VERİ MADENCİLİĞİ TEKNİKLERİYLE SALDIRI TESPİTİ
VE
BİR UYGULAMA

Elmas YILDIZ

YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

Mayıs 2010
ANKARA

Elmas YILDIZ tarafından hazırlanan VERİ MADENCİLİĞİ TEKNİKLERİYLE SALDIRI TESPİTİ VE BİR UYGULAMA adlı bu tezin Yüksek Lisans/Doktora tezi olarak uygun olduğunu onaylarım.

.....

Yrd. Doç. Dr. Nursal ARICI

Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile..... Anabilim Dalında Yüksek lisans/Doktora tezi olarak kabul edilmiştir.

Başkan : Doç. Dr. Ahmet COŞAR

Üye : Yrd. Doç. Dr. Nursal ARICI (Danışman)

Üye : Yrd. Doç. Dr. Nurettin DOĞAN

Üye : _____

Üye : _____

Tarih : ...../……/……

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Elmas YILDIZ

VERİ MADENCİLİĞİ TEKNİKLERİYLE SALDIRI TESPİTİ

VE

BİR UYGULAMA

(Yüksek Lisans Tezi)

Elmas YILDIZ

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Mayıs 2010

ÖZET

Bilginin her geçen gün daha da önem kazanması ile birlikte arttırılan güvenlik önlemleri arasında saldırı tespit sistemleri de yer almaktadır. Saldırı Tespiti Sistem'leri; bilgisayar sistemine ve ağ kaynaklarına olan saldırıları tespit etmek, saldırıların kimden geldiğini tanımak, sistemi izleyip anormal olan durumları saptamak ve bunlara karşı gerekli önlemleri almayı amaçlayan güvenlik sistemleridir. Sistemlerdeki anormal olan durumları saptamak ise veri madenciliği konusu içinde yer almaktadır.

DoS atakları; bir ağa ya da kaynağa ulaşımı engellemek için kullanılır ve kötü niyetli kullanıcıların saldırılarını kolaylıkla gerçekleştirebilmeleri açısından ilk deneyecekleri saldırı tipleridir. Bu ataklardan birisi olan Brute Force saldırısı; sistemin kullanıcı hesabını ve şifresini kırmak için tahminlere dayalı deneme yanılma yöntemini kullanır.

Bu tezde, veri madenciliği tekniğini kullanarak nasıl saldırı tespit edileceği açıklanmış ve DoS ataklarından biri olan Brute Force tipi bir saldırının veri

madenciliği teknikleri ile tespit edilip engellendiği bir saldırı tespit uygulaması tasarlanmıştır. Tasarım Visual Studio.NET 2005 C# ve Microsoft Office 2007 Access programları kullanılarak geliştirilmiştir. Uygulamada geliştirilen saldırı tespit sistemi saldırıyı ip tabanlı tespit etmesi ve gerçek zamanlı bir saldırı tespit sistemi olmasından dolayı önemlidir.

Bilim Kodu **:** 704.3.006

Anahtar Kelimeler **:** Saldırı Tespit Sistemleri, Veri Madenciliği, DoS Saldırıları

Sayfa adedi **:** 96

Tez yöneticisi **:** Yrd. Doç. Dr. Nursal ARICI

INTRUSION DETECTION SYSTEMS WITH DATA MINING TECHNICS AND

AN APPLICATION

(M. Sc.Thesis)

Elmas YILDIZ

GAZİ UNIVERSITY

INSTITUTE of INFORMATICS

May 2010

ABSTRACT

As importance of information is increasing day by day, security measures are also enhanced simultaneously. Intrusion detection systems are among them. Intrusion detection systems are security systems that detect attacks on computer systems and network sources, identify the source of attacks, track system to identify abnormal situations and aim to take necessary measures against them. Detection of abnormal situations in systems is included in data mining topic.

Dos attacks ,one of the intrusion types, block access to a network or to a source and they would be easily performed and tried first by malicious users. One of these attacks - Brute Force attack- uses estimation based trial and error method to crack system's user accounts and passwords.

In this thesis, how to use data mining techniques to detect intrusions is presented. An intrusion detection application in which one of the DoS attacks - Brute Force type attack - detected and blocked by data mining systems is proposed.Proposed application is developed using C# on Microsoft Visual Studio .NET 2005 and Microsoft Office 2007 Access. Importance of intrusion

detection system in this proposed application arises from detection of attacks ip-based and being a real-time intrusion detection system.

Science Code :704.3.006

Keywords : Intrusion Detection Systems, Data Mining, DoS attacks

Page number : 96

Adviser : Assist. Prof. Dr. Nursal ARICI

TEŞEKKÜR

Bu çalışmanın gerçekleşmesine katkılarından dolayı ve danışmanım olarak tezin yazılmasında yol gösterdiği için sayın hocam Yrd. Doç. Dr. Nursal Arıcı' ya içtenlikle teşekkür ederim. Çalışmalarım boyunca manevi desteği ve değerli katkıları ile her zaman yanımda olan eşim Erdem YILDIZ' a teşekkür ederim. Ayrıca tezimi, bu çalışma esnasında dünyaya gelen sevgili kızım Zeynep YILDIZ' a ithaf ediyorum.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

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

Kısaltmalar	Açıklama
ART	Adaptive Resonance Theory (Adaptif Rezonans Teorisi)
BP	Back Propagation (Geri Yayılım)
DARPA	Defence Advanced Research Projects Agency (Savunma İleri Araştırma Projeleri Teşkilatı)
DBD	Delta Bar Delta
DoS	Denial of Service (Hizmet Engelleme)
FTP	File Transfer Protocol (Dosya Aktarma Kuralları)
IIS	Internet Information Server (İnternet Bilgi Sunucusu)
IP	Internet Protocol (İnternet Kuralları)
KDD	Knowledge Discovery and Delivery (Bilgi Keşfi ve Teslimatı)
LM	Levenberg-Marquardt
LVQ	Learning Vector Quantisation (Vektör Kuantalama Öğrenme)
MIT	Massachusetts Institute of Technology
MLP	Multilayer Perceptron (Çok Katmanlı Perseptron)
QP	Quick Propagation (Hızlı Yayılım)

RBFN	Radial Basis Function Networks (Radyal Tabanlı Fonksiyon Ağları)
RP	Resillient Propagation (Esnek Yayılım)
ROC	Receiver Operating Characteristic (Alıcı Karakteristiği)
SOM	Self-Organizing Maps (Kendi Kendini Düzenleyen Haritalar)
STS	Saldırı Tespit Sistemi (Intrusion Detection System)
TCP	Transmission Control Protocol (Transfer Kontrol Kuralları)
VM	Veri madenciliği (Data Mining)
VPN	Virtual Private Network (Sanal Özel Ağ)
YSA	Yapay Sinir Ağı (Artificial Neural Network)

1. GİRİŞ

Sayısal veya mantıksal her türlü değer bir veridir. Bilgi, insanların çeşitli yollarla edindikleri ve zihinsel bir süreçten geçirerek içselleştirdikleri düşüncelerin, yeniden anımsanması ya da başkalarıyla paylaşılması amacıyla çeşitli ortamlarda kayıt altına alınmasıyla ortaya çıkan işlenmiş verilerdir [1].

Verilerin dijital ortamda saklanmaya başlanması ile birlikte, yeryüzündeki bilgi miktarının yaklaşık olarak her yirmi ayda bir kendini iki katına çıkardığı öne sürülmektedir [1]. Veritabanı teknolojisinin de artan veri miktarına paralel olarak gelişmesi ile birlikte veri saklama eskiye oranla kolaylaşmıştır. Böylece kurum ve kuruluşların veri tabanlarında çok büyük miktarlarda veri yığınları oluşmuştur.

Bilgisayar sistemleri ile üretilen bu veriler tek başlarına değersizdir. Çünkü çıplak gözle bakıldığında bir anlam ifade etmezler. Bu veriler belli bir amaç doğrultusunda işlendiği zaman bir anlam ifade etmeye başlar. Bu yüzden büyük miktardaki verileri işleyebilen teknikleri kullanabilmek büyük önem kazanmaktadır. Bu ham veriyi bilgiye veya anlamlı hale dönüştürme işlemleri veri madenciliği (VM) ile yapılabilmektedir [2].

Karmaşık formatlardaki veri yığınları içinden değerli bilgi parçaları ile gerçek bilgiyi ayıklamak ve kullanılır hâle getirmede veri madenciliği tekniği önemli bir yer tutmaktadır. Bundan dolayı, günümüzde gerçek bilgiye ulaşmak kadar, ulaşılan bilginin sağlıklı yorumlanması, yani bilgi yığınına saklanmış altın bilginin çıkartılması da oldukça önem arz etmektedir. Bu yapılmadığı takdirde, insanlar bilgi kirlenmesine maruz kalacak, üretilen ve toplanan veriler de manasız bilgi yığınlarına dönüşmüş olacaktır.

“Günümüzde veri her geçen gün katlanarak artmakta ve her türlü veriye erişim kolaylaşmaktadır” sözünü hemen her yerde duymaktayız. Ancak aradığımız bilgiye erişim o kadar da kolay olmamaktadır. Her türlü veri elimizin altında iken bilgiye ulaşmak oldukça zor olabiliyor. Çünkü bilgiyi ararken İnternet’te veri kümeleri

arasında kayboluyoruz. Bu yüzden gerçek bilgi değerlidir ve korunması gerekmektedir.

Bilgi çağını yaşadığımız şu günlerde, e-devlet, e-imza, e-ticaret gibi kavramlardan oldukça sık bahsedilmektedir. Gerek hız ve verimlilik artışı, gerekse kolaylık sağlaması nedeniyle birçok bilgi elektronik ortamlara aktarılmıştır. Ancak, kişisel veya kurumsal açıdan önemli bir bilginin, başkalarının eline geçmesi ile maddi ve manevi zararlara yol açabileceği görülmüştür. Bu nedenle, elektronik ortamların yaygınlaşarak kullanılmaya başlaması ile birlikte, zaten önemli olan bilgi ve bilgisayar güvenliği kavramının önemi günümüzde daha da artmıştır. Birçok kurumsal ve ticari firma bilgi güvenliğine verdikleri önemi öncelikli hale getirmiştir. Geliştirilen e-devlet, e-kurum gibi projelerde güvenliğin en üst düzeyde tutulması ulusal bir amaç haline gelmiş, bu konuda hukuki ve teknolojik önlemler geliştirilmiştir [3].

Bilginin bu kadar değerli olduğu günümüzde bilgi güvenliği, veritabanlarında kritik bilgiler bulunan tüm kamu kurumlarının, üniversitelerin, bankaların ve özel kuruluşların en önemli konusu haline gelmiştir. Dolayısıyla bu bilgilerin korunması için bilgi işlem yöneticileri güvenlik duvarları, antivirüs yazılımları, elektronik imza uygulamaları, saldırı tespit sistemleri ve saldırı engelleme sistemleri vb. teknolojilerini de kullanarak güvenlik politikalarını oluşturmaktadırlar.

