X_i\|^2) \qquad for\ \gamma > 0 \qquad (3.14)$$

Sınıflandırma doğruluğunu arttırmak için denklemde, denklem 3.14'teki ϒ parametresi seçilmelidir. Değer doğru ayarlanmamışsa, iki sınıf arasındaki sınır doğru şekilde belirtilmez ve sınıflandırma hatası artar.

Örnek olarak; şekil (3.5)'de iki sınıfın noktaları gösterilmektedir. Destek vektör makinesini kullanarak, iki sınıf arasındaki sınırı belirtmek planlanmaktadır. Destek vektörü, iki sınıf arasındaki sınırı belirtmek için kullanılır. Destek vektörlerinin iki sınıfa yakın olması amaçlanmaktadır.



[24] Kernel

Şekil 3.5. Koordinat Alanındaki Noktaları Görüntüsü

Tablo (3.4)'de, şekil (3.5) noktalarının koordinatları iki sınıf için gösterilmiştir. Her nokta bir örneği temsil etmektedir. Birinci ve ikinci sınıflardaki örneklerin sayısı sırasıyla 4 ve 2 bulunmuştur.

Tablo 3.4. İki Sınıf Noktaların Koordinatlarının Görüntüleri

Nokta	**xy**	**1**	**2**	**3**	**4**
Sınıf1	x	3	3	6	6
	y	1	-1	1	-1
Sınıf2	x	1	0	0	-1
	y	0	1	-1	0

Şekil (3.5)'e dayanarak, üç destek vektörü bulunmuştur. Bu üç vektör, komşuları nedeniyle seçilmiştir ve amaç, destek vektör noktalarına dayanan sınıflar arasındaki sınırları belirlemektir.

$$\{S_1 = \begin{pmatrix}1\\0\end{pmatrix}, S_2 = \begin{pmatrix}3\\1\end{pmatrix}, S_3 = \begin{pmatrix}3\\-1\end{pmatrix}\}$$

Üç destek vektörünün tümünün bias (b) girişi, bire eşittir.

$$S_1 = \begin{pmatrix}1\\0\end{pmatrix} => \tilde{S}_1 = \begin{pmatrix}1\\0\\1\end{pmatrix} \quad S_2 = \begin{pmatrix}3\\1\end{pmatrix} => \tilde{S}_2 = \begin{pmatrix}3\\1\\1\end{pmatrix} \quad S_3 = \begin{pmatrix}3\\-1\end{pmatrix} => S_3 = \begin{pmatrix}3\\-1\\1\end{pmatrix}$$

Bir sonraki adımda, üç parametre α_1, α_2 ve α_3 bulunmalıdır. Belirtilen parametreleri bulmak için destek vektörü kullanılmalıdır.

$$\alpha_1\tilde{S}_1.\tilde{S}_1 + \alpha_2\tilde{S}_2.\tilde{S}_1 + \alpha_3\tilde{S}_3.\tilde{S}_1 = -1$$
$$\alpha_1\tilde{S}_1.\tilde{S}_2 + \alpha_2\tilde{S}_2.\tilde{S}_2 + \alpha_3\tilde{S}_3.\tilde{S}_2 = +1$$
$$\alpha_1\tilde{S}_1.\tilde{S}_3 + \alpha_2\tilde{S}_2.\tilde{S}_3 + \alpha_3\tilde{S}_3.\tilde{S}_3 = +1$$

Denklem (15 ila 17)'de, sayıların nasıl yerleştirileceği tanımlanmış ve değerleri hesaplanmıştır.

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}.\begin{pmatrix}1\\0\\1\end{pmatrix}+\alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}.\begin{pmatrix}1\\0\\1\end{pmatrix}+\alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}.\begin{pmatrix}1\\0\\1\end{pmatrix}=-1\rightarrow 2\alpha_1+4\alpha_2+4\alpha_3==-1 \quad (3.15)$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}.\begin{pmatrix}3\\1\\1\end{pmatrix}+\alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}.\begin{pmatrix}3\\1\\1\end{pmatrix}+\alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}.\begin{pmatrix}3\\1\\1\end{pmatrix}=+1\rightarrow 4\alpha_1+11\alpha_2+9\alpha_3=+1 \quad (3.16)$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}.\begin{pmatrix}3\\-1\\1\end{pmatrix}+\alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}.\begin{pmatrix}3\\-1\\1\end{pmatrix}+\alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}.\begin{pmatrix}3\\-1\\1\end{pmatrix}=1\rightarrow 4\alpha_1+9\alpha_2+11\alpha_3=+1 \quad (3.17)$$

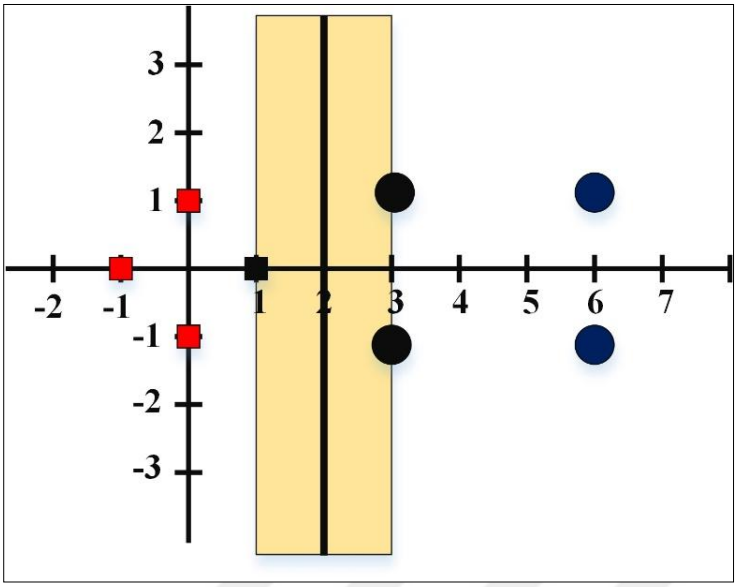
Denklemlere (19 ila 20) göre, üç parametre α_1, α_2 ve α_3 sırasıyla-3.5, 0.75 ve 0.75 bulunmuştur. Hiper düzlemin (sınır çizgisi) değeri denklem (3.18) 'ye göre elde edilmiştir.

$$\tilde{w}=\sum_i \alpha_i \tilde{S}_i \quad (3.18)$$

$$\tilde{w}=a_1 \begin{pmatrix}1\\0\\1\end{pmatrix}+\alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}+\alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix} \quad (3.19)$$

$$\tilde{w}=(-3.5).\begin{pmatrix}1\\0\\1\end{pmatrix}+(0.75).\begin{pmatrix}3\\1\\1\end{pmatrix}+(0.75).\begin{pmatrix}3\\-1\\1\end{pmatrix}=\begin{pmatrix}-3.5\\0\\-3.5\end{pmatrix}+\begin{pmatrix}2.25\\0.75\\0.75\end{pmatrix}+\begin{pmatrix}2.25\\-0.75\\0.75\end{pmatrix}=\begin{pmatrix}1\\0\\-2\end{pmatrix} \quad (3.20)$$

Hiper düzlem tarafından en uygun şekilde ayrılan vektörlerin toplanması, en az hataya sahip olmalı ve en yakın nokta olan hiper düzlem arasındaki mesafe en üst düzeye çıkarılmalıdır. Eşitliği (3.19) çözerek, optimal b ve w değerlerini bularak, optimum hiper düzlem bulunabilir. Bu nedenle hiper düzlemin ayrılması amacıyla, w = (10) ve b = -2 değerine eşitlenmiştir. Sınıflar için ayırma şeması şekil (3.6)'da gösterilmiştir.

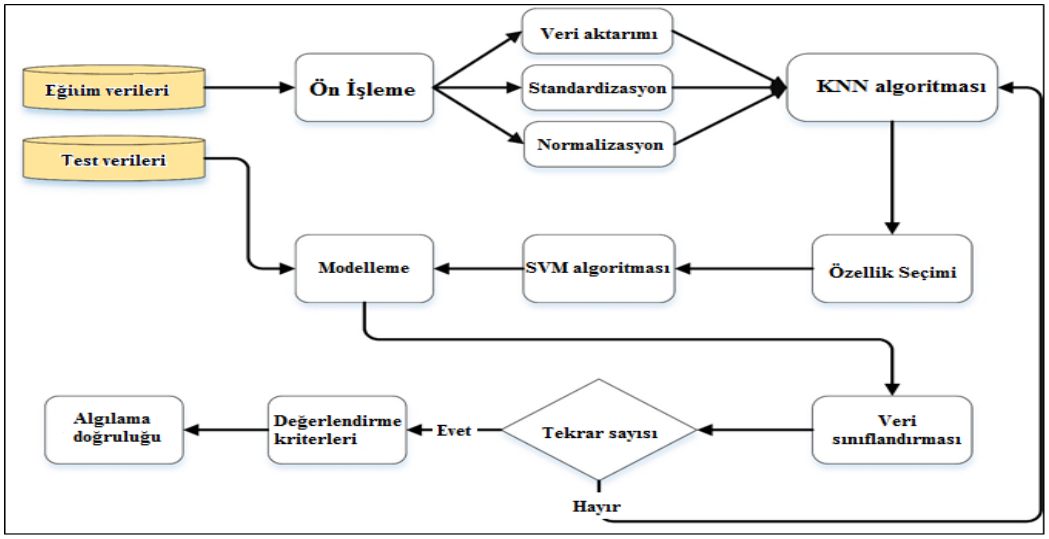


Şekil 3.6. Sınıf Ayırma Şeması ve Kenar Çizgisi Oluşturma

Daha sonra deney aşamasında, destek vektör ile giriş alanından elde edilen vektör arasındaki skaler çarpma çekirdeğine dayanarak, giriş vektör sınıfı belirlenmiştir. Destek vektör makinesi, verileri kategorize etmek ve bölmek için çok güçlü bir modeldir. Özellikle karar ağacı ve k en yakın komşu algoritması gibi diğer makine öğrenim yöntemleri ile birleştirildiğinde güçlü bir model oluşturulmuştur.

