
GAZİ ÜNİVERSİTESİ
Gazi
1926

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANARAK HİZMET AKSATMA SALDIRILARI TESPİTİ

Emin DOĞRU

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2020

Emin DOĞRU tarafından hazırlanan "MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANARAK HİZMET AKSATMA SALDIRILARI TESPİTİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Necaattin BARIŞÇI

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. …….

Başkan: Prof. Dr. A.Kürşad TÜRKER

Endüstri Mühendisliği Ana Bilim Dalı, Kırıkkale Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……..

Üye: Doç. Dr. Aydın ÇETİN

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……..

Tez Savunma Tarihi: 25/06/2020

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

…….…….

Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

EMİN DOĞRU

25/06/2020

MAKİNE ÖĞRENMESİ TEKNİKLERI KULLANARAK HİZMET AKSATMA SALDIRILARI TESPİTİ

(Yüksek Lisans Tezi)

Emin DOĞRU

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2020

ÖZET

Bu çalışmada, her geçen gün artan siber saldırılara karşı elektronik ortamdaki verilerin korunması amacıyla geliştirilen ve en önemli araçlardan biri olan Saldırı Tespit Sistemleri (STS) üzerine araştırmalar yapılmıştır. Bu kapsamda makine öğrenmesi teknikleri çalışılmış, saldırı türleri araştırılmış, daha önce geliştiren STS'ler ve STS'ler üzerine yapılmış akademik çalışmalar incelenmiş, en çok kullanılan veri setleri tespit edilmiştir. Yapılan araştırmalar neticesinde günden güne gelişmekte ve çoğalmakta olan saldırı türlerinin tespit edilebilmesi için geleneksel makine öğrenmesi tekniklerinin dışında tekniklere başvurulması gerektiği tespit edilmiştir. Bu konuda en iyi sonuçları hibrit makine öğrenmesi tekniklerinin verdiği görülmüş ve en hızlı ve başarılı sonuç alınabilecek hibrit makine öğrenmesi teknikleri araştırılmıştır. Bu bilgiler ışığında çalışmada, hibrit makine öğrenmesi tekniklerinden Karar Tablosu ve Naive Bayes (KTNB), Lojistik Model Ağacı (LMA), Fonksiyonel Ağaç (FA), Naive Bayes Ağacı (NBA) teknikleri, veri seti olarak en çok kullanılan KDD'99 veri seti, yazılım olarak ise açık kaynak kodlu olan WEKA yazılımı tercih edilmiştir. Yapılan çalışma sonucunda hibrit algoritmalar ile hem başarı oranının arttığı hem de veriyi işleme süresinin azaldığı tespit edilmiştir. En başarılı sonucu ise %99,95'lik sonuç ile Karar Tablosu ve Naive Bayes tekniklerinin hibrit olarak kullanıldığı teknik vermiştir.

Bilim Kodu	:	92432
Anahtar Kelimeler	:	Makine Öğrenmesi, Hizmet Aksatma Saldırıları Tespiti
Sayfa Adedi	:	65
Danışman	:	Prof. Dr. Necaattin BARIŞÇI

DENAIL OF SERVICE ATTACK DETECTION USING MACHINE LEARNING TECHNIQUES

(M. Sc. Thesis)

Emin DOĞRU

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2020

ABSTRACT

In this thesis, research has been conducted on Intrusion Detection Systems (IDS), which is one of the most important tools developed to protect the data in electronic environment against increasing cyber-attacks. In this context, machine learning techniques have been studied, the types of attacks have been investigated, the academic studies on IDS and IDS that have been developed before have been examined, and the most used datasets have been determined. As a result of the researches, it has been determined that in order to identify the types of attacks that are developing and increasing day by day, techniques other than traditional machine learning techniques should be used. Hybrid machine learning techniques have been shown to give the best results in this regard, and hybrid machine learning techniques that can achieve the fastest and most successful results have been investigated. In the light of this information, KDD'99, which is the most used dataset, the Decision Table and Naive Bayes (KTNB), Logistics Model Tree (LMA), Functional Tree (FA), Naive Bayes Tree (NBA) techniques, and WEKA software, which is open source, was preferred as software. As a result of the study, it was determined that hybrid algorithms both increase the success rate and decrease the data processing time. The most successful result is the technique using the Decision Table and Naive Bayes techniques as a hybrid with a 99.95% result.

Science Code	:	92432
Key Words	:	Machine Learning, Denail Of Service Attack
Page Number	:	65
Supervisor	:	Prof. Dr. Necaattin BARIŞÇI

TEŞEKKÜR

Çalışmalarım boyunca değerli katkılarını, yardımlarını esirgemeyen, tecrübesini ve ilmini cömertçe aktarmaya çalışan değerli tez danışmanım Prof. Dr. Necaattin BARIŞÇI hocama teşekkürü bir borç bilirim.

Ayrıca; her türlü zorluklarla mücadele ederek bu günlere gelmemde emeği geçen Annem ve Babama, başarılı olmam için desteğini hiçbir zaman esirgemeyen, yaşantıma renk katan, sırdaşım, yoldaşım, hayat arkadaşım, değerli eşim Meryem DOĞRU ile hayatıma anlam veren, gözümün nuru, biricik kızım Öykü DOĞRU'ya şükranlarımı sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

SİMGELER VE KISALTMALAR

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

Kısaltmalar	Açıklamalar
ACK	Acknowledgement Flag
ARFF	Attribute Relationship File Format
DARPA	Savunma İleri Araştırma Projeleri Teşkilatı
DDoS	Dağıtık Hizmet Aksatma Saldırısı
DoS	Hizmet Aksatma Saldırısı (Denial of Service)
FA	Fonksiyonel Ağaç
FTP	File Transfer Protokol
GA	Genetik Algoritma
GAP	Genetik Ağ Programlama
IP	İnternet Protokolü
İSS	İnternet Servis Sağlayıcıları
KDD	Bilgi Keşfi ve Teslimatı
KTNB	Karar Tablosu ve Naive Bayes
LAN	Yerel Alan Ağını (Local Area Network)
LMA	Lojistik Model Ağacı
MAE	Ortalama Mutlak Hata
MIT	Massachusetts Teknoloji Enstitüsü
NB	Naive Bayes
NBA	Naive Bayes Ağacı
PING	Packet İnternet Gopher
R2L	Uzaktan Yerele (Remote to Local)
RMSE	Kök Kareler Karesi
SOM	Ön düzenleyici Haritalar (Self-Organization Map)
STS	Saldırı Tespit Sistemi
SYN	Senkronize (Synchronous)
TCP	Transmission Control Protocol
UDP	Kullanıcı Veri Birimi Protokolü

Kısaltmalar	Açıklamalar
U2L	Kullanıcıdan Yerele (User to Local)
U2R	Kullanıcıdan Yöneticiye (User to Root)
WEKA	Waikato Environment for Knowledge Analysis
YSA	Yapay Sinir Ağları

1. GİRİŞ

Teknolojinin hızla gelişmesi, bilgilerin elektronik ortama aktarılmasına hız kazandırmış ve bu aktarımı zorunlu hale getirmiştir. Ayrıca aktarılan bu verilerin dışında günümüzde yeni üretilen veriler de kâğıt ortamından çok elektronik ortamda üretilmektedir. Bütün bilgilerimiz sanal ortamda iken bunların korunması da büyük öneme sahiptir. Korumak için öncelikle saldırıları tespit edebilmek ve daha sonrasında bunlara karşı tedbirler almak gerekmektedir.

Her geçen gün daha da artan virüslerin ve bilgisayar korsanlarının sayısı, kullanmakta olduğumuz telefonlara habersizce yüklenebilen zararlı mobil yazılımlar, bilginin saklanmasını, taşınmasını ve kopyalanmasını zorlaştırmaktadır. Bunlara tedbir olarak Saldırı Tespit Sistemleri geliştirilerek virüs daha veriye ulaşamadan tanımlanarak ortaya çıkarılmalı ve yok edilmelidir.

Saldırı Tespit Sistemleri erken uyarı sistemi olarak görev yapmakta ve bizleri önceden uyarmaktadır. Sadece yazılımsal veya donanımsal olabileceği gibi iki özelliği bir arada kullanabilen STS'ler de günümüzde mevcuttur.

STS'lerin hızlı, kararlı ve kesin sonuçlar ortaya koyabilmesi için kullanılan veri setinin her türlü saldırı örneği bulundurması gerekmektedir. Yeterince çeşitlilik sağlanamaz ise öğrenme gerçekleşmeyeceği için STS doğru sonuçlara ulaşamayabilecektir.

Hali hazırda STS geliştirilmesi için birçok çalışma yapılmaktadır. Bu çalışmaların büyük bir bölümünde KDD'99 veri seti kullanılmaktadır. KDD'99 veri seti kendisini ispat edebilmiş, ulaşılması kolay, makine öğrenmesi tekniklerinin birçoğu ile uyumlu ve araştırmacı için geçmiş çalışmalar ile karşılaştırma yapılabilme imkânı sunmaktadır.

Güvenlik duvarları da saldırı tespiti için kullanılabilmektedir ancak sadece paket ve port bazında filtreleme yaptığından gelen paketler zararsız paketler olabilmekte ancak güvenlik duvarından geçememektedir. Onun yerine saldırı tespit sistemleri kullanarak gelen paketin zararlı mı zararsız mı olduğu analiz edilebilmekte ve buna göre paketler yönlendirilmektedir.

Günümüzde hizmet aksatma saldırıları, sistemi belli bir süre ile işlevsiz hale getirerek kullanıcıların bu sistemi kullanamamalarına neden olmaktadır. Bu tarz saldırılar hem prestij kaybına ve hem de maddi kayıplara neden olmaktadır. Bu saldırı türünü yapmak artık bilgisayardan az da olsa anlayan kötü niyetli insanlar için kolay hale gelmiştir.

Şirketler, kamu kurum ve kuruluşları, sosyal medya sağlayıcıları hem prestij ve müşteri kaybetmemek hem de maddi ve manevi olarak zarara uğramamak için bu saldırı türüne önlem almaya çalışmakta ve buna büyük önem vermektedirler.

Bu çalışmada hizmet aksatma saldırılarının tespit edilmesine çalışılmıştır. Bu kapsamda geçmiş yıllarda yapılan araştırmalar incelenmiş olup büyük bir ilerleme kat edildiği görülmesine rağmen daha da geliştirilebileceği ve saldırının keşif süresinin kısaltılabileceği ve başarı oranının arttırılabileceği öngörülmektedir.

Çalışmada WEKA (Waikato Environment for Knowledge Analysis) yazılımı [1] üzerinde bulunan hibrit makine öğrenmesi teknikleri kullanılarak hangisinin daha doğru sonuç vereceği ve daha hızlı olacağı incelemiştir. Analiz sırasında hibrit makine öğrenmesi tekniklerinden olan Karar Tablosu ve Naive Bayes (KTNB), Lojistik Model Ağacı (LMA), Fonksiyonel Ağaç (FA), Naive Bayes Ağacı (NBA) sınıflandırıcıları kullanılmıştır.

Bu tez çalışması 6 bölümden oluşmaktadır. Birinci bölümde STS'ler hakkında genel bilgiler verilmiş, neden STS'lere ihtiyaç duyulduğundan bahsedilmiş ve bu tezde hangi yazılımın, veri setinin ve algoritmaların kullanılacağına değinilmiştir.

İkinci bölümde STS'lerin detaylı tanımına yer verilmiş, tarihsel gelişimden bahsedilmiştir. STS'lerin bilgi güvenliği açısından önemi vurgulanmıştır. STS'lerin sınıflandırılmasından da bu bölümde bahsedilmiş ve detaylı olarak açıklanmıştır. Ayrıca Hibrit STS'ler detaylı olarak incelenmiş, başarı durumlarına bakılmış ve tarihsel gelişimleri listelenmiştir.

Üçüncü bölümde, ilk olarak günümüzde yapılmakta olan saldırılar türlerine göre genel olarak açılanmıştır. Bunun yanı sıra tezin asıl üzerinde durduğu Hizmet Aksatma Saldırıları detaylı olarak incelenmiştir. Hizmet Aksatma Saldırılarının türleri ve özellikleri hakkında detaylı açıklamalara yer verilmiştir.

Dördüncü bölümde, makine öğrenmesi teknikleri incelenmiş, tarihsel gelişimlerinden bahsedilmiştir. Klasik makine öğrenmesi tekniklerine örnekler verilmiş ve açıklanmış, tez için kullanılacak hibrit makine öğrenmesi teknikleri detaylı olarak açıklanmıştır.

Beşinci bölümde, öncelikle kullanılacak yazılım ve veri tabanı detaylı olarak incelenmiştir. Daha sonrasında toplanan veriler üzerinde çalışmalar yapılarak kullanacağımız yazılıma uygun hale getirilmiş, düzenlenen bu veriler ile dördüncü bölümde bahsedilen makine öğrenmesi teknikleri denenmiş ve en başarılı makine öğrenmesi tekniği belirlenmiştir.

Altıncı bölümde ise karşılaşılan problemlere yer verilmiş, çözüm yolları konusunda yönlendirmeler yapılmış, çalışma sonucunda elde edilen sonuçlar açıklanmış ve tezin genel olarak başarı durumuna değinilmiştir.

2. SALDIRI TESPİT SİSTEMİ (STS)

Bilgisayar, telefon ve tablet gibi elektronik aletlerdeki teknolojik gelişmelerin hızlanması ve internetin yaygınlaşması ile birlikte verilerimiz elektronik ortamlarda barındırılmaya, aktarılmaya ve işlenmeye başlamıştır. Bu sebeple verilerin büyüklüğü ve önemi günden güne artmaktadır. E-ticaret, E-devlet, bankacılık ve sosyal medya gibi elektronik ortamda yapılan işlemlerde en özel ve önemli bilgiler kullanılmakta olduğundan bu verilerin korunması da zorunlu hale gelmektedir.

Teknoloji olumlu yönde çok hızlı gelişirken olumsuz yönde de bir o kadar artış yaşanmaktadır. Elektronik ortamlarda yapılan saldırılar günden güne artmakta, tespit edilmesi ve karşı konulması zorlaşmaktadır. Ancak bu saldırıların yapıldıktan sonra tespit edilmesinin ötesinde saldırı daha olmadan tedbir almak veya saldırı olurken saldırının kim tarafından yapıldığını, saldırı türünü, ağımızdaki hangi zafiyeti kullanarak yapıldığını ve zamanlama bilgilerini tespit etmek önem arz etmektedir.

Bu saldırılara karşı koyabilmek amacıyla akademik ve bilimsel çalışmaların yanında bu çalışmaları son kullanıcının hizmetine sunmak için de teknolojik çalışmalar yapılarak Saldırı Tespit Sistemleri geliştirilmektedir.

Saldırı tespit sistemleri için değişik tanımlamalar yapmak mümkündür. Bunlardan bazıları;

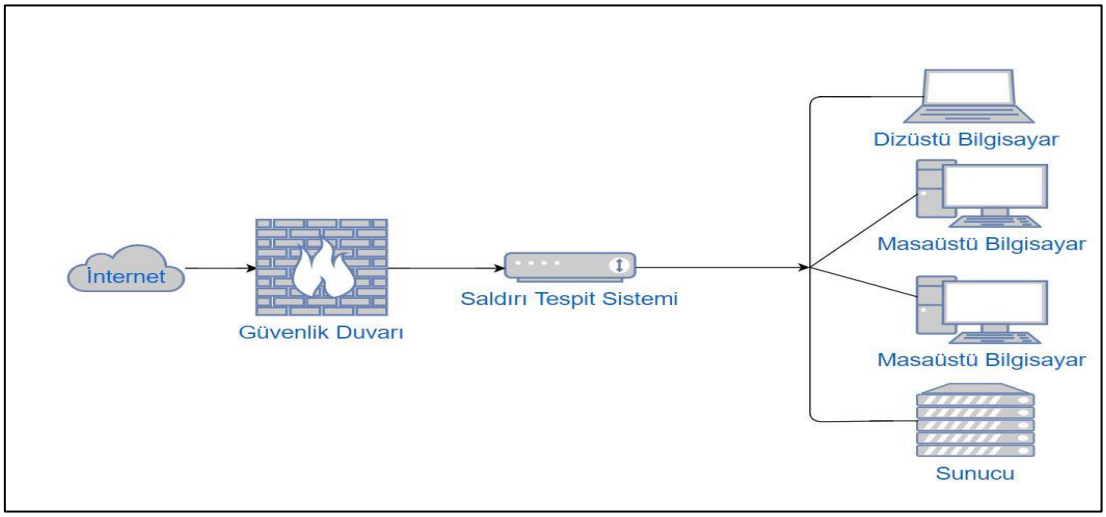
- Saldırı tespit sistemleri basit anlamda hırsız alarmlarıdır [2].
- Saldırı tespit sistemleri, bir ağdaki trafik modellerini ve mesaj türlerini analiz eden ve daha sonra potansiyel sızma girişimleri hakkında uyarı ve alarmlar sağlayan trafik toplayıcıları olarak tanımlanır [3].
- Saldırı tespit sistemleri, bir bilgisayar sisteminin veya ağın davetsiz misafirlerin yetkisiz kullanımını veya yetkili kişiler tarafından yanlış veya kötüye kullanımını tanımlamaktır [4].
- Saldırı tespit sistemleri, bir bilgisayar sistemindeki veya ağındaki olayları izleme ve izinsiz giriş belirtilerini bulmak için bunları analiz eden sistemlerdir [5].

Saldırı Tespit Sistemi ilk olarak Ekim 1972 tarihinde James P. Anderson tarafından hazırlanan makalede gündeme gelmiştir. Anderson bu makalede ABD Hava Kuvvetleri

birliklerinde bilgisayar güvenliğinin önemini ortaya koymuştur. Bu tarihte ABD Hava Kuvvetleri kullanıcı tabanlı bir karmaşık bilgisayar sistemine sahiptir. 40 yıl önce sahip olunan bu sistem hala günümüzde farklı bakış açıları ile kullanılmakta olup aynı güvenlik risklerini içermekte, hatta bu güvenlik riskleri katlanarak artmıştır [6].

1980 yılında James P. Anderson [7] bilgisayar güvenliği, denetimi ve gözetimini iyileştirmenin yollarını açıklayan bir çalışma yayınlanmıştır.

Araştırmacı Dorothy Denning ve Peter Neumann 1984-1986 yılları arasında ilk STS'yi araştırmış ve geliştirme yollarını keşfetmişlerdir [8]. Bu çalışmalarda bilgisayar sistemlerindeki güvenlik sorununa bir çözüm olarak saldırı tespit kavramını önermişlerdir. Buradaki temel fikir bilgisayar sistemine izinsiz girişlerin tespit edilmesidir. Ortaya atılmış olan ilk fikirdeki saldırı tespit sistemi tasarımı Şekil 2.1'de verilmiştir.



Şekil 2.1. Basit saldırı tespit sistemi tasarımı

Dorothy Denning [9], tarafından geliştiren ilk STS "Uzman Saldırı Tespit Sistemi" olarak adlandırılmıştır ve kötü amaçlı etkinliği tespit etmek için kural tabanlı çalışan bir uzman sistem olarak tasarlanmıştır.

1989 yılında Haystack Laboratuvarları tarafından Stalker isimli bir saldırı tespit sistemi geliştirilmiştir. Bu STS, denetim verilerini manuel ve otomatik denetleme imkânı bulunan, çok güçlü arama yetenekleri olan, patern karşılaştırması yöntemiyle çalışan sunucu tabanlı bir saldırı tespit sistemi olarak ortaya çıkmıştır. 1990 yılının başarında Stalker ilk ticari saldırı tespit sistemi olarak piyasaya sürülmüştür [10].

Bu tarihten sonra ticari yönünün de gelişmesi ile STS'nin önemi daha çok kavranmış olup araştırma geliştirme faaliyetleri için daha büyük bütçeler ayrılmıştır. Ayrıca akademik anlamda da yapılan çalışmalarda katlanarak artmıştır.