Günümüzün vazgeçilmez bir parçası olan bilgi işlemde bilginin gizliliği, özgünlüğü ve bütünlüğü devamlı bir saldırı altındadır. Bilgisayar korsanları (hacker) tarafından devamlı bu sistemler saldırıya uğramakta ve bilgiler ele geçirilmeye çalışılmaktadır. Bu nedenle kamu kurum ve kuruluşları ile büyük firmalar saldırılara karşı çok büyük yatırımlar yapmaktadır.

Saldırı, bilginin mahremiyetini, bütünlüğünü ve erişilebilirliğini tehlikeye atabilecek girişimlerin kümesi olarak tanımlanmaktadır [4]. Saldırı tespiti ise, bir bilgisayar sisteminde veya ağda meydana gelen olayları izleyerek, bilginin mahremiyetini, bütünlüğünü ve erişilebilirliğini bozmak ya da sistemin güvenlik mekanizmalarını

aşmak için yapılan hareketler olarak tanımlanan saldırı işaretlerini analiz etme işlemidir [5].

Günümüzde bilgi ve bilgisayar güvenliğinin öneminin kavranmasıyla, geliştirilen araçlardan biri olan Saldırı tespit sistemleri (STS), saldırılara karşı sistemimizde alarm niteliği taşıyan yazılım ve donanımlardır. STS' lerin kullanılması ile sistemlere yapılan yetkisiz erişimler ve kötüye kullanımlar tespit edilerek, bunların yol açabileceği zararlar engellenmiş olur. Bilgisayar sistemlerinde STS' lerin kullanılması ile birlikte, sisteme ne tür saldırıların daha çok yapıldığı, sistemdeki mevcut açıklar ve saldırganlar hakkında daha detaylı bilgiler elde edilebilir [6].

İlk olarak 1980' de Anderson' un yaptığı çalışmalar sonucunda ortaya çıkan STS' ler, ardından yapılan birçok çalışma ile hızla gelişmesini devam ettirmiştir [7]. STS' lerin geliştirilmesinde günümüze kadar istatistiksel yöntemlerin dışında, kural tabanlı (rule based), eşik değeri belirleme (threshold value), durum geçiş diyagramları (state transition diagrams), yapay sinir ağları (artificial neural networks), yapay bağışıklık sistemi (artificial immune system), bulanık mantık (fuzzy logic), veri madenciliği (data mining) gibi farklı birçok yaklaşım uygulanmaktadır [6].

Veri Madenciliği (VM), büyük miktarlardaki verinin içinden geleceğin tahmin edilmesinde yardımcı olacak anlamlı ve yararlı bağlantı ve kuralların bilgisayar programlarının aracılığıyla aranması ve analizidir. VM aracılığıyla, büyük veri kümelerinden oluşan veritabanı sistemleri içerisinde gizli kalmış bilgilerin çekilmesi sağlanır. Bu işlem, istatistik, matematik disiplinleri, modelleme teknikleri, veritabanı teknolojisi ve çeşitli bilgisayar programları kullanılarak yapılır [8].

Ağ güvenliğinin devamlı olarak sağlanmasının yolu yetkisiz erişimleri gerçek zamanlı olarak tespit etmektir. Etkili bir saldırı tespit sistemi, atak ve şüpheli ağ erişimlerini yetkin bir şekilde tespit ederek kurumsal ağ güvenliğinin bir bacağını oluşturur. Ama bu istenmeyen trafiğin sadece saptanması yeterli değildir. Kullanılan saldırı tespit uygulaması, istenmeyen bağlantılara anında yanıt verebilmeli ve ağ kaynaklarına yetkisiz erişimi engellemelidir. İyi dizayn edilmiş bir saldırı tespit uygulaması, gerçek zamanlı olmasının yanında kapsamlı kayıt tutabilme, komple

denetim ve gerektiğinde ilgili kişileri ikaz edebilecek gelişmiş uyarı mekanizmalarına sahip olmalıdır.

Denial of Service (DoS) saldırıları ağın veya cihazların hizmet dışı kalmasını sağlayan bir atak tekniğidir. Amacı bilgiyi çalmak değil, ağın veya cihazların iş göremez hale gelmesidir. DoS ataklarından biri olan Brute Force saldırısı ise sistemin kullanıcı hesabını ve şifresini kırmak için tahminlere dayalı deneme yanılma yöntemini kullanan bir saldırı tekniğidir. Bu tez çalışması kapsamında geliştirilen yazılım projesinde DoS atakları seçilmiştir. Çünkü DoS atakları çok basit ve ilkel bir yöntem olmasına rağmen hala güncel bir saldırı tekniğidir. DoS atakları arasından Brute Force tipi saldırının seçilmesinin nedeni ise veri madenciliği tekniği ile saldırı tespitinin rahatlıkla uygulanabilmesidir. Uygulama projesindeki saldırı tespit yazılımı geliştirilip diğer DoS atakları da tespit edilebilir.

Literatürde yapılan çalışmalar incelendiğinde genellikle mevcut STS' ler ve bunların tasarlanmasında veri madenciliği tekniğinden nasıl faydalanıldığı anlatılmıştır. Çalışmalara bakıldığında STS' ler geliştikçe, yapılan saldırıların ve saldırı türlerinin de arttığı görülmüştür. STS'ler için geliştirilen, literatürdeki veritabanı uygulamaları incelendiğinde, araştırmacıların uygulamalarında kullanılabilecekleri güncel saldırı türlerini içeren bir veritabanına rastlanmamıştır. Güncel bir veri kümesi olmayışı veya oluşturulamayışı STS' lerin başarısını engelleyen bir etken olarak karşımıza çıkmaktadır.

Gerçek zamanlı STS çalışmalarının pek çoğunda önceden hazırlanmış veri kümelerinin kullanılmış olduğu tespit edilmiştir. Yapılan bu çalışma ile birlikte gerçek zamanlı bir STS tasarımının nasıl yapılabileceği, STS tasarımını yaparken kullanılabilecek güncel bir veri kümesinin nasıl oluşturulabileceği ile ilgili yaklaşımlar ortaya konulmaya çalışılmıştır.

Bu tez çalışması kapsamında veri madenciliği tekniğini kullanarak DoS ataklarından Brute Force tipi saldırıları kullanıcı bazlı ve IP tabanlı tespit eden gerçek zamanlı bir saldırı tespit sistemi yazılım projesi geliştirilmiştir. Bu projede amaç, tespit edilen kullanıcı ve IP adreslerinin sisteme girişini engellemek ve e-posta ile sistem

yöneticisine bildirmektir. Bu amaç doğrultusunda geliştirilen saldırı tespit sistemi yazılımı; kullanıcı adı, IP adresi, tarih ve saat gibi bilgileri sistem yöneticisine sunmaktadır.

Bu tez çalışmasındaki amaç, STS' lerde veri madenciliği kullanmanın işlevselliğini bir uygulama yazılımı ile göstermektir. Bu çalışma ve geliştirilen yazılım projesi bilgi güvenliği ile ilgili çalışan sistem yöneticilerine ve uzmanlarına fikir verecek ve kullandıkları STS'nin ağlarının güvenlik seviyesine göre geliştirebilmelerine yardımcı olacak niteliktedir.

Tez çalışması yedi bölümden oluşmuştur. Tezin ikinci bölümünde, bilgi güvenliği konusunun önemli kavramlarından olan saldırı ve saldırı tipleri, üçüncü bölümünde saldırı tespit sistemleri açıklanmaktadır.

Dördüncü bölümde saldırı tespit sistemlerinde kullanılan teknikler hakkında bilgi verilmiş, beşinci bölümde bu tekniklerden biri olan veri madenciliği konusu ele alınmıştır.

Altıncı bölümde veri madenciliği tekniği ile gerçekleştirilen bir saldırı tespit sistemi yazılım projesi açıklanmıştır.

Son olarak yedinci bölümde saldırı tespit sistemlerinin öneminden bahsedilmiş ve saldırı tespit sistemi yazılımında neden veri madenciliği tekniklerinin kullanıldığı açıklanmıştır.

2. SALDIRI ve SALDIRI TİPLERİ

2.1. Saldırı Tanımı

Bilgi ve bilgisayar güvenliğinde; karşı taraf, genel olarak kötü niyetli olarak nitelendirilen kişiler (korsanlar) ve yaptıkları saldırılardır. Var olan bilgi ve bilgisayar güvenliği sistemini aşmak veya atlatmak; zafiyete uğratmak; kişileri doğrudan veya dolaylı olarak zarara uğratmak; sistemlere zarar vermek, sistemlerin işleyişini aksattırmak, durdurmak, çökertmek veya yıkmak gibi kötü amaçlarla bilgisayar sistemleri ile ilgili yapılan girişimler, saldırı veya atak olarak adlandırılmaktadır. Saldırganlar, amaçlarına ulaşmak için çok farklı teknikler içeren saldırılar gerçekleştirmektedirler. Saldırı türlerinin bilinmesi, doğru bir şekilde analiz edilmesi ve gereken önlemlerin belirlenmesi, bilgi güvenliği için büyük bir önem arz etmektedir [9].

Kurum ve şahısların sahip oldukları bilgilere yetkisiz erişmek, zarar vermek, maddi ve manevi kazanç sağlamak vb. için bilişim sistemleri kullanılarak yapılan her türlü hareket saldırı olarak nitelendirilir [10].

Saldırı; yetkisiz erişimlerle sistemin kırılmaya veya kaynakların yanlış kullanılmaya çalışılmasıdır [11].

2.2. Saldırı Sebepleri

Saldırılar genellikle; hatalı ve eksik yetkilendirmelerde, zayıf şifreler kullanıldığında, sistem yanlış yapılandırıldığında ve yazılım kaynaklı açıklıklar bulunduğunda meydana gelir. Başarılı saldırı girişimleri nüfuz olarak tanımlanır. Saldırıların önlenmesinde saldırı tespit sistemleri güvenlik duvarının arkasında, ağ içerisinde çalışırlar [10].

Bilgi ve bilgisayar sistemlerine saldırılar, teknolojik gelişmelerdeki artışa paralel olarak artmaktadır. Saldırganların ulaşmak istedikleri bilgi ya da zarar vermek istedikleri bilgisayar sistemleri bazen çok küçük şeyler olabileceği gibi çok büyük hedefler de olabilir. Genel olarak değerlendirildiğinde, amaçları; merak, kendini

tatmin etme, şaka yapma, zarar verme, etkisiz hale getirme, para kazanma, itibarını azaltma, kişisel mahremiyeti öğrenme, politik çıkar elde etme, kişisel, kurumsal veya ulusal çıkarları tahrip etme ve/veya kendi çıkarlarına hizmet etme gibi konuları kapsamaktadır.

2.3. Saldırı Çeşitleri

Bilgisayar sistemlerinde en genel anlamda sızma ya da saldırı makine başından yapılacak izinsiz erişimlerden başlar ve çok geniş bir spektruma yayılır. Bilgisayar ağları söz konusu olduğunda ise saldırılar sadece bu tip kullanıcı ve erişim temelli saldırılar ile sınırlı kalmaz. Ağ üzerinden yapılan saldırılar günümüzde en sık karşılaşılan problemlerdir.

Ağ üzerinden yapılan saldırılar 4 temel kategoriye ayrılırlar [12].