3.6. ÖNERİLEN MODEL

Bu tez çalışmasında önerilen model, KNN algoritması ve destek vektör makinesinin birleşimidir. KNN algoritması, özellik seçmek için, destek vektör makinesi ise sınıflandırma için kullanılacaktır. Şekil (3.7)'de önerilen model akış şeması gösterilmiştir. KNN algoritması, özellik seçmek için en yakın mesafe kriterini kullanmaktadır. Özellikler arasındaki mesafe hesaplanmakta ve daha küçük mesafeye sahip özellikler üstün özellikler olarak seçilmektedir. Önerilen modelde, ön işleme başlangıçta eğitim verileri üzerinde yapılmaktadır. Ön işlem aşamasında veri transferi, standardizasyon ve normalizasyon işlemleri gerçekleştirilmektedir. Daha sonra KNN algoritmasını kullanarak, özellik seçimi yapılmaktadır. Önemli özellikler seçildikten sonra, sınıflandırma destek vektör makinesi algoritması kullanılarak yapılacaktır.



Şekil 3.7. Önerilen Akış Şeması

3.7. ÖN İŞLEM AŞAMASI

Veri ön işlemi, sınıflandırmada çok önemli bir adımdır, böylece tespitin doğruluğunu arttırma ön işleme faktörüne bağlıdır. Ön işlemenin amacı, sınıflandırma için doğru, eksiksiz ve tutarlı örneklere sahip olmaktır. Eksik değerli örnekler için, diğer örneklerin sahip olduğu değerlere bakarak uygun değer ataması yapılması gerekmektedir.

3.7.1. Veri Aktarımı

Veri tabanında, bazı özelliklerin dize değerlerinin sayıya dönüştürülmesi gerekmektedir. Örneğin, sınıf özellikleri ve protokol türleri sayısal tipe dönüştürülmelidir. Tablo (3.5) 'te, bir dize tipinin bir sayıya tipine dönüştürülmesi gösterilmektedir.

Tablo 3.5. Bir Dize Türünü Bir Sayıya Dönüşmesi

0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.	Dize tipi
0,1,19,10,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,22.	Sayı tipi

3.7.2. Standardizasyon

Bu yöntemde, verilerin ortalama ve standart sapmasına göre, orantısal bir ölçek ile değeri azaltılmaktadır. Standardizasyonun amacı veri özellikleri için geçerli değerler oluşturmaktır.

$$X_{ij} = \frac{x_{ij} - m_j}{\sigma_j} \tag{3.21}$$

Denklem (3.21) 'de, mj değişkeni j'nin i istatistik dönemi boyunca ortalaması anlamına gelmektedir, σ ve i istatistiksel dönemi boyunca j değişkenin standart sapmasıdır.

A= {34,28,32,45,38,35,62,43,37,26,36,50,44,25,55,34,28,65,33,26}

Elemanlarından oluşan bir A kümesinin son elemanı için yukarıdaki standardizasyon yapılacak olunur ise;

Kümede bulunan verilerin ortalaması 38.8, standart sapması ise 11.4 olarak hesaplanacaktır.

1) İlk olarak değer ortalamadan çıkarılmalıdır: 26-38.8=-12.8

2) Daha sonra bulunan sonuç standart sapmaya bölünmelidir: 12.8 / 11.4 =-1.12

3.7.3. Normalizasyon

Normalizasyonun amacı örneklerin kalitesini arttırmaktır, böylece örnekler üzerindeki işlemlerin sınıflandırılması daha az hata ve daha iyi kalite ile gerçekleştirilir. Genellikle, her veri kümesinde gerçek değerlerden uzak veriler vardır. Bu veriler için doğru miktar bulunmalıdır. Özelliklerin normalleştirilmesi denklem (3.22)'e göre yapılmaktadır.

$$x_i = \frac{v_i - \min(v_i)}{\max(v_i) - \min(v_i)} \tag{3.22}$$

Denklem (3-22)'de v parametresi özelliklerin gerçek değeridir, min(v) ve maks(v), karakteristiğin minimum ve maksimum değerleridir. X, özelliklerin

normalleştirilmesinin sayısal değerini içermektedir. Tablo (3.6), 11 örnek üzerindeki normalizasyon işlemlerini göstermektedir.

Tablo 3.6. 11 Veri Örneğinin Normalizasyonu

No.	**x**	**y**	**Sınıf**
1	25	40000	N
2	35	60000	N
3	45	80000	N
4	20	20000	N
5	35	120000	N
6	52	18000	N
7	23	95000	Y
8	40	62000	Y
9	60	100000	Y
10	48	220000	Y
11	33	150000	Y
Normalizasyon			
No.	**x**	**y**	**Sınıf**
1	0.125	0.11	N
2	0.375	0.21	N
3	0.625	0.31	N
4	0	0.01	N
5	0.375	0.50	N
6	0.8	0.00	N
7	0.075	0.38	Y
8	0.5	0.22	Y
9	1	0.41	Y
10	0.7	1.00	Y
11	0.325	0.65	Y

3.8. ÖZELLİK SEÇİMİ

Büyük bir kümeden özelliklerin bir alt kümesini (veri tabanındaki veri sütunlarını) bulmak, birçok alanda var olan bir sorundur. Özelliklerin sayısının arttırılması, bir sistemin hesaplanma maliyetini arttırdığından dolayı en az sayıda özelliğe sahip sistemlerin tasarlanması ve uygulanması gerekli görülmektedir. Öte yandan, bu konuya odaklanmak çok önemlidir, böylece sisteme kabul edilebilir bir performans sağlamak için etkin bir özellik alt kümesi tanımlanmış olur. Özellik seçiminde düşük veri ve parazitli özellikler silinerek, veri kümesindeki öğrenme yöntemlerinin verimliliğini artıran bir özellik seçilmiş olacaktır. Özellik seçimi, makine öğrenme algoritmalarının performansı üzerinde olumlu bir etkiye sahip olmaktadır. Modelin yapım aşamasında ilgisiz ve gereksiz özelliklere sahip olması, düşük performansa ve yüksek hesaplamalara yol açabilmektedir.

KNN algoritmasında, örneklerin en yakın sınıflara verildiği varsayılmaktadır. Mesafeyi tespit etmek için en yaygın kriter Öklid mesafesini kullanmaktır. İki örnek arasındaki öklid mesafesi $x =< x_1, x_2, \ldots x_n >$, $y =< y1, y2, \ldots, yn >$ denklem (3.23)' e göre tanımlanmaktadır.

$$d(x,y) = \sqrt{(x_1 - y_1) + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2} \qquad (3.23)$$

KNN sınıflandırmasında, üzerinde işlem yapılan örnek birey üyeliği en yakın sınıfa atanır. Her bir nesnenin K' ya yakın komşular arasındaki en yaygın sınıfa atanması amaçlanmıştır (k pozitif bir tamsayıdır). Eğer k=1 ise, komşusunun en yakın sınıfına atanmaktadır. Örneğin, Tablo (3.7) 'de birbiriyle yakından ilişkili üç özelliği seçmek için KNN algoritmasının kullanılması düşünülmektedir. Özellikleri seçmek için, önce örnekler arasındaki mesafe hesaplanmalıdır. Daha sonra numunelerin ortalaması, her özellik için elde edilen toplam mesafeden hesaplanmaktadır. Sonunda, diğer özelliklere göre daha düşük toplam mesafeye sahip olan özellikler seçilmiş olacaktır.

Tablo 3.7. 11 Numuneye Göre 11 Özellik Seçimi

No.	**X1**	**X 2**	**X3**	**X4**	**X5**	**Sınıf**	**X1**	**X2**	**X3**	**X4**	**X5**
1	2	3	5	6	8	N	(1)^2	(2)^2	(3)^2	(3)^2	(4)^2
2	3	2	1	5	7	N	(0)^2	(3)^2	(1)^2	(2)^2	(3)^2
3	**4**	**3**	**6**	**5**	**5**	Y	(1)^2	(2)^2	(4)^2	(2)^2	(1)^2
4	**2**	**6**	**7**	**2**	**3**	Y	(1)^2	(1)^2	(5)^2	(1)^2	(1)^2
5	**3**	**7**	**4**	**6**	**2**	N	(0)^2	(2)^2	(2)^2	(3)^2	(2)^2
6	**5**	**5**	**2**	**4**	**6**	Y	(2)^2	(0)^2	(0)^2	(1)^2	(2)^2
7	**2**	**4**	**3**	**5**	**7**	N	(1)^2	(1)^2	(1)^2	(2)^2	(3)^2
8	**4**	**3**	**4**	**5**	**7**	N	(1)^2	(2)^2	(2)^2	(2)^2	(3)^2
9	**6**	**5**	**6**	**2**	**3**	Y	(3)^2	(0)^2	(4)^2	(1)^2	(1)^2
10	4	2	2	3	5	Y	(1)^2	(3)^2	(0)^2	(0)^2	(1)^2
11	3	2	3	7	8	N	(0)^2	(3)^2	(1)^2	(4)^2	(4)^2
Test verileri örneği							Ortalama mesafesi				
	3	5	2	3	4	?	1.72	4.09	7	4.81	6.45

3.9. DESTEK VEKTÖR MAKİNESİ İLE SINIFLANDIRMA

Destek vektör makinesi denetleme algoritmalardandır ve eğitim örneklerine ihtiyacı vardır. Öğrenme aşamasında örnekler bir hiper düzlemle ayrılır ve hiper düzlem, ayırma sınırı (her bir grubun hiper düzlemle en yakın noktaları arasındaki mesafe) en üst düzeye çıkarılacak şekilde hesaplanır, optimize edilmiş hiper düzlem bulunur daha sonra hiper düzlemde en küçük mesafeye sahip iki sınıfın örnekleri elde edilen eğitim verilerinin en küçük alt kümesi olarak tanımlanır, tanımlanan bu küme destek vektörü olarak adlandırılır. İşlem sonunda sadece deney aşamasında bulunan örnekler kullanılmaktadır. Deney aşamasında destek vektör ile giriş alanından elde edilen vektör arasındaki skaler çarpma çekirdeğine dayanarak giriş vektörü sınıfı belirlenir. Veri sınıflandırması Kernel fonksiyonu ile yapılır. Kernel fonksiyonu, veriyi giriş uzayından daha yüksek boyutlu bir alana eşler ve böylece o alandaki verilerin

doğrusal bir şekilde ayrılması mümkün olur. Daha sonra, o alandaki optimize edilmiş hiper düzlem tarafından aktarılan eğitim verileri ayrılır.