2010 yılında, G. Schaffrath ve arkadaşları tarafından [11], akış tabanlı izinsiz giriş tespiti alanında araştırmalar yapılmıştır. Bu araştırmalar, akış tabanlı saldırı tespitinin neden gerekli olduğu konusunu irdelemektedir. Akış kavramı açıklanır ve ilgili standartlar belirlenir. Araştırmada saldırıların ve savunma tekniklerinin nasıl sınıflandırılabileceği ve akış tabanlı tekniklerin Tarama, Solucan, Botnet ve Hizmet Aksatma saldırılarını tespit etmek için nasıl kullanılabileceği gösterilmektedir.

2012 yılında P. Prasenna ve arkadaşları tarafından [12], geleneksel ağ güvenliğinde, saldırı tespit sisteminin sadece matematiksel algoritmalara ve etkisiz karşı önlemlere dayandığı öne sürülmüştür. Ancak bu matematiksel algoritma ve etkisiz karşı önlemlerin uygulamaya dökülmesi zor olduğundan yazarlar, çok sayıda kural üretmek yerine Genetik Ağ Programlama (GAP) gibi evrim optimizasyon tekniklerinin kullanılabileceğini bir yöntem öne sürmüştür. GAP, yönlendirilmiş grafiğe dayanmaktadır. Veri madenciliği tabanlı bir STS'yi gerçek zamanlı bir ortamda kullanmaya ilişkin güvenlik sorunlarına odaklanılmıştır.

Rocky K. ve C. Chang tarafından, 2002 yılında Dağıtılmış Hizmet Aksatma Saldırıları için saldırı tespit yaklaşımları önerilmiştir [13]. Bu Dağıtılmış Hizmet Aksatma Saldırılarını tespit yöntemlerinde, mevcut savunma mekanizmalarının sistematik olarak gözden geçirilmesi ve değerlendirilmesi ele alınmıştır. Yazarlar, saldırı kurbana ulaşmadan çok önce, bir savunma yapılmasının nasıl kurgulanabileceği ve gerçek hayata geçirilebileceğini değerlendirmişlerdir. Sonuç olarak yapılan saldırıya karşı üç savunma hattının olması gerektiği ve bunların;

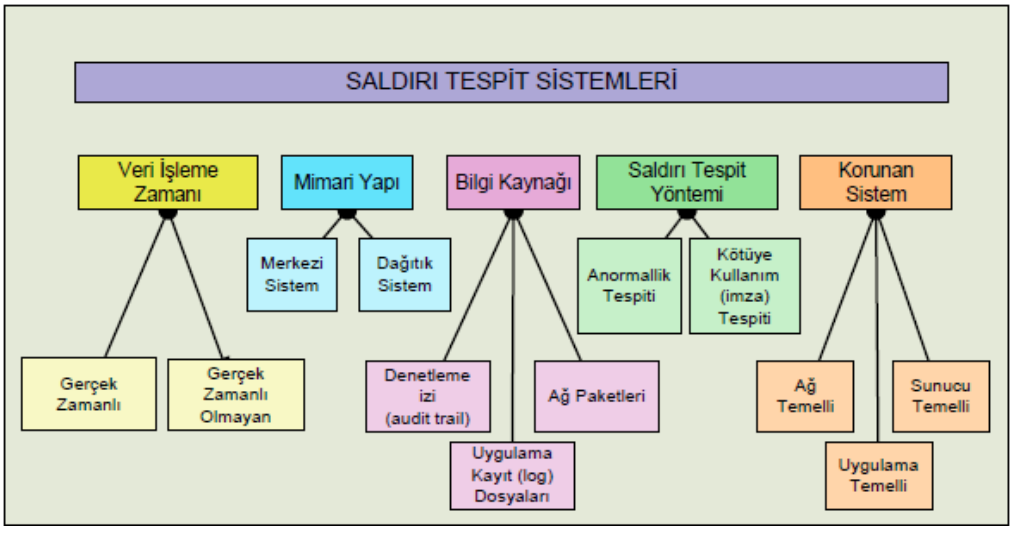
- Saldırı önleme (saldırıdan önce),
- Saldırı algılama ve filtreleme (saldırı sırasında),
- Saldırı kaynağı izleme ve tanımlama (saldırıdan sonra),

şeklinde birbirleri ile uyumlu çalışan bir savunma sistemi olmasının uygun olacağını göstermişlerdir.

Munivara Prasad tarafından, 2011 yılında bir konakçı veya ağa dayanan çeşitli anomali tabanlı saldırı tespit teknikleri önerilmiştir [14]. Davranışsal modelle ilgili işleme türüne bağlı olarak birçok farklı teknik kullanılmıştır. Profil oluşturma ve kullanıcı davranışları, bir bilgisayar sistemine yapılan saldırıları tespit etmek için etkili bir yaklaşım olarak değerlendirilmiştir. Anomali saldırı tespitinde, program ve kullanıcı davranışlarını belirlemek için negatif olmayan matris çarpanlarına ayırma kullanılarak yeni bir model tabanlı yöntem ele alınmıştır. Bu yöntemde, bilgi kaynağı olarak sistem çağrıları dizilerinden ve UNIX komutlarından elde edilen denetim veri akışları kullanılmıştır. Denetim verileri sabit uzunlukta bölümlere ayrılmıştır. Program ve kullanıcı davranışları, verilerin her bir bölümüne gömülü, ayrı sistem çağrıları veya komutların frekansları ile ölçülmüştür. Ayrıca normal davranışlarla ilişkili denetim verisi bloklarından özellikleri çıkarmak için negatif olmayan matris çarpanlarına ayırma uygulanmıştır.

2.1. STS'lerin Sınıflandırılması

Saldırı Tespit Sistemleri ile ilgili farklı kaynaklarda değişik sınıflandırmalar yapılmış olmasına rağmen Saldırı Tespit Sistemlerini Şekil 2.2'de gösterildiği şekilde sınıflandırmak mümkündür [15]. Diğer kaynaklarda daha fazla veya az bir sınıflandırma yapılmasına rağmen en doğru ve optimum sınıflandırmanın bu olduğu değerlendirilmiştir. Daha fazla olması birbirlerine benzeyen ve tekrar eden sınıflara, daha az olması ise çok genel sınıflar oluşmasına neden olacaktır.



Şekil 2.2. Saldırı tespit sistemi türleri

2.1.1. Saldırı tespit yöntemlerine göre

Anormallik tespiti: Ağın normal akışını tespit ederek bu akışın dışında oluşan herhangi bir olayı saldırı olduğunu değerlendiren yöntemdir. Yöntem kullanıcı grubu veya her bir kullanıcı için oluşturulan profillere dayanmaktadır. Saldırı Tespit Sistemi bu profillerin olağan aktivitelerini taban çizelgesi (baseline) kullanarak belirler. Herhangi bir profil bu taban çizelgesinden uzak bir aktivite oluşturursa alarm tetiklenir.

Avantajları;

- İç ağda oluşan saldırıları tespit etmede çok etkili,
- Saldırganın taban çizelgesinde oluşan olağan profili bilmesinin zor olduğundan hangi saldırıyı alarm tetiklenmeden yapabileceğini bilmemesi,
- Profil belirlendikten sonra yapılacak yeni saldırıların tespitinin kolay olmasıdır [16].

Dezavantajları ise;

- Profil oluşurken yapılacak saldırıların olağan trafik olarak değerlendirilmesi,
- Profil oluşması için ağın belli bir süre dinlenilmesi ve değerlendirilmesi,
- Saldırının normal trafik akışına uygun olması durumunda alarm tetiklenmemesi,
- Kullanıcı hesabı yönetiminin zor ve zaman alan bir süreç oluşu olarak sıralanabilir [16].

İmza temelli: İç ağda oluşabilecek atakları belirlemede kullanılmakta olan bir yöntemdir. Bu yöntemde ağda oluşabilecek zararlı akışlar daha önceden yapılmış atakların imzalarından oluşan veri tabanı ile karşılaştırılır. Eğer uyumlu olan trafik var ise alarm tetiklenir.

Avantajları;

- İmza veri tabanının daha önceden tespit edilmiş olan bütün atakları içeren bir imza veri tabanın olması,
- Kurulum tamamlandıktan sonra tespit işleminin başlaması,
- İmza veri tabanına yeni atakların eklenebilmesi,
- Sistemin kullanım ve yönetiminin kolay olmasıdır [17].

Dezavantajları ise;

- Saldırının tespit edilebilmesi için saldırı imzasının veri tabanında olması zorunluluğu,
- Ortaya çıkan her saldırı imzasının veri tabanına eklenmesi, veri tabanın güncel olması zorunluluğu

- Veri tabanının optimum seviyede tutulması zorunluğu, veri tabanı çok büyük olursa yanlış uyarı (false alarm) alma olasılığı artacak, veri tabanı çok küçük olursa da saldırı tespiti yapılamayacaktır [17].

2.1.2. Mimari yapılarına göre

Merkezi sistem: Bu yöntemde sistem ağdaki router, sunucu ve ağ anahtarı olabilecek bir ana elemana yüklüdür. Tüm ağ trafiği, tek bir düğümde bulunan saldırı tespit sistemi tarafından kontrol edilerek giriş-çıkış yapar.

Avantajları;

- Tek bir saldırı tespit sistemi geniş bir alt ağı (subnet) izleyebilir,
- Giriş- çıkış verilerini izleyen, sistem üzerindeki etkisi çok az olan pasif bir yöntemdir [18].

Dezavantajı ise;

- Yoğun bir ağda tüm paketleri denetlemek zordur [18].

Dağıtık sistem: Bu yöntemde sistem birçok düğüme veya tüm düğümlere ayrılmış olarak kurulur ve bu düğümlere ajan adı verilir. Bu ajanlar düğüm üzerinden geçen trafiği izleyerek uygun sonuçlar üretir. Bu sonuçlar daha merkeze gönderilerek değerlendirme yapılmaktadır. Bu yönetim sistemi ajanlar ile koordineli olarak çalışmakta ve uygun paketler için alarm üreterek ağ üzerinde yayınlamaktadır.

Avantajı;

- Tüm paketleri tek bir ana sunucu üzerinde değerlendirme sorunu bu sistemde çözülmüştür [18].

Dezavantajları;

- Her düğüm için yapılandırılması gerektiğinden yönetimi zordur,
- Ajanları kendi aralarında koordine etmesi zordur [18].

2.1.3. Veri kaynağına göre

Denetleme kayıtları: Bu yöntem denetim izlerini (audit trail) değerlendirerek saldırı olasılığının değerlendirilmesine dayanmaktadır. Denetleme izleri kullanıcı veya kullanıcı

gruplarının hangi veriye, ne zaman, nasıl eriştiği, günlük faaliyetlerini nasıl yaptığı gibi bilgileri toplamaktadır. Bu bilgilerin değerlendirilmesi sonucu kullanıcı profilleri oluşturulmakta ve herhangi bir zamanda bu profiller dışındaki bir hareketin saldırı olma olasılığı değerlendirilmektedir.

Uygulama kayıt dosyaları: Bu yöntem uygulama katmanında tutulan log dosyalarındaki bilgileri değerlendirmektedir.

Ağ trafiği: Ağ trafiği sürekli olarak denetlenerek günlükler oluşturulmakta ve saldırılar engellenmeye çalışılmaktadır. Ağ paketlerinin incelenmesi ile oluşturulan bu günlükler içerisindeki bilgiler değerlendirilir ve saldırı olasılığı hesaplanır. Bu yöntem en çok hizmet engelleme saldırıları tespitinde işimize yaramaktadır. Bu yöntemin etkili sonuçlar verebilmesi için ağ cihazları ve saldırı tespit cihazlarının çok iyi bir değerlendirme ile yerleştirilmesi gerekmektedir.

2.1.4. Veri işleme zamanına göre

Gerçek zamanlı: Bu yöntemde sistem çevrimiçi (online) çalışır, yani bu sistem anormal etkinlikleri tespit etmek için ağdan paketler toplar. Bu sistemin performansı gelen paketlerin özellikleri ile kontrol edilmesi için seçilen özelliklerin karşılaştırılmasına bağlıdır. Seçilen özelliklerin sayısı gerçek zamanlı sistemin kaynak tüketimi doğrudan etkilemektedir.

Avantajı;

- Gerçek zamanlı sistemler Saldırı Tespit Sisteminden istenen anormal davranışı algılamaktadır [19].

Dezavantajları;

- Daha fazla kaynak tüketimine neden olmaktadır,
- Sistem darboğazlarına neden olabilmektedir [19].

Veri madenciliği: Bu yöntem çevrimdışı (offline) çalışmaktadır. Yani bu sistemler KDD veri seti gibi kaydedilmiş veri kümelerini işlemektedir. Sistem saldırı hakkında bilgi sağlamakta ve saldırının neden olduğu hasarı onarmaya yardımcı olmaktadır. Gelen saldırılar veri tabanındaki saldırı kimlikleri ile eşleştiğinden aynı tip saldırı ihtimali azalmaktadır.

2.1.5. Korunan sisteme göre

Ağ temelli: Bu yöntem ağ trafiğini izleyerek, geçen trafiği saldırılara karşı analiz eder. Bir saldırının tanımlanması durumunda veya anormal bir davranış algılandığında yöneticiye alarm verir. Bu tür saldırı tespit sistemleri 4 ana saldırı türünü de tespit edebilir.
Avantajları;

- Çapraz Platformlara uygulanabilir,
- Merkezi olarak yönetilir [20].

Dezavantajları ise;

- Eğitim zamanına ihtiyacı vardır,
- Yüksek band genişliği kullanmaktadır,
- Başarısızlık oranı yüksektir [20].

Sunucu temelli: Sistemin veya tek bir bilgisayarın güvenliğini içeriden veya dışarıdan gelen saldırılara karşı izleyen bilgisayarlara özgü bir yöntemdir. İç saldırılar herhangi bir programın hangi kaynağa eriştiği ve herhangi bir güvenlik zafiyeti oluşturup oluşturmadığının tespit edildiği durumu ifade etmektedir. Word programının bir anda sistem parola veri tabanına erişmeye ve onu değiştirmeye çalışması bu duruma örnektir. Dış saldırılar ise sisteme gelen ve giden paketleri denetleyerek, analiz edilmesi durumudur. Bu yöntem her faaliyeti kaydeder ve yetkili makama iletir.
Avantajları;

- Çok yönlüdür,
- Yerel Alan Ağını (LAN) koruyabilir,
- Ağ Temelli Saldırı Tespit Sistemine göre daha az eğitim gerektirir,
- Bant genişliğine ihtiyaç duymaz
- Kayıt defteri (Registry) taraması yapar [21].

Dezavantajları ise;

- Bir olayı, saldırı göstergesi olarak tanımlanmasını beklemek zorunda olan, proaktif olarak önleyemeyen pasif bir sistemdir,
- Veri toplama her bir ana bilgisayar için ayrı ayrı gerçekleşmektedir,
- Günlüğü kaydetme ve raporlama aktivitesi ağa fazladan yük getirecektir,

- Saldırganlar saldırı anında bu tür saldırı tespit sistemlerini devre dışı bırakarak işlem süresi, depolama, hafıza ve diğer sistem kaynaklarını hiçbir program ile paylaşmadan kullanabilirler [21].

2.2. Saldırı Tespit Aşamaları

Saldırı tespit sisteminin çalışması, bilgisayar sisteminin kötü amaçlı yazılım, casus yazılım, spam ve çok daha fazlası gibi tehlikeli tehditlerden korunmasını önlemek için kullanılan diğer programlarla oldukça benzerdir. Saldırı tespit sisteminin işi, sorun hakkındaki bilgilerin kaydedilmesinden başlar ve tehdidin oluşumunu ve doğasını kontrol eder. Sistem sorunu izlediğinde ve bu konuda veri topladığında, bu bilgileri saldırı tespit sisteminin yönetim bölümüne göndererek sistemi korumak ve sistemi güvenli ellerde tutmak için birkaç önleyici tedbir alır. Saldırı tespit sistemleri önemli olayları izleyerek değişik şekillerde çalışabilirler. Bu önemli olaylar;

- Ağ faaliyetinin ve ağdaki tehdit faaliyetinin izlenmesi,
- Virüsleri, kötü amaçlı yazılımları, casus yazılımları ve farklı virüs formlarını tespit etme yeteneği,
- Geri yükleme noktalarını bulabilmesi,
- Farklı ağ programlarının, kimliği doğrulanmamış ve yetkisiz kullanımını gözlemleyerek çalışabilmesidir.

Yani, saldırı tespit sisteminin tüm çalışması, bu tür ağ olaylarının incelenmesine dayanmaktadır.

İdeal bir saldırı tespit sistemi, hangi mekanizmaya bağlı olduğuna bakılmaksızın aşağıdaki sorunları ele almalıdır:

- Sistem, insan gözetimi olmadan sürekli olarak çalışmalıdır. Gözlenen sistemin arka planında çalışmasına izin verecek kadar güvenilir olmalıdır,
- "Kara kutu" olmamalıdır. Yani, iç işleri dışarıdan incelenebilir olmalıdır,
- Sistem çökmesinden sonra hayatta kalması ve bilgi tabanının yeniden başlatma sırasında yeniden oluşturulmaması anlamında hataya dayanıklı olmalıdır,
- Yıkılmaya direnmelidir. Sistem, bozulmadığından emin olmak için kendini izleyebilmelidir,

- Sisteme minimum yük yüklemelidir. Bilgisayarı yavaşlatan bir tarama sistemi kullanılmayacak hale getirebilecektir,
- Normal davranıştan sapmaları gözlemlemelidir,
- Sisteme kolayca uyarlanmalıdır. Her sistemin farklı bir kullanım modeli vardır ve savunma mekanizması bu kalıplara kolayca uyum sağlamalıdır,
- Yeni uygulamalar eklendikçe, zaman içinde sistem davranışını değiştirmekle ilgilenmelidir. Sistem profili zamanla değişecektir,
- Kandırmak zor olmalı.

Yukarıda listelenenlerin tümü, ideal bir Saldırı Tespit Sisteminin sahip olması gereken özelliklerdir. Böylece sistem saldırıları ve sızmalarını savunmak için mükemmel hale gelir.

2.3. Hibrit Saldırı Tespit Sistemleri

Saldırı Tespit Sistemlerinin amacı, uygun kimlik doğrulama işlemi yapılmadan izinsiz giriş yapan işlemlerin tespiti olarak tanımlanabilir. Bu saldırlar farklı teknikler ile yapılmaktadır. Bu tekniklerin güçlü oldukları yönleri kadar güçsüz oldukları yönler de vardır. Hibrit yöntemler bu tekniklerin güçlü oldukları yönleri kullanarak güçsüz yönlerini en aza indirebilmektedir.

Hai ve arkadaşları tarafından [22] sensör ağları için bir saldırı tespit sistemi önerilmiştir. Bu saldırı tespit sisteminde şema hem anomali modeline hem de yanlış kullanım tekniklerine dayalı izinsiz giriş çerçevesi sağlamak ve hiyerarşik bir ağ oluşturmak için tasarlanmış olup küme tabanlı protokolden yararlanılmıştır. Şemalarında STS'yi, yerel ve global aracı olmak üzere iki algılama modülünden oluşturmuşlardır. Yazarlar, modellerini iki ajan arasındaki iş birliğini daha doğru bir şekilde tespit etmek için uygulamışlardır. Bununla birlikte, bu şemanın dezavantajı, imza belleğindeki keskin artıştır, bu da düğüm belleğinin aşırı yüklenmesine yol açabilmektedir.

2007 yılında Kai Hwang tarafından [23], dağıtılmış hizmet aksatma saldırısı için kullanılan sel saldırılarını trafik akışı seviyesindeyken tespit edebilmek amacıyla yeni bir dağıtık yaklaşım sunulmuştur. Bu yaklaşım Kaynak Sonu Savunma Sistemi olarak adlandırılmıştır. Bu sistemin, internet servis sağlayıcıları (İSS) tarafından işletilen çekirdek

ağlar üzerinde verimli olduğu gösterilmiştir. Kaynak sonu savunmasının, sel saldırılarına karşı tam bir çözüm olmadığı, bu savunmayı dağıtık olarak kullanmayan ağların yine de başarılı saldırılar karşısında savunmasız kalabileceği ortaya konulmuştur. Yine de kaynak sonu savunma sistemi saldırı trafiği ve normal bağlantıların birbirinden ayrılması için gerekli olduğu da belirtilmiştir. Bu durum, kaynak sonu savunmasını, tüm hizmet aksatma saldırıları için çözümünün temel yapı taşlarından biri ve internet güvenliğini teşvik etmek için ise gerekli hale getirmiştir.