2.3.1. Bilgi tarama (probe ya da scan)

Bu saldırılar bir sunucunun ya da herhangi makinanın, geçerli IP adreslerini, aktif giriş kapılarını (port) veya işletim sistemini öğrenmek için yapılan saldırılardır [12].

Örneğin;

- Belirli bir portu sürekli tarama saldırısı (ipsweep).
- Bir sunucu üzerindeki hizmetleri bulmak için tüm portları tarar (Portsweep).

2.3.2. Yönetici hesabı ile yerel oturum açma (remote to local - r2l)

Kullanıcı haklarına sahip olunmadığı durumda misafir ya da başka bir kullanıcı olarak izinsiz erişim yapılmasıdır [12].

Örneğin;

- Unix işletim sistemi üzerinde çalışan bir trojan saldırısıdır (Sshtrojan).
- Tahmini kolay şifreleri bularak sisteme girilmesidir (guest).

2.3.3. Kullanıcı hesabının yönetici hesabına yükseltilmesi (user to root - u2r)

Bu tip saldırılarda sisteme girme izni olan fakat yönetici olmayan bir kullanıcının yönetici izni gerektirecek işler yapmaya çalışmasıdır [12].

Örneğin;

- Solaris üzerinde eject programı ile tampon taşmasına (overflow) yol açıp, yönetici haklarına sahip olunmasıdır (Eject).

- Sql veritabanı kurulu Linux makinalarda sunucuya bağlanan kullanıcının belirli komutlarla yönetici hakları ile komut satırı elde etmesidir (Sqlattack).

2.3.4. Hizmet engelleme (Denial of Service - DoS)

TCP/IP protokol yapısındaki açıklardan faydalanarak veya bir sunucuya çok sayıda istek yönelterek sunucunun iş göremez hale gelmesini sağlayan saldırılardır. Denial of Service (DoS) atakları İnternet'e bağlı olan ağları ve cihazları hedef alır. Bu tip atakların amacı bilgiyi çalmak değil, ağları veya cihazları iş göremez hale getirmektir. Bu sayede kullanıcılar artık ağ kaynaklarına erişemeyeceklerdir. DoS saldırıları günümüzde en yaygın kullanılan saldırı biçimlerinden biridir. Çünkü birçok saldırı yöntemini içinde barındırır. DoS saldırıları genelde bir ağa ya da kaynağa ulaşımı engellemek için kullanılır [12]. Kendi içinde mantıksal olarak ikiye ayırmak mümkündür:

a. Program tabanlı DoS saldırıları

b. Network tabanlı DoS saldırıları

Saldırılarda kullanılan iki yöntem vardır. Bunlar "flooding" ve "exploiting" dir. "Flooding" yönteminde kullanıcı kurbana çok sayıda paket yollayarak belli bir işlemi çalışamaz duruma getirir. "Exploiting" yönteminde ise kurbanda çalışan herhangi bir program hedef alınıp, bu programın açıklarından faydalanarak program kullanılamaz hale gelir. Bu saldırı yöntemleri program ve network tabanlı DoS saldırılarında kullanılır.

DoS saldırıları Şekil 2.1' deki gibi kendi içinde gruplara ayrılır [12]:

Şekil 2.1. DoS saldırı tipleri [12]

4 çeşit DoS atağı vardır. Bunlar:

a. TCP/IP uygulamasındaki kusurları istismar eden ataklar. Örneğin Ping of Death ve Teardrop.

b. TCP/IP' deki zayıflıkları kullanan ataklar. Örneğin SYN Flood ve LAND atakları.

c. Brute-force atakları. Örneğin Smurf atağı. Bu tip ataklar networkü gereksiz data ile istila ederler.

d. IP Spoofing.

Ping Of Death Atağı

Ping of Death atağı PING uygulamasını kullanarak IP tanımlamasında izin verilen 65535 byte data sınırını aşan IP paketleri oluşturur. Normalden büyük paket daha sonra networke gönderilir. Sistemler çökebilir, durabilir veya kapanıp açılabilir [13].

Teardrop Atağı

Teardrop atağı IP paketlerinin tekrar birleştirilmesindeki zayıflığı istismar eder. Data, networkler içinden iletilirken genellikle daha küçük parçalar ayrılır. Herbir parça orjinal paket gibi görünür. Ama istisna olarak offset alanı vardır. Teardrop programı bir dizi IP paket parçaları oluşturur. Bu parçalar örtüşen offset alanlarına sahiptir. Bu parçacıklar varış noktasında tekrar birleştirildiklerinde bazı sistemler çökebilir, durabilir veya kapanıp açılabilir [13].

Syn Flood Atağı

SYN atakları hedef sistemi ard arda gelen SYN paketleri ile istila eder. Herbir paket hedef sistemin SYN-ACK yanıtı yanılmasına sebep olur. Hedef sistem SYN-ACK yanıtına karşılık ACK yanıtı beklerken, vadesi dolmuş (yanıt alınmamış) SYN-ACK yanıtlarını backlog queue'de beklemeye alır. SYN-ACK yanıtları kuyruktan sadece ACK yanıtı geldiği zaman veya iç zamanlayıcı TCP üç yollu el sıkışmayı sonlandırdığı zaman çıkarılır. Kuyruk dolduğunda sistem gelen bütün SYN isteklerini reddedecektir. Bu yüzden sistem normal kullanıcılar için kullanılamaz olacaktır [13].

Land Atağı

Bu atak çeşidinde saldırganlar (hacker) hedef sistemin IP adresini, source (kaynak) IP adresi olarak kullanarak networkü SYN paketleri ile istila ederler. Bu durumda host bilgisayar sanki paketleri kendi kendine göndermiş gibi görünür. Bu durumda hedef sistem kendi kendine yanıt vermeye çalışırken sistem kullanılamaz duruma gelir [13].

Brute-Force Atağı

Brute Force saldırıları sistemlerde kullanıcı hesaplarını ve şifreleri kırmak için tahminlere dayalı deneme yanılma yöntemiyle yapılan bir saldırı çeşididir. Burada amaç kullanıcının şifresini ele geçirmek ve sisteme o kullanıcı üzerinden sızmaktır. Sözlük saldırılarına çok benzer. Bir dosyadan tahmini şifreleri alır ve tek tek dener.

Uygulamalarda bu tip saldırıları algılayan ve önlem alan bir sistem yoksa korsan, hesaba ait şifreyi bulana kadar saldırıya devam edebilir [14].

Ip Spoofing Atağı

Ip Spoofing, sistemlere girmek için, saldırganın kimliğini gizleyebilmesi için veya DoS atağının etkisini büyütmek için kullanılır. IP Spoofing router'u veya firewall'u kandırarak isteğin güvenilir networkten geldiğini sağlamaya çalışan bir tekniktir. Bu sayede sistemlere yetkisiz erişim sağlanır. Saldırgan bunu yapmak için paketin header'ını değiştirir. Bu sayede paket güvenilir networkten geliyormuş gibi gözükür ve router veya firewall bu paketlere izin verir [13].

3. SALDIRI TESPİT SİSTEMLERİ (STS)

Bilgi ve bilgisayar teknolojilerinin hızlı gelişimi sonucu, elektronik ortamların kullanım oranının gün geçtikçe artması ve sağladığı kolaylıkların yanında, bu ortamlarda saklanan bilgilerin güvenliğinin sağlanması bir ihtiyaç haline gelmiştir. Korunacak bilginin değerine göre farklılık gösterebilecek olan koruma sistemlerinin aslında tek amacı, saldırganlara ve saldırılara karşı önlem alarak, bilginin mahremiyetinin korunmasıdır. Bilgisayar sistemlerine yönelik tehditler ve bu sistemlerdeki zayıflıklar olduğu sürece, saldırıları tespit etmek, bilgi güvenliğinde önemli rol oynayacaktır [6].

3.1. Saldırı Tespit Sistemleri Tanımı

Saldırı Tespiti Sistemleri (Intrusion Detection Systems), bilgisayar sistemine ve ağ kaynaklarına olan saldırıları tespit etmek, saldırıların kimden geldiğini tanımak, sistemi izleyip anormal olan durumları saptamak ve bunlara karşı gerekli önlemleri almayı amaçlayan güvenlik sistemleridir.

Diğer bir tanımda ise;

“Saldırı tespiti, bir sisteme yapılan izinsiz müdahalelerin mümkünse önceden, olay anında veya daha sonra; sistem günlüklerinden ve/veya bilgisayar ağı trafiğinden alınan verilerden yola çıkılarak çeşitli metotların yardımı ile analizidir” [10].

Saldırı tespitinin amacı; bu müdahaleleri mümkünse önlemek ve saldırıları kayıt altına alarak e-posta, kısa mesaj, konsol uyarı mesajı vb. gibi yollardan yetkili kişileri bilgilendirmektir [10].

Saldırı Tespiti Sistemi, bir sisteme karşı yapılan saldırıları tespit etme amaçlı kullanılan sistemlerdir. Burada asıl amaç, bir saldırı olduğunda bilgilendirme amaçlı uyarılar vermektir. Bu şekilde sistem yöneticileri potansiyel zayıf noktalardan ve sisteme karşı yapılan saldırılardan haberdar olabilir; buna uygun şekilde önlemler alabilirler. Kısaca STS, amacı uyarmak olan bir savunma sistemidir. Bu amaçla,

bütün ağ trafiği STS tarafından dinlenebilir veya önemli bileşenler, bileşen bazında dinlenerek STS' ne yönlendirilebilirler [15].

STS' nin en önemli özelliği pasif bir koruma sağlamasıdır. Yani, STS herhangi bir saldırıyı sadece kaydetmekle ve alarm göndermekle yetinir. O saldırıyı engellemek için hiçbir girişimde bulunmaz. Dolayısıyla, saldırıdan korunmak isteniyorsa bunu, STS kayıtlarını takip eden sistem yöneticileri gerçekleştirecektir. Bu tür bir savunma mekanizması, sistem yöneticilerine fikir vermek ve potansiyel tehditleri görmek amaçlıdır. Bu da kayıtları düzenli olarak izlenmeyen bir STS' nin bir işlevinin olmadığı sonucunu doğurmaktadır [15].

3.2. Saldırı Tespit Sistemi Tarihçesi

Saldırı tespiti kavramı ilk olarak Anderson'un "Bilgisayar Güvenliği Tehdit Gözetleme ve İzleme (Computer Security Threat Monitoring and Survaillance)" makalesi ile 1980' de ortaya atılmıştır [7]. Bu çalışma, STS' lerin tanımlanması ve tanınması açısından büyük bir öneme sahiptir.

İlk nesil STS' ler, basit bilgisayar sistemleri üzerine düşünülmüştür. İkinci neslinde ise günümüzde STS' lerin vazgeçilmezi olan denetleme izi (audit trail) kavramı ve veritabanı mantığının bilgi güvenliği alanındaki önemi ortaya çıkmıştır. Bunu izleyen çalışmalarda, güvenlik konusundaki çalışmalara yardımcı olmak amacıyla günlük denetleme verilerinin otomatik araçlar ile elde edilmesi konusunda çalışmalar yapılmıştır [6].