3.10. DEĞERLENDİRME KRİTERLERİ

Önerilen modelin değerlendirilmesi, denklem (3.24) ila (3.29)'a (Farnaaz vd., 2016) dayanmaktadır. Denklem (3.24-3.29)'da TP, TN, FP ve FN parametreleri sırasıyla doğru pozitif, doğru negatif, yanlış pozitif ve yanlış negatifi temsil etmektedir.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.24}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.25}$$

$$\text{F} - \text{Measure} = \frac{2*\text{Precision}}{\text{Precision}+\text{Recall}} \tag{3.26}$$

$$\text{Detection Rate (DR)} = \frac{TP}{TP+TN} \tag{3.27}$$

$$\text{False Alarm Rate (FAR)} = \frac{FP}{FP+TN} \tag{3.28}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \tag{3.29}$$

- **Hassasiyet Kriteri**

Bu kriter sınıflandırılan öğelerin, tüm örneklerinin doğru tespit oranını temsil etmektedir. Bu endeks, önerilen modelin yüzdesine göre sınıflandırılmış örneklerin sayısını belirlemede gösterdiği performansı ifade edebilir. Hassasiyet kriterleri TP ve FP parametrelerini içermektedir. Bu kriterde yanlış pozitif örneklerinin, doğru pozitif algılama doğruluğu üzerinde olumlu etkisi vardır. Bu kriter, doğru bir şekilde tanımlanmış negatif cevapların bir kısmını göstermektedir.

- **Hatırlama Kriteri**

Bu kriter, doğru tespit edilen örneklerin, veri setindeki örnek sayısına oranını ifade etmektedir. Hatırlama kriterleri TP ve FN parametrelerini içermektedir. Bu kriterde, yanlış pozitif numunelerinin algılama doğruluğu üzerinde etkisi bulunmaktadır. Bu kriter, doğru bir şekilde belirlenen pozitif yanıtların bir kısmını

oluşturmaktadır. Örneğin, gerçekten hastalık teşhisi konulmuş kişilerin yüzdesinin doğru tespit edilmesi oranıdır.

- **F-Ölçü Kriteri**

F- Ölçü kriteri, hassasiyet ve hatırlama kriterlerinin ortalama değerlendirmeleri için kullanılmaktadır. Sistemi değerlendirmek için F-Ölçü kriteri bir miktardır, hassasiyet ve hatırlama kriterlerinin sonucu olarak kullanılır. F- Ölçü, hassasiyet ve hatırlama kriterlerine göre değerlendirme yapmaktadır.

- **Algılama Doğruluğu (DR)**

Algılama doğruluğu oranı aşağıdaki denklemden elde edilmektedir:

Sınıflandırılmış birbiri ile ilişkili örneklerin sayısı / (Gerçek pozitif örneklerin sayısı + gerçek negatif örneklerin sayısı).

- **Yanlış Uyarı Oranı (FAR)**

Yanlış uyarı oranı aşağıdaki denklemden elde edilmektedir:

Yanlış pozitif örnek sayısı / (yanlış pozitif örnek sayısı + negatif gerçek örnek sayısı).

- **Doğruluk Kriteri**

Doğruluk kriteri tanı için ana kriterdir ve amacı tüm veri örneklerinin doğruluğunu değerlendirmektir. Bu kriter, tanının doğruluğu için temel kriterlerden biridir. Bu kriter, ölçülen değerin gerçek değere yakın olduğu anlamına gelir. Doğruluk oranının yüksek olması için hassasiyetin yüksek olması gerekir, ancak bu gerekli değildir.

Tablo (3.8)'de TP, FP, TN ve FN parametrelerini belirlemek için varsayımsal (Hipotetik) bir veri kümesi gösterilmektedir. Test1 ve Test2 için sıfır değeri hasta olmama ve bir değeri hasta olma anlamına gelmektedir.

Tablo 3.8. Doğruluk Kriteri Parametrelerinin Tanımlanması İçin Varsayımsal Veri Seti

Numara	Test1	Test2	Numara	Test1	Test2
1	1	0	**21**	0	1
2	1	0	**22**	1	1
3	0	0	**23**	0	0
4	0	1	**24**	0	0
5	1	1	**25**	1	1
6	1	1	**26**	0	0
7	1	1	**27**	0	0
8	0	0	**28**	0	0
9	1	0	**29**	0	0
10	1	1	**30**	1	1
11	1	1	**31**	0	0
12	1	1	**32**	0	0
13	0	0	**33**	0	0
14	0	0	**34**	0	0
15	1	1	**35**	1	1
16	1	1	**36**	0	0
17	1	1	**37**	1	1
18	1	0	**38**	1	0
19	0	0	**39**	0	0
20	0	1	**40**	0	0

Tablo (3.9)'da TP, FP, TN ve FN parametreleri için gerekli koşullar tanımlanmıştır. Tanımlanan koşula bağlı olarak TP, FP, TN ve FN parametrelerinin değerleri belirlenmektedir.

Tablo 3.9. Doğruluk Kriterinin Parametreleri İçin Belirleme Koşullarının Tanımlanması

```
if test1=1 then
do;
if test2=1 then result_c12="TP"
else if test2=0 then result_c12="FN"
end;
else if test1=0 then
do;
if test2=1 then result_c12="FP"
else if test2=0 then result_c12="TN"
end;
```

Tablo (3.10)'da varsayımsal veri seti için parametre tipi belirtilmektedir. Parametre tipi, koşullara göre komutlarla belirlenmektedir.

Tablo 3.10. Doğruluk Kriteri Parametrelerinin Tipini Tespit Etmek İçin Varsayımsal Veri Seti

Numara	Test1	Test2	Tip	Numara	Test1	Test2	Tip
1	1	0	FN	**21**	0	1	FP
2	1	0	FN	**22**	1	1	TP
3	0	0	TN	**23**	0	0	TN
4	0	1	FP	**24**	0	0	TN
5	1	1	TP	**25**	1	1	TP
6	1	1	TP	**26**	0	0	TN
7	1	1	TP	**27**	0	0	TN
8	0	0	TN	**28**	0	0	TN
9	1	0	FN	**29**	0	0	TN
10	1	1	TP	**30**	1	1	TP
11	1	1	TP	**31**	0	0	FN
12	1	1	TP	**32**	0	0	TN
13	0	0	TN	**33**	0	0	TN
14	0	0	TN	**34**	0	0	TN
15	1	1	TP	**35**	1	1	TP
16	1	1	TP	**36**	0	0	TN
17	1	1	TP	**37**	1	1	TP
18	1	0	FN	**38**	1	0	FN
19	0	0	TN	**39**	0	0	TN
20	0	1	FP	**40**	0	0	TN
TP=14; TN=17; FN=6; FP=3							
Hassasiyet $= \frac{\text{TP}}{\text{TP+FP}} = \frac{14}{14+3} = 0.8235 * 100 = 82.35$							
Hatırlama $= \frac{TP}{TP+FN} = \frac{14}{14+6} = 0.7 * 100 = 70$							
F − Ölçüsü $= \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * 82.35}{82.35 * 70} = 75.67$							
Detection Rate (DR) $= \frac{\text{TP}}{\text{TP} + \text{TN}} = \frac{14}{14 + 17} = 45.16$							
False Alarm Rate (FAR) $= \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{3}{3 + 17} = 15.00$							
Doğruluk $= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{14 + 17}{14 + 17 + 3 + 6} = 0.7750 * 100 = 77.5$							

DÖRDÜNCÜ BÖLÜM

DEĞERLENDİRME VE SONUÇLAR

4.1.DEĞERLENDİRME VE SONUÇLAR

Bu bölümde, KNN ve destek vektör makinesinin bir kombinasyonu olarak önerilen modelin değerlendirmesi ve sonuçları gösterilmektedir. En yakın komşu algoritmasında, özellik seçimi ve örnek sınıflandırması için destek vektör makinesi kullanılmıştır. Program, farklı tekrarların yanı sıra farklı değerler temelinde uygulanmakta ve test edilmektedir. Tablo (4.1), K' nın farklı değer ve100 kez tekrara göre önerilen modelin sonucunu göstermektedir. Tablo (4.1)'de görüleceği üzere k = 3 doğruluk değeri ile diğerlerinden daha büyüktür. K = 5 ve k = 7 için doğruluk değeri sırasıyla 91.89 ve 90.58 bulunmuştur. Ayrıca k = 3, k = 5 ve k = 7 için DR değeri sırasıyla 92.06, 91.23 ve 90.95'tir. K = 3 için FAR değeri 8.56'dır ve sırasıyla k = 5 ve k = 7 için 9.15 ve 9.48'dir. Tablo (4.1)'in sonuçlarına dayanarak k değerinin doğruluğu belirlemede etkili olduğu sonucuna ulaşılmaktadır. FAR değeri ile doğruluk değeri arasındaki ilişkiye dikkat edilmelidir. FAR değeri küçüldükçe doğruluk değeri artmaktadır.

Tablo 4.1. Önerilen Modelin 100 Kez Tekrarlara Dayanarak Karşılaştırılması

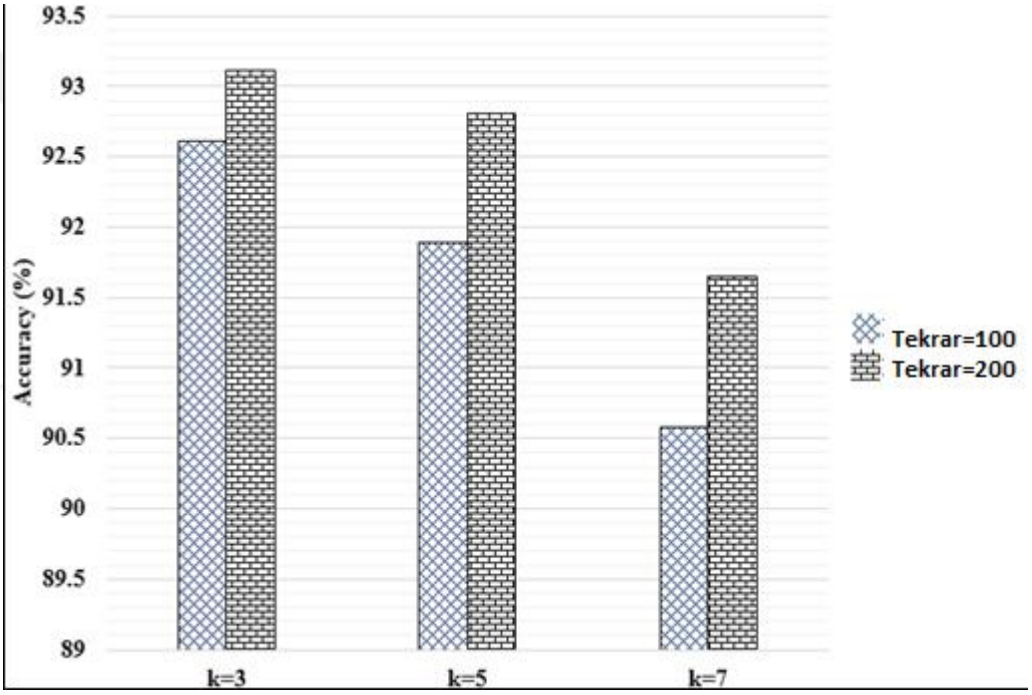
K Sayısı	Hassasiyet	Hatırlama	F-Ölçüsü	DR	FAR	Doğruluk
k=3	90.02	92.62	91.30	92.06	8.56	92.61
K=5	89.18	91.33	90.24	91.23	9.15	91.89
K=7	88.47	90.12	89.29	90.95	9.48	90.58

Tablo (4.2)'de, önerilen modelin K’nın farklı değer ve 200 tekrara göre sonucu gösterilmektedir. Tablo (4.2)'de görüleceği üzere k = 3 doğruluk değeri ile diğerlerinden daha büyüktür. K = 5 ve k = 7 için doğruluk değeri sırasıyla 92.81 ve 91.65'dir. Ayrıca k = 3, k = 5 ve k = 7 için DR değeri sırasıyla 92.78, 91.68 ve 91.05'tir. K = 3 için FAR değeri 8.03 bulunmuştur ve sırasıyla k = 5 ve k = 7 için 8.82.15 ve 9.12'dir. Geçmiş çalışmalarda K değeri için genellikle 3,5,7 kullanıldığı için bu çalışmada yine 3,5 ve 7 değerleri kullanılmıştır.