Patel ve arkadaşları tarafından [24], Saldırı Tespit Sistemi için otonom hesaplama, bulanık mantık, ontoloji ve risk yönetimi kavramlarını birleştiren bir model oluşturulması gündeme getirilmiştir. Ayrıca araştırmacılar tarafından bulut ortamı için Saldırı Tespit Sistemi özelliklerinin de kullanılmasının uygun olacağına değinilmiştir. Böyle bir teklif için yöntem veya algoritma yazar tarafından önerilmemiştir. Önerilen konseptin uygulanması makalenin gelecekteki kapsamıdır.

Parag ve arkadaşları tarafından [25], ağın dışında kalan ve buluta gelen tüm istekleri kontrol eden donanıma sahip bir Saldırı Tespit Sistemi önerilmiş ve uygulanmıştır. Araştırmacılar bu çalışmalarında Hatalı İzinsiz Giriş tespit yöntemini kullanmışlardır. Konuşlandırılan sistem Ağ İzinsiz Giriş Tespit Sistemi olarak adlandırılmıştır. Sistemin sahip olduğu donanım bileşeni ağın dışında olduğundan, saldırganların donanıma saldırması kolay hale gelmiştir. Bu da kurgulanan sistemin dezavantajı olarak belirtilmiştir.

Derpen ve arkadaşları tarafından hem anomali hem de yanlış kullanım tespiti kullanan akıllı bir hibrit saldırı tespit sistemi önerilmiştir. Bu çalışmada altı özellik manuel olarak seçilmiş, anomali tespiti için Ön Düzenleyici Haritalar (SOM) yapısı, imza tespiti için J48 Karar Ağacı kullanılmıştır. Sonuçları kural tabanlı bir karar destek sistemi ile birleştirilmiştir [26].

Koshal ve Bag tarafından, C4.5 Karar Ağacı ve Destek Vektör Makinesi (DVM) algoritmalarını birleştiren ağ tabanlı bir Saldırı Tespit Sistemi önerilmiştir. Bu iki algoritmanın avantajlarını birleştirerek daha iyi bir performans ve düşük yanlış pozitif oranı hedeflenmiş, korelasyon bazlı özellik seçimi ise filtre olarak uygulanmıştır. İlk olarak işlenmiş eğitim iki farklı sınıfa ayırılmış, daha sonra karar ağaçlarından gelen tahminler destek vektör makine bloğundan geçirilmiştir. Son adım olarak sadece karar destek

makinesinden gelen ve hibrit yapıdan gelen sonuçları birleştirilmiştir. [27].

2010 yılında Gang Wang ve arkadaşları tarafından [28], Yapay Sinir Ağlarının bazı geleneksel yöntemlerle karşılaştırıldığında önemli ölçüde gelişmiş STS performansı sağlayabileceği sonucuna varılmıştır. Yüksek doğruluk oranı ve daha az yanlış alarm oranı elde etmek açısından STS'nin performansını arttırmak için Bulanık Kümeleme ve Yapay Sinir Ağına dayanan yeni bir yaklaşım önerilmiştir. Çalışma prensibi olarak öncelikle farklı eğitim alt kümeleri oluşturmak için bulanık kümeleme uygulanması ve daha sonra oluşturulan bu farklı eğitim alt kümesinde farklı temel model formüle etmek için farklı YSA modelleri kullanılması denenmiştir. Sonuç olarak ise bu sonuçları birleştirmek için bulanık toplama modülü kullanılmıştır.

Mostaque Md. Morshedur Hassan tarafından [29], Genetik algoritma ve bulanık mantığın bir arada ve hibrit olarak kullanıldığı bir sistem tasarımı yapılmış ve denenmiştir. Önermiş olduğu STS, yeni izinsiz girişler tespit edildikçe yeni kurallar oluşturabilmekte ve kendini güncelleyebilmektedir. Ancak önerilen bu STS'de iki sorun göze çarpmaktadır. Birincisi yanlış alarm oranının fazlalığı, ikincisi ise büyük boyutlu veriler için hazırlanması gereken kuralların zorluğudur.

Wathiq Laftah Al-Yaseen ve arkadaşları tarafından [30], K-Means ve Karar Destek Makineleri hibrit olarak kullanıldığı bir yaklaşım denenmiştir. Yapılan çalışma sonucunda yüksek başarı oranı ve az sayıda yanlış alarm oranı elde edilmiştir. Ancak bilinmeyen saldırıların tespit edilmesindeki başarı oranı çok yüksek seviyelerde değildir. Bu çalışma oldukça başarılı sonuçlar vermiş olmasına rağmen yazarlar tarafından da dile getirilen ve cevaplanması gereken çok fazla soru vardır.

Shadi Aljawarneh ve arkadaşları tarafından [31], J48, Meta Pagging, Rastgele Ağaç, Karar Ağacı, AdaBoostM1, Karar Kütüğü and Naive Bayes yöntemlerinin birlikte kullanıldığı bir sistem tasarımı yapılmıştır. Çalışma sonucunda Hizmet Aksatma Saldırılarının tespitindeki başarı oranı çok yüksek seviyelerde olmuştur. Ayrıca saldırı olmayan paketlerin saldırı olarak tespiti ve saldırıların normal olarak tespit edilmesi gibi yanlış tespitler konusunda da geliştirilme yapılmış olup yanlış oranı oldukça düşürülmüştür.

Farid ve arkadaşları tarafından [32], Naive Bayes ve Karar Ağacının hibrit kullanıldığı bir

STS çalışması yapılmıştır. Çalışmada ayrıca KDD’99 veri seti içerisindeki niteliklerin azaltılarak yeni veri setleri oluşturulmuş ve bu veri setleri de denenmiştir. Genel olarak çok yüksek başarı oranı elde edilmiştir. Zaten çok yüksek başarı oranı elde edildiğinden nitelik azaltımının etkisi tam olarak görülememiştir. Çalışmadaki tek sorun ise yanlış alarmların fazlalığıdır.

2011 yılında Z. Muda ve arkadaşları [33], tüm saldırı tiplerinin doğru şekilde tespit edilemediği mevcut anomali tespiti sorununu tartışmışlardır. Bu sorunun üstesinden gelmek için, K Ortalamalar kümelenmesi ve Naïve Bayes sınıflandırma kombinasyonu ile karma bir öğrenme yaklaşımı önermişlerdir. Önerilen yaklaşımda, sınıflandırma amacıyla bir sınıflandırıcı uygulamadan önce tüm veriler ilgili gruba ayrıştırılmıştır. KDD'99 veri kümesini kullanarak önerilen yaklaşımın performansını değerlendirmek için bir deney yapmışlardır. Sonuç olarak, önerilen yaklaşımın doğruluk, algılama oranı ve makul yanlış alarm oranı açısından daha iyi performans gösterdiğini tespit etmişlerdir.

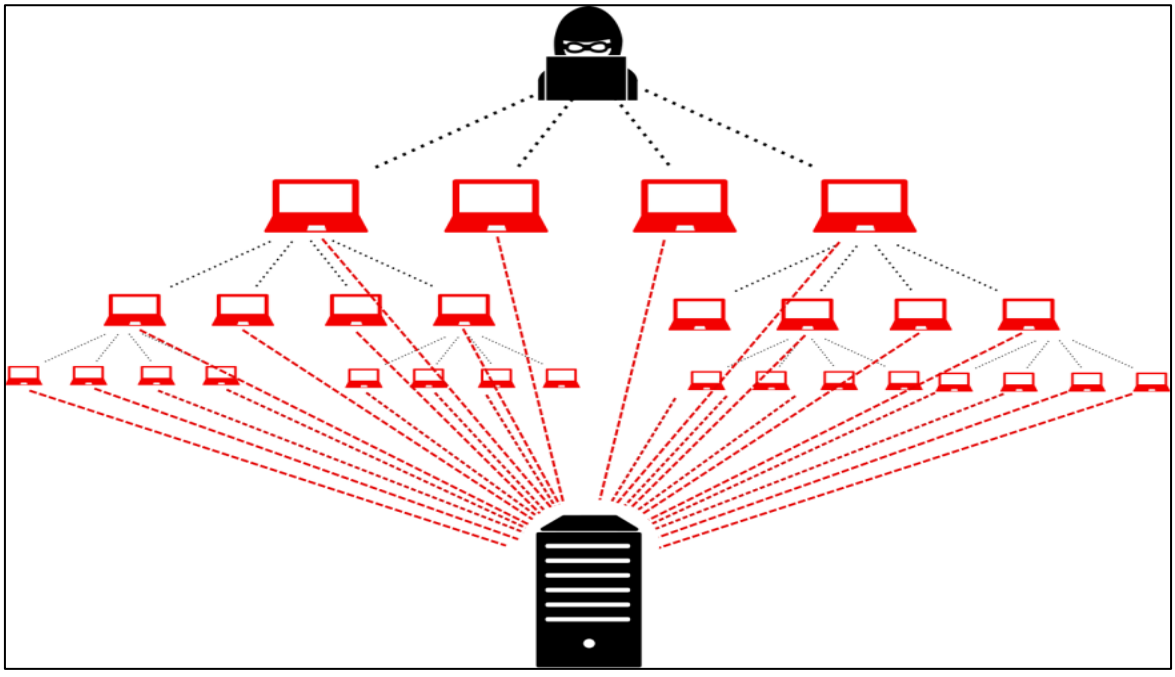
Soodeh Hosseini ve arkadaşları tarafından [34], Genetik Algoritma, Karar Destek Makineleri, Parçacık Sürü Optimizasyonu ve Yapay Sinir Ağlarının hibrit şekilde kullanıldığı bir STS önerilmiştir. Yapılan çalışmada kullanılan makine öğrenmesi teknikleri değişik ikili kombinasyonlar halinde denenerek hibrit şekilde kullanılmıştır. Neredeyse tüm kombinasyonlar denenmiş olmasına rağmen en başarılı sonucun 4 makine öğrenmesi tekniğinin birlikte kullanıldığı yöntemin verildiği tespit edilmiştir. Bu çalışmada tüm saldırı türlerinin tespitine çalışılmış ve en başarı sonuçlar Hizmet Aksatma Saldırıları tespitinde alınmıştır.

3. SALDIRI TİPLERİ VE ÖZELLİKLERİ

3.1. Hizmet Aksattırma Saldırıları (Denail Of Service-DOS)

Hizmet aksatma saldırıları hedef alınan sunucu veya kaynağa kullanıcıların erişimini engellemek, kaynağı kullanılamaz hale getirmek için yapılan bir saldırı türüdür. Her ağın kaldırabileceği bir yük miktarı bulunmaktadır. Saldırganlar bu yük miktarının sınırlarını zorladıklarından sistem normal kullanıcılara hizmet verememektedir.

Saldırgan bu saldırıyı tek bir istemciden yapabileceği gibi birden çok istemciden de yapabilmektedir. Bu tür birden çok istemciden yapılan saldırılar Dağıtık Hizmet Aksatma Saldırısı (Distributed Denial of Service-DDoS) olarak adlandırılmaktadır. Bu tür saldırıları yapabilmek için Şekil 3.1'de görüldüğü üzere öncelikle saldırgan zombi olarak isimlendirilen başka makineleri ele geçirmek zorundadır. Sonrasında master olarak isimlendirilen makinesi ile bu zombi istemciler üzerinden hedef alınan sunucu veya kaynağa saldırmaktadır. Bu tür saldırılarda saldırıyı tespit etmek zordur. Çünkü saldırgan dünyanın her tarafından zombi istemci ele geçirmiş olabilir ve hedefe saldırıyor olabilir. Hedef sunucu veya kaynak birçok istemciden gelen paket ve istekleri normal olarak algılayabilmektedir.



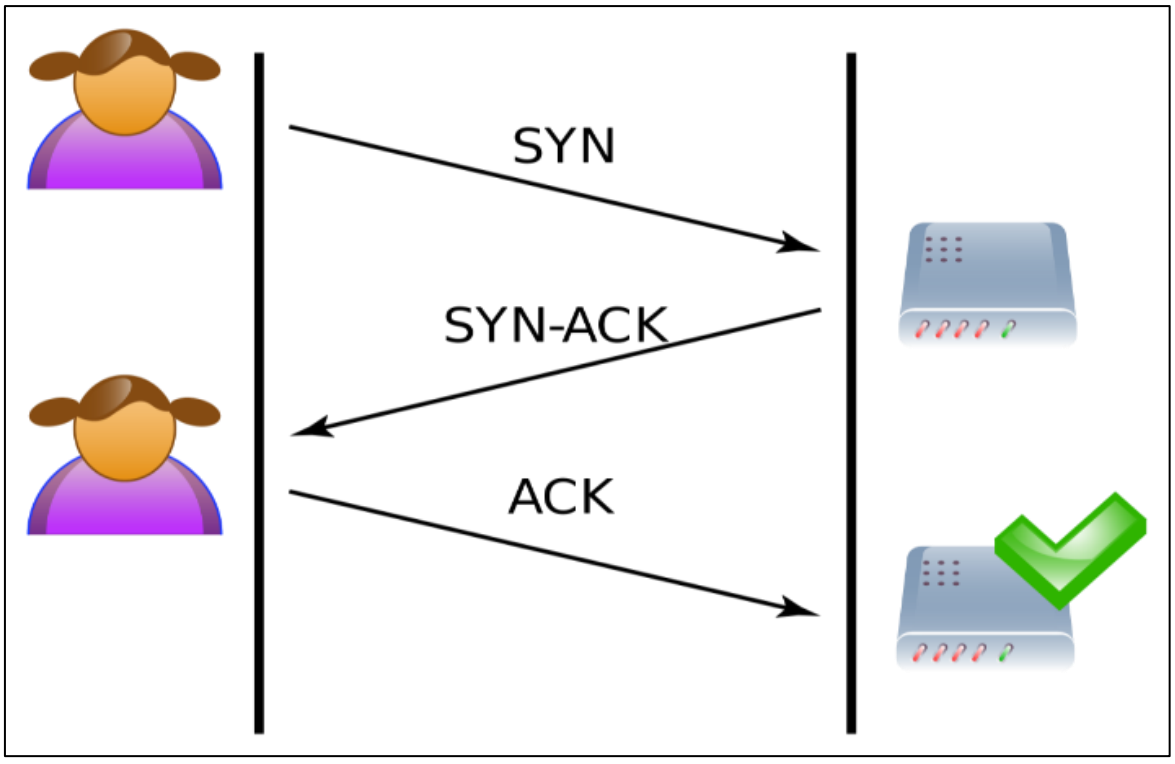
Şekil 3.1. Basit hizmet aksatma saldırısı

Çoğu hizmet aksatma saldırısı TCP/IP protokolündeki zayıflıkları kullanmaktadır. Bu tip saldırılara Syn Seli, ICMP Seli, UDP Seli, Smurf, Ölümcül Ping, Gözyaşı Saldırısı vb. örnek olarak verilebilir.

3.1.1. SYN seli saldırısı

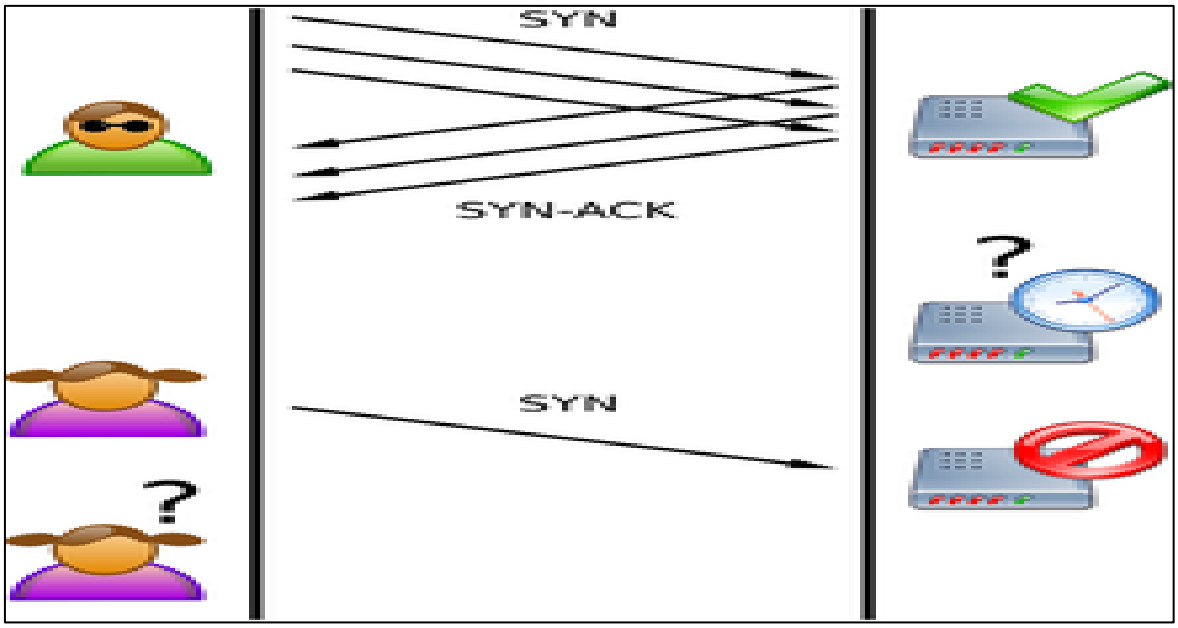
İstemci makine bir sunucu ile bağlantı kurmak isteği zaman her iki makine de bir dizi iletiyi sırasıyla birbirlerine gönderirler. Bu işlem Üçlü El Sıkışma (Three Way Handshaking) olarak adlandırılır. Şekil 3.2'de görüldüğü gibi;

- Öncelikle istemci SYN (senkronizasyon) mesajını sunucuya gönderir,
- Daha sonra sunucu, SYN mesajının onayı olarak SYN-ACK (acknowledgment) mesajını istemciye gönderir,
- Son olarak istemci bağlantının gerçekleşmesini sağlamak amacıyla ACK mesajı ile cevap verir.



Şekil 3.2. Üçlü el sıkışma

Böylece sunucu ve istemci arasındaki bağlantı açılarak veri alışverişi gerçekleşmeye başlar.



Şekil 3.3. SYN seli saldırısı

SYN Seli atağı yapmak isteyen bir saldırgan Şekil 3.3'teki gibi SYN-ACK mesaj alışveriş mantığını kullanarak saldırı gerçekleştirir. Saldırgan sahte kaynak IP adresleri bulunan bir dizi SYN mesajı gönderir. Sunucu ise bu mesajlara SYN-ACK mesajı ile yanıt vererek, istemciden gelecek olan ACK mesajını bir süre beklemeye başlar. Ancak saldırgan sahte kaynak IP adresleri kullandığından herhangi bir ACK mesajı iadesi yapılamaz. Böylece, kısmen açılan bu bağlantılar bağlantı kuyruğunu ve bellek tamponlarını doldurur ve gerçek kullanıcıların bu sunucuyu kullanmasını engeller [35].

3.1.2. UDP seli saldırısı

Bu, SYN seli saldırısından sonra en popüler ikinci hizmet aksatma saldırısı yöntemidir. UDP bağlantılarında üçlü el sıkışma mekanizması bulunmamaktadır. Bu açıdan üçlü el sıkışma mekanizmasını kullanan ataklara karşı güvenlidir. UDP Seli saldırılarındaki temel fikir, paketlere yanıt verdiği bilinen UDP hizmetlerinden yararlanmaktır. Bilgisayar korsanı, sahte UDP paketleri gönderdiği yayın adreslerinin bir listesini elinde bulundurmaktadır. Çoğu durumda paketler, hedef makinelerde 7'inci porta (ECHO portu) yönlendirilir. Ancak, kötü niyetli kullanıcının chargen portuna paketleri göndermesi gereken saldırılar vardır. "Chargen" portu, test amacıyla kullanılan ve aldığı her paket için bir dizi karakter üreten bir porttur. Bir ana makinenin chargen hizmetini aynı veya başka bir makinedeki "ECHO" portuna bağlayarak, etkilenen tüm makineler etkili bir şekilde

hizmet dışı bırakılabilir, çünkü çok fazla sayıda paket üretilecektir. Ayrıca, iki veya daha fazla ana bilgisayar bu kadar bağlıysa, araya giren ağ da tıkanabilir ve trafiği bu ağdan geçen tüm ana bilgisayarlara paketler iletilemeyebilir [36].