Denning çalışmasında 3 farklı istatistiksel model tanımlamıştır. Bu modeller;

a. Kullanıcının belirli aralıklarla bir işlemi tekrar etmesine izin veren ve eşik değerine göre anormallik olduğunu tespit eden model,

b. İstatistiksel momentlerin bilindiği varsayılarak tespit edilen sapmalar ile anormallik olduğunu tespit eden model,

c. Anormalliklerin tek olaya değil bir diziye bağlı olduğu Markov modeli olarak verilmektedir [16].

STS yazılımlarının test edilmesinde saldırı veritabanları kullanılmaktadır. MIT' nin Lincoln Laboratuarlarında 1998 yılında oluşturulan DARPA Saldırı Tespit Değerlendirme veri kümesi bilinen en kapsamlı veritabanıdır. DARPA verileri "tcpdump" komutu ile alınan paket başlıklarını içermektedir. Bu verilerle çalışmak oldukça fazla ön işlemden geçirilmesi gerekmiştir. Bu yüzden DARPA verileri ön işlemden geçirilerek "KDD'99" veri kümesi oluşturulmuştur.

Günümüzde ise; STS' lerin geliştirilmesinde istatistiksel yöntemlerin dışında, kural tabanlı, eşik değeri belirleme, durum geçiş diyagramları, yapay sinir ağları, yapay bağışıklık sistemi, bulanık mantık, veri madenciliği gibi farklı birçok yaklaşım uygulanmaktadır. Sistem yöneticileri bu tekniklerden bilgi güvenliği politikasına en uygun olanı kullanmaktadır.

3.3. STS' lerin Sınıflandırılması

Saldırı tespit sistemleri, tüm tedbirlere karşın bilgisayar sistemlerine yapılan saldırıları gerçekleşirken ya da gerçekleştikten sonra tespit etmek, İnternet veya yerel ağdan gelebilecek, ağdaki sistemlere zarar verebilecek, çeşitli paket ve verilerden oluşan bu saldırıları fark etmek üzere tasarlanmış sistemlerdir ve bu saldırılara yanıt vermeyi amaçlayan bir güvenlik teknolojisidir. Saldırı tespit sistemleri bir nevi alarm sistemi olarak düşünülebilir [17]. Bunları pek çok farklı şekillerde kategorilere ayırmak mümkündür. Saldırı tespit sistemleri; Şekil 3.1' de görüldüğü gibi konumuna göre, saldırıyı tanıma yöntemlerine göre ve veri işleme zamanına göre sınıflandırılır.

Şekil 3.1. Saldırı tespit sistemleri tipleri

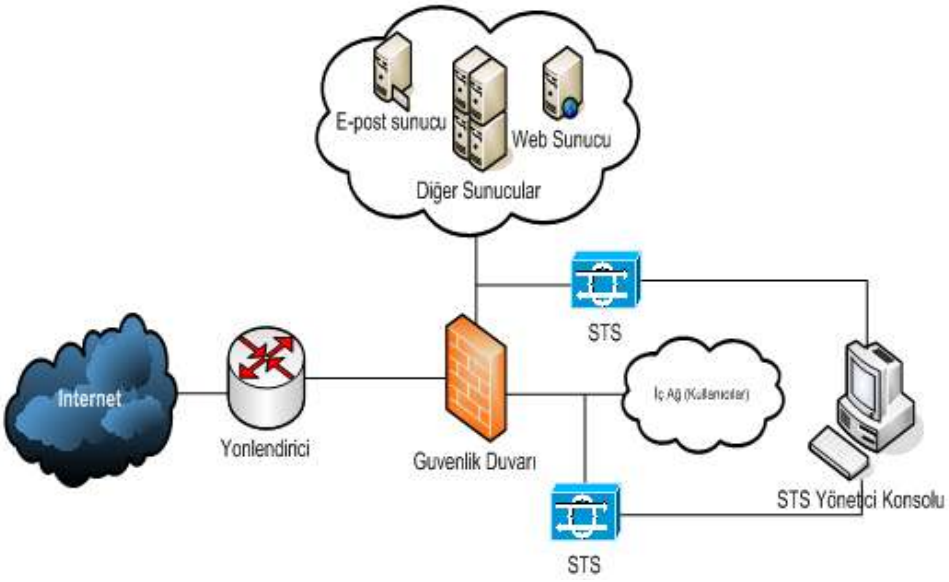
3.3.1. Konumuna göre STS

Konumuna göre STS' nin ağ yerleşimi ve neyi dinlediği ile ilgili bir sınıflandırmadır [15]. Bunlar ağ tabanlı ve bileşen tabanlı STS' dir.

Ağ Tabanlı STS

Ağ tabanlı saldırı tespit yönteminde sensörler yardımıyla bütün bir ağ trafiği yakalanmakta ve bu trafik incelenmektedir. Eğer bir saldırı tespit edilirse bu bilgi merkez konsola gider. Sistem yöneticisi hangi sensörün hangi saldırıyı yakaladığını bu merkezi konsoldan gözleyebilir. Bu merkezi konsol, aynı zamanda sensörlerin idaresi işlevini de yerine getirir. Resim 3.1' de görüldüğü gibi sensör sadece bulunduğu ağ segmentini dinleyebilir. Farklı ağ segmentlerinin dinlenmesi isteniyorsa her birine bir sensör koymak gerekmektedir [15].

Sensörün ağ paketlerini inceleyebilmesi için ağ segmentinin bütününün bir huba bağlanması gerekir. İkinci bir alternatifse mirror işlemi yapabilen bir switch kullanmak olabilir. Switch'teki bütün trafik, mirror portuna yollanmalı ve sensör de bu porta bağlanmalıdır [15]. Resim 3.1 de ağ tabanlı STS topolojisi görülmektedir.

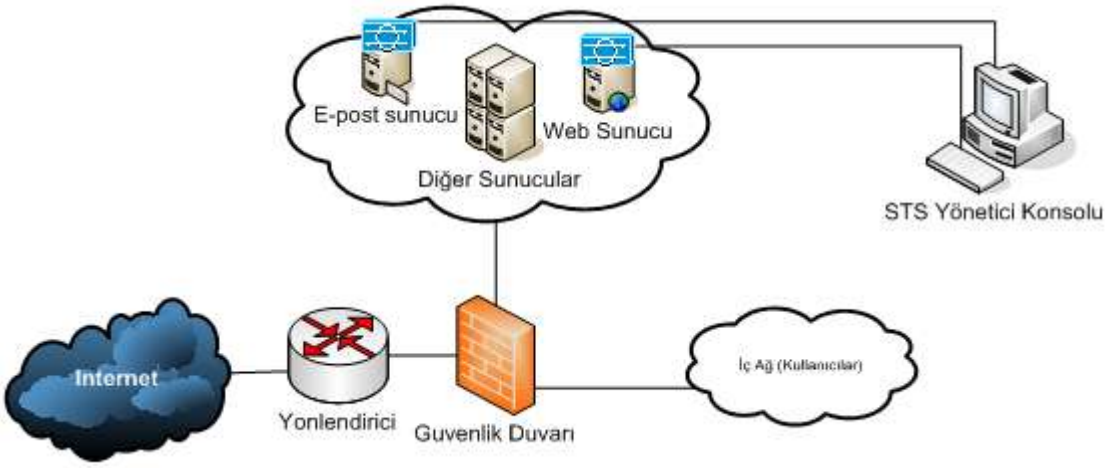


Resim 3.1. Ağ tabanlı STS [15]

Bileşen Tabanlı STS

Host bazlı STS' ler ağ segmentini dinlemezler. Sadece üzerine yüklendikleri bilgisayara olan saldırıları gözlerler. Bazı host bazlı sistemler sadece ilgili bilgisayara gelen ağ paketlerini incelerler. Diğerleri ise ağ paketlerini değil de bilgisayar üzerindeki dosya erişim ve program çalıştırma kayıtları ya da syslog, veritabanı, web sunumcusu gibi uygulama yazılımı kayıtlarını incelerler. Bileşen tabanlı STS' lerin en büyük avantajı yüksek hızlı ağlarda bilgi kaçırmamasıdır [15].

Bileşen tabanlı STS' lerde her bir host üzerine sensörler yerleştirilmiştir. Bu sensörler sadece üzerinde yüklendikleri bilgisayarları hedef alan saldırıları tespit etmektedirler ve yakaladığı saldırıları yönetim konsoluna iletirler. Bu tür bir sistemin en büyük dezavantajı, uç bilgisayar sayısının çok olduğu ortamda yönetilmesinin zor olmasıdır [15]. Resim 3.2 de bileşen tabanlı STS topolojisi görülmektedir.



Resim 3.2. Bileşen tabanlı STS [15]

3.3.2. Tanıma yöntemine göre STS

STS'ler bir alarmı tespit etme şekline göre ikiye ayrılır. Bunlar imza tabanlı ve anomali tespit eden STS' lerdir. Hibrid teknolojiler mevcuttur [15].

İmza tabanlı STS

İmza tabanlı STS' lerde iki tür saldırı yakalama yöntemi bulunmaktadır. Bu yöntemlerin ilki imza bazlı saldırı yakalamadır. STS'de bir çok saldırı imzası tanımlıdır. Bu imzalar şu ana kadar tespit edilmiş saldırıların bir tür karakteristik özelliklerini yansıtmaktadırlar. Ağ paketinin protokol çeşidi, protokol işaretleri (flags), kaynak ya da hedef IP ve port numaraları, paketin paket verisi (payload) kısmı imzaların karakteristik özellikleridir. Bu bilgiler bir imza veritabanında tutulur. STS sensörü tarafından ağdan dinlenen her paket bütün imzalarla karşılaştırılır. Herhangi biriyle uyum sağlarsa, o imza ile ilgili alarm mesajı STS yönetici birimine iletilir [15].

Bu sistemlerin en büyük eksikliği bir saldırının yakalanabilmesi için ilgili imzaya ihtiyacı olmasıdır. Yeni çıkmış bir saldırının eğer imzası veritabanına ilave edilmemişse, sistemin bu saldırıyı tanımasına imkân yoktur. Hatta bazı deneyimli saldırganlar, halen kullanılmakta olan saldırı yöntemlerinde bazı ufak değişiklikler yaparak rahatlıkla gözetlenen sisteme girebilirler. Dolayısıyla STS' ler kesin çözüm

değildirler ama önemli bir güvenlik bileşenidirler. Sistem veya güvenlik yöneticileri kendilerince kritik olabilecek durumlar için saldırı imzaları da oluşturmalıdır [15].

Anomali tespit eden STS

Bu yöntemde ağ trafiği belli bir zaman incelenir ve kullanıcıların ya da bileşenlerin profilleri belirlenir. Dosya erişimleri, CPU kullanımı, erişim zamanları ve ağ trafiğinin protokol dağılımı gibi karakteristik özellikler bu profilleri oluşturur. Profillerin oluşmasından sonra ağ trafiği bu profillerle karşılaştırılır. Eğer uymuyorsa anormallik olarak tespit edilir [15].