Tablo 4.2. Önerilen Modelin 200 Kez Tekrarlara Dayanarak Karşılaştırılması

K Sayısı	Hassasiyet	Hatırlama	F-Ölçüsü	DR	FAR	Doğruluk
K=3	91.08	91.95	91.51	92.78	8.03	93.11
K=5	90.15	91.72	90.93	91.68	8.82	92ç81
K=7	89.35	90.64	89.99	91.05	9ç12	91.65

Şekil (4.1)'de görüleceği üzere önerilen modelin sonuçları; 100 ve 200 tekrar ile doğruluk değeri diğerlerinden daha büyük bulunmuştur.



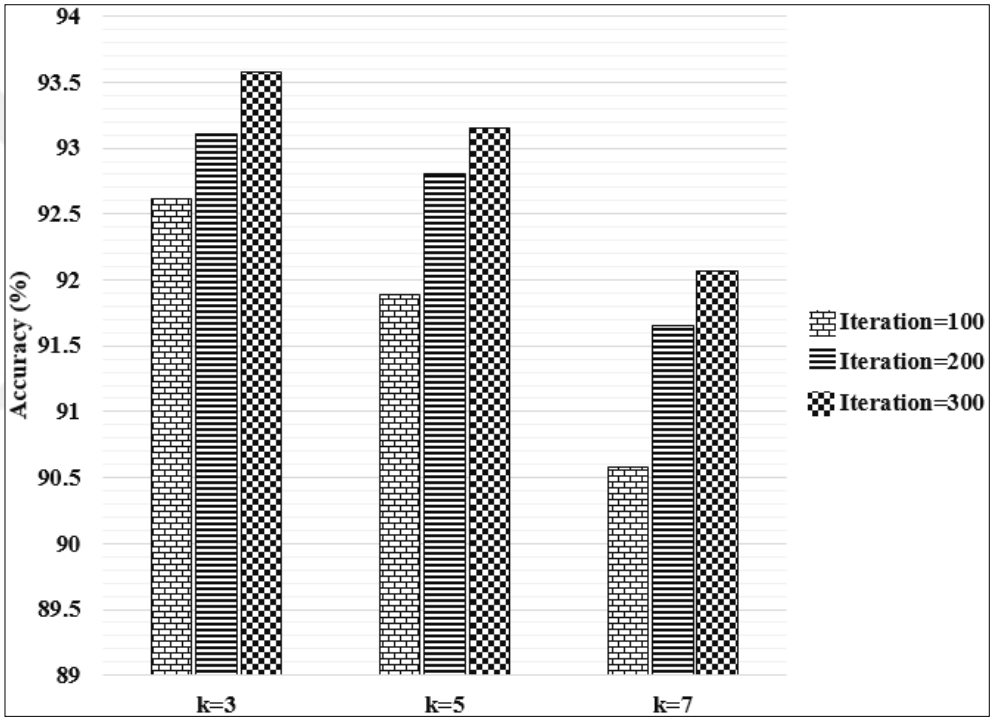
Şekil 4.1. 100 ve 200 Tekrar Karşılaştırılması

Tablo (4.3)'te, önerilen modelin sonuçları k'nın farklı değerlere ve 300 tekrarlamasına dayanılarak gösterilmiştir. Tablo (4.3)'te, görüleceği üzere doğruluk değeri k = 3'te diğerlerine kıyasla daha büyüktür. K = 5 ve k = 7 için doğruluk değeri sırasıyla 93.15 ve 92.07'dir. Ayrıca k = 3, k = 5 ve k = 7 için DR değeri sırasıyla 93.12, 92.25 ve 91.68'dir. K = 3 için FAR değeri 7.14 ve k = 5 ve k = 7 için sırasıyla 7.92 ve 8.16'dır.

Tablo 4.3: Önerilen Modelin Farklı K Değerlerine ve 300 Tekrarına Göre Uygulanması

K	Precision	Recall	F-Measure	DR	FAR	Accuracy
K=3	92.13	92.84	92.48	93.12	7.14	93.58
K=5	91.75	92.06	91.90	92.25	7.92	93.15
K=7	90.52	91.15	90.83	91.68	8.16	92.07

300 tekrarla değerlendirmenin sonucu, 100 ve 200 tekrarla karşılaştırıldığında daha yüksek doğruluk oranına sahip olduğu görülmektedir.



Şekil 4.2. 100 Tekrarın 200 ve 300 Tekrar Temel Alınarak Karşılaştırılması

4.2. ÖNERİLEN MODELİN ÖZELLİK SEÇİMİNE GÖRE DEĞERLENDİRİLMESİ

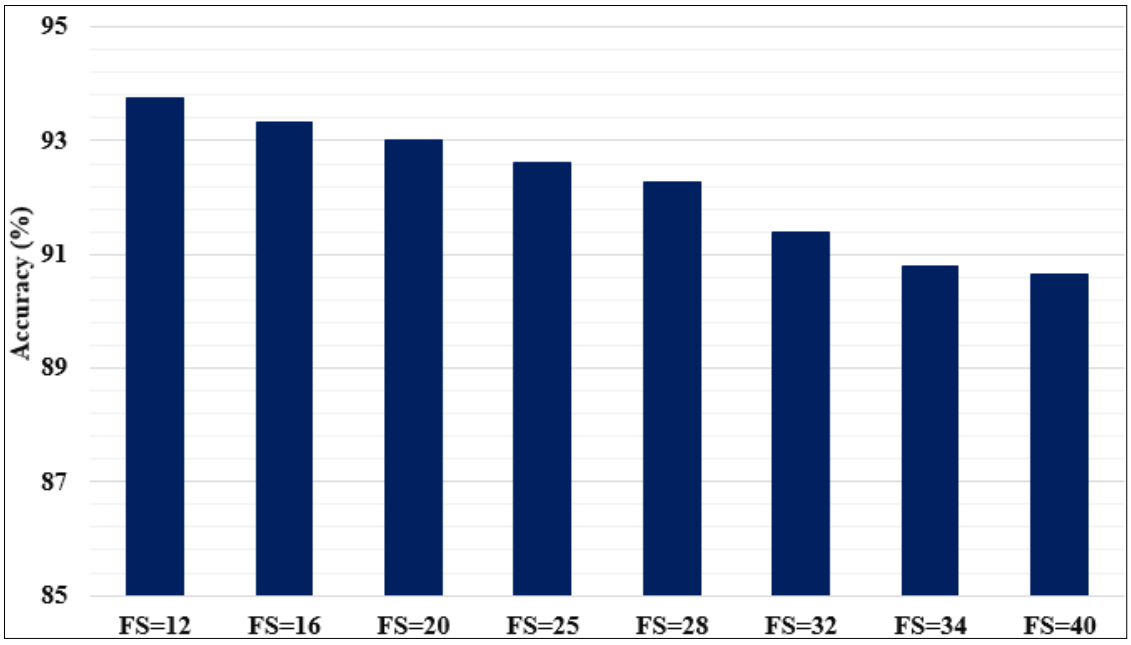
Tablo (4.4)'de, önerilen modelin sonuçları özellik seçimine bağlı olarak gösterilmiştir. Tablo (4.4)'de, özellik sayısının az olduğu durumda algılama doğruluğunun değeri daha büyüktür. Destek vektör makinesi algoritması daha az özelliğe sahip olduğundan dolayı ilişkileri bulmak ve örnekleri sınıflandırmakta daha yeteneklidir. 12 özellikli doğruluk değeri 93.76, 16 özellikli doğruluk değeri 93.34 ve

40 özellikli doğruluk değeri 90.65'e eşittir. Ayrıca, 12 özelliğe sahip DR değeri 93.95 ve 40 özelliğe sahip DR değeri 89.24'tür. İlk olarak 12 özellikli program uygulanmıştır. Kullanılan veri setindeki bireyi tanımlayan özellik sayısının yüksek tutulması algoritmanın doğruluk yüzdesini düşürdüğü görülmüştür. Yapılacak çalışmalarda bu kriter göz önünde bulundurulmalıdır.

Tablo 4.4. Önerilen Modelin, 100 Tekrar Özellik Seçimi ile Karşılaştırılması

Kriterler	**Özellik Seçimi**							
	12	**16**	**20**	**25**	**28**	**32**	**34**	**40**
Hassasiyet	93.81	92.89	92.33	91.71	91.50	90.03	89.44	89.68
Hatırlama	94.26	92.97	93.57	91.84	91.89	91.58	90.17	90.26
F-Ölçüsü	94.03	92.93	92.90	91.77	91.69	90.80	89.80	89.97
DR	93.95	93.31	93.04	91.18	91.10	90.59	90.15	89.24
FAR	8.22	9.03	9.85	10.24	10.94	11.89	13.96	14.73
Doğruluk	93.76	93.34	93.02	92.61	92.28	91.39	90.78	90.65

Şekil 4.2'de, önerilen modelin özellik seçimi ve 100 tekrar ile karşılaştırılması gösterilmiştir. Özellikler arttırıldığında şekilde de görüleceği üzere grafik aşağıya doğru eğilim almıştır.