3.1.3. Smurf saldırısı

Bu saldırı ping (packet internet gopher) mantığı ile çalışmaktadır. Ping aslında bir sunucu veya istemcinin çevrim içi olup olmadığını kontrol etmek amacıyla kullanılmaktadır. Smurf saldırı ise şu şekilde yapılmaktadır;

- Saldırgan bir ping paketi oluşturur. Bu ping paketinin içerisinde bulunan kendi IP'sini kurbanın IP'si ile değiştirerek kendini gizlemiş olur,
- Bu ping paketleri ağa hızlı bir şekilde yayılarak, ağ meşgul edilir, ayrıca yayılan paketlerdeki kaynak IP'si de değiştirildiğinden paketler tekrar kurban sunucu veya kaynağa yönlenir,
- Kendisine gelen her ping paketine karşılık vermeye çalışan kurbanın göndermiş olduğu cevaplar da kendisine geldiğinden sonsuz bir döngü oluşur. Böylece kurban isteklere cevap veremez ve kullanılamaz.

Bu atak hedef sunucuyu kullanılamaz hale getirmek için yapıldığından ve ping paketleri ile gerçekleştiğinden çoğu sunucu ping paketlerine cevap vermemektedir [37].

3.1.4. Gözyaşı saldırısı

Paketlerin ağ üzerinden iletilmesi paketlerin küçük parçalara ayrılması şeklinde yapılmaktadır. Parçalanmış paketler hedefe gönderilir ve hedef parçaları tekrar bir araya getirerek paket bütününü elde eder. Paketlerin yeniden birleştirilebilmesi için parçaların her birinde ofset değerleri bulunmaktadır. Gözyaşı saldırısında her parçada olan ofset değerleri değiştirilerek paketlerin yeniden birleştirilmemesi veya birleştirilirken hata alınması sağlanmaktadır. Paketleri birleştiremeyen ve hata alan hedef sunucu veya kaynak sağlıklı şekilde çalışma kabiliyetini kaybetmektedir [38].

3.1.5. Ölümcül ping saldırısı

Saldırganın hedef sunucu veya kaynağa büyük boyutlu ping paketleri göndermesi ile yapılmaktadır. Normal şartlar altında gönderilebilecek en büyük paket boyutu 65.535

byte'dır. Saldırgan bu paket boyutundan daha büyük boyutta paket göndermektedir. Sistem bu paketi nasıl parçalayacağını bilemediğinden çakışma olmaktadır. Günümüzde hem ağ cihazları hem de işletim sistemleri bu saldırının tedbirini almışlardır [39].

3.2. Yönetici Hesabı Kullanarak Yerel Oturum Açma

Bu saldırı kavramsal olarak Kullanıcı Oturumunun Yönetici Oturumuna Yükseltilmesi Saldırısına benzemektedir ancak nihai hedefi çok farklıdır. Bu tür bir saldırı, saldırganın sistemin açıklarını önceden belirlemesine dayanmaktadır. Saldırgan herhangi bir kullanıcı yetkisi olamamasına rağmen misafir veya başka bir kullanıcı olarak sisteme erişebilmekte ve böylece sistem açıklarını inceleyebilmektedir. Bu saldırının en önemli türleri Dictionary, Ftp-write, imap, xlock, guest, Named, xnsnoop, phf ve Sendmail saldırılarıdır. Günümüzde bu saldırı türü için saldırganların elinde birçok araç olduğu için her sistem yöneticisi kendi sistemini çok iyi incelemeli ve varsa güvenlik açıklarını kapatmalıdır. Kapatılamayan açıklıkların ise misafir ve normal kullanıcılar tarafından görülmesi veya incelenebilmesi engellenmelidir [40].

3.3. Kullanıcı Oturumunun Yönetici Oturumuna Yükseltilmesi

Bu saldırı saldırganın, kullanıcı hesabı ile sisteme giriş yaptıktan sonra sistemde yönetici hesabı ile erişilebilecek bilgilere ulaşması mantığına dayanmaktadır. Saldırgan kendisine veya bir başkasına ait olan kullanıcı yetkisine sahip bir hesap ile sisteme giriş yapar, daha sonrasında yetkisinin olmadığı işlemleri gerçekleştirebilmek için bu kullanıcı hesabının yetkilerini yönetici yetkileri ile donatmaya çalışmaktadır. Günümüzde bu kapsamada en çok yapılan saldırılar Eject, Ffbconfig, Fdformat, Loadmodule, Perl, Ps, Xterm saldırılarıdır [41].

3.4. Bilgi Tarama

Bu saldırı türü sistemin tamamını tarayarak hangi saldırıya açık olduğunun belirlenmesine dayanmaktadır. Saldırgan Sistemin tamamını tarayarak sistemin açıklık ve zayıflıkları belirlemeyi, sistemdeki güvenlik duvarı, saldırı tespit/önleme sistemi ve antivirüs sistemi hakkında bilgi edinmeyi, erişime açık olan portları öğrenmeyi, işletim sistemi hakkında bilgi almayı, sanal veya özel ağ durumunu öğrenmeyi amaçlamaktadır. Bu saldırıyı

gerçekleştirmek için kullanılan araçlar ile siber güvenlik uzmanlarının kullanmakta oldukları yazılımlar aynıdır. Güvenlik uzmanları da tıpkı saldırganlar gibi sistemin açıklarını ve zayıflıklarını tespit etmektedirler. Günümüzde bu kapsamada en çok yapılan bilgi tarama saldırıları Ipsweep, Mscan, Nmap, Saint, Satan saldırılarıdır [42].

4. MAKİNE ÖĞRENMESİ TEKNİKLERİ

Günümüzde teknolojinin gelişmesi ile bilgisayarların kabiliyetleri ve performansları da artış göstermiştir. Daha önceleri tasarlanmış ancak yeterli performans alınamamış makine öğrenmesi teknikleri günümüzde performanslı bir şekilde kullanılabilmektedir. Bu makine öğrenmesi tekniklerinin dışında yeni makine öğrenmesi teknikleri geliştirilmektedir.

Makine öğrenmesi, matematiksel yöntemler ve istatistiksel veriler kullanarak çıkacak sonuçları tahmin etmek olarak tanımlanabilir. Burada genel bir algoritma oluşturulur ve makineye giriş verileri verilir. Makine bu verileri işleyerek nasıl bir yapıda sonuç üretmesi gerektiğini öğrenir. Bu öğrenmenin gerçekleşmesi ile birlikte makine yeni verilerin sonuçları tahmin etmeye hazır hale gelmiş olacaktır.

Makine öğrenmesinde, öğrenme yöntemleri temel olarak iki ana gruba ayrılır. Bunlar gözetimli öğrenme ve gözetimsiz öğrenmedir.

Gözetimli öğrenmede, veri seti içerisindeki verilerin sonuçları da makineye verilerek kendi çıkardığı sonuç ile gerçek sonucu karşılaştırması ve yapmış olduğu hataları düzelterek öğrenme sağlaması hedeflenmektedir. Örneğin; erik, elma ve portakal verilerinden oluşan bir veri setini makineye verdiğimizde makinenin ilk dikkat etmesi gerekenler, büyüklük, renk, içerdiği vitaminler vb. olacaktır. Biz her veri bilgisi için sonuç bilgisinin de erik, elma veya portakal olduğunu belirtmemiz gerekmektedir. Öğrenme sonrasında vereceğimiz verilerin hangi sınıfa ait olduğunu rahatlıkla bulabilecektir.

Gözetimli öğrenmede, veri seti içerindeki girdiler ve çıktıların çeşidi hangi tür algoritma kullanacağımızı da belirlemektedir. Örneğin; yukarıdaki örnekte olduğu gibi bir kategori söz konusu ise Sınıflandırma Algoritmaları, eğer çıkış verisi sayısal değer ise Regresyon Algoritmaları kullanılır.

Gözetimsiz öğrenme, çıktı verileri verilmeden algoritmanın kendi kendine öğrenmesi ve çıkarımlarda bulunmasıdır. Burada algoritma çıkarımlar ve varsayımlarda bulunarak veri içerisindeki örüntü ve düzeni keşfeder.

Gözetimsiz öğrenmede, Kümeleme, Boyut Azaltımı ve Anomali tespiti algoritmaları kullanılmaktadır.

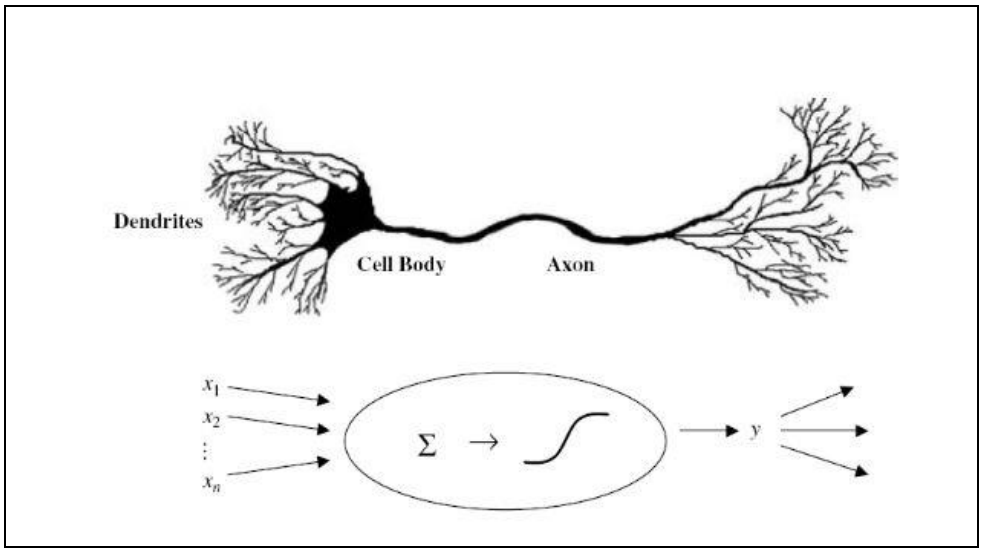
4.1. Klasik Makine Öğrenmesi Teknikleri

Yapay zekâ araştırmalarının nihai hedefi, insan gibi düşünebilen makineler oluşturabilmektir. Ancak günümüz koşullarında bu çok da gerçekçi bir hedef olamamaktadır. Bu yüzden araştırmalar daha gerçekçi bir hedef olan "makinelere belli sorunları çözebilecek çözüm yollarını öğretmek" üzerine yoğunlaşmaktadır. Yazılım ve kodlama ile çözümü çok zor olabilecek sorunlar makine öğrenmesi teknikleri ile çok kolay bir şekilde çözüme kavuşmaktadır. Günümüzde en çok kullanılan makine öğrenmesi teknikleri Yapay Sinir Ağları, Genetik Algoritma, Bulanık Mantık ve Sezgisel Algoritmalardır.

4.1.1. Yapay sinir ağları

Yapay sinir ağları düşünebilen bir varlık olan insanın, beynini örnek alarak, yeni bilgiler edinme, öğrenme, öğrenilen bilgiden sonuçlar çıkarma, yeni bağlantıları keşfedebilme gibi yetenekleri bilgisayar sistemlerine aktarmayı hedeflemektedir. Buradaki nihai hedef insan gibi düşünerek ortaya çıkan problemleri çözebilecek bir yapı geliştirmektir. Günümüzde henüz bu yapılamamış olsa da belli problemlere çözüm olabilecek bilgisayar sistemlerin oluşturulması başarılmıştır.

Yapay sinir ağları oluşturulurken insan beyni dikkate alındığından, insanın biyolojik sinir sistemini incelemek gerekmektedir. İnsan sinir hücresinin örneği ve Yapay Sinir Ağlarına uyarlaması Şekil 4.1'de gösterilmiştir. Burada insan beynindeki bir sinir hücresinin yapısının açıklanması gerekmektedir. Bir sinir hücresinin içerisinde Akson, Sinaps, Dentrit, Nöron ve hücre gövdesi bulunmaktadır. İnsan sinir ağında girdiler elektriksel yük olarak dentritlerle toplanmakta ve hücre gövdesine iletilmekte, hücre gövdesinden ise aksonlar aracılığıyla dışarı gönderilmektedir [43].



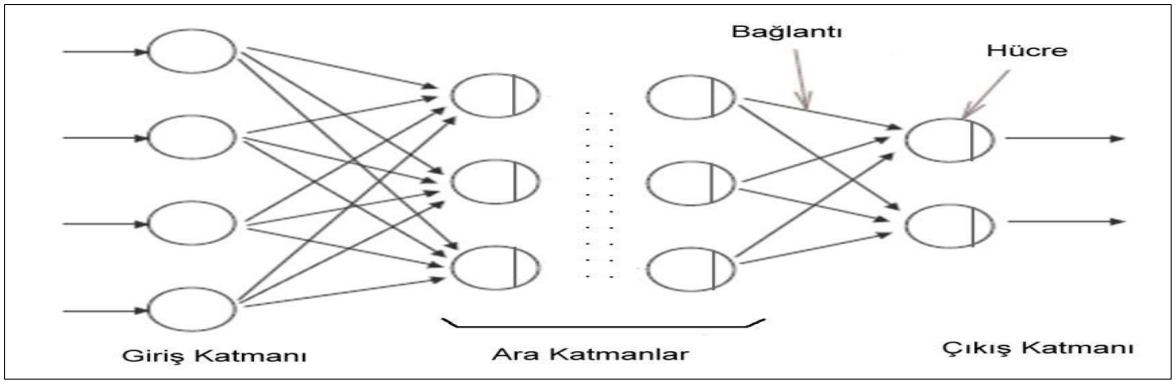
Şekil 4.1. Sinir ağı ve yapay sinir ağı uyarlaması

Çizelge 4.1'de gösterildiği gibi insan sinir ağındaki yapının bir benzeri yapay sinir ağlarında karşımıza çıkmaktadır. İnsan sinir hücresinde bulunan bölümlerin her birinin işlevini yerine getirecek şekilde yapay sinir ağları tasarlanmıştır [43].

Çizelge 4.1. İnsan sinir yapısı ve ysa'nın kısımlarının karşılaştırılması

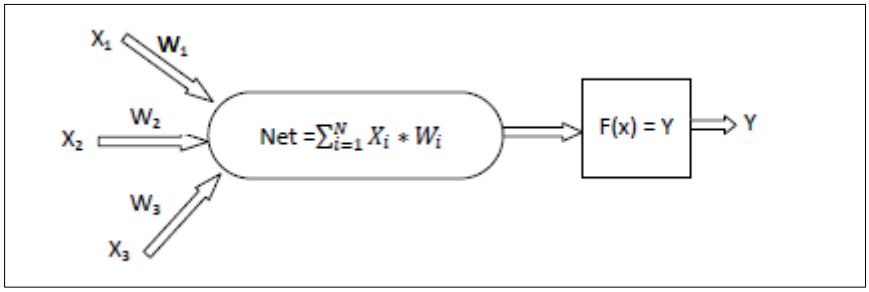
İnsan Sinir Ağı	Yapay Sinir Ağı
Nöron	İşlem Yapan Eleman
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Aksonlar	Çıktılar
Sinapslar	Ağırlıklar

Yapay Sinir Ağlarının yapısı Şekil 4.2'de gösterildiği gibi girdilerin alındığı Girdi Katmanı, sonuçların daha tutarlı ve doğru verilmesi amacıyla kullanılan ara katmanlar ve çıktıların alındığı Çıkış katmanından oluşmaktadır. Buradaki her bir eleman Nöron olarak adlandırılmakta ve nöronlar arasındaki bağlantıların ağırlıkları hesaplanarak sonuca gidilmektedir. Hesaplama işlemi farklı fonksiyonlar ve hesaplama teknikleri kullanılarak yapılmaktadır. Fonksiyon ve hesaplama tekniğinin seçimi tamamen problemin nasıl daha uygun şekilde çözüleceği ile alakalıdır. Farklı problemlerde farklı fonksiyon ve yöntem tercih edilmektedir.



Şekil 4.2. YSA'nın yapısı

Yapay sinir ağlarının matematiksel gösterimi ise Şekil 4.3'te gösterilmiştir. Burada X'ler girdileri, W'lar ağırlıkları, $Net = \sum_{i=1}^{N} x_i * w_i$ Toplam fonksiyonunu, $F(x) = Y$ Aktivasyon fonksiyonunu, çıktıları ise Y ifade etmektedir.



Şekil 4.3. YSA'nın matematiksel gösterimi

YSA'da giriş ve çıkış katmanındaki nöron sayıları, problemin karmaşıklığına göre belirlenir, ancak ara katman sayısı ve ara katmanlarda bulunacak nöron sayısı "deneme yanılma" yapılarak belirlenir [44].

4.1.2. Genetik algoritma

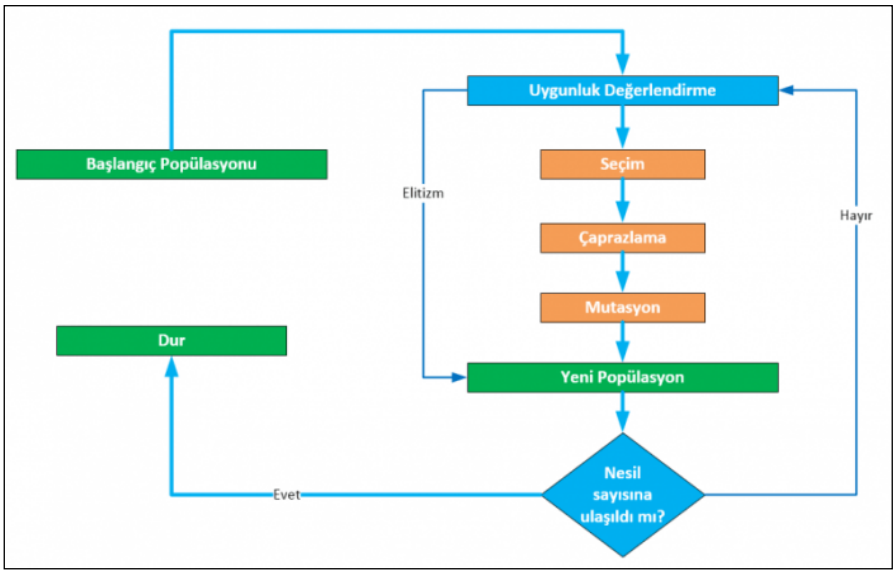
Genetik algoritmanın ilk temelleri John Holland tarafından 1975 yılında atılmıştır. John Holland, Darwin tarafından ortaya atılan evrim yasalarından esinlenerek çözüme gidebilecek en iyi yolun bulunması amacıyla genetik algoritmalardan kitabında bahsetmiştir.

Goldberg tarafından yapılan tanıma göre genetik algoritma rastlantısal arama yöntemleri kullanılarak çözüm üretilmeye çalışılan sezgisel arama yöntemlerinden biridir [45].

Genetik algoritma iyi nesillerin hayatlarına devam ederken kötü nesillerin hayatlarının sonlanması ilkesine dayanmaktadır. Genetik algoritmalar olasılık üzerine dayalı ve sezgisel bir arama yöntemidir. Genetik algoritma optimizasyon amacıyla birçok uygulama alanında kullanılmaktadır [46].

Genetik algoritma, problemin çözümüne çeşitli çözüm yollarını değerlendirerek başlar, uygunluk fonksiyonu, oluşturulan bu çözüm yollarının uygunluğunu değerlendirir, sonrasında yeni seçim, çaprazlama ve son olarak mutasyon işlemleri ile sonuca gitmeye çalışır [47].

Genetik algoritmaların çalışma prensibi Şekil 4.4'te verilmiştir.