Örneğin bir kurum düşünelim. Bu kurum çalışanları sabah 9:00 akşam 18:00 arasında çalışıyor olsun. Bu zaman aralıklarında kullanıcıların İnternet'e bağlandıklarını, e-postalarını okuduklarını varsayalım. Buna bağlı olarak da profiller oluşacaktır. Eğer akşam 21:00' de bir http bağlantısı gerçekleştiyse bu profile uymamaktadır. Dolayısıyla anormal trafik yakalama yöntemini kullanan STS bu durumu kaydedecektir [15].

Ağ trafiği zamanla değişmektedir. Böylece, *normal* olarak öğretilen ağ trafiği belirli bir zaman sonra çok değişecektir. Bu da, cihazın devamlı olarak *normal* olan trafiği öğrenmesi gerekliliğini doğurmaktadır. Bu yüzden, anomali tespit eden STS' leri çok fazla false-pozitif verebilir [15].

3.3.3. Veri işleme zamanına göre STS

STS' ler için veri işleme zamanı, izlenen olaylar ve olayların analizi arasında geçen zamanı ifade eder. STS' ler gerçek zamanlı ve gerçek zamanlı olmayan şeklinde ikiye ayrılırlar [6].

Gerçek zamanlı STS

Gerçek zamanlı sistemlerde; veri, iletişim anında analiz edilir ve eğer bir saldırı tespit edilirse saldırıya cevap olarak gereken çevre değişimleri gerçekleştirilir. Bu tip STS' ler, yoğun bilgi akışı olan ağlarda uygulanması zor ve maliyetli bir yöntem

olmakla birlikte aktif olarak cevap üretilmesi gereken durumlarda da tek çözüm olan sistemlerdir. Ticari uygulamalar genellikle gerçek zamanlı sistemlerdir [6].

Gerçek zamanlı olmayan STS

Gerçek zamanlı olmayan sistemlerde; alınan veriler depolanarak, analiz edilmesi için ilgili STS' ye gönderilir. "*Depola" ve "yolla"* mantığıyla çalışan bu sistemlerde belirlenen tehditler ve olası saldırılar, sistem kullanıcısının dikkatini çekecek şekilde açılır pencerelerde alarm olarak gösterilir. Saldırının profilinin çıkarılması ve bu tür saldırılara bir daha maruz kalınmaması için, geçmişe yönelik depolanan veriler üzerinde yapılan analizlerle sonuçlar veya saldırılar elde edilir. STS' nin gerçek zamanlı olması şartı aranmadığı durumlarda kullanılan bu yöntem, sistemin belirlenen güvenlik açıklarını kapatmak için kullanılır [6].

3.4. Saldırı Tespit Sistemi Özellikleri

Tam güvenlik ancak teorik olarak mümkündür. Hiçbir güvenlik ürünü mucizeler yaratmaz. STS' ler de, dağıtık ortamlarda, sistem ve iletişim ağı güvenlik sorumlularını saldırılardan ve ortamlardaki anormal değişimlerden haberdar etmeyi hedeflemektedir. Bu ürün ailesinin kurumda güvenliği arttırması, ancak ürünlerin ve sistemlerin iyi tanınması, gereksinimlere göre konfigüre edilmesi ve düzenli güncellenmesi durumunda gerçekleşir. Bunun için gereken ön çalışma ihmal edildiği takdirde, ürünlerden istenen verim alınamaz. Bu durum, bilgi işlem personeli için ise yalnızca bakım gerektiren yeni sunucuları olması anlamına gelecektir. STS, izinsiz girişin giderek arttığı günümüzde, Güvenlik duvarı ve diğer koruyucu ürünlerle beraber bir kurumun vazgeçilmez güvenlik cephesini oluşturmaktadır.

3.4.1. Doğru saldırı tespit sistemi konfigürasyonu

Doğru STS' lerin yönetimi için her gün sistem yöneticisinin zaman ayırıp alarmlarını kontrol etmesi gerekmektedir. Alarmların kontrol edilmemesi STS' lerin fonksiyonelliğini büyük ölçüde yok etmek demektir. Birçok organizasyonda güvenlik yöneticisi kadrosunun eksik olduğunu göz önünde bulundurursak STS' lerin

yönetiminin mümkün olduğu kadar kolaylaştırılması gerekmektedir. Kolaylaştırmanın da temel şartı az fakat doğru alarm elde etmektir [15].

Az fakat doğru alarm için ilk olarak sistemimizde bulunan bileşenlerle ilgili saldırı imzalarının STS' de yüklü olması gerekmektedir. Örneğin sistemimizde IIS web sunumcusu varsa Apache web sunumcusu ile ilgili saldırılar aktif olmamalıdır. Ayrıca dikkat edilmesi gereken diğer konu ise güvenlik duvarınca bloke edilen saldırı imzalarının aktif hale getirilmemesidir. Mesela sisteminizde FTP sunumcusu yoksa ve güvenlik duvarında dışarıdan herhangi bir FTP isteğine cevap verme engellenmiş ise, FTP sunumcusu ile ilgili saldırı imzaları STS' de bulunmamalıdır [15].

STS' nin saldırı yakalama kabiliyeti imza veritabanının içeriğiyle direk olarak ilgilidir. Dolayısıyla imza veritabanının devamlı güncellenmesi gerekmektedir [15].

3.4.2. Saldırı tespit sistem yazılımı özellikleri

1. Bilgisayar sistemleri ve ağlar arasındaki trafiği dikkat çekmeden analiz edebilmelidir. Gizli (stealth) modunu desteklemelidir.
2. Sisteme yapılan saldırılara karşı koyabilmeli ve ağ yöneticisine bildirebilmelidir.
3. Gerçek zamanlı saldırı tespiti yapabilmelidir.
4. Saldırı tespitlerini ağ ve işletim sistemlerini inceleyerek yapabilmelidir.
5. Ağda dolaşan paketleri görüntüleyebilmeli ve saldırı içerikli olup olmadığını tespit edebilmelidir.
6. İşletim sisteminin kayıtlarına (log) bakarak belli bir sisteme yapılan izinsiz erişimleri ve sistemde uygulanan izinsiz aktiviteleri kontrol edebilmelidir.
7. Kullanıcı grafik ara yüzü ile kolayca yönetilebilmelidir.
8. Kullanıcı grafik ara yüzü ile olası alarmları görüntüleyebilmeli, algılayıcıların konfigürasyonlarını kontrol edebilmeli, verileri toplayabilmeli, bu verileri

veritabanı ortamında saklayabilmeli ve ağ veya sistem aktivitelerini raporlayabilmelidir.

9. Saldırılara karşı nasıl tespit yapılacağı ve nasıl karsı koyulacağı konularında var olan veritabanını İnternet aracılığı ile güncelleyebilmeli ve bu teknikleri kullanıcı tanımları ile kontrol edebilmelidir. Kullanıcı kendi belirlediği kuralları ekleyebilmelidir.

10. Saldırı tespit sistemini oluşturan modüller ve yönetim birimleri arasında veri akış güvenliği açısından VPN'i, güçlü ve zayıf şifreleme yapılabilmelidir.

11. Saldırı durumunda konsola alarm gönderebilmeli (SNMP), e-posta atabilmeli, aktif oturumu görüntüleyebilmeli, raporlama yapabilmeli, saldırı tehdidi içeren bağlantıları kesebilmeli, durdurabilmeli, kullanıcı tarafından kontrol edilebilmeli ve gerekirse güvenlik duvarı üzerinde belirlenmiş olan ağ güvenlik politikasını değiştirebilmeli ve güvenlik duvarı ile entegre çalışabilmelidir.

12. İzinsiz ve yetkisiz erişimleri tespit edebilmeli ve bu erişimler hakkında kayıt tutabilmelidir.

13. Sunucu işletim sistemlerinden, MS Windows, IBM AIX ve Sun Microsystem Solaris vb. işletim sistemlerini desteklemelidir [18].

3.4.3. Saldırı tespit sistemlerinin olası sorunları

Merkezi yönetim sırasında gönderilen paketlerin iletim güvenliği sorun yaratabilir. Bu bilgilerin ortaya çıkması sisteme yapılan saldırıların üçüncü kişilerce öğrenilmesine neden olabilir. Bu noktada, topolojideki yönetim konsolunun yeri de önem kazanmaktadır. Bu sunucuya erişim, güvenlik duvarı kurallarıyla kısıtlanmalı veya konumlandırılması dikkatlice düşünülmelidir [15].

Ağ trafiğinin STS' nin kaldırabileceği yoğunluktan fazla olması durumunda STS' nin yakalama kabiliyetinin yetersiz kalması nedeniyle alarm olarak kaydedilmeden kaçan paketler olabileceği göz önünde bulundurulmalı, ağ trafiği yoğunluğuna uygun performans seviyelerini karşılayacak bir STS seçilmelidir. Bu nedenden dolayı

saldırganlar, asıl saldırı ile birlikte, STS' nin tespitini azaltmak için servis dışı bırakma saldırılarını da yapabilirler. En önemli nokta ise STS' lerin düzenli olarak kontrol edilmesi zorunluluğudur. Bir STS, kurulduğu halde kontrol edilmiyorsa hiçbir işe yaramaz. STS' ler, pasif savunma cihazları oldukları için düzenli olarak kontrol edilmelidir [15].

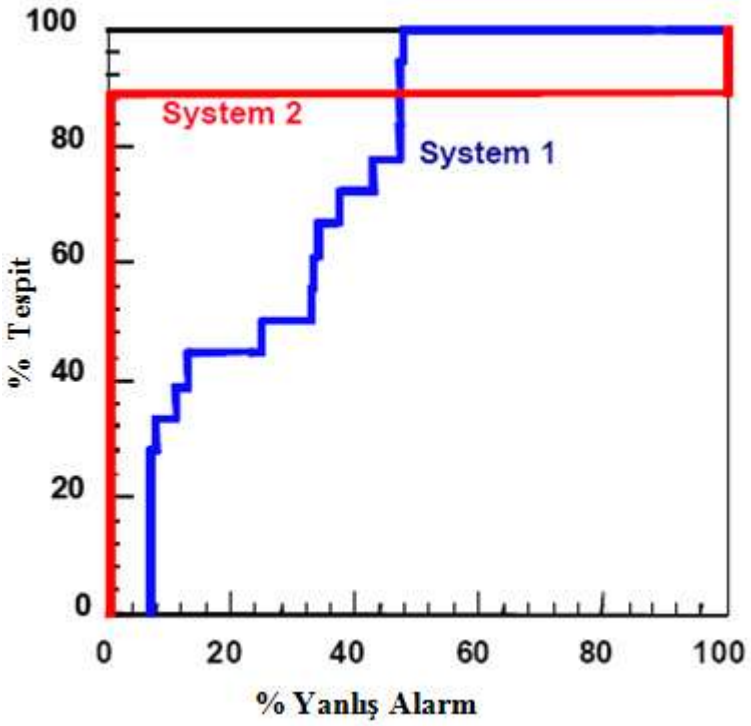
3.4.4. Saldırı tespit sistemlerinde yanlış alarmlar

Saldırı Tespit Sistemleri güvenliği artırıcı özelliğe sahiptir. Ancak bir takım sorunları da beraberinde getirir. Günümüzde STS' lerin problemleri, saldırı olmayan aktiviteleri bazen saldırı olarak algılaması (false-negative) ve saldırı olan aktiviteleri ise bazen saldırı olarak algılamaması (false-positive) olayıdır. Bu yüzden, STS uygulanmasında insan faktörü ve bilgi çok önemlidir. *Saldırı Tespit Analisti* dediğimiz bilgili kişiler tarafından kayıtların incelenmesi gerekmektedir. Kayıtların doğruluğu tespit edildikten sonra, harekete geçilip çok kolay önlem alınabilir. Bazı STS' ler otomatik olarak müdahale etmeye olanak sağlamaktadır. Yani, saldırı kaydı sistemde tespit edilir edilmez, insan faktörü olmadan tepki vermek veya gerekli cihazlarda konfigürasyon değişikleri yapmaya olanak sağlar. STS' lerin doğruluk yüzdesinin %80' nin üzerine çıkmadığı durumlarda otomatik müdahalelerin çok doğru olmayacağı bir gerçektir [15].