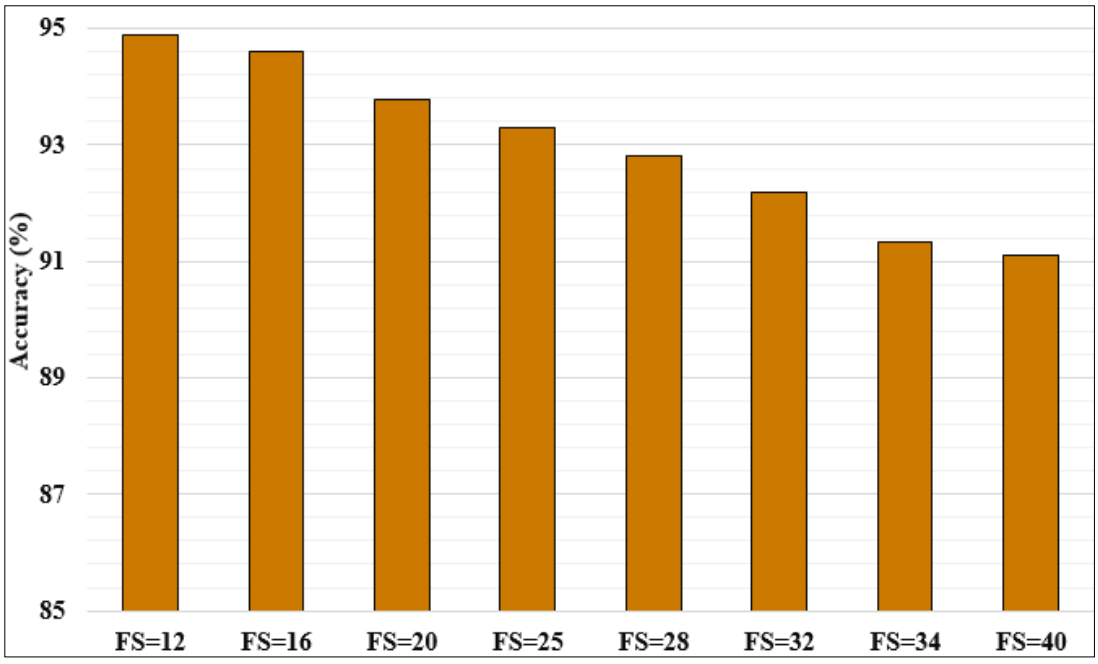
Şekil 4.3. Önerilen Modelin Özellik Seçimi ve 100 Tekrarı Temel Alınarak Karşılaştırılması

Tablo (4.5)'te önerilen modelin sonuçları, özellik seçimi ve 200 tekrarlı olarak gösterilmektedir. Tablo (4.5)'te 12 özellikli doğruluk oranının 94.90, 16 özellikli doğruluk oranının 94.61, 20 özellikli doğruluk oranının 93.79 ve 40 özellikli doğruluk oranının 91.12'ye eşit olduğu görülmektedir. DR değeri 200 tekrar ve 40 özellikte 90.36 ve FAR değeri 13,49 olarak bulunmuştur.

Tablo 4.5. 200 Tekrar ile Özellik Seçimine Dayalı Olarak Önerilen Modelin Karşılaştırılması

Kriterler	**Özellik Seçimi**							
	12	**16**	**20**	**25**	**28**	**32**	**34**	**40**
Hassasiyet	94.15	93.71	93.38	92.59	92.07	91.26	91.04	90.29
Hatırlama	94.92	94.14	93.46	92.78	92.68	91.79	91.36	90.78
F-Ölçüsü	94.53	93.92	93.42	92.68	92.37	91.52	91.20	90.53
DR	94.02	94.62	93.26	93.05	92.44	91.56	91.20	90.36
FAR	7.48	8.19	9.011	10.31	10.89	11.73	12.09	13.49
Doğruluk	94.90	94.61	93.79	93.31	92.81	92.20	91.32	91.12

Şekil (4.3), önerilen modelin özellik seçimine ve 200 tekrarına göre karşılaştırılmasını göstermektedir.



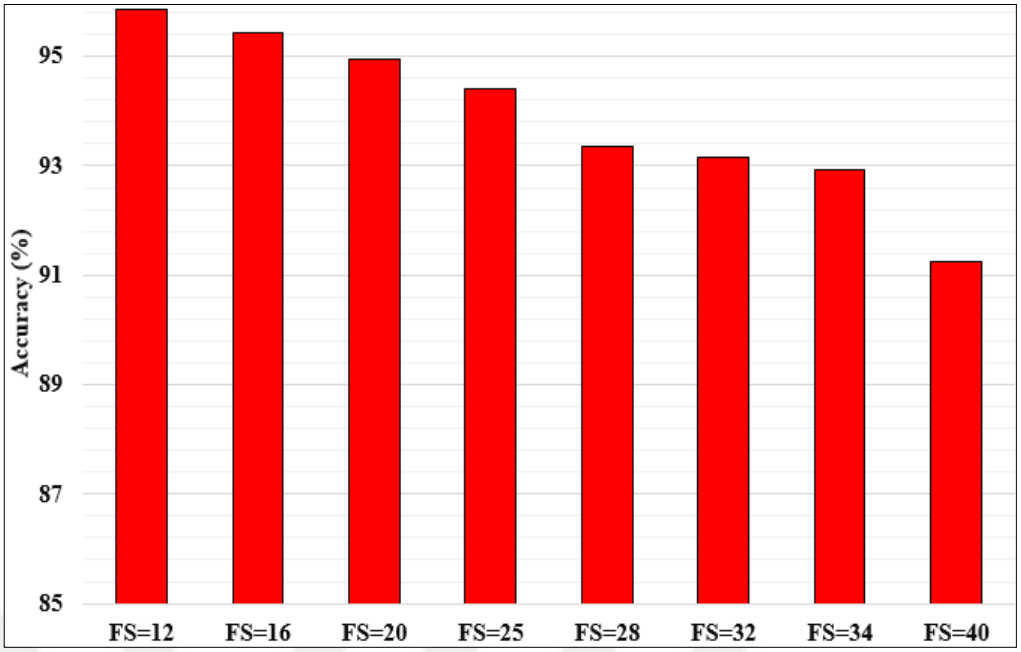
Şekil 4.4. Önerilen Modelin Özellik Seçimine ve 200 Tekrara Göre Karşılaştırılması

Tablo (4.6) 'da önerilen modelin sonuçları özellik seçimine ve 300 tekrara göre gösterilmiştir. Tablo (4.6), 12 özellik ile doğruluk oranının 95.87, 16 özellik ile doğruluk oranının 95.42, 20 özellik ile doğruluk oranının 94.95 ve 40 özellik ile doğruluk oranının 91.26'ya eşit olduğunu göstermektedir. DR değeri 300 tekrar ve 40 özellik ile 91.07 ve FAR değeri 12.18'dir.

Tablo 4.6. 300 Tekrar ile Özellik Seçimine Dayalı Olarak Önerilen Modelin Uygulanması

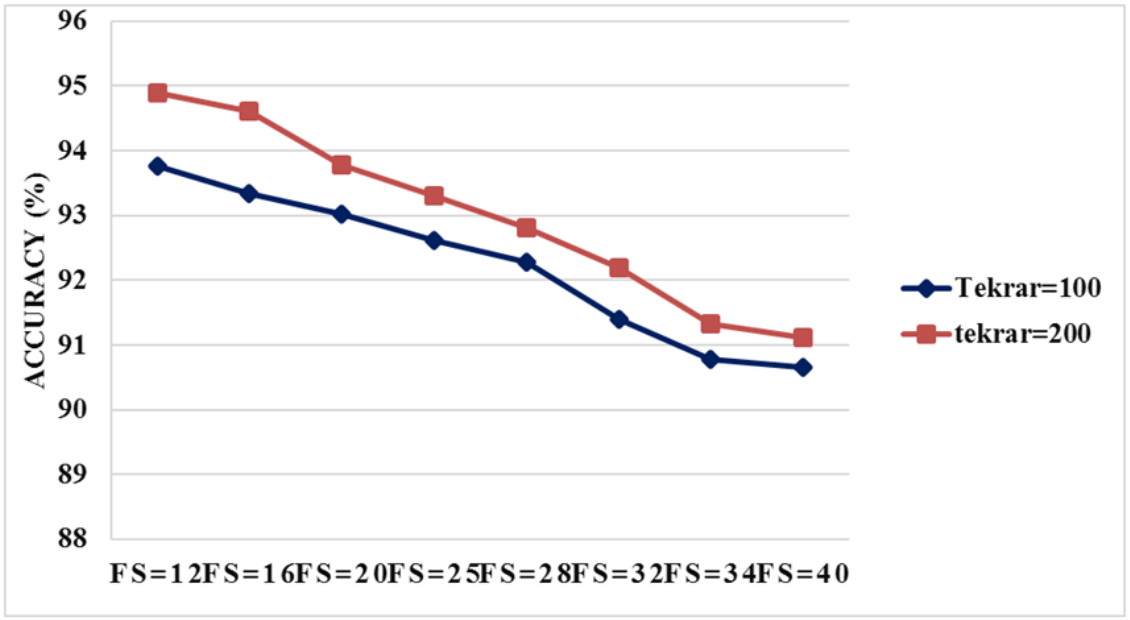
Kriterler	**Özellik Seçimi**							
	12	**16**	**20**	**25**	**28**	**32**	**34**	**40**
Hassasiyet	95.47	95.23	94.52	94.11	93.24	92.63	92.29	91.51
Hatırlama	95.73	95.82	94.91	94.76	93.97	92.81	92.57	91.77
F-Ölçüsü	95.60	95.52	94.71	94.43	93.60	92.72	92.43	91.64
DR	95.23	95.14	94.35	94.16	93.27	93.11	92.63	91.07
FAR	6.68	7.94	8.18	9.69	10.23	11.84	11.26	12.18
Doğruluk	95.87	95.42	94.95	94.41	93.35	93.58	92.94	91.26

Şekil (4.5), önerilen modelin özellik seçimi ve 300 tekrara göre karşılaştırılmasını göstermektedir. Şekil (4.4)’te görüldüğü üzere 12 özellik ile doğruluk değeri %95,87'dir, bu da diğer durumlardan daha fazla olduğunu göstermektedir.