Şekil 4.4. Genetik algoritmanın çalışma prensibi

Çok çeşitli alanlarda en iyi çözümün bulunması amacıyla kullanılan genetik algoritmaların işleyiş adımları:

- Arama uzayında olabilecek tüm çözümler hesaplanarak dizi olarak belirlenir.
- Genellikle rastlantısal bir çözüm kümesi tercih edilir ve başlangıç populasyonu olarak bu çözüm kümesi ele alınır.
- Her bir dizinin çözüm kalitesini belli etmek amacıyla uygunluk değeri hesaplanır.
- Diziler gruplar halinde işleme sokularak olasılık değerine göre rastlantısal olarak seçilir ve daha sonra çoğalma işlemine tabi tutulur.

- Yeni bireylerin çözüm kalitesi hesaplanarak uygunluk değerleri belirlenir, daha sonra çaprazlama ve mutasyon işlemleri gerçekleştirilir.
- Daha önce karar verilen kuşak sayısı kadar yukarıdaki işlemler tekrarlanır.
- İterasyon, daha önce karar verilen kuşak sayısına ulaştığında işlem tamamlanır.
- Amaç fonksiyonu dikkate alınarak en uygun dizi belirlenir [48].

4.2. Hibrit Makine Öğrenmesi Teknikleri

4.2.1. Karar tablosu ve naive bayes (KTNB)

Karar Tabloları, Çizelge 4.2'de olduğu gibi Koşul, Koşul Kombinasyonları, Eylem ve Sonuçlar olmak üzere 4 bölümden oluşmaktadır. Bu algoritma problem mantığını modellemek, Karar verme aşamalarını netleştirmek ve kararı etkileyebilecek olan bütün seçenekleri değerlendirmek için kullanılır. Tekniğin çalışma aşamaları ise;

- Bütün koşullar ve eylemler sırası ile yazılır,
- Bu koşul ve eylemlere göre bütün kombinasyonlar değerlendirilerek buna göre sütunlar oluşturulur,
- Yukarıda değerlendirilen bütün kombinasyonlara göre sütunlar doldurulur,
- Tablonun geneli incelenerek ihtiyaç varsa sadeleştirme işlemi yapılır.

Çizelge 4.2. Karar tablosu çizelgesi

KOŞUL	KOŞUL KOMBİNASYONLARI
EYLEM	SONUÇLAR

Navie Bayes ise veri kümeleri üzerinde eğitim ve test verilerini kullanarak çalışmaktadır. Bu algoritma her veri için tüm olasılıkları hesaplamaktadır. Bu olasılıklardan değeri en büyük olan olasılık verinin sınıfı olarak atanmaktadır.

1998 yılında Mark Hall ve Eibe Frank tarafından ortaya atılmış olan Karar Tablosu ve Naive Bayes Yönteminin hibrit olarak kullanılmasıyla bu algoritma oluşturulmuştur. Algoritma, öznitelikleri biri Naïve Bayes tarafından, diğeri de karar tablosu tarafından modellenen iki ayrı alt kümeye ayırmaktadır. Navie Bayes tarafından kullanılacak öznitelikler Karar Tablosu tarafından modellenmektedir. Bu iki modelden üretilen sonuçlar bayes kuralı ile tek bir sonuç altında toplanmaktadır [49].

Karar Tablosu ve Naive Bayes hibrit yöntemi Karar Tablosu sınıflandırıcısı ile aynı şekilde ilerlemektedir. Aramadaki her noktada, biri Karar Tablosu için, diğeri Naive Bayes için olacak şekilde niteliklerin iki ayrık alt kümeye bölünmesiyle ilişkili sonucu değerlendirir. Her adımda, seçilen niteliklerin Naive Bayes tarafından, geri kalanının Karar Tablosu tarafından ve tüm özelliklerin başlangıçta Karar Tablosu tarafından modelleneceği bir ileri seçim yöntemi kullanılır.

Genel tahmini oluşturmak için Karar Tablosu ve Naive Bayes'in olasılık tahminleri birleştirilmelidir. Örneğin; $X^{\perp}$ Karar Tablosunda bir nitelik ve $X\perp$'de Naive Bayes'de bir nitelik olarak varsayıldığında algoritma değerlendirmesi

$$Q(y|X) = \alpha \times Q_{DT}(y|X^{\top}) \times Q_{NB}(y|X^{\perp}) / Q(y) \tag{4.1}$$

olarak hesaplanır. "Eş. 4.1" de ve Karar Tablosu ve Naive Bayes sınıflarından elde edilen olasılık tahminleri, α normalizasyon sabiti ve $Q(_Y)$ ise öncelikli olasılıktır. Tüm olasılıkların değerlendirilmesinde Laplace kullanılmıştır [50].

4.2.2. Lojistik model ağacı (LMA)

Bu algoritma, lojistik regresyon algoritması ile karar ağacı algoritması birleştiren ve hibrit şekilde kullanılmasını sağlayan gözetimli bir sınıflandırma algoritmasıdır. Lojistik Model Ağacı, Karar Ağacı'nın geçmiş versiyonlarına dayanmaktadır. Hibrit olarak kullanılmayan Karar Ağacı'nın yaprakları parçalı bir sabit model oluşturmaktadır. Ancak Lojistik Regresyon ve Karar Ağacı'nın birlikte kullanıldığı bu modelin yapraklarında doğrusal regresyon modeli kullanılarak parçalı bir doğrusal regresyon elde edilir [51]. Lojistik varyasyonda, LogitBoost algoritması ağaçtaki her düğümde bir Lojistik Regresyon modeli

üretmek için kullanılır; ardından düğüm C4.5 kriteri ile bölünür.

Bir Lojistik Model Ağacı, temelde yapraklarda regresyon fonksiyonlarına sahip bir regresyon ağacı gibi yapraklarda lojistik regresyon fonksiyonlarına sahip standart bir karar ağacı yapısından oluşur. Sıradan karar ağaçlarında olduğu gibi, niteliklerden biri üzerinde yapılan bir test her iç düğümle ilişkilendirilir. K değerlerine sahip bir nominal öznitelik için, düğümün k alt düğümleri vardır ve örnekler, öznitelik değerine bağlı olarak k dallarından birine göre sıralanır. Sayısal öznitelikler için, düğümün iki alt düğümü vardır ve sınama, öznitelik değerini bir eşikle karşılaştırmaktan oluşur. Söz konusu öznitelik değeri eşik değerden küçükse, örnek sol dalda, aksi takdirde sağ dalda sıralanır.

Daha açıklayıcı olarak, bir lojistik model ağacı, bir dizi iç veya terminal olmayan düğümden (N) ve bir dizi yapraktan veya terminal düğümünden (T) oluşan bir ağaç yapısından oluşur. S, verilerde bulunan tüm özniteliklerin kapsadığı tüm örnek alanını belirtir. Daha sonra ağaç yapısı, S'nin S_t bölgelerine ayrık bir alt bölümü verir ve her bölge ağaçtaki bir yaprak ile temsil edilir:

$$S = \cup_{t \in T}\, S_t\,, \quad S_t \cap S_{t'} = \emptyset \quad için \quad t \neq t' \tag{4.2}$$

Sıradan karar ağaçlarının aksine, $t \in T$ yaprakları sadece sınıf etiketi yerine ilişkili lojistik regresyon fonksiyonuna sahiptir. Regresyon işlevi f_t, verilerde bulunan tüm özniteliklerin bir $Vt \subseteq V$ alt kümesini dikkate alır (burada nominal özniteliklerin regresyon amacıyla ikili hale getirildiğini varsayarız) ve sınıf üyelik olasılıklarını şu şekilde modeller;

$$\Pr(G = j|X = x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{j} e^{F_k(x)}} \tag{4.3}$$

ise

$$Fj(x) = \alpha_0^j + \sum_{\upsilon \in V_t} \alpha_\upsilon^j . \upsilon \tag{4.4}$$

veya aynı şekilde

$$Fj(x) = \alpha_0^j + \sum_{k-1}^{m} \alpha_{v_k}^j . v_k \quad (4.5)$$

ile ifade edilir. Eğer $\alpha_{v_k}^j = 0$ ise $v_k \notin V_t$ dir. Bu durumda tüm lojistik model ağacının temsil ettiği nokta :

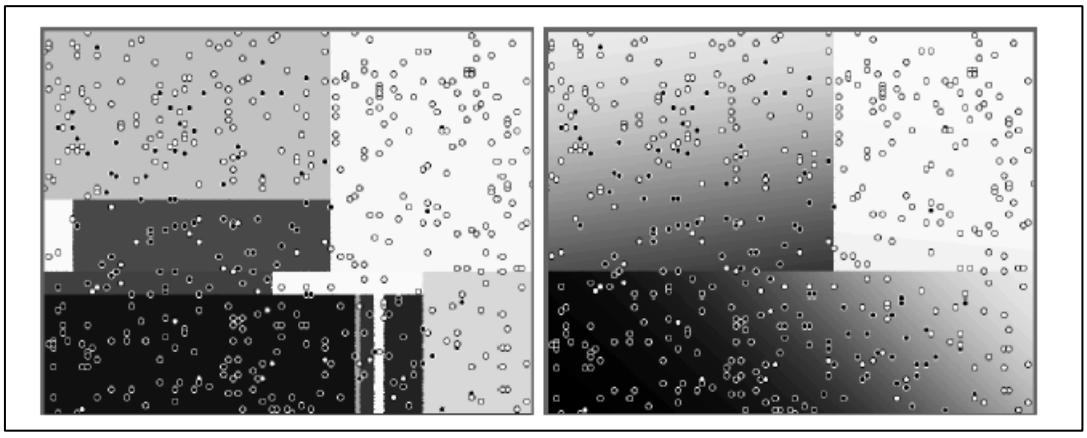
$$f(x) = \sum_{t \in T} f_{t(x)} . I(x \in S_t) \quad (4.6)$$

şeklinde bulunur. Burada $I(x \in S_t)$ değeri eğer $x \in S_t$ ise 1 değilse 0'dır.

Hem bağımsız lojistik regresyonu hem de sıradan karar ağaçları, özel birer lojistik model ağaçları durumlarıdır. Birinci durum köke geri bulan bir lojistik model ağacı, ikincisi ise tüm $t \in T$ için $Vt = \emptyset$ olan bir ağaçtır.

İdeal olarak, basit bir doğrusal modelin en iyi yanlılık dengesini sunduğu küçük veri kümeleri için, lojistik modeli ağacı sadece kök yani tek bir lojistik regresyon modelinden oluşmalıdır. Diğer veri kümeleri için daha ayrıntılı bir ağaç yapısı uygun olacaktır.

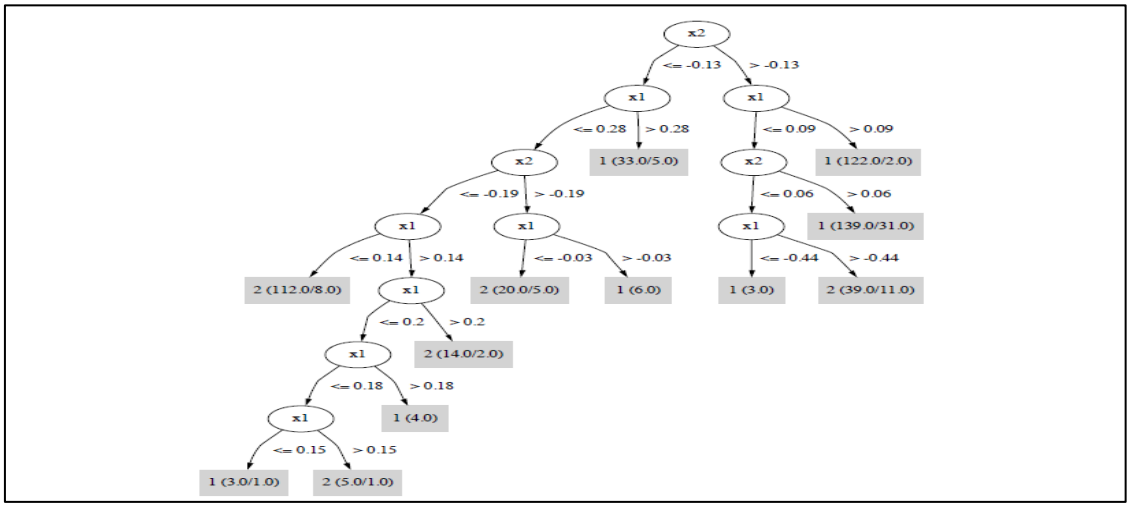
Aynı gerekçe, ağacı oluştururken karşılaşılan orijinal veri kümesinin alt kümeleri için de geçerlidir. Ağaç indüksiyonunun bölme ve fethetme şeklinde çalışmakta olduğundan: bir grup örnek için bir sınıflandırıcı, bir bölünme gerçekleştirerek ve sonra ortaya çıkan iki alt küme için ayrı sınıflandırıcılar oluşturarak çalışmaktadır. Küçük veri kümelerinin çok olması durumu için ağaç oluşturmanın genellikle iyi bir fikir olmadığını gösteren güçlü kanıtlar vardır, daha basit modeller kullanmak daha iyidir [52].



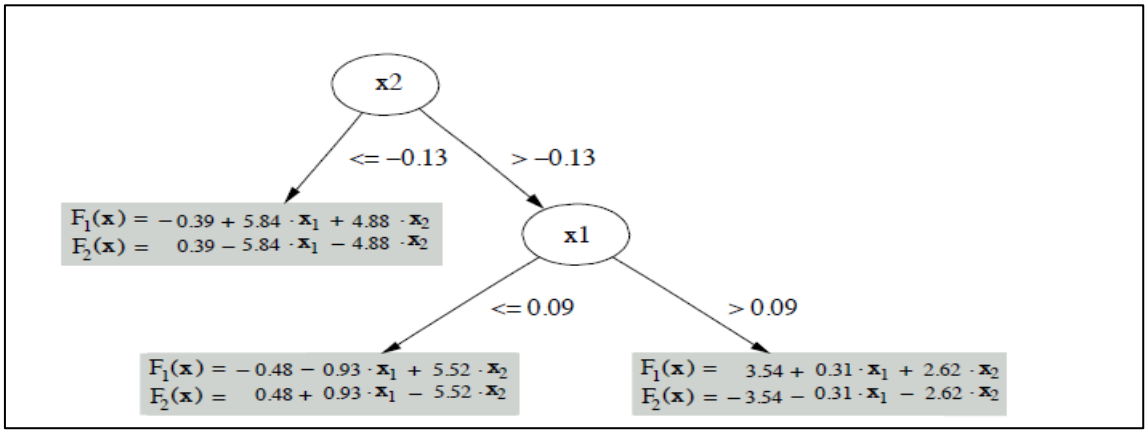
Şekil 4.5. C4.5 ağacı ve LMA modeli görselleri [53]

Ağaçta daha düşük seviyelerde karşılaşılan alt kümeler küçüldüğünden, bir noktada ağaç yetiştirme prosedürünü özyinelemeli çağırmak yerine doğrusal bir lojistik model oluşturmak tercih edilebilir. Bu, lojistik model ağacı algoritması için bir temel dayanaktır.

Şekil 4.5'de bir lojistik model ağacının ve bir C4.5 karar ağacının sınıf olasılık tahminleri, aynı örnek veri seti kullanılarak görselleştirilmiştir. Lojistik model ağacı başlangıçta örnek alanını 3 bölgeye ayırır ve bölgeler içindeki alt modelleri oluşturmak için lojistik regresyon işlevlerini kullanır. C4.5 ağacı ise örnek alanını 12 bölgeye ayırır. C4.5 tarafından oluşturulan ağacın eğitim verilerinde, özellikle de örnek alanının sağ alt bölgesinde bazı desenleri aştığı açıktır. Şekil 4.6'da C4.5 algoritması tarafından oluşturulan Karar Ağacı ve Şekil 4.7'de de LMA tarafından oluşturulan Lojistik Model Ağacı gösterilmektedir [53].



Şekil 4.6. C4.5 algoritması tarafından oluşturulan karar ağacı örneği [53]



Şekil 4.7. LMA tarafından oluşturulan lojistik model ağacı örneği [53]

Lojistik model ağacının yapraklarında F1, F2 fonksiyonları, sınıf üyelik olasılıklarını "Eş. 4.7" ve "Eş. 4.8" ile gösterilir;

$$P_T(G = 1|X = x) = \frac{e^{F_1(x)}}{e^{F_1(x)} + e^{F_2(x)}}, \tag{4.7}$$

$$P_T(G = 2|X = x) = \frac{e^{F_2(x)}}{e^{F_1(x)} + e^{F_2(x)}}. \tag{4.8}$$

4.2.3. Fonksiyonel ağaç (FA)

Fonksiyonel Ağaç Modelinin kullanılmakta olduğu bir problemde çoklu kararlar, hem sınıflandırma probleminde olduğu gibi ağacın ara düğümlerinde, hem de regresyon probleminde olduğu gibi yaprak kısımlarında alınmaktadır. Yani bu model her iki yaklaşımın ortak kümesi olarak gösterilebilir [54]. Karar ağaçlarının bölünmesi ve dallanması sırasında tek bir nitelik üzerinden işlem yapılırken Fonksiyonel Ağaç Modelinde yaprak ve düğümlerin hepsinde eldeki öznitelik değerlerinin tamamı kullanılmaktadır [55].

Fonksiyonel Ağaç modelinde yukarıdan aşağıya dallanma ve bölünme ağaç algoritması kullanılmaktadır. Ağaç yapısının düğümlerinde bölünme işlemi tekli olmasına rağmen veri setindeki bütün nitelikler ile çoklu lojistik regresyon algoritması sonucu ortaya çıkan yeni nitelikler bir arada kullanılır. Bulunan bu yeni niteliklerin değerleri regresyon fonksiyonundan elde edilen tahminlerdir [53].

Ağaç yapısı oluşturma işlemi sonucunda budama yapılır. Bu budama aşağıdan başlayarak yukarı doğru yapılır. Eğer bir düğümün yaprağı yoksa üç durum uygulanabilir. Birincisi, budama işlemi gerçekleştirilmez. İkincisi, tek sonuç değerine veya sabit bir değere sahip olan bir yaprakla yer değiştirilir. Üçüncüsü de bir fonksiyon değerine sahip bir yaprak ile değiştirilir [54].

Fonksiyonel Ağaç Algoritması ile hangi sınıfa ait olduğu belli olmayan örnekler için ağaç yapısındaki kök düğümünden itibaren yapraklara doğru yönelmesi ile tahminler yapılır. Ağaç yapısındaki karar düğümlerinde ayrı ayrı regresyon fonksiyonları ile işlem yapılır. Regresyon fonksiyonu ile nitelik sayıları çoğaltılır ve ortaya çıkan olasılıklara göre ağaç

yapısındaki hangi taraftan devam edilmesi gerektiğine karar verilir. İşlem sırasında yaprağa gelinmesi durumunda bu yaprak sabit şeklinde sınıflandırılabilir veya yapraktaki başka regresyon fonksiyonu değerlendirilir.

Fonksiyonel Ağaç Algoritması ile sonuçların veya sınıfların tahmini aşağıdaki şekilde yapılır;

- İlk olarak eğitim veri seti ile Fonksiyonel Ağaç yapısı tasarlanır,
- Sonucu belli olmayan bir örnek ile birlikte işlem başlatılarak, ağacın kökünden itibaren yaprağa gidinceye kadar ilerlenir,
- Örneğimize ait olan niteliklerin tamamı, ağacın her düğümünde, veri setinde bulunan toplam sınıf kadar oluşturulmuş regresyon fonksiyonuna uygulanır,
- Ortaya çıkan olasılıklar değerlendirilerek ağaç üzerinde hangi taraftan devam edilmesi gerektiği belirlenir,
- Yaprağa ulaşıldığında, eğer yaprak bir sınıfa ait ise örneğimiz de o sınıfa ait demektir. Eğer yaprakta başka regresyon modeline ulaşılmış ise örneğimize ait nitelikler bu regresyon modeline uygulanarak sonuçta çıkacak olan olasılığa göre örneğin sınıfı belirlenir. [56]

Lojistik regresyon analizi, örüntü hesaplama problemlerinde fazlaca kullanılmakta ve Fonksiyonel Ağaç Algoritmasının temelini oluşturmaktadır [57]. Regresyon problemin, eğer ikili sonucu ancak tek bağımsız değişkeni varsa Lojistik regresyon analizi, eğer ikili sonucu ancak birden çok değişkeni varsa çoklu lojistik regresyon ile çözüme gidilir.