Yanlış Alarm Seviyesi

Yanlış alarm, tespit edilmesi planlanan büyüklük için yapılan yanlış değerlendirmeleri içerir. Yanlış değerlendirmeler iki çeşit olabilir. Biri, var olan bir değeri kaçırmak diğeri, var olmayan bir değeri varmış gibi tespit etmektir. Yanlış alarm ikincisidir. Genelde STS sistemlerinin başarımını ölçmekte önemli bir parametredir. Çünkü bir sistemde izin verilen yanlış alarm sayısı ve doğru tespit miktarı birbiriyle ilintilidir. Yanlış alarmlara izin verildikçe doğru tespit oranı artmaktadır. Sistem parametreleri ikisinin de optimum olduğu noktaya ayarlanmalıdır. Burada yapılacak analiz ve başarım hesaplamasında ROC (Receiver Operating Characteristic– Alıcı Karakteristiği) eğrileri kullanılır. ROC adını radar uygulamalarından almıştır. Şekil 3.2' de iki sistem için ROC eğrileri verilmiştir.

Burada örneğin %40'lık bir yanlış alarma izin verilirse sistemin doğru yakalama şansı %50 olur. Eğer %60'lık yanlış alarm kabul edilirse %100'luk bir doğru tespit şansı vardır [19].

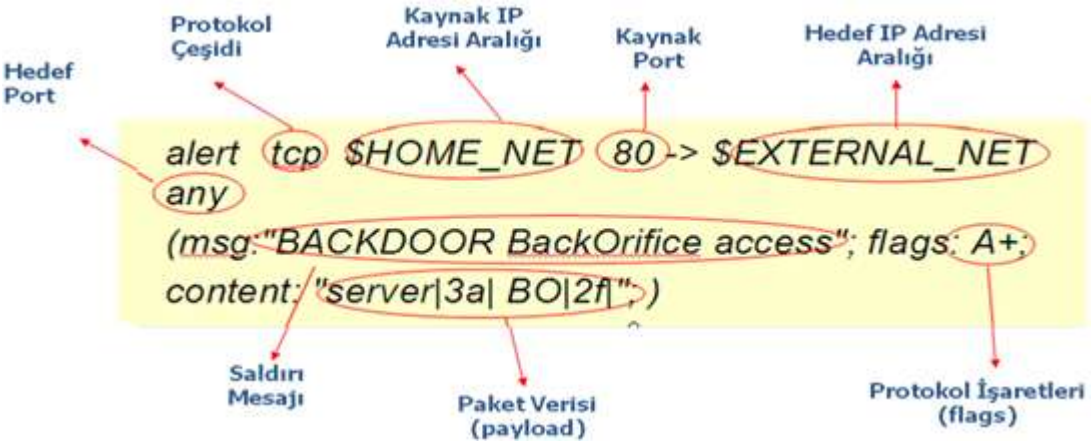


Şekil 3.2. Örnek ROC eğrileri [19]

Yanlış alarmlar tespit edildiğinde ilk olarak kuralın değiştirilerek yanlış alarmı önleyecek şekilde düzeltilmesi düşünülebilir. Eğer kural değiştirilemeyecek gibi görünüyorsa, o zaman kaynak IP adresine özel olarak bir ihmal etme kuralı konulabilir. Ancak birden fazla IP'den aynı yanlış alarm gelmekteyse bunu yönetmek zor olabilir. Daha sonra kuralın tamamen silinmesi düşünülebilir. Ancak bu durum, gerçekten o kuralla ilgili bütün alarmların yanlış alarm olduğu durumlarda düşünülmelidir [15].

3.4.5. Saldırı imzası

Resim 3.3'te açık kaynak kodlu bir STS olan Snort' ta kullanılan bir imza çeşidi gösterilmektedir.



Resim 3.3. Saldırı imzası [15]

Saldırı imzasında bulanan kelimeler ve ifade ettikleri anlamlar şunlardır:

"alert" bu imzanın bir alarm ürettiğini belirtmektedir.

İkinci kelimesi ise protokol çeşidini gösterir. Örnekteki saldırı imzası *tcp* protokolüne ait ağ paketlerini ilgilendirmektedir.

"HOME_NET" değişkeni kaynak IP adres aralığını ifade etmektedir. Bu değişkene STS yöneticisi daha önceden korumayı düşündüğü ağın adres aralığını girmiş olmalıdır.

Kaynak port numarası 80' dir.

"EXTERNAL_NET" ise aynen HOME_NET gibi önceden değeri atanmış bir değişkendir ve güvensiz ağ aralığını kapsamaktadır.

"msg" kelimesinden sonra tırnak içinde yazılmış kısım ise alarm mesajını ifade etmektedir. Eğer ağ paketi saldırı imzasına uyarsa bu mesaj STS yönetici birimine iletilmektedir.

“flags” kelimesi protokol işaretlerini göstermektedir.

“A” harfi ACK işaretini temsil eder. Yani “Ağ paketinin TCP protokol işaretinin ACK değeri 1 olmalıdır” demektir.

“content” kelimesinden sonra gelen tırnak içerisindeki ifade paketin paket verisi (payload) kısmında aranacak bir veridir. Bütün bir paket verisi bu kelime için taranır.

Bu imza genel olarak şu şekilde ifade edilebilir: Kaynak adresi güvenli ağ, hedefi güvensiz ağ, kaynak portu 80, hedef portu herhangi bir port olan, protokol işareti ACK olarak belirlenmiş ve payload kısmında server|3a| BO|2f| kelimesi bulunan ağ paketlerini BACKDOOR BackOrifice access saldırısı olarak tanımla.

Bu saldırı imzası bir arka kapı (backdoor) çeşidi olan BackOrifice’i tanımaktadır. Güvenli ağdaki bir bilgisayara yüklenmiş BackOrifice arka kapı yazılımı dışarıya bilgi sızdırmak için açılacak bağlantıyı bu imzaya uygun bir ağ paketi ile yapmaktadır. Paket yönü içeriden dışarıya olabilmesi için kaynak IP aralığı güvenli ağ, hedef IP aralığı güvensiz ağ olarak belirlenmiştir.

Bu örnekte Snort’un kullandığı imza formatı açıklanmıştır. Diğer STS sistemleri de buna yakın bir format kullanmaktadırlar. Çoğu sistem, kullanıcılarına yeni saldırı imzaları yazma imkânı sunabilmektedir. Yani güvenlik yöneticisi kritik olarak gördüğü bazı ağ trafiklerini de gözlemleyebilir [15].

3.4.6. Saldırı tespit sistemi imzalarının güncellenmesi

İmza tabanlı STS’ leri sürekli güncellemek gerekir. Yeni bir saldırı olduğunda bu saldırıyla ilgili imza geliştirilir, daha sonra imza veritabanı bu yeni imzayı da kapsayacak şekilde güncellenir. İmza güncelleme konusunda iki tür yaklaşım vardır [15]:

a. Toplu güncelleme: Haftada bir, bütün yeni imzalar toplanarak toplu bir şekilde güncelleme yapılır.

b. Özel imza güncellemeleri: Çok önemli bir saldırı olduğunda, toplu güncelleme beklenmeden o saldırıya özel imzalar çıkarılarak kullanıcılara ulaştırılır.

3.4.7. Saldırı tespit sistemine kural ekleme

STS' den çıkarılması gereken kurallar olduğu gibi STS' ye eklenmesi gereken kurallar da olabilir. Bu kurallar sistem ihtiyaçlarına göre belirlenip düzenli olarak gözden geçirilmeli, herhangi bir sistem değişikliği olduğu takdirde uygun bir şekilde güncellenmelidir. STS imzalarının eklenmesi önemlidir. Yeni imzalar, yeni saldırılar çıktığı zaman bunların da fark edilmesini sağlamak amacıyla, ya da sisteme özel bir tasarımda saldırıların bulunması amacıyla kullanılabilir [15].

3.4.8. Saldırı tespit sistemi kural optimizasyonu

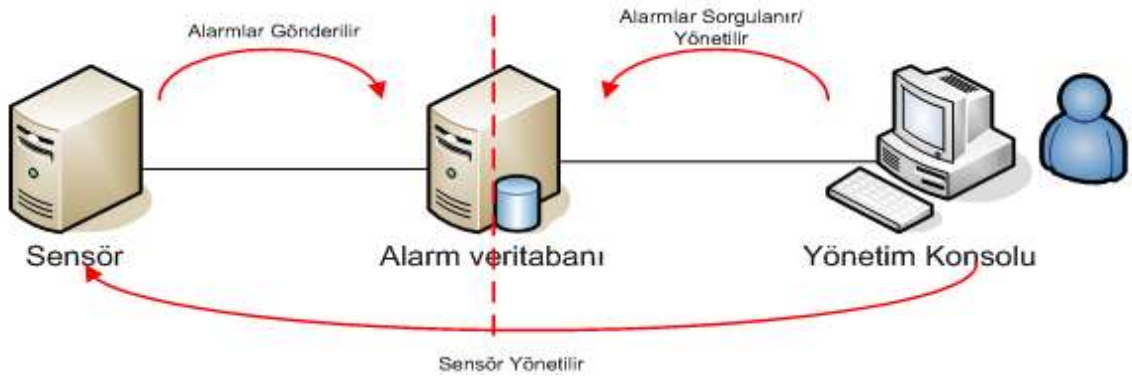
Paket başlıkları dışında kalan mesaj kısmının içeriğini ifade eden content kısmında açıklıkla ilgili verilebilecek en detaylı bilgiler verilmelidir. Bu şekilde yanlış alarmların önlenmesi sağlanır. Ayrıca, saldırıya göre imza yazmak yerine, açıklığa göre imza yazmak tercih edilmelidir. Bu şekilde yapılmadığı takdirde, o açıklığı kullanarak yapılan her farklı saldırı için ayrı bir imza oluşturmak gereklidir [15].

Ayrıca, birbirini kapsayan kurallar da ayıklanmalıdır. Böylece, STS daha az imza üzerinden karar vereceği için performansı artar.