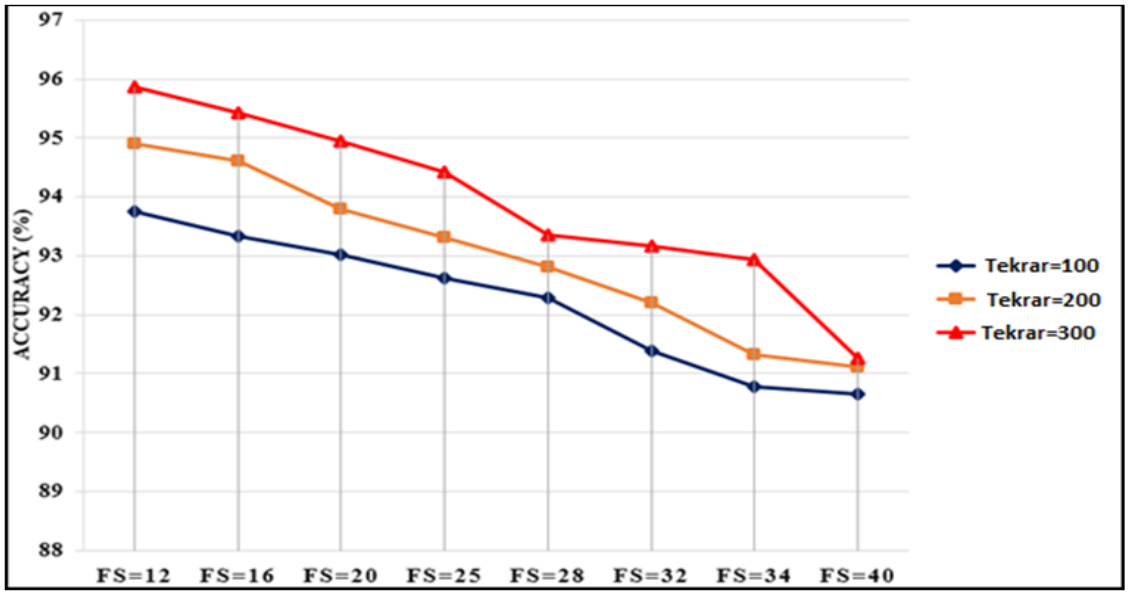
Şekil 4.5. Özellik Seçimine Dayalı Olarak Önerilen Modelin Uygulama Şeması ve 300 Kere Tekrarlanması

Şekil (4.6) önerilen modelin özellik seçimine dayanarak, 100 ve 200 tekrarlamanın karşılaştırılmasını göstermektedir. Şekil (4.4)'e göre 200 tekrarlı modelin doğruluk oranı daha fazla bulunmuştur.



Şekil 4.6. Önerilen Modelin Özellik Seçimine ve Farklı Tekrarlamaya Göre Karşılaştırılması

Sonuçlar; az tekrarla karşılaştırıldığında, daha yüksek tekrarlarla önerilen modelin daha doğru olduğunu ve örneklerin daha iyi kategorilere ayrıldığını göstermektedir. Ayrıca özelliklerin seçimi, daha az sayıda özelliğe sahip önerilen modelin daha fazla algılama doğruluğu olduğunun tespit edildiğini göstermiştir. FAR değeri 200 tekrarda 13.49 bulunmuştur ve bu değer 100 tekrar değerinden daha azdır.



Şekil 4.7. Önerilen Modelin Özellik Seçimine ve Farklı Tekrarlara Göre Karşılaştırılması

Önerilen model 100, 200 ve 300 tekrara dayandırılarak değerlendirilmiştir. Her tekrar için farklı sonuçlar elde edilmiştir. 300 tekrarlama ile önerilen model sonuçlarının 100 ve 200 tekrardan daha iyi olduğunu ve hata oranının daha düşük olduğunu göstermiştir. Ayrıca 400 tekrar ve üzeri bir değerlendirme yapılmış ancak bu tekrarlarda sonuçlar çok fazla değişkenlik göstermediği için tablo ve şekil olarak gösterilmemiştir. Değerlendirme kriterlerindeki yüzde değişim sadece 300 tekrar için kabul edilmiş ve üzeri tekrarlar için tekrarlanan sonuçlar elde edilmiştir. Önerilen modeldeki yüzde değişim 300 kere tekrarlandığında sonuca ulaşılmıştır.

Kriterlerdeki yüzde değişim sadece 300 tekrar için kabul edilmiştir.

Ayrıca, 300 kere tekrarlanan durumda FAR değeri yüzde 12,18'dir ve 100 ve 200 tekrarlamadan daha az olduğu bulgusuna ulaşılmıştır.

BEŞİNCİ BÖLÜM

SONUÇLAR VE ÖNERİLER

5.1. SONUÇ

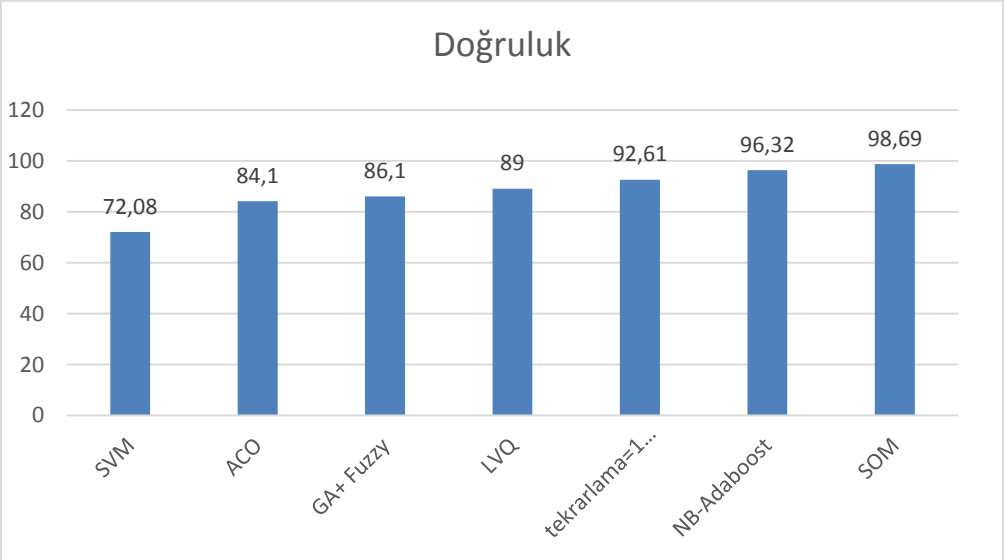
Bilgisayar ağlarındaki saldırıların tespit edilmesi ve önlenmesi, güvenlik tartışmalarındaki temel konulardan biridir. Bu konuda, saldırı tespit sistemleri gibi saldırıları önlemek amacıyla çeşitli yöntemler uygulanmıştır. Saldırı tespit sistemleri genellikle saldırı türlerini tanımlar ve bu saldırıları belirli gruplarda sınıflandırır. Bu tez çalışmasında, k en yakın komşu ve destek vektör makinesinin kombinasyonu saldırı tespit sisteminin uygulaması için kullanılmıştır. Önerilen modeli göstermek ve test etmek için KDD99 veri tabanı kullanılmıştır. Önerilen modelde, en yakın komşu algoritması özellik seçimi, örneklerin sınıflandırılması için ise destek vektör makinesi kullanılmıştır. Özellik seçiminin amacı algılama doğruluğunu arttırmaktır. Sonuçlar, önerilen modelin doğruluğunun 92.61 olduğunu göstermiştir. Bu değer en iyi değere eşit bulunmuştur.

Bu bölümde, önerilen modelin sonuçları diğer modellerle karşılaştırılmıştır. Tablo (5.1)'de görüldüğü üzere önerilen model CSVAC, ACO, SVM, GA + Fuzzy ve LVQ ile karşılaştırılmıştır.

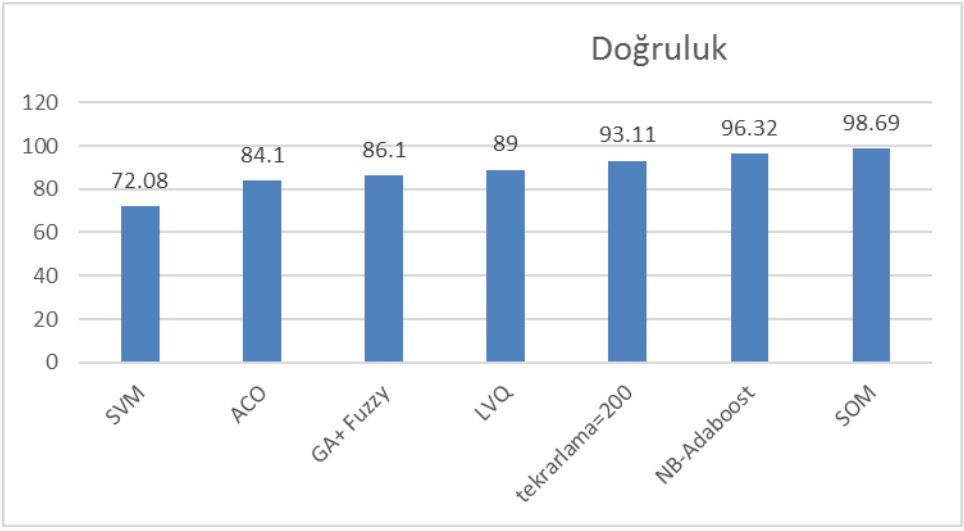
Tablo (5.1)'de önerilen modelin doğruluk oranı, CSVAC, ACO, SVM, GA + Fuzzy ve LVQ modellerinin doğruluk oranından daha yüksektir. Ayrıca, düşük sayıda özelliğe sahip önerilen modelin doğruluk oranı SOFM ve NB + Adaboost modellerinin doğruluk oranından fazla bulunmuştur. Önerilen modelin doğruluğunda üzerindeki tekrar sayısı da etkili olmaktadır.