Bir regresyon probleminin ρ sayısı kadar bağımsız değişkene sahip olduğunu ve değişkenlerin X= (x1,…,xp) şeklinde bir tane vektörle ifade edildiği değerlendirilsin. Bu değerlendirme sonucunda çoklu regresyon değeri için logit fonksiyonu y(x) ile ifade edilir ve “Eş. 4.9” da olduğu gibidir.

$$y(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \tag{4.9}$$

Bu eşitlik sonucunda lojistik regresyon modeli ise

$$p(x) = \frac{e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p}} \quad (4.10)$$

şeklinde ifade edilir. “Eş. 4.10” da, ρ(x) model sonucunda oluşan olasılığı, x1,…,xρ bağımsız değişkenleri, β0,β1,…,βρ katsayıları ise regresyon modelinin katsayılarıdır.

4.2.4. Naive bayes ağacı (NBA)

Naive Bayes Ağacı, ana yapı olarak Karar Ağaçlarını, yapraklarında ise Naive Bayes sınıflandırıcısını kullanan gözetimli bir algoritmadır [58]. Naive Bayes sınıflandırıcısının, eğer veri setleri az veriden oluşuyor ise Karar Ağaçlarından daha verimli çalıştığı, yapılan çalışmalarda görülmektedir. Bu nedenle Karar Ağacı yapısı oluştururken birkaç nitelik ilerledikten sonra yapraklarda Naive Bayes Sınıflandırıcısı kullanmak, nitelikleri bölerek işleme devam etmekten daha verimlidir.

Ağaç yapısı yukarıdan aşağıya doğru, bilgi kazanımına göre seçilen tek değişkenli bölünmelerle oluşturulmaktadır. Ağacın oluşumu sırasında, her bir düğümde, o düğümdeki verilerin bölünüp bölünmeyeceğine karar verilir veya bu düğümdeki veriler üzerinde, eğitilmiş bir yerel Naive Bayes modelini dikkate alan bir budama işlemi yapılır. Her bir düğümdeki budama kararı, bir düğümdeki yerel Naive Bayes modelinin çapraz doğrulanma sonucu (ayrık veriler kullanılarak hesaplanır), düğümde ayrılmadan kaynaklanan yaprakların sonucunun ağırlıklı toplamı ile karşılaştırılarak verilir. Bir düğümde en az 30 örnek varsa ve bölünmeyle elde edilen hatadaki göreceli azalma yüzde beşin üzerindeyse, bölünme dikkate alınır.

Naive Bayes Ağaçlarının genellikle standart Karar Ağaçları veya Naive Bayes’e göre performansının daha iyi olduğu gözlemlenmiştir, ancak Karar Ağaçları ve Naive Bayes algoritmalarının da Naive Bayes Ağaçlarına göre daha iyi olduğu birkaç durum vardır [58].

Naive Bayes Ağacı Algoritması klasik özyinelemeli bölümleme şemalarına benzer, yapraklarında tek bir sınıfın sonucuna ulaşmak yerine Naive Bayes vardır [58]. Roh Kovani tarafından önerilmiş Navive Bayes Ağacı Algoritmasının çalışma prensibi şu şekilde çalışmaktadır.

- Her nitelik için, niteliğindeki olabilecek bölünmenin faydası değerlendirilir ve sürekli özellikler için bu aşamada bir eşik değeri de hesaplanır.
- En yüksek sonuca sahip nitelik bırakılır.
- Bırakılan nitelik geçerli düğümün faydasından önemli ölçüde daha iyi değilse, geçerli düğüm için bir Naive Bayes sınıflandırıcısı oluşturulur ve geri dönülür.
- Nitelik üzerindeki teste göre nitelik sürekli ise, bir eşik ayrımı kullanılır; nitelik ayrık ise olası değerler için çok yönlü bir ayrım yapılır.
- Bu işlem özyinelemeli olarak her bir veri satırı için uygulanır.

Kararlı ağaçlarında olduğu gibi bu algoritmada da standart entropi minimizasyon tekniği kullanılarak sürekli nitelikler için bir eşik değeri belirlenir. Bir düğümün faydası, verilerin ayrıklaştırılması ve düğümde Naive-Bayes kullanarak 5 katlamalı çapraz doğrulama sonuç tahmininin hesaplanmasıyla bulunur. Bir bölünmenin faydası ise düğüme verilen ağırlığın o düğüme giden örnek sayısı ile orantılı olduğu düğümlerin faydasının ağırlıklı toplamıdır.

Sezgisel olarak, her bir yapraktaki bir Naive-Bayes sınıflandırıcısı için genelleme doğruluğunun, mevcut düğümdeki tek bir Naive Bayes sınıflandırıcısından daha yüksek olup olmadığını tahmin edilmesine çalışılır.

Nitelikleri seçmek için çapraz doğrulamayı doğrudan kullanmak, büyük ek yük getirmesi nedeniyle yaygın olarak kullanılmamıştır. Ancak, veriler ayrıksa, Naive-Bayes kullanılarak örnek sayısı, nitelik sayısı ve etiket değerleri sayısında çapraz doğrulama yapılabilir. Bunun nedeni, örneklerin kaldırabilir, sayaçların güncelleyebilir, sınıflandırabilir ve farklı bir örnek kümesi için tekrarlanabilir olmasıdır [59].

Örnek sayılarını m, öznitelik sayılarını n ve etiket değerlerini l temsil ederse, ayrık öznitelikler için öznitelik seçim aşamasının karmaşıklığı $O\,(m.n^2.\mathrm{l})$ formülü ile hesaplanır. Öznitelik sayısı $O(\log m)$'den azsa ve etiket sayısı küçükse, çapraz doğrulama kullanarak öznitelik seçiminde harcanan süre, örnekleri her özniteliğe göre sıralamak için harcanan süreden daha azdır. Bu nedenle Naive Bayes Ağacının büyük veri setleri için verimli olacağını bekleyebiliriz [58].

5. DENEYSEL ÇALIŞMALAR

5.1. WEKA (Waikato Environment for Knowledge Analysis) Yazılımı

WEKA (Waikato Environment for Knowledge Analysis) yazılımı Yeni Zelanda'da bulunan Waikato Üniversitesi tarafından geliştirilmiş açık kaynak kodlu bir yazılımdır. Bu yazılım Java tabanlı olarak geliştirilmiştir. Java programlama dili ile geliştirildiğinden hali hazırda kullanılmakta olan modern bilgi işlem platformlarında da kullanılabilmekte, taşınabilmekte ve platform bağımsız çalışabilmektedir [60].

Dosya biçimi olarak ARFF (Attribute Relationship File Format) dosya biçimini kullanmaktadır. ARFF dosya yapısı metin şeklinde olup Weka yazılım aracı için geliştirilmiştir. Yazılım aracı içerisinde birçok veri seti olduğu gibi kendi veri setinizi de kullanabilmektesiniz. Kullanımı kolay olduğundan birçok araştırma kullanılmaktadır. KDD'99 veri setinin ARFF dosyası şekline dönüştürülmüş halinden bir görüntü Resim 5.1'de gösterilmiştir.

```
@relation train

@attribute duration numeric
@attribute protocol_type {tcp,udp,icmp}
@attribute service {...,domain_u,...,http,...,telnet,...}
@attribute flag {OTH,REJ,RSTO,RSTOS0,RSTR,S0,S1,S2,S3,SF,SH}
@attribute src_bytes numeric
@attribute dst_bytes numeric
...
...
...
@attribute dst_host_rerror_rate numeric
@attribute dst_host_srv_rerror_rate numeric
@attribute label {dos,u2r,r2l,probe,normal}

@data
0,tcp,http,SF,181,5450,.....,0.00,0.00,normal
0,tcp,http,SF,239,486,.....,0.00,0.00,normal
184,tcp,telnet,SF,1511,2957,.....,0.00,0.00,u2r
305,tcp,telnet,SF,1735,2766,.....,0.00,0.00,u2r
23,tcp,telnet,SF,104,276,.....,0.00,0.00,r2l
0,tcp,telnet,RSTO,125,179,.....,0.50,0.50,r2l
0,tcp,http,SF,54540,8314,.....,0.00,0.00,dos
0,tcp,http,SF,54540,8314,.....,0.00,0.00,dos
0,udp,domain_u,SF,1,0,.....,0.00,0.00,probe
3,tcp,telnet,SF,54,114,.....,0.01,0.00,probe
```

Resim 5.1. Veri seti ARFF dosyası görüntüsü

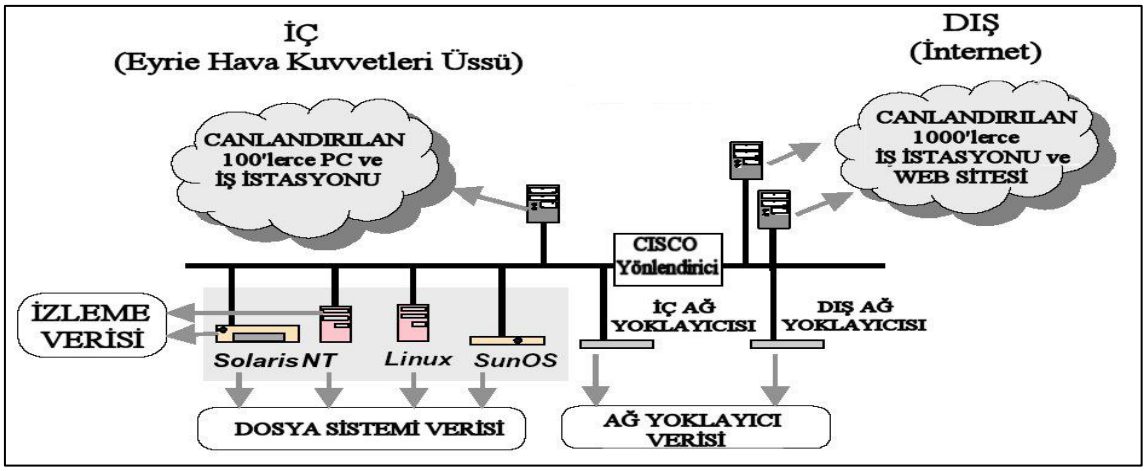
ARFF dosyasının ilk satırında “@relation” ile gösterilen bölümde bağlantı bilgisi, sonraki satırlarında “@attribute” ile gösterilen bölümlerde her bir nitelik bilgisi ve gelebilecek değerler, “@data” ile gösterilen bölümden sonra da her bir veri için bir satır olacak şekilde veriler bulunmaktadır. Her veri satırındaki her bir nitelik virgül ile ayrılmaktadır.

Weka yazılım aracı ile sınıflandırma, veri ön işleme, regresyon, gruplandırma, özellik seçimi ve çıkarımı gibi birçok makine öğrenmesi tekniği ve istatistiksel yöntem kullanılabilmektedir.

5.2. KDD’99 Veri Seti

KDD'99 Saldırı Tespit Değerlendirme Veri Kümesi, Üçüncü Uluslararası Bilgi Keşfi ve Veri Madenciliği Araçları Yarışması için kullanılan veri kümesidir. Yarışmanın amacı, bağlantıları sınıflandıran ve normal trafiği saldırı trafiğinden veya izinsiz girişlerden ayıran bir saldırı tespit sistemi yaratmaktı.

DARPA veri seti MIT (Massachusetts Institute of Technology) Lincoln Laboratuarı Bilisim Sistemleri Teknoloji Grubu tarafından Şekil 5.1’de gösterildiği gibi Amerikan Hava Kuvvetleri’nin ağ yapısına benzer bir yapıda tasarlanan simülasyondan alınan verileri içermektedir. Dokuz haftalık ağ trafiğinden alınan TCP dökümü verileri kaydedilmiştir. Yedi haftalık veriler eğitim seti, iki haftalık veriler ise test seti olarak kullanılmıştır. Eğitim setinde yaklaşık beş milyon, test setinde iki milyon bağlantı bulunmaktadır [61].



Şekil 5.1. DARPA 99 ağ yapısı [62]

5.2.1. KDD veri seti içeriği

KDD veri seti içerisindeki her bağlantının son verisi bağlantı etiketinden oluşmaktadır. Diğer bir deyişle bağlantının hangi sınıfa ait olduğunu belirlemektedir. KDD veri seti içerisinde 4 çeşit saldırı sınıfı ve bir de normal bağlantıların olduğu bir sınıf bulunmaktadır. Bunlar;

- Hizmet Aksattırma Saldırıları (Denail Of Service-DOS)
- Yönetici Hesabı Kullanarak Yerel Oturum Açma Saldırıları (Remote to Local)
- Kullanıcı Oturumunu Yönetici Oturumuna Yükseltme Saldırıları (User to Root)
- Bilgi Tarama Saldırıları (Probe veya Scan)
- Normal Bağlantılar

Veri seti içerisinde yukarıdaki saldırı sınıflarına ait olan 39 saldırı çeşidi vardır. Bunlar Çizelge 5.1'de sınıflarına göre ayrıştırılarak gösterilmiştir. Burada Eğitim veri setinde 22 adet saldırı türü bulunmasına rağmen test veri setinde eğitim veri setindeki saldırı türleri dışında 17 adet daha saldırı türü vardır.

Çizelge 5.1. KDD'99 veri setindeki saldırı türleri

Saldırı Sınıfı	Eğitim Setindeki Saldırı Türü	Test Veri Setindeki Ek Saldırı Türleri
Hizmet Aksattırma Saldırıları	Teardrop, back, smurf, land, pod, neptune,	apache2, udpstorm, processtable, mailbomb
Yönetici Hesabı ile Yerel Oturum Açma Saldırıları	loadmodule, rootkit, perl, bufferoverflow,	ps, sqlattack, xterm
Kullanıcı Hesabının Yönetici Hesabına Yükseltme Saldırıları	ftpwrite, spy, guesspasswd, imap, warezmaster, phf, multihop, warezclient	httptunnel, named, sendmail, snmpgetattack, snmpguess, xlock, xsnoop, worm
Bilgi Tarama Saldırıları	nmap, ipsweep, satan, portsweep	saint, mscan,

5.2.2. KDD veri seti bağlantı nitelikleri

KDD'99 veri seti içerisindeki her bir bağlantı satırı 41 nitelikten oluşmaktadır. Bu nitelikleri 4 ana grupta sınıflandırmak mümkündür [63]. Bu gruplar;

- Basit Nitelikler: Paket başlıklarında bulunan nitelikler.
- İçerik Nitelikleri: Bazı saldırı türleri ardışık kalıplar oluşturmaz çünkü bunlar sadece bir bağlantı içerir ve paketin yükünde gizlidir. İçerik özellikleri, alan bilgisine dayanan ve yükteki şüpheli davranışı araştırmak için kullanılan özelliklerdir.
- Zaman Kaynaklı Trafik Nitelikleri: İki çeşit zaman kaynaklı trafik niteliği vardır. Bu nitelikler sadece son iki saniyedeki bağlantıları denetlerler. "Aynı Sunucu" niteliği aynı sunucu kaynağına sahip bağlantıları analiz ederken, "Aynı Servis" niteliği ise aynı servis kaynağına sahip bağlantıları analiz eder.
- Sunucu Kaynaklı Trafik Nitelikleri: Bazı saldırılar büyük zaman aralıklarında gerçekleşir. Bu nedenle bağlantıları zaman aralıklarına göre değil belli sayıdaki bağlantı aralıklarına göre değerlendirmek gerekmektedir. Sunucu kaynaklı nitelikler aynı sunucudan gelen 100 bağlantıyı değerlendirerek oluşturulur.

Çizelge 5.2. KDD'99 veri seti bağlantı nitelikleri listesi

Nitelik Grubu	Nitelik İsmi	Nitelik Türü	Tanım
Basit Nitelikler	duration	Sürekli	Bağlantı uzunluğu
	protocol-type	Ayrık	Protokol tipi
	service	Ayrık	Servis tipi
	src-bytes	Sürekli	Kaynaktan hedefe veri
	dst-bytes	Sürekli	Veri byte sayısı
	flag	Ayrık	Bayrak
	land	Ayrık	Kaynak ve hedef IP aynı ise 1 değilse 0
	wrong-fragment	Sürekli	Yanlış parçalama
	urgent	Sürekli	Acil paket sayısı
İçerik Nitelikleri	Hot	Sürekli	"hot" göstergesi
	num-failed-logins	Sürekli	Hatalı giriş sayısı
	logged-in	Ayrık	Giriş başarılı ise 1 değilse 0
	num-compromised	Sürekli	Gizliliğin ihlal edilme sayısı

Çizelge 5.2. (devam) KDD'99 veri seti bağlantı nitelikleri listesi

İçerik Nitelikleri	root-shell	Ayrık	“Root Shell” elde edildiyse 1 değilse 0
	su-attempted	Ayrık	“Su Root” komutu girildiyse 1 değilse 0
	num-root	Sürekli	“Root” erişim sayısı
	num-file-creations	Sürekli	Dosya oluşturma işlemleri sayısı
	num-shells	Sürekli	Shell promptlarının sayısı
	num-access-files	Sürekli	Kontrol dosyalarına erişim işlemleri sayısı
	num-outbound-cmds	Sürekli	FTP oturumunda giden komut
	is-hot-login	Ayrık	Giriş “hot” listesindeyse 1 değilse 0
	is-guest-login	Ayrık	Giriş “guest” ise 1 değilse 0
Zaman Kaynaklı Trafik Nitelikleri	count	Sürekli	Aynı sunucuya önceki iki bağlantıyla aynı bağlantıların sayısı
	serror-rate	Sürekli	“SYN” hata bağlantılarının yüzdesi
	rerror-rate	Sürekli	“REJ” hata bağlantılarının yüzdesi
	same-srv-rate	Sürekli	Aynı servise bağlantıların yüzdesi
	diff-srv-rate	Sürekli	Farklı servislere bağlantıların yüzdesi
	srv-count	Sürekli	Aynı servise önceki iki bağlantıyla aynı bağlantıların sayısı
	srv-serror-rate	Sürekli	“SYN” hata bağlantılarının yüzdesi
	srv-rerror-rate	Sürekli	“REJ” hata bağlantılarının yüzdesi

Çizelge 5.3. (devam) KDD'99 veri seti bağlantı nitelikleri listesi

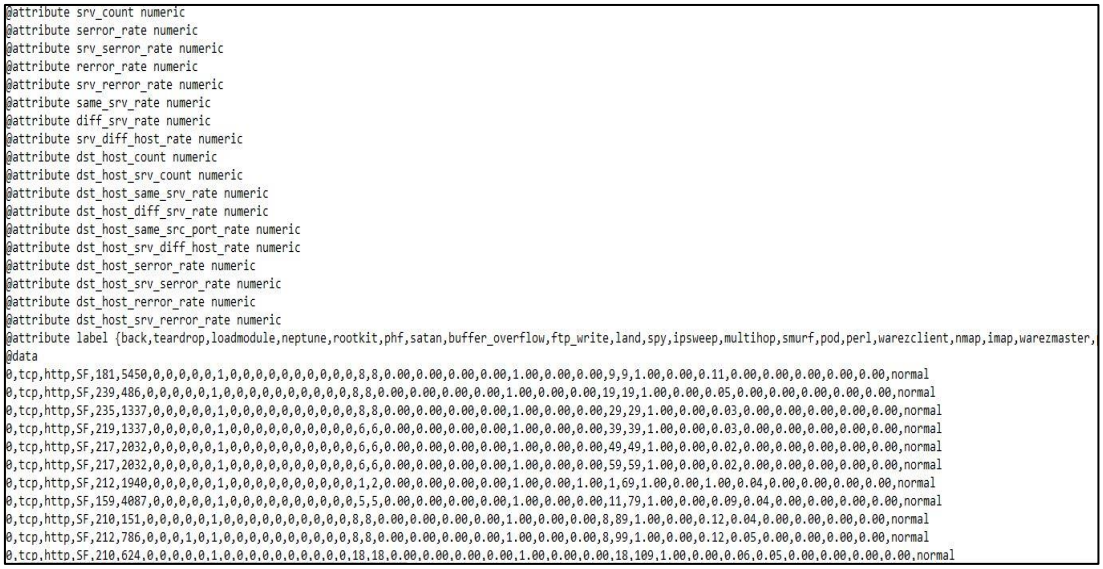
	srv-dif-host-rate	Sürekli	Farklı servislere bağlantıların yüzdesi
Sunucu Kaynaklı Trafik Nitelikleri	dst-host-count	Sürekli	Aynı hedef sunucusu IP'sine sahip bağlantı sayısı
	dst-host-srv-count	Sürekli	Aynı hedef sunucusu IP'sine ve aynı servise sahip bağlantı sayısı
	dst-host-same-srv-rate	Sürekli	Aynı hedef portuna ve aynı servise sahip bağlantı yüzdesi
	dst-host-diff-srv-rate	Sürekli	Sunucudaki farklı servislerin yüzdesi
	dst-host-same-src-port-rate	Sürekli	Sunucudaki aynı kaynak porta sahip bağlantı yüzdesi
	dst-host-srv-diff-host-rate	Sürekli	Farklı kaynaktan gelen servislerin bağlantı yüzdesi
	dst-host-serror-rate	Sürekli	"S0" hatasına sahip bağlantı yüzdesi
	dst-host-srv-serror-rate	Sürekli	"S0" hatasında sahip sunucu ve belirli servislerin bağlantı yüzdesi
	dst-host-rerror-rate	Sürekli	Sunucudaki "RST" hatasında sahip bağlantıların yüzdesi
	dst-host-srv-rerror-rate	Sürekli	"RST" hatasında sahip sunucu ve belirli servislerin bağlantı yüzdesi

Çizelge 5.2'de KDD'99 veri seti içerisindeki bütün nitelikler gruplarına göre açıklamaları ile birlikte verilmiştir. Bu nitelikler sayesinde öğrenme işlemi daha kararlı ve kolay olacaktır. Bu niteliklerin bazıları makine öğrenmesi işleminde kullanılmayabilir. Bunun öğrenme zamanı ve test zamanına etkisi olacaktır ancak çıktıların doğruluk oranını düşürebilecektir. Bu nedenle bu çalışmada niteliklerin azaltılması yapılmamıştır.