3.4.9. Saldırı tespit sistemi yönetim ve denetimi

Saldırı Tespit Sistemi Yönetim Konsolu

STS'ler genelde saldırıların tutulduğu veritabanını incelemeye yarayan, sensörleri uzaktan yöneten konsollarla birlikte gelirler. Bu konsollar, kullanım kolaylığı sağlayarak yönetim işlerini kolaylaştırırlar. Saldırı bilgisi trafiği hassas bilgi içerdiği için uygun bir şekilde korunmalıdır. Bunun için aradaki trafiğin şifrelenmesi tavsiye edilir. Bu şifreleme için iki cihaz arasında tünel yapılabilir. Ayrıca, çoğu STS bu özelliği ürünlerine otomatik olarak eklemektedir [15]. Resim 3.4' te örnek bir STS yönetim konsolu gösterilmektedir.



Resim 3.4. STS yönetim konsolu [15]

Saldırı Tespit Sistemi Denetimi

STS kurulduktan sonra belirli periyotlarda etkinliğinin gözlemlenmesi ve sağlıklı çalıştığının kontrolü için denetlenmesi tavsiye edilmektedir [15].

Denetim sırasında:

STS sensörlerinin ve yönetim konsolunun var olan ağın topolojisine uygun bir şekilde konumlandırılıp konumlandırılmadığına bakılmalıdır.

STS kurallarının etkinliği kontrol edilmelidir. Gereksiz olan kurallar kaldırılmalı, yakalanan trafiğe göre belirlenebilecek ek kurallar da gözden geçirilmelidir.

STS bileşenleri arasında alarm alışverişinin şifreli yapıldığı kontrol edilmelidir. Örneğin sensörün veritabanına gönderdiği alarm bilgilerinin şifreli olması, aynı şekilde veritabanından çekilen bilgilerin de şifreli ulaştırılması gerekmektedir. Alarm bilgilerinin hassas bilgiler olduğu unutulmamalı, ağ saldırıları hakkında bilgi veren bu ağ paketleri buna göre şifrelenerek korunmalıdır.

STS yönetim konsoluna politikalara uygun bir şekilde erişim kontrolü yapılıp yapılmadığı kontrol edilmelidir. Alarmların kimler tarafından gözlenip raporlanabileceği, kimler tarafından silinip değiştirilebileceği belirlenmeli ve her personel ayrı hesaplar kullanarak oturum açmalıdır.

STS' ne özel konfigürasyon dosyaları (örneğin Snort'ta "snort.conf" dosyası) gözden geçirilmeli ve bir aksaklık olup olmadığı kontrol edilmelidir.

Ayrıca, belki de en önemlisi STS alarmlarının ne kadar sıklıkla ve kim tarafından izlendiği, ne kadar sıklıkla raporlama gibi faaliyetlerde bulunulduğu kontrol edilmelidir. Daha önce de belirtildiği gibi, kontrol edilmeyen bir STS' nin güvenlik açısından katacağı bir fayda olamaz [15].

4. SALDIRI TESPİT SİSTEMLERİNDE KULLANILAN TEKNİKLER

STS' lerde, anormallik ve kötüye kullanım (imza) tabanlı yaklaşımları modellemek için günümüze kadar birçok teknik kullanılmıştır. Bu teknikler, elde edilen verilerin modellenmesi, sınıflandırılması veya kural tablolarının oluşturulması için geliştirilmiştir. Kullanılan tekniklerden elde edilen veriler sayesinde, saldırı tespit yaklaşımlarının uygulanması için gerekli olan platform oluşturulmuştur [6]. Aşağıdaki kısımda bu teknikler özetlenmektedir;

4.1. Veri Madenciliği

Büyük miktardaki veri içerisinden anlamlı bilginin çıkarıldığı tekniğe veri madenciliği adı verilir. Veri madenciliği ile saldırı tespiti birbirine yakın konulardır. Çünkü veri madenciliğinde yapılan işlerin bir kısmı anormal durumların tespiti ile ilgilidir [11].

Veri madenciliği kullanılarak geliştirilen saldırı tespit sistemleri anormallik tabanlı yaklaşıma yakındır. Anormalliklerin tespiti için veri madenciliği teknikleri kullanılabilir. Veri madenciliğinin doğasında yer alan özetlemeler yardımıyla veri indirgeme sayesinde gerçek zamanlı saldırı tespitinin de yapılması mümkün olabilecektir. Ağ trafiğinden elde edilen normal trafik profilleri veri tabanlarında tutulur. Yeni girişimlerle normal trafik karşılaştırılarak saldırı tespiti yapılır [11].

4.2. Kural Tabanlı (Rule Based) Sistemler

Kural-tabanlı karşılaştırma tekniğinde bilinen saldırı yöntemlerinin tanımları olan kurallar ya da imzalar kullanılır. Anılan imzalar, yeni saldırı tekniklerini içerecek şekilde sürekli güncellenmelidir. Bilgisayar sisteminde veya bilgisayar ağında gerçekleşen etkinlikler, belirlenmiş saldırı kuralları ile karşılaştırılır ve etkinlik ile imzanın benzerliği kontrol edilir. Bir kurala uygunluk saldırı olarak kabul edilir [20].

4.3. Tanımlayıcı İstatistikler (Descriptive Statistics)

Kullanıcı veya sistem davranışları farklı değişkenlere göre ölçülerek istatistiksel bir model oluşturulur. Bu değişkenlerden bazıları; kullanıcı oturum girişi, oturum

kapatma, belli bir zaman periyodunda erişilen dosya sayısı, kullanılan disk alanı ve hafıza olarak sıralanabilir. Kullanıcı profilleri ve hesap izleri kullanılarak normal davranışların modeli oluşturulur ve anormallik tespit edilir. Kullanıcı profilinin basit istatistiklerle oluşturulup, buradan uzaklık vektörlerini (distance vector) kullanarak karar alan sistemlerdir. Davranış profili oluşturulurken, kullanılan işlemci zamanı, bir zaman periyodundaki ağ bağlantı sayısı gibi farklı ölçütler de kullanılabilir. İstatistiksel yaklaşımların dezavantajlarından biri, saldırganın bu istatistikleri öğrenerek ona göre davranış sergileyebilmesidir [6].

4.4. Eşik Değeri Tespiti

Anormallik tespiti yaparken eşik değerinin ne olması gerektiği önemli bir konudur. Çünkü bu değer tanıma doğruluğunu etkileyen konulardan biridir. Eşik değeri verilirken genellikle uzman görüşünden faydalanılır. Yani sistemin kurulması esnasında o sistem için hangi eşik değerinin normali verdiği bilinerek sabit bir değer verilir. Daha sonra bu değer doğru tanıma oranına bakılarak aşağı yukarı hareket ettirilir. Yinelemeli bir yapıda ve deneysel olarak bu değerin optimumu bulunur. Ayrıca her anormallik tespiti sistemi için eşik değeri farklı olacaktır [21].

4.5. Durum Geçiş Analizi

Durum geçiş analizi tekniği, durum değişimi serileri oluşturularak gerçekleştirilir. Bir işin yapılması için birbirini takip eden durum sırası olduğu varsayılır ve buna göre bir seri oluşturulur. Sızmaların senaryosu çıkarıldıktan sonra, anahtar hareketler, imza hareketler olarak tanımlanır. İmza hareketler, saldırının tamamlanması için gereken en küçük hareket kümesidir. Durumlar, geçişler ve imzalar, durum geçiş diyagramı olarak grafiksel biçimde sunulur. Burada tüm davranışlar durumlara karşı düşer. Eğer bir davranış daha önceden tanımlı durumlara ve durum geçişlerine denk düşen hareketler yapıyorsa saldırı olarak tanınır [6].

4.6. Uzman Sistemler

Belirli bir alanda sadece o alan ile ilgili bilgilerle donatılmış ve problemlere o alanda uzman bir kişinin getirdiği şekilde çözümler getirebilen bilgisayar programları olarak

tarif edilebilir. Sızma belirleme sistemlerinin ilkleri kural-tabanlı (rule based) uzman sistemlerdir [22].

4.7. Örüntü Eşleme (Pattern Matching)

Sistemde daha önceden tanımlanmış ve karşılaşılmaması gereken bazı sözcüklerin tanınması için kullanılır. Esnek değildir fakat basittir. Örneğin "parola dosyasını kopyala" komutu görüldüğünde bunun bir saldırı olduğunu en basit şekilde bu yöntem tespit eder [22].

5. VERİ MADENCİLİĞİ

Veritabanı ve bilgi teknolojileri 1960’ lardan beri ilkel dosya işlem sistemlerinden büyük güçlü veritabanı sistemlerine doğru sistematik olarak gelişiyor. Bu gelişme 1970’ lerden itibaren ilişkisel veritabanı sistemlerinin oluşmasına, 1980’ lerin ikinci yarısından itibaren multimedya uzay verileri gibi hacimli verileri tutmaya olanak sağlayan nesne-tabanlı, geliştirilmiş-ilişkisel veri tabanları gibi gelişmiş veritabanı sistemlerinin oluşmasına neden oldu. Veritabanı sistemlerindeki bu gelişmeler 1980’ lerin sonunda veri ambarları ve veri madenciliği gibi kavramların oluşmasını sağladı [23].

5.1. Veri Madenciliği Tanımı

Veri madenciliği, veri ambarlarında tutulan verilerden otomatikleşmiş modeller sayesinde anlamlı bilgileri, ilişkileri ve davranışları ortaya çıkarma süreci olarak da tanımlanmaktadır. Bu süreçte, veri içinde önceden pek fazla bilinmeyen veya görülemeyen desenler (pattern) öncelikle ortaya çıkarılmaktadır. Bu desenler genellikle bilgiler arasındaki ilişkilerin, sıralamanın, sınıflandırmanın, veri birlikteliğinin ve tahminlemenin sonucunda elde edilmektedir [24].

Veri madenciliği ile ilgili değişik tanımlar vardır. Bunlar:

- Veri madenciliği, geniş veritabanlarından bilgi çıkartabilmek amacıyla makine öğrenmesi, örüntü tanıma, istatistik, görselleştirme gibi alanların tekniklerini bir araya getiren disiplinler arası bir alandır [25].

- Veri madenciliği, ham verinin tek başına sunamadığı bilgiyi ortaya çıkaran veri analizi sürecidir [26].

- Veri madenciliği, büyük hacimli veriler içerisindeki örüntüleri araştıran matematiksel algoritmaları kullanır. Veri madenciliği hipotezleri keşfeder, sonuçları birleştirmek için insan yeteneğini kullanır. Veri madenciliğinin sadece bir bilim olmadığı, aynı zamanda bir sanat olduğu da söylenebilir [27].

- Veri madenciliği, geniş veritabanlarındaki birliktelikleri araştırır [28].

- Veri madenciliği, istatistik, veritabanı teknolojisi, örüntü tanıma, makine öğrenme ile etkileşimli yeni bir disiplin ve geniş veritabanlarında önceden tahmin edilemeyen ilişkilerin ikincil analizi olarak tanımlamıştır [29].

- Veri madenciliği, eldeki verilerden üstü kapalı, çok net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılmasıdır. Diğer bir ifadeyle, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir [30].

Veri madenciliği tanımları incelendiğinde, bu tanımların ortak unsurlarının ilki çok fazla miktarlarda verinin veri ambarlarında tutulması, ikincisi ise bu verilerden anlamlı bilgiler elde edilmesidir [8].