Tablo 5.1. Önerilen Modelin Doğruluk Oranının Diğer Modellerle Karşılaştırılması

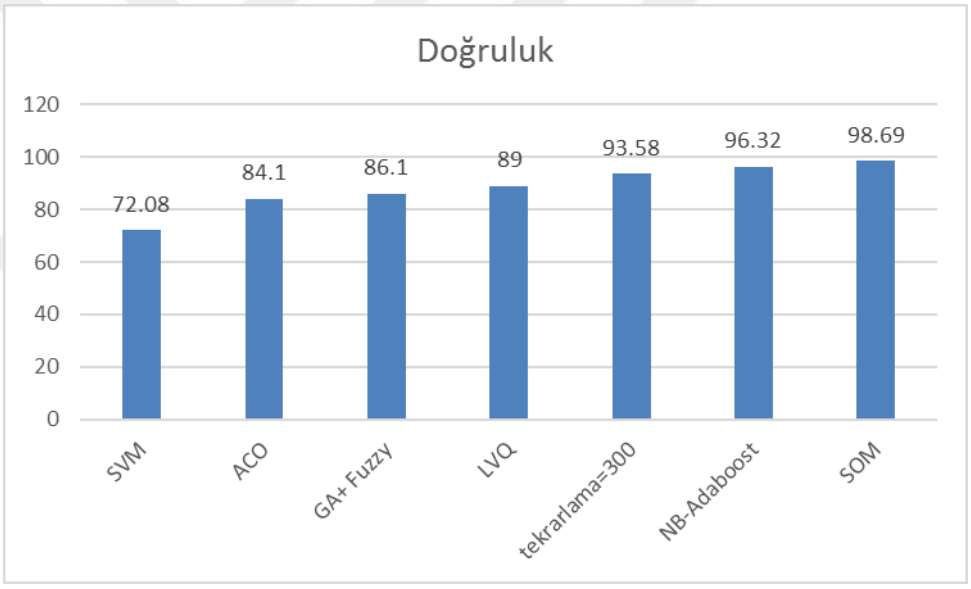
Referanslar	**Modeller**		**Doğruluk oranı**
(Subaira vd., 2014)	CSVAC		88.04
	ACO		84.10
	SVM		72.08
(Morshedu vd., 2013)	GA+ Fuzzy		86.10
(Dagli, 2012)	SOFM		98.69
(Degang vd., 2007)	LVQ		89
(Li vd., 2010)	NB-Adaboost		96.32
-	Önerilen model	Tekrarlama=100	92.61
		Tekrarlama=200	93.11
		Tekrarlama=300	93.58



Şekil 5.1. Önerilen Modelin 12 Özellik Seçimi ve 100 Tekrarının Diğer Modeller ile Karşılaştırılması



Şekil 5.2. Önerilen Modelin 12 Özellik Seçimi ve 200 Tekrarının Diğer Modeller İle Karşılaştırılması



Şekil 5.3. Önerilen Modelin 12 Özellik Seçimi ve 300 Tekrarının Diğer Modeller İle Karşılaştırılması

Önerilen modelin tekrar sayısı, k sayısı gibi farklı kriterlere göre değerlendirilmiştir. Her bir değerlendirmede, kriterler için farklı sonuçlar elde edilmiş, 300 tekrarda doğruluk yüzdesi %93,58 bulunmuştur. Daha fazla tekrarda, önerilen model daha yüksek bir doğruluk oranına ve daha az hata oranına sahiptir. Ayrıca, k değerinin daha düşük olması durumunda önerilen modelin doğruluk yüzdesi artmaktadır. Önerilen modelin doğruluk oranı diğer modellerin doğruluk oranından daha yüksektir ve bazı durumlarda doğruluk yüzdesi daha az bulunmuştur. Sonuçlar,

özellik sayısının düşük olduğunda, doğruluk yüzdesinin yüksek olduğunu ve özellik sayısının yüksek olması durumunda, doğruluk yüzdesinin düşük olduğunu göstermektedir. Özellik sayısının fazla olduğu durumlarda, benzer örneklerin algoritma tespit doğruluğu daha az olacaktır. Sonuçlar, önerilen modelin 100 ve 200 tekrarda doğruluk oranının sırasıyla %92,61 ve %93,11 olduğunu göstermiştir. Önerilen modelin 300 tekrarda ve k = 3 ile doğruluğu 93.58, hata oranı yüzdesi 7,14 bulunmuştur.

Sonuçlar önerilen modelin 300 tekrarının, 100 ve 200 tekrarla karşılaştırıldığında doğruluk oranının daha fazla olduğunu göstermiştir. Karşılaştırmalar sonucunda, önerilen modelin CSVAC, ACO, SVM, GA + Fuzzy ve LVQ modellerinden de doğruluk oranının daha yüksek olduğu bulgusuna ulaşılmıştır.

5.2. ÖNERİ VE GELECEKTEKİ ÇALIŞMALAR

Destek vektör makinesi, sınıflandırma problemlerinde bir araç olarak kullanılmaktadır. Destek vektör makinelerinin dayandığı matematiksel prensipler gereği sınıflandırma problemlerinin çözümünde elde edilen sonucun doğruluk yüzdesi yüksektir. Destek vektör makinesinin amacı, iki sınıf verilerini iyi bir şekilde bölebilecek maksimum marjinal bir hiper düzlem oluşturmaktır. Destek vektör makinesinin zayıf yönlerinden biri, bölünmüş hiper düzlem oluşturulmasında yapısal verilerine dikkat etmemesidir. Bu nedenle, destek vektör makinesinin doğruluğunu artırmak için veriler arasındaki yerel bağlılık (korelasyon) sürdürülmelidir. Greedy algoritması, destek vektör makinesinde kullanılan optimizasyon yöntemlerinden biridir. Bu yöntem, her bir sınıfın numunesi için ortalama matrisleri ve kovaryansı kullanarak tüm olası seçimleri elde eder ve en kötü örnek dağılımı için sınıflandırma doğru bir şekilde yapılır. Sınıflandırma etkinliğini arttırmak için, eğitim sırasında veriler arasındaki yerel bağlılık (korelasyon) dikkate alınmalıdır. Destek vektör makinesinin nispeten yüksek hesaplama karmaşıklığı ile başa çıkmak için, tüm özellikleri kullanmak yerine sınıflandırmayı hızlandıracak niteliklerin bir alt kümesini kullanmak yeterli olacaktır.

Veri setindeki sınıf örneklerinin eşit olmadığı sınıflandırma problemlerinde, bu durum verimliliği olumsuz yönde etkilemektedir. Bir sınıfın örneklerinin sayısı, diğer sınıftan çok daha az olduğunda oluşmaktadır. Toplam hatanın büyüklüğünü en aza indirmeye dayalı normal sınıflandırma yöntemleri, dengesiz veri üzerinde iyi

çalışamazlar çünkü sınıfların dengeli dağılımı olduğu varsayılmaktadır. Bu sorunu düzeltmenin yollarından biri, iki katmanlı bir destek vektör makinesi kullanmaktır. Dengesiz verileri tanımlamak için ilk katmanda birçok örnek alınır ve ikinci katmanda, eğitim ve test için ağırlıklı örnekler seçilir.

En yakın komşuluk algoritmasındaki en önemli en önemli sınıflandırma problemlerinden biri mesafe kriteridir. Mesafe kriteri, bir test numunesinin komşularının hesaplanmasındaki en temel parçadır. Bir sınıflandırma algoritmasının eğitim örnekleri olarak en yakın komşuları kullanmak, veri sıralama için verimli bir yöntemdir. Bu tezde, mesafe kriteri kullanılarak uygun komşular temsilci olarak seçilmiştir. Daha sonra bir ayırıcı hiper düzlem kullanılarak; sınıflandırma algoritmasının, sınıflandırma vektörünü desteklemesi sağlanmıştır. Bu çalışmada saldırı tespiti için hibrit bir sınıflandırma yöntemi kullanılmıştır. Denetlenen sınıflandırmada, sınıflandırma sayısı için eğitim verisi olarak, sınıfların sayısı ve türü gibi örneklerden temel bilgiler alınmıştır.

Sınıflandırma sorunlarından biri, özellik alanı boyutlarının yüksek olmasıdır. Özelliklerin çoğu ilgisiz ve gereksizdir bu yüzden sınıflandırma verimliliği üzerinde olumsuz bir etkiye sahiptir. Bu nedenle özellik seçimi, özellik alanını azaltmak ve sınıflandırma verimliliğini artırmak için kullanılmaktadır. Gelecekteki çalışmalarda üç adımlı yöntemlerin kullanılması planlanmaktadır. Yani öncelikle, akıllı bir yöntem kullanılarak önemli özellikler ayıklanmalı ve daha az etkisi olan özellikler, özellik kümesinden kaldırılmalıdır. İkinci adımda, bulanık mantık modelleri kullanılarak ilk adımda kaldırılmayan özellikler tanımlanacaktır. İlk veriler belirsiz olduğundan dolayı bunlar bulanık kümeler olarak tanımlanır. Daha sonra bulanık kümeler algoritmaya kategorize edilecektir. Son adımda, özelliklerin boyutlarını azaltmak için bir sınıflandırma algoritması kullanılacaktır.

KAYNAKÇA

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

Aziz, A.S.A, EL-Ola Hanafi, S., Hassanien, A.E. (2017). "Comparison of Classification Techniques Applied for Network Intrusion Detection and Classification", *Journal of Applied Logic*, 24 (A), 109-118.

Bace, R., & Mell, P. (2001). "NIST special publication on intrusion detection systems.", *Booz-allen and hamilton inc mclean va*.

Barbara, D., Couto, J., Jajodia, S., Popyack, L., & Wu, N. (2001). "ADAM: Detecting intrusions by data mining". *In Proceedings of the IEEE Workshop on Information Assurance and Security*.

Bukac, V., Tucek, P., Deutsch, M. (2012). "Advances and Challenges in Standalone Host-Based Intrusion Detection Systems", *Trust, Privacy and Security in Digital Business*. , 105-117.

Cai, C., Yuan, L. (2013). "Intrusion Detection System Based on Ant Colony System", *Journal of Networks*, 8(4), 888-894.

Chen, P.J., Chen, Y.W. (2015). "Implementation of SDN Based Network İntrusion Detection and Prevention System", *International Carnahan Conference on Security Technology (ICCST)*, 141-146.

Chen, Z., Chen, Z., Delis, A. (2007). "An Inline Detection and Prevention Framework for Distributed Denial of Service Attacks", *the Computer Journal*, 50(1), 7-40

Colom, J.F., Gil, D., Mora, H., Volckaert, B. (2018). "Scheduling Framework for Distributed Intrusion Detection Systems Over Heterogeneous Network Architectures", *Journal of Network and Computer Application*s, In press, Accepted Manuscript, Available online 10 February.

Dagli, C.H. (2012). "Applying Variable Coefficient functions to Self-Organizing Feature Maps for Network Intrusion Detection on the 1999 KDD Cup Dataset", *Procedia Computer Science*, 8, 333-337.

Debar, H., Viinikka, J. (2005). “Intrusion Detection: Introduction to Intrusion Detection and Security Information Management”, *Foundations of Security Analysis and Design*, *Springer,* 207-236.

Degang, Y., Guo, C., Hui, W., Xiaofeng, L. (2007). “Learning Vector Quantization Neural Network Method for Network Intrusion Detection, Wuhan” *University Journal of Natural Sciences*, 12(1) , 147-150.