5.3. Veri Setlerinin Oluşturulması

5.3.1. Veri seti oluşturulması ön işlemleri

Uygulamada KDD'99 veri setinin %10 luk kısmından oluşan veri seti kullanılmıştır. Bu veri seti içerisinden sadece hizmet aksatma saldırıları çeşitleri ve normal bağlantıların bulunduğu veriler seçilerek alınmıştır. Eğitim veri seti 64955 adet veri ve test veri seti 65456 adet veri içerecek şekilde hazırlanmıştır. KDD'99 veri setinin %10'luk kısmından alınan verilerin ilk hali Resim 5.2'de gösterilmiştir.



Resim 5.2. KDD'99 veri seti görüntüsü

Alınan veri setinin her bir satırında 41 nitelik ve sonuç değerinden oluşan bir bağlantı verisi bulunmaktadır. Bu nitelikleri birbirlerinden ayırarak makinenin anlamlandırabileceği hale getirilmesi gerekmektedir. Bu işleme normalizasyon denilmektedir. İlk olarak her bir satırın son niteliği saldırı türünün belli etmektedir. Bu değerler metin halinden sayısal hale getirilmiştir. Çizelge 5.3'te hangi saldırının sonuç niteliğinde hangi sayısal değer ile gösterildiği belirtilmiştir. Veri seti içerisinde 6 çeşit hizmet aksatma saldırısı bulunmakta olmasına rağmen her birinde eşit sayıda bulunmamaktadır.

Çizelge 5.4. KDD'99 veri setindeki saldırıların normalizasyonu

Saldırı Türü	Sayısal Değeri
Normal	0
Neptune	1
Smurf	2
Teardrop	3
Land	4
Pod	5
Back	6

Normalizasyon işlemleri yapılırken sayısal değerler olduğu gibi bırakılmış, sadece metinsel değerler sayısala çevrilmiştir. Diğer metinsel değer olan Bayrak Türü değerinin normalize edilmiş hali Çizelge 5.4'de gösterilmiştir. Bayrak Türü niteliği her satırda virgül ile ayrılmış dördüncü niteliktir. KDD'99 veri Seti içerinde 11 tür bayrak varken, bu çalışmada hizmet aksatma saldırıları ve normal bağlantılar alındığından 9 adet bayrak türü normalize edilmiştir.

Çizelge 5.5. KDD'99 veri setindeki bayrak türü değerinin normalizasyonu

Bayrak Türü	Sayısal Değer
OTH	0
REJ	1
RSTO	2
RSTR	3
S0	4
S1	5
S2	6
S3	7
SF	8

Protokol Tipi değerlerinin normalize edilmiş hali Çizelge 5.5'te gösterilmiştir.

Çizelge 5.6. KDD'99 veri setindeki protokal tipi değerinin normalizasyonu

Protokol Tipi	Sayısal Değeri
ICMP	0
TCP	1
UDP	2

Servis Türü değerlerinin normalize edilmiş hali Çizelge 5.6'da gösterilmiştir. KDD'99 veri seti içerisinde 66 adet Servis Türü olmasına karşın bu çalışmada hizmet aksatma saldırıları ve normal bağlantılar alındığından 62 adet Servis Türü normalize edilmiştir.

Çizelge 5.7. KDD'99 veri setindeki servis türü değerinin normalizasyonu

Servis Türü	Sayısal Değer	Servis Türü	Sayısal Değer
auth	0	netbios_dgm	31
bgp	1	netbios_ns	32
courier	2	netbios_ssn	33
csnet_ns	3	netstat	34
ctf	4	nnsp	35
daytime	5	nntp	36
discard	6	ntp_u	37
domain	7	other	38
domain_u	8	pop_2	39
echo	9	pop_3	40
eco_i	10	printer	41
ecr_i	11	private	42
efs	12	remote_job	43
exec	13	rje	44
finger	14	shell	45
ftp	15	smtp	46
ftp_data	16	sql_net	47
gopher	17	ssh	48
hostnames	18	sunrpc	49
http	19	supdup	50

Çizelge 5.6. (devam) KDD'99 veri setindeki servis türü değerinin normalizasyonu

http_443	20	systat	51
IRC	21	telnet	52
imap4	22	tim_i	53
iso_tsap	23	time	54
klogin	24	urh_i	55
kshell	25	uucp	56
ldap	26	uucp_path	57
link	27	vmnet	58
login	28	whois	59
mtp	29	X11	60
name	30	Z39_50	61

KDD’99 veri seti normalize edilerek makine öğrenmesi teknikleri için kullanılabilir hale getirilmiştir. Çalışmanın devamında Eğitim Veri Seti ve Test Veri Seti oluşturulacaktır.

5.3.2. Eğitim veri setinin oluşturulması

Eğitim veri seti içerisinde toplam 64955 adet veri bulunmaktadır. Eğitim veri seti oluşturulurken KDD’99 veri seti içerisindeki Hizmet Aksatma Saldırıları dışında kalan saldırı türleri veri setinden çıkarılmıştır. Bu yüzden saldırı olarak tespit edilen veri örnekleri azalmıştır. Ancak öğrenme işlemi sağlayabilecek kadar veri sayısı eğitim veri seti içerisinde bulunmaktadır.

Eğitim veri setindeki veri sayılarının saldırı çeşitlerine göre dağılımı Çizelge 5.7’de gösterilmiştir.

Tablo üzerinde de görüldüğü gibi 6 tür Hizmet Aksatma Saldırısı bulunmaktadır. Bu aynı zamanda KDD’99 veri içerisindeki bulunan Hizmet Aksatma Saldırısını ifade etmektedir. Burada eğitimin sağlıklı bir şekilde doğru ve güvenilir olması için yaklaşık üçte bir oranında normal veri de bulunmaktadır.

Çizelge 5.8. Eğitim veri seti dağılımı

Saldırı Türü	Saldırı Çeşidi	Veri Sayısı	Toplam Veri Sayısı
Hizmet Aksatma Saldırıları	Neptune	21174	64955
	Smurf	21001	
	Teardrop	728	
	Land	14	
	Pod	164	
	Back	1276	
Normal		20598	

Burada dikkat çeken en büyük problem bazı örneklerden çok fazla varken bazılarından çok az örnek bulunmasıdır. Bu KDD'99 veri setinden kaynaklanan başlıca problemlerden biridir. KDD'99 veri seti içerisinde her tür saldırıdan eşit miktarda örneğe yer verilmemiştir. Bu çalışmada bazı saldırılardan az olmasına rağmen, eğitim ve test veri setlerine eşit olarak dağıtılmıştır.

5.3.3. Test veri setinin oluşturulması

Test Veri Seti içerisinde toplam 65456 adet veri bulunmaktadır. Bu veri setindeki verilerin saldırı çeşitlerine göre dağılım Çizelge 5.8'de gösterilmiştir.

Çizelge 5.9. Test veri seti dağılımı

Saldırı Türü	Saldırı Çeşidi	Veri Sayısı	Toplam Veri Sayısı
Hizmet Aksatma Saldırıları	Neptune	18736	65456
	Smurf	23262	
	Teardrop	251	
	Land	7	
	Pod	100	
	Back	927	
Normal		22173	

Test veri seti oluşturulurken Eğitim Veri Setinde olan saldırı sayıları kadar örnek olmasına dikkat edilmiştir. Buradaki en büyük dezavantaj Eğitim Veri Setinde olduğu gibi bazı saldırı türü örneklerinden az sayıda bulunmasıdır. Ancak KDD Veri Seti kullanırken bunun önüne geçme imkânı bulunmaktadır. Bu çalışmada elde olan imkanları en iyi değerlendirecek şekilde veri sayıları paylaştırılmıştır.

5.4. Yapılan Simülasyon Çalışmaları

Simülasyon çalışmalarına başlamadan önce WEKA yazılımı ve KDD'99 veri seti incelenmiştir. İnceleme sırasında hangi makine öğrenmesi tekniğinin hangi sonuçları verdiği denenmiştir. Çalışma sırasında hibrit algoritmalar üzerine çalışılmış ancak klasik algoritmaların da nasıl sonuçlar verdiğine bakılmıştır.

Çalışma sırasında her bir algoritma için öncelikle ağ, eğitim veri seti kullanılarak eğitilmiş, daha sonra da test veri seti kullanılarak test edilmiştir. Genel olarak bakıldığında her bir hibrit makine öğrenmesi tekniğinde öğrenme işlemi %100 veya yakın başarı ile sağlanmıştır. Ancak eğitim sonucunun çok yüksek başarı ile sağlanması test işleminin de bu sonuçlar ile olacağı anlamını taşımamaktadır. Böyle bir genelleme yapmak bilimsellikten uzak bir yanılgıdan ileri gidemez.

Simülasyon çalışması sırasında ilk olarak KMNB algoritması üzerinde çalışma yapılmıştır. Burada öncelikle eğitim veri seti kullanılarak ağ eğitilmiştir. Eğitilmiş ağın öğrenme durumu oldukça yüksektir. Daha sonrasında eğitilmiş olan ağa test verileri yüklenerek ağın başarı durumu test edilmiştir. Test sonucunda alınan başarı durumu %99,95 olmuştur.

İkinci olarak Lojistik Model Ağacı algoritması üzerinde simülasyon çalışılması yapılmıştır. Oluşturulmuş olan eğitim ve test veri setleri bu algoritma üzerinde uygulanmıştır. Eğitim sonrası %100 başarı oranı elde edilmiştir. Bu işlem sonrasında test verileri ağa yüklenerek sonuçlar elde edilmiştir. Test sonucunda %99,55'lik bir başarı oranı elde edilmiştir.

Üçüncü olarak Fonksiyonel Ağaç Algoritması üzerinde çalışılmıştır. Oluşturulmuş olan eğitim veri seti kullanılarak %100 başarı ile eğitim sonuçlandırılmıştır. Daha sonrasında test veri seti bu algoritma üzerinde uygulanmış ve %99,76 gibi yüksek bir başarı oranı elde edilmiştir.

Dördüncü ve son olarak Naive Bayes Ağacı algoritması üzerinde çalışma yapılmıştır. Oluşturulmuş olan eğitim veri seti algoritma üzerinde uygulanarak %100 başarı ile eğitim sağlanmıştır. Daha sonrasında test veri seti bu algoritma üzerinde denenmiş ve %99,86'lık bir sonuç elde edilmiştir.

5.5. Eğitim ve Test Sonuçları

Yapılan çalışma sırasında eğitim için ve test için aynı veri setleri kullanılacak şekilde farklı makine öğrenmesi teknikleri üzerinde çalışmalar yapılmıştır. Algoritmalar üzerinde yapılan eğitim işlemi sonucunda çok yüksek başarı oranları elde edilmiştir. Eğitim işlemi sonucunda alınan sonuçlar toplu olarak Çizelge 5.9'da sunulmuştur. Buradan da görüldüğü üzere %100 gibi yüksek bir başarı oranı elde edilmiştir. Bu başarının altında Hibrit Makine Öğrenmesi tekniklerinin başarılı olmasının yanı sıra veri setlerinin normalize edilmesi ve KDD veri setindeki bazı aksaklıkların düzeltilmeye çalışılması da yer almaktadır.

Çizelge 5.10. Eğitim sonuçları tablosu

Kullanılan Algoritmalar	Doğru Sınıflandırılan Veri Sayısı	Yanlış Sınıflandırılan Veri Sayısı	Ortalama Mutlak Hata (MAE)	Kök Kareler Karesi (RMSE)	Başarı
KMNB	64943	12	0,0002	2,2839	%99,98
LMA	64955	0	0	0,0013	%100
FA	64955	0	0	0,0847	%100
NBA	64955	0	0	0,198	%100

Eğitim işlemi sonrasında her bir algoritma için test veri seti ile test işlemi yapılmış ve algoritmaların başarı durumunu gözlemlenmiştir. Burada eğitimde olabilecek ezberin önüne geçebilmek amacıyla ayrı bir test veri kullanılmıştır. Bunun için önceki bölümlerde de bahsedildiği üzere yeni saldırı ver normal verilerin olduğu başka bir veri seti oluşturulmuştur. Test veri seti ile elde edilen başarı durumu Çizelge 5.10'da gösterilmiştir.

Çizelge 5.11. Test sonuçları tablosu

Kullanılan Algoritmalar	Doğru Sınıflandırılan Veri Sayısı	Yanlış Sınıflandırılan Veri Sayısı	Ortalama Mutlak Hata (MAE)	Kök Kareler Karesi (RMSE)	Başarı
KMNB	65426	30	0,0004	0,0112	%99,95
LMA	65166	290	0,0013	0,0354	%99,55
FA	65305	151	0,0008	0,0254	%99,76
NBA	65368	88	0,0004	0,0184	%99,86

Bu çalışmada elde edilen sonuçların oldukça başarılı olduğu görülmüştür. Simülasyon çalışmasından KDD'99 veri setinde az bulunan bazı saldırı örneklerinin dezavantaj yaratabileceği düşünülmüş olmasına rağmen yapılan çalışma sonucunda tüm makine öğrenmesi tekniklerinde öğrenmenin sağlandığı ve bu az olan saldırı türlerinin de oldukça tespit edilebildiği görülmüştür.

5.6. Sonuçların Literatür ile Karşılaştırılması

Literatür tarandığında saldırı tespit sistemleri ile ilgili birçok çalışma yapıldığı ve en optimum çözümün bulunmaya çalışıldığı görülecektir. İlk çalışmalar klasik makine öğrenmesi teknikleri kullanılmış ancak saldırıların sayısının ve çeşidinin artmaya başlaması klasik makine öğrenmesi tekniklerinin yetersiz kalmasına sebep olmaya başlamıştır.

Klasik makine öğrenmesi tekniklerinin yetersiz kalması hibrit makine öğrenmesi tekniklerinin gelişimine neden olmuştur. Saldırıların tespit edilebilmesi için klasik makine öğrenmesi teknikleri birbirlerinin dezavantajını en aza indirecek şekilde hibrit olarak kullanılmıştır. Geçmiş dönemlerde hibrit makine öğrenmesi teknikleri kullanılarak yapılan çalışmalar Çizelge 5.11'de verilmiştir. Bu tablo oluşturulurken, simülasyon sonrası sonuçlar ile tam bir kıyas yapılabilmesi amacıyla sadece hizmet aksatma saldırıları başarı oranı ve KDD veri seti kullanan çalışmalar baz alınmıştır.

Çizelge 5.12. Literatürdeki hibrit makine öğrenmesi çalışmaları

S.No.	Çalışma Yazarı	Çalışma Yılı	Makine Öğrenmesi Tekniği	Veri Seti	Başarı Oranı
1	Mostaque Morshedur Hassan [29]	2013	Bulanık Mantık ve Genetik Algoritma	KDD Cup 99	%85
2	Wathiq Laftah Al-Yaseen [30]	2017	K-Means ve Karar Destek Makineleri	KDD Cup 99	%99,54
3	Shadi Aljawarneh [31]	2018	J48, Meta Pagging, Rastgele Ağaç, Karar Ağacı, AdaBoostM1, Karar Kütüğü and Naive Bayes	KDD Cup 99	%99,9
4	Dewan Md. Farid [32]	2010	Naive Bayes ve Karar Ağacı	KDD Cup 99	%99,75
5	Zaiton Muda [33]	2011	K-Means ve Naive Bayes	KDD Cup 99	%99,6
6	Soodeh Hosseini [34]	2020	Genetik Algoritma-Karar Destek Makineleri-Parçacık Sürü Optimizasyonu-Yapay Sinir Ağları	KDD Cup 99	%99,80
7	E. Doğru, N. Barışçı	2020	Karar Masası ve Naive Bayes	KDD'99	99,95

Tablodan da görüleceği gibi başarı oranları oldukça yüksektir. Ancak bir saldırı tespit sisteminin canlı ve gerçek bir sistemde kullanılması için yeterli değildir. Bu hata oranları gerçek bir sistem için kabul edilebilir değildir.

Literatürdeki çalışmalar ile bizim çalışmamızın sonuçları karşılaştırıldığında, literatürdeki çalışmaların en yüksek başarı oranı elde etmiş çalışması ile bizim çalışmamızda elde edilen en kötü sonucun birbirlerine oldukça yakın olduğu görülmektedir.

6. SONUÇ VE ÖNERİLER

Saldırı Tespit Sistemleri günümüzün en önemli konularından biri haline gelmiştir. Günümüzde siber saldırılar arttığı ve saldırı türleri farklılaştığından saldırı tespit sistemlerinin de gelişmesi ve bu saldırıları tespit edebilmesi gerekmektedir. Bütün verilerimiz dijital ortama aktarılırken bunları korunması kritik öneme sahiptir.

Türkiye'de kullanılan e-devlet platformu içerisinde Türkiye vatandaşı olan her bireyin bilgileri bulunmaktadır. Bu bilgiler ile her türlü işlem yapılabilir. Bu yüzden bu bilgilerin hiçbir kötü niyetli insanın eline geçmemesi gerekmektedir.

İnternet üzerinden alışveriş günden güne popüler bir hale gelmektedir. Yapılan istatistiklerde Türkiye'de internet üzerinden alışveriş yapan kişi sayısı 40 milyon civarındadır. Ülkemiz nüfusunun ortalama 80 milyon olduğu değerlendirildiğinde, nüfusun yarısı internet alışverişinin aktif olarak kullandığı görülmektedir.

İnternet alışverişi sırasında internet siteleri ile çok az bilgimizi bile paylaşsak, yine de isim, soy isim, adres, kredi kartı bilgileri, eposta adresi, telefon numarası bilgilerini paylaşmak zorunluluğu doğmaktadır. Bu bilgiler kötü niyetli insanların eline geçmesi durumunda bizleri zor duruma düşürecek sonuçlar doğurabilir. Tüm bunların önüne geçmek için internet üzerinden alışveriş hizmeti sunan bu firmaların saldırı tespit sistemleri kullanmaları kritik öneme sahiptir.