Depren, O., Topallar, M., Anarim, E., Ciliz, M.K. (2005). “An Intelligent Intrusion Detection System (IDS) for Anomaly and Misuse Detection in Computer Networks”, *Expert Systems with Applications*, 29, 713–722.

Dhyaram, L.P., Vishnuvardhan, B. (2018). “Classification Performance Improvement Using Random Subset Feature Selection Algorithm For Data Mining”, *Big Data Research*, In Press, Accepted Manuscript, Available Online 25 April .

Ding, S., Jia, H., Du, M., Xue, Y. (2018). “A Semi-Supervised Approximate Spectral Clustering Algorithm Based on HMRF model”, *Information Sciences*, 429, 215-228.

Drugan, M.M. (2018). “Reinforcement Learning Versus Evolutionary Computation: A Survey on Hybrid Algorithms”, *Swarm and Evolutionary Computation, in Press*, Corrected proof, Available Online 29 March.

Farnaaz, N., Jabbar, M.A. (2016). Random Forest Modeling for Network Intrusion Detection System”, *Procedia Computer Science*, 89, 213-217.

Fischetti, M. (2016). “Fast Training of Support Vector Machines With Gaussian Kernel”, *Discrete Optimization*, 22 (A), 183-194.

Gastaldo, P., Bisio, F., Gianoglio, C., Ragusa, E., Zunino, R. (2017). “Learning With Similarity Functions: A novel Design for the Extreme Learning Machine”, *Neurocomputing*, *261*, 37-49.

Gueorguieva, N., Valova, I., Georgiev, G. (2017). “M&MFCM: Fuzzy C-means Clustering with Mahalanobis and Minkowski Distance Metrics”, *Procedia Computer Science*, 114, 224-233.

Gunupudi, R.K., Nimmala, M., Gugulothu, N., RGali, S. (2017). "CLAPP: A self-Constructing Feature Clustering Approach for Anomaly Detection", *Future Generation Computer Systems,74*, 417-429.

Hamed, T., Dara, R., Kremer, S.C. (2018). "Network Intrusion Detection System Based on Recursive Feature Addition and Bigram Technique", *Computers & Security*, *73*, 137-155.

https://www.slideshare.net/DerisStiawan/network-attack-and-intrusion-prevention-system

http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

https://nl.mathworks.com/help/stats/pdist.html

https://www.cisco.com/c/en/us/about/security-center/guide-ddos-defense.html

https://www.coherentnews.com/cloud-ids-and-ips-are-methodology-used-to-escalate-the-security-level-of-networks/

Jadhav, A., Jadhav, A., Jadhav, P., Kulkarni, P. (2013). "A Novel Approach for the Design of Network Intrusion Detection System(NIDS)", *PROCEEDINGS OF 2013 International Conference on Sensor Network Security Technology and Privacy Communication System*, 22-27.

Kaur, D., Punja, R. (2014). "A Comparative Study of Various Distance Measures for Software Fault Prediction", *International Journal of Computer Trends and Technology (IJCTT*), 17(3), 1-4.

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

Kim, B.K., Jang, J.S., Chung, T.M. (2002). "Design of Network Security Control System for Cooperative Intrusion Detection", *International Conference on Information Networking: Wireless Communications Technologies and Network Applications, Springer*, 389-398.

Kim, G., Lee, S., Kim, S. (2014). "A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection With Misuse Detection", *Expert Systems With Applications*, 41 (4), 1690-1700.

Kizza, J.M. (2017). “System Intrusion Detection and Prevention, Guide to Computer Network Security”, 275-301.

Kshirsagar, D., Sawant, S., Wadje, R., Gayal, P. (2017). “Distributed Intrusion Detection System for TCP Flood Attack”, *Proceeding of International Conference on Intelligent Communication, Control and Devices*, Springer, 951-958.

Li, W., Li, Q.X. (2010). “Using Naive Bayes with AdaBoost to Enhance Network Anomaly Intrusion Detection”, *Third International Conference on Intelligent Networks and Intelligent Systems*, 486-489.

MacQueen, J.B. (1967). “Some Methods for Classification and Analysis of Multivariate Observations”, *In Proceedings of 5th Berkley Symposium on Mathematical Statistics and Probability, 1*, 281-297.

Mantur, B., Desai, A., Nagegowda, K.S. (2015). “Centralized Control Signature-Based Firewall and Statistical-Based Network Intrusion Detection System (NIDS) in Software Defined Networks (SDN)”, *Emerging Research in Computing, Information, Communication and Applications, Springer*, 497-506.

Martin, F. (1995). *Instance-Based Learning: Nearest Neighbour with Generalisation*. (Doctoral Dissertation), University of Waikato.

Mazzeo, G.M., Masciari, E., Zaniolo, C.. (2017). “A Fast and Accurate Algorithm for Unsupervised Clustering Around Centroids”, *Information Sciences*, 400-401, 63-90.

Mkuzangwe, N.N.P., Nelwamondo, F.V. (2017). “A Fuzzy Logic Based Network Intrusion Detection System for Predicting the TCP SYN Flooding Attack”, *Intelligent Information and Database Systems, Springer*, 14-22.

Mohamadi, H., Habibi, J., Abadeh, M.S. (2008). “Misuse Intrusion Detection Using a Fuzzy-Metaheuristic Approach”, *Second Asia International Conference on Modelling & Simulation (AMS)*, 439-444.

Morshedur Hassan, M.Md., 2013). “Network Intrusion Detection System Using Genetic Algorithm and Fuzzy Logic”, 1(7), 1435-1445.

Nejad, T.R., Abadi, M.S.A. (2014). "Intrusion Detection in Computer Networks Through a Hybrid Approach of Data Mining And Decision Trees" *WALIA Journal* ,30(S1), 233-237.

Nguyen, H.A., Choi, D. (2008). "Application of Data Mining to Network Intrusion Detection: Classifier Selection Model, APNOMS 2008, LNCS 5297", *Springer-Verlag Berlin Heidelberg*, 399-408

Niksefat, S., Kaghazgaran, P., Sadeghiyan B. (2017). "Privacy Issues in Intrusion Detection Systems: A Taxonomy, Survey and Future Directions", *Computer Science Review*, 25, 69-78.

Obimbo, C., & Jones, M. (2012). Applying Variable Coe_cient functions to Self-Organizing Feature Maps for Network Intrusion Detection on the 1999 KDD Cup Dataset. Procedia Computer Science, 8, 333-337.

Papamartzivanos, D., Marmol, F.G., Kambourakis, G. (2018). "Dendron: Genetic Trees Driven Rule Induction for Network Intrusion Detection Systems", *Future Generation Computer Systems*, 79 (2), 558-574.

Qadeer, M.A, Iqbal, A., Zahid, M., Siddiqui, M.R. (2010). "Network Traffic Analysis and Intrusion Detection Using Packet Sniffer", *Second International Conference on Communication Software and Networks*, 313-317.

Sengupta, N., Sil, J. (2012). "Comparison of Supervised Learning and Reinforcement Learning in Intrusion Domain", *Wireless Networks and Computational Intelligence*, 546-55.

Shah, S.A.R., Issac, B. (2018). "Performance Comparison of Intrusion Detection Systems and Application of Machine Learning to Snort System", *Future Generation Computer Systems, 80*, 157-170.

Silva, A.P.D. (2017). "Optimization approaches to Supervised Classification*", European Journal of Operational Research*, 261 (2), 772-788.

Singh, P., Tiwari, A. (2015). "An Efficient Approach for Intrusion Detection in Reduced Features of KDD99 Using ID3 and Classification with KNNGA", *Second International Conference on Advances in Computing and Communication Engineering*, 445-452.

Subaira, A.S., Anitha, P., Anjusree, S., Vinothini, C. (2014). "An Approach For Ids by Combining Svm and Ant Colony Algorithm", *International Journal of Research in Engineering and Technology*, 3(3), 459-465.

Sullivan, R. (2011). "Introduction to Data Mining for the Life Sciences", *Machine-Learning Techniques*, 363-454.

Tran, N.T., Tomiyama, S., Kittitornkun, S., Vu, T.H. (2012). "TCP Reassembly for Signature-Based Network Intrusion Detection systems", *Computer, Telecommunications and Information Technology*, IEEE, 1-4.

Tylman, W. (2008). "Misuse-Based Intrusion Detection Using Bayesian Networks", *Third International Conference on Dependability of Computer Systems DepCoS-RELCOMEX*, 203-210.

Verma, A., Ranga, V. (2018). "Statistical Analysis of CIDDS-001 Dataset for Network Intrusion Detection Systems using Distance-Based Machine Learning", *Procedia Computer Science*, *125*, 709-716.

Wang, W., Guan, X., Zhang, X. (2004). "A Novel Intrusion Detection Method Based on Principle Component Analysis in Computer Security, International Symposium on Neural Networks", *Advances in Neural Networks, Springer*, 657-662.

Wang, X. (2014). "Ethernet Firewall Intrusion Detection Research and Application", *Proceedings of the 2012 International Conference on Cybernetics and Informatics*, 65-71.

Yang, L., Weng, D. (2012). "Snort-based Campus Network Security Intrusion Detection System", *Information Engineering and Applications, Springer*, 824-831.

Yousufi, R.M., Lalwani, P., Potdar, M.B. (2017). "A Network-Based Intrusion Detection and Prevention System With Multi-Mode Counteractions", *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 1-6.

Yuan, Y., Huo, L., Hogrefe, D. (2017). "Two Layers Multi-class Detection Method for Network Intrusion Detection System", *IEEE Symposium on Computers and Communications (ISCC)*, 767-772.

Yunfeng, Y. (2012). “Research on the System Model of Network Intrusion Detection”, *Proceedings of the 2012 International Conference of Modern Computer Science and Applications*, 185-190.

Zaman, S., Karray, F. (2009). “TCP/IP Model and Intrusion Detection Systems”, *International Conference on Advanced Information Networking and Applications Workshops*, IEEE, 90-96.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Meysam AHAN PANJEH
Doğum Yeri ve Tarihi	İran- Urmia 1981
Eğitim Durumu	
Lisans Öğrenimi	URUMİYEH AZAd Üniversitesi Mühendislik Fakültesi Bilgisayar Bölümü
Y. Lisans Öğrenimi	Yönetim Bilişim Sistemleri Bölümü
Bildiği Yabancı Diller	İngilizce –Farsca
Bilimsel Faaliyetleri	
İş Deneyimi	
Stajlar	
Projeler	
Çalıştığı Kurumlar	Jahad daneshgahi- Üniversite Otomasyon Müdürü
İletişim	
E-Posta Adresi	meysam.ah@gmail.com
Tarih	Temmuz 2018