Bu çalışmada Saldırı Tespit Sistemleri genel olarak ve detaylı incelenmiştir. Daha önceki çalışmalarda neler yapıldığı, kullanılan veri setlerinin neler olduğu, tasarımlarının nasıl yapıldığı, hangi makine öğrenmesi tekniğinin uygulandığı araştırılmıştır.

Bu araştırmalar sonucunda klasik makine öğrenmesi tekniklerinin saldırıları tespit etmedeki olumlu ve olumsuz yanları görülmüştür. Buradaki olumlu yanları arttıracak, olumsuz yanları ise azaltabilecek bir makine öğrenmesi tekniğinin saldırı tespit sistemlerine uygulanması denenmiştir.

Literatür çalışması sonucunda;

- Klasik makine öğrenmesi tekniklerinin o günün teknolojisine göre başarılı olduğu ancak günümüzde saldırı tiplerinin değiştiği,
- Kullanılan veri setinin makine öğrenmesi aşamasında çok önemli olduğu, farklı saldırı çeşitlerini barındıran veri setlerinin daha doğru sonuçlar verdiği,
- Veri seti üzerinde yapılan ön işlemlerin gerekli olduğu ve doğruluk oranına büyük etkisi olduğu,
- Veri setindeki veri sayısının çok olmasının eğitim ve test zamanını uzattığı, veri sayısının az olmasının da doğruluk oranını düşürdüğü,
- Geçmiş çalışmalarda olduğu gibi günümüzde de veri setlerini normalize edebilecek bir uygulamanın hali hazırda bulunmadığı, bütün bu işlemlerinin araştırmacılar tarafından yapıldığı ve bu normalizasyon işleminin bir standardının olmadığı,
- Literatürdeki çalışmaların büyük bir kısmında KDD'99 veri setinin kullanıldığı ve günümüz şartlarında ve ihtiyaçlarında bir veri setinin oluşturulmasına ihtiyaç olduğu,
- Kullanılan makine öğrenmesi tekniklerinin kullanılma aşamasında araştırmacıya çok fazla tercih sunulduğu ve bu tercihlerin sonuca büyük etkilerinin olduğu, daha kesin ve doğru sonuçlar alabilmek için kullanıcıya bırakılan tercihlerin azaltılması gerektiği sonucuna varılmıştır.

Yapılan çalışmada WEKA yazılımında yararlanılmıştır. Bu yazılım içerisindeki hazır halde gelen klasik ve hibrit algoritmalar ile daha sonra eklenebilen algoritmalar denenmiştir. Klasik algoritmaların belli bir noktaya kadar verimli olduğu ancak yüzde yüze yakın bir sonuç veremediği görülmüştür. Ayrıca hibrit algoritmalar kullanmanın zamandan tasarruf sağladığı görülmüştür. Bu tasarruf hem eğitim aşamasında hızlı öğrenme hem de test aşamasında hızlı sonuç alma olarak çözüme yansımıştır.

Tez çalışmasında java ile yazılan ve platform bağımsız çalışan bir yazılımın olumlu etkileri görülmüştür. Yazılım açık kaynak kodlu olduğundan, günden güne daha fazla geliştirildiğinden, imkân ve kabiliyetleri arttığından akademik araştırmalarda ve öğrencilerin derslerinde kullanımı oldukça kolay ve verimlidir. Ayrıca sonuçların görsel ve matematiksel çıktılar şekilde sunulması, sonuçların yorumlanmasını kolaylaştırmaktadır.

Bu çalışmada KDD’99 veri seti kullanılmıştır. KDD’99 veri setinin kullanılmasındaki amaç bu çalışmada elde edilen sonuçların literatürdeki diğer çalışmalar ile daha doğru şekilde karşılaştırılabilmesidir.

Burada dikkat çeken en büyük problem, bazı örneklerden çok fazla varken bazılarından çok az örnek vardır. Bu KDD’99 veri setinden kaynaklanan başlıca problemlerden biridir. Her tür saldırıdan eşit miktarda örnek alınmamıştır. Bu durum ağın eğitilmesinde, öğrenmesinde problemler oluşturabilecektir. Ancak bu durumun tek bir çözüm yolu vardır o da KDD’99 veri setinin kullanılmamasındır. Ancak bu durumda da literatürdeki diğer çalışmalar ile karşılaştırma şansı olmayacaktır.

Benzer bir çalışma 2019 yılında yapılmış olup çalışma sırasında benzer sorunlar görülmüş ve çözümler geliştirilmiştir. Bahse konu çalışmada da weka yazılımı ve KKD’99 veri seti kullanılmış olup çok başarılı sonuçlar elde edilmiştir. İki çalışmada hibrit makine öğrenmesi tekniklerinin başarı oranının klasik makine öğrenmesi tekniklerine göre daha başarılı olduğunu ortaya çıkarmıştır [64].

Çalışmada ilk olarak KDD’99 veri setinin %10’luk bölümü olan veri seti seçilmiştir. Bu veri seti içerisinden öncelikle normal ve hizmet aksatma saldırıları dışında kalan saldırılar çıkarılmıştır. Böylece kullandığımız hibrit algoritmaların hizmet aksatma saldırıları üzerindeki sonuçları daha kesin olmuştur. Daha sonra bu veri seti normalize edilmiştir. Metin olarak bulunan nitelikler normalizasyon sırasında sayısal değerlere dönüştürülmüştür. Böylece daha hızlı ve doğru sonuçlar elde edilmiştir.

İkinci olarak bu veri seti eğitim ve test veri seti olarak iki parçaya ayrılmıştır. Bu ayrım sırasında her iki sette de yaklaşık olarak eşit sayıda saldırı ve normal bağlantı olmasına dikkat edilmiştir.

Üçüncü olarak oluşturulan veri setleri her algoritma üzerinde denenmiş ve sonuçlar kayıt altına alınmıştır. Burada hem klasik algoritmalar hem de hibrit algoritmalar ile deneme yapılmıştır. Ancak literatürdeki klasik algoritmalar ile yapılan çalışmaların sonuçları daha yüksek oranda doğru olduğundan bu çalışmada bulunan klasik algoritma sonuçları değil literatürdeki araştırma sonuçları örnek olarak verilmiştir.

Dördüncü olarak Fonksiyonel Ağaç algoritması ile denemeler yapılmıştır. Bu algoritmada Lojistik Regresyon algoritması ile Karar ağacı algoritması hibrit olarak kullanılmıştır. Eğitim sırasında %100 öğrenme sağlanmasına rağmen test aşamasında belli oranda bir düşüş olduğu görülmüştür. Yine de test işlemi sonucunda %99,76'lık bir başarı elde edilmiştir.

Beşinci olarak Lojistik Model Ağacı algoritması ile denemeler yapılmıştır. Bu algoritmada lojistik regresyon algoritması ile karar ağacı algoritması hibrit olarak kullanılmıştır. Eğitim aşamasında %100 başarı oranı elde edilmiş ancak test aşamasında bu algoritmada da düşüş olduğu görülmüştür. Bu düşüşe rağmen test sonucunda %99,55'lik bir başarı elde edilmiştir.

Altıncı olarak Naive Bayes Ağacı algoritması ile denemeler yapılmıştır. Bu algoritmada Naive Bayes algoritması ile Karar Ağacı algoritması hibrit olarak kullanılmıştır. Bu algoritma da öğrenmenin %100 olduğu bir algoritmadır. Ancak bu algoritmada da belli oranda düşüş vardır. Bu düşüş oranı çok düşüktür ve test sonucunda %99,86'lık bir başarı elde edilmiştir.

Yedinci olarak Karar Masası ve Naive Bayes (KMNB) algoritmalarının hibrit olarak kullanıldığı algoritma ile denemeler yapılmıştır. Bu algoritma da eğitim aşamasında çok yakın olmasına rağmen %100 başarı oranı elde edilememiştir. Ancak diğer algoritmalarda olduğu gibi test başarı oranı eğitim başarı oranından düşük elde edilmiştir. Başarı oranında düşüş olmasına rağmen test sonucunda %99,95'lik bir başarı elde edilmiştir.

Yapılan çalışmada genel olarak ülkemizde ve dünyada devam eden araştırmalara yakından bakılmış ve gelişmeler takip edilmiştir. Bu noktada görülmüştür ki ülkemizde bu konuda akademik çalışmalar çok olmasına rağmen ticari bir ürün geliştirme konusunda büyük eksiklikler vardır. Bu konuda büyük bir atılım gerekmektedir.

KAYNAKLAR

1. Akyol, A., Hacıbeyoğlu, M., Karlık, B. (2016). Desing of multilevel hybrid classifier wiyh variant feature sets for intrusion detection system. *IEICE Transactions on Information and Systems,* 99(7), 1810-1821.

2. Richards, K. (1999). Network based intrusion detection: a review of technologies. *Computers & Security,* 18, 671-682.

3. Hancock, B. (1998). *Automated ıntrusion detection systems and network security*. Network Security, 14-15.

4. White, G., Pooch, V. (1996). Cooperating security managers: distributed intrusion detection systems. *Computers & Security,* 15, 441-450.

5. Mahdavi, E., Fanian, A., Amini, F. (2020). A real-time alert correlation method based on code-books for intrusion detection systems. *Computers & Security,* 89, 51-62.

6. Anderson, J. P. (1972). Computer security technology planning study. *Electronic System Division*, 2, 7-11.

7. Anderson, J. P. (1980). *Computer security threat monitoring and surveillance technical report,* James P. Anderson Co.

8. Bace, R. G. (2000). *Intrusion detection* (Second edition). Michigan: Macmillan Technical Publishing, 223-250.

9. Denning, D. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering,* 13, 222-232.

10. Shaker A. A., Sharad, G. (2011). Intrusion detection system (IDS): case study. *International Conference on Advanced Materials Engineering*, 15.

11. Schaffrath, G., Sadre, R., Morariu, C., Pras, A. and Stiller, B. (2010). *An overview of IP flow-based intrusion detection.* Communications Surveys & Tutorials.

12. Prasenna, P., Krishna, K. R., Ramana, R. and Devanbu, A. (2012). Network Programming And Mining Classifier For Intrusion Detection Using Probability Classification. *Pattern Recognition, Informatics and Medical Engineering*, 21-23.

13. Chang, R. K. C. (2002). Defending against flooding-based distributed denial-of-service attacks: a tutorial. *Communication Magazine,* 40(10).

14. Jyothsna, V., Prasad V. V. R., Prasad, K. M. (2011). A review of anomaly based ıntrusion detection systems. *International Journal of Computer Applications,* 28(7), 26-35.

15. Axelsson, S. (2000). Intrusion detection systems: a survey and taxonomy. *Dept. of Computer Eng., Chalmers University of Technology*, Göteborg, Sweden, 1-23.

16. Shijoe, J. Malathi, D., Bharath, R., Dorathi J. (2018) A survey on anomaly based host ıntrusion detection system. *Journal of Physics: Conf. Series*, 1000, 1-10.

17. Hung-Jen, L., Richard, L. C., Ying-Chih, L., Kuang-Yuan, T. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36, 16–24.

18. Mohammed, O. S., Alsohybe, N. T., Ba-Alwi, F. M., Thabit, Z. A. (2018). Survey on intrusion detection system types. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, 7(4), 444-462.

19. Liao, H., Lin, C. R., Lin, Y., Tung, K. (2013). Intrusion detection system: a comprehensive review. *Journal of Network and Computer Applications*, 36, 16–24.

20. Santos, K. B., Phani, R. T. C. S., Ratnakar, M., Dawood, B. S., Sudhakar, N. (2013). Intrusion Detection System- Types and Prevention. *International Journal of Computer Science and Information Technologies*, 4(1), 77 – 82.

21. Abhishek, P., Harsha, Bhat, Vaibhav, S., Nalini M. (2015). Classification of Intrusion Detection System. *International Journal of Computer Applications*, 118(7).

22. Tran, H., Huh, E., Minho, J. (2009). A lightweight ıntrusion detection framework for wireless sensor network. *Wireless Communications and Mobile Computing*, 559-572.

23. Yu Chen, Kai Hwang, Wei-Shinn Ku (2007). Collaborative detection of ddos attacks over multiple network domains. *IEEE Transactions on Parallel and Distributed Systems*, 18(12).

24. Patel, A., Taghavi, M., Bakhtiyari, K., Júnior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications*, 36(1), 25-41.

25. Shelke, P. K., Sontakke, S., Gawande, A. D. (2012). Intrusion detection system for cloud computing. *International Journal of Scientific & Technology Research*, 1(4), 67-71.

26. Depren, O., Topallar, M., Anarim, E., and 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.

27. Koshal, J. and M. Bag, (2012). Cascading of C4.5 decision tree and support vector machine for rule based ıntrusion detection system. *International Journal of Computer Network and Information Security (IJCNIS)*, 4(8), 8-20.

28. Wang, G., Hao, J., Ma, J., Huang, L. (2010). A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert Systems with Applications*, 37, 6225–6232.

29. Mostaque, M. M. H. (2013). Network ıntrusion detection system using genetic algorithm and fuzzy logic. *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, 1(7).

30. Al-Yaseen, W. L., Othman, Z. A., Nazri, M. Z. A. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems With Applications*, 67, 296-303.

31. Aljawarneh, S., Aldwairi, M., Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, 25, 152-160.

32. Farid, D. M., Harbi, N., Rahman, M. Z. (2010). Combining naive bayes and decision tree for adaptıve ıntrusion detection. *International Journal of Network Security & Its Applications (IJNSA)*, 2(2).

33. Muda, Z., Yassin, W., Sulaiman, M.N., Udzir, N.I., (2011). Intrusion detection based on k-means clustering and naive bayes classification. *7th International Conference on IT in Asia (CITA).*

34. Hosseini, S., Zade, B. M. H. (2020). New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN. *Computer Networks*, 173(22).

35. Bani-Hani, R. M., Zaid, A. (2013). SYN flooding attacks and countermeasures: a survey. *The 4th International Conference on Information and Communication Systems.*

36. Saravanan, K., Asokan, R. (2011). Distributed denial of service (ddos) attacks detection mechanism. *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, 1(5).

37. Choudhary, K., Meenakshi, S. (2011). Smurf attack: attacks using ICMP. *International Journal of Computer Science and Technology*, 2(1).

38. Khatak, A., Maini, R. (2016). Comparative analysis of different denial of service attacks. *The International Journal Of Engineering And Science (IJES)*, 5(5), 32-35.

39. Fekadu, Y., Eman, A. Ammar, O. (2018). Analysis of ping of death DoS and DDoS attacks. *Proceedings of IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, USA.

40. Mukkamala, S., Janoski, G., Sung, A. (2002). Intrusion detection using neural networks and support vector machines. *IEEE International Joint Conference on Neural Networks, IEEE Computer Society Press*, 1702-1707.

41. Iftikhar, A., Azween, B. A., Abdullah, S. A. (2009). Artificial neural network approaches to intrusion detection: a review. *Telecommunications and Informatics, Included in ISI/SCI Web of Science and Web of Knowledge*, 200-205.

42. Labib, K., Vemuri, V.R. (2004). Detecting and visualizing denial-of-service and network probe attacks using principal component analysis. *Third Conference on Security and Network Architectures.*

43. Shiruru, K. (2016). An introduction to artificial neural network. *International Journal of Advance Research and Innovative Ideas in Education,* 27-30.

44. Sağıroğlu, Ş., Beşdok, E., Erler, M. (2003). *Mühendislikte yapay zekâ uygulamaları-1: Yapay sinir ağları*. Kayseri: Ufuk Kitabevi, 10-100.

45. Karaboğa, D. (2017). *Yapay Zekâ Optimizasyon Algoritmaları* (Beşinci Baskı). Ankara: Nobel Kademik Yayıncılık, 145-180.

46. Javidi, M. M., Fard, R. H. and Jampour, M. (2015). Research in random parameters of genetic algorithm and ıts application on tsp and optimization problems. *Walailak J Sci & Tech,* 12(1), 27-34.

47. Kumar, N., Karambir, R. K. (2012). A comparative analysis of pmx, cx and ox crossover operators for solving travelling salesman problem. *International Journal of Latest Research in Science and Technology*, 1(2), 98-101.

48. Engin, O. (2001). *Akış tipi çizelgeleme problemlerinin genetik algoritma ile çözüm performansının arttırılmasında parametre optimizasyonu*. Yayınlanmamış Doktora Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

49. Masud, K., Rashedur, R. (2013). Decision tree and naive bayes algorithm for classification and generation of actionable knowledge for direct marketing. *Journal of Software Engineering and Applications,* 6, 196-206.

50. Hall, M., Frank, E. (2008). Combining naive bayes and decision tables. *Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS),* 318-319.

51. Shaker, A. A., Sharad, G. (2011). Intrusion detection system (IDS): case study. *International Conference on Advanced Materials Engineering*, 15.

52. Perlich, C., Provost, F., Simonoff, J. (2003). Tree inductions vs. logistic regression: a learning-curve analysis. *Journal of Machine Learning Research 4*, 211–255.

53. Landwehr, N., Hall, M., Frank, E. (2005). Logistic model trees. *Machine Learning,* 59(1-2), 161-205.

54. Gama, J. (2004). Functional trees. *Machine Learning,* 55(3), 219-250.

55. Vasconcellos, E. C., de Carvalho, R. R., Gal, R. R., LaBarbera, F. L., Capelato, H. V., Velho, H. F. C., Trevisan, M., Ruiz, R. S. R., (2011). Decision tree classifiers for star/galaxy deparation. *Astronomical Journal,* 141(6), 1-17.

56. Rodriguez, J. J., Garcia-Osorio, C., Maudes, J., Diez-Pastor, J. F. (2010). An experimental study on ensembles of functional trees. *Multiple Classifier Systems*, 5997, 64-73.

57. Witten, I. H., Frank, E., Hall, M. A. (2011). *Data mining: practical machine learning tools and techniques* (Third edition), Burlington (MA): Morgan Kaufman.

58. Kohavi, R. (1996). Scaling up the accuracy of naive bayes classifiers: a decisiontree hybrid. *Proc 2nd International Conference on Knowledge Discoveryand Data Mining*, 202–207.

59. Kohavi, R. (1995). *Wrappers for performance enhancement and oblivious decision graphs.* Yayımlanmamış Doktora Tezi, Stanford University, Computer Science Department, Kaliforniya.

60. Prabha, S. L., Shanavas, A. R. M. (2015). Application of educational data mining techniques in e-learning- a case study. *International Journal of Computer Science and Information Technologies,* 6(5), 4440-4443.

61. Aggarwal, P., Sharma, S. K. (2015). Analysis of kdd dataset attributes-class wise for ıntrusion detection. *Procedia Computer Science*, 57, 842-851.

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

63. Jyothsna, V., Prasad, V. V. R., Prasad, K. M. (2011). A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, 28(7).

64. Doğru, E., Barışçı, N. (Aralık 2019). *Makine öğrenmesi teknikleri kullanılarak hizmet aksatma saldıları tespiti*. 3'üncü Anadolu Uluslararası Uygulamalı Bilimler Kongresi, Diyarbakır.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : DOĞRU, Emin

Uyruğu : T.C.

Doğum tarihi ve yeri : 01.01.1988, Mersin

Medeni hali : Evli

Telefon : 0 (546) 494 73 45

E-mail : dogruemin@hotmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Mühendisliği	Devam ediyor
Lisans	Hava Harp Okulu / Bilgisayar Mühendisliği	2010
Lise	Işıklar Askeri Lisesi	2006

İş Deneyimi

Yıl	Yer	Görev
2010-Halen	Hava Kuvvetleri Komutanlığı	Subay

Yabancı Dil

İngilizce

Yayınlar

1. Doğru, E., Barışçı, N. (Aralık 2019). *Makine öğrenmesi teknikleri kullanılarak hizmet aksatma saldıları tespiti. 3'üncü Anadolu Uluslararası Uygulamalı Bilimler Kongresinde sunuldu*, Diyarbakır.

Hobiler

Kitap okumak, Yan flüt, Yüzme

GAZİ GELECEKTİR...