5.2. Veri Madenciliği Süreci

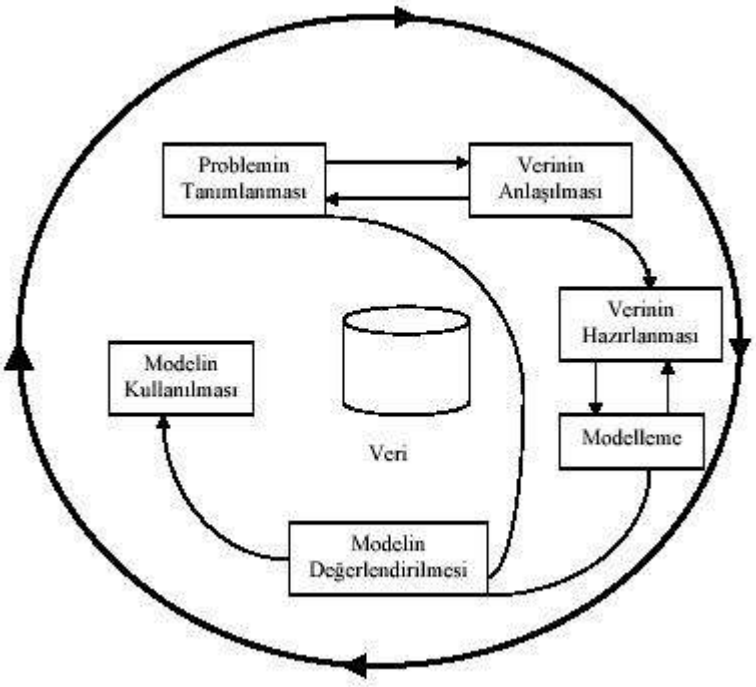
1995 yılında birincisi düzenlenen Veritabanları Bilgi Keşfi konferansı bildiri kitabında, bilgi teknolojilerinin oluşturduğu veri dağları aşağıdaki cümleler ile ifade edilmiştir [31].

“Dünyadaki bilgi miktarının her 20 ayda bir ikiye katlandığı tahmin edilmektedir. Bu ham veri seli ile ne yapmamız gerekmektedir? İnsan gözleri bunun ancak çok küçük bir kısmını görebilecektir. Bilgisayarlar bilgelik pınarı olmayı vaat etmekte, ancak veri sellerine neden olmaktadır.”

Veri madenciliği süreci çok hızlı bir biçimde karmaşık hale gelebilir. Bu sebeple veri madenciliği için standart bir süreç söz konusudur. Bu standart süreç bir konsorsiyum tarafından belirlenmiştir. The Cross- Industry Standard Process for Data Mining (CRISP-DM) konsorsiyumu, 1996 yılının sonlarına doğru Daimler Chrysler, SPSS ve NCR firmaları tarafından kurulmuştur [32].

Bu gelişmelerden bir yıl sonra, sözcüklerin baş harfleri Cross-Industry Standard Process for Data Mining açılımında olan CRISP-DM konsorsiyumu oluşturulmuş, Avrupa Komisyonundan fon elde edilmiş ve başlangıç fikirleri oluşturulmaya başlanmıştır. CRISP-DM süreci 6 aşamadan oluşmaktadır. Bu aşamalar şekil 5.1' de gösterilmiştir [33]:

- Problemin Tanımlanması
- Verinin Hazırlanması
- Modelin Kurulması
- Modelin Değerlendirilmesi
- Modelin Kullanılması
- Modelin İzlenmesi



Şekil 5.1. Veri madenciliği süreçleri [33]

5.2.1. Problemin tanımlanması

Veri madenciliği çalışmalarında başarılı olmanın en önemli şartı, projenin hangi işletme amacı için yapılacağının açık bir şekilde tanımlanmasıdır. İlgili işletme amacı, işletme problemi üzerine odaklanmış ve açık bir dille ifade edilmiş olmalı, elde edilecek sonuçların başarı düzeylerinin nasıl ölçüleceği tanımlanmalıdır. Ayrıca yanlış tahminlerde atlanılacak olan maliyetlere ve doğru tahminlerde kazanılacak faydalara ilişkin tahminlere de bu aşamada yer verilmelidir [30].

5.2.2. Verinin hazırlanması

Verinin hazırlanması aşaması dört temel aşamadan oluşmaktadır. Bu aşamalar:

- Verinin toplanması,
- Verinin birleştirilmesi,
- Verinin temizlenmesi,
- Verinin dönüştürülmesidir.

Veri hazırlanması sürecinde, veri toplanması en önemli adımlardan birisidir. Bu aşamada verilerin belirlenen amaçlara uygun elde edilmesi, gerek veri hazırlama aşamasını gerekse tüm veri madenciliği sürecini doğrudan etkilemektedir.

Veri birleştirme sürecinde, farklı kaynaklardan toplanan verilerin aynı formatta çevrilmesi gerekmektedir. Böylece farklı kaynaklardan alınan verilerin hepsine aynı analizler uygulanabilmektedir.

Veri temizleme sürecinde, verilere uygulanan yöntem esas olarak analizi yanlış yönlere sürükleyebilecek olan eksik ya da aykırı verilerin veri topluluğundan çıkarılmasıyla veri madenciliği sürecine etkisi ortadan kaldırılmaktadır.

Veri dönüştürme sürecinde ise, verilerin farklı formlarını analize uygun olacak şekilde dönüştürülmesi sürecidir [34].

5.2.3. Modelin kurulması ve değerlendirilmesi

Tanımlanan problem için en uygun modelin bulunabilmesi, olabildiğince çok sayıda modelin kurularak denenmesi ile mümkündür. Bu nedenle veri hazırlama ve model kurma aşamaları, en iyi olduğu düşünülen modele ulaşılıncaya kadar yenilenen bir süreçtir.

Bir modelin doğruluğunun test edilmesinde kullanılan en basit yöntem basit geçerlilik testidir. Bu yöntemde tipik olarak verilerin %5 ile %33 arasındaki bir kısmı test verileri olarak ayrılır ve kalan kısım üzerinde modelin öğrenimi gerçekleştirildikten sonra, bu veriler üzerinde test işlemi yapılır. Bir sınıflama modelinde yanlış olarak sınıflanan olay sayısının, tüm olay sayısına bölünmesi ile hata oranı, doğru olarak sınıflanan olay sayısının tüm olay sayısına bölünmesi ile ise doğruluk oranı hesaplanır.

(Doğruluk Oranı = 1 – Hata Oranı)

Sınırlı miktarda veriye sahip olunması durumunda, kullanılabilecek diğer bir yöntem ise çapraz geçerlilik (Cross Validation) testidir. Bu yöntemde veri kümesi tesadüfi olarak iki eşit parçaya ayrılır.

Bir diğer önemli değerlendirme kriteri ise modelin anlaşılabilir olmasıdır. Bazı uygulamalarda doğruluk oranlarındaki küçük artışlar çok önemli olsa da, bir çok işletme uygulamasında ilgili kararın niçin verildiğinin yorumlanabilmesi çok daha büyük önem taşıyabilir [35].

5.2.4. Modelin kullanılması

Veri madenciliği sürecinin son aşaması, kurulan ve geçerliliği kabul edilen modelin kullanılmasıdır. Bu doğrudan bir uygulama olabileceği gibi bir başka modelin alt parçası olarak da kullanılabilir [8].

5.2.5. Modelin izlenmesi

Zaman içerisinde bütün sistemlerin özelliklerinde ve dolayısıyla ürettikleri verilerde ortaya çıkan değişiklikler, kurulan modellerin sürekli olarak izlenmesini ve gerekiyorsa yeniden düzenlenmesini gerektirecektir. Tahmin edilen ve gözlenen değişkenler arasındaki farklılığı gösteren grafikler model sonuçlarının izlenmesinde kullanılan yararlı bir yöntemdir [30].

5.3. Veri Madenciliğinde Karşılaşılabilecek Önemli Sorunlar

Veri madenciliğinde ortaya çıkan sorunların temelinde iki unsur yatmaktadır. Bunlar ilk olarak işletmenin hangi amaçla veri madenciliği yaptığı, diğeri ise elde bulunan verilerden kaynaklanmaktadır. Karşılaşılan bazı problemler şunlardır [8].

- Veritabanının boyutları,
- Dinamik veri yapısı,
- Eksik veri,
- Gürültü,
- Eksik değerlerdir.

5.3.1. Veritabanının boyutları

Veritabanın boyutlarının veriler için yeterli olmaması durumunda, yapılan analizlerin uygulanabilirliği yoktur. Bu problemin çözümünde ise, yapılan örnekleme tekniklerinin ve örnek miktarının azaltılmasıyla bu problem aşılabilmektedir [34].

5.3.2. Dinamik veri yapısı

Veri tabanları dinamik olma eğilimindedir. İçerikleri her zaman ekleme, düzeltme ve silme işlemleri ile değişim halindedir. Veri madenciliği bakımından bu problem, bugüne kadarki en doğru bilgiye uygun kuralların nasıl temin edileceğidir [36].

5.3.3. Eksik veri

Bir veritabanı sıklıkla veri madenciliği ve basit öğrenme işlerini sağlayan özellik veya nitelikleri sunmak gibi farklı amaçlar için tasarlanır. Veritabanlarında toplanan bilgiler eksik olabilir. Eksik veri problemlere sebep olmaktadır. Çünkü bazı veriler geçerli etki alanında sunulamaz. Örneğin; hasta veritabanı, kırmızı kan hücreli hasta bilgilerini barındırmıyorsa, hasta veritabanından sıtma teşhisi yapılamaz. Sorunu çözmek için bu tür veriler geliştirilir [8].

5.3.4. Gürültü

Veri özellikleri ya da sınıflarındaki hatalara gürültü adı verilir. Veri tabanlarındaki eksik bilgi ve bu yanlışlardan dolayı veri madenciliği amacına tam olarak ulaşmayabilir. Bu bilgi yanlışlığı, ölçüm hatalarından, ya da öznel yaklaşımdan olabilir [37].

5.3.5. Eksik değer

Yapılan her analiz için büyük bir sorun olan veri değerlerinin hatalı olması, tüm analizi işlevsiz kılabilecek büyük bir sorundur. Genellikle veri toplama ya da girilme aşamasında oluşan bu hata analizi doğrudan etkilemektedir [34].

5.4. Veri Madenciliğinde Kullanılan Modeller

Veri madenciliğinde kullanılan modeller, temel olarak tahmin edici ve tanımlayıcı olmak üzere iki ana baslık altında toplanabilir. Tahmin edici modeller ile tanımlayıcı modeller arasındaki fark kesin sınırlarla ayrılmamıştır. Tahmin edici modeller anlaşılabilir olduğu ölçüde tanımlayıcı model olarak, tanımlayıcı modeller de tahmin edici model olarak kullanılabilirler [38].

Tahmin edici modellerde, sonuçları bilinen verilerden hareket edilerek bir model geliştirilmesi ve kurulan bu modelden yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır. Örneğin, bir banka önceki dönemlerde vermiş olduğu kredilere ilişkin gerekli tüm verilere sahip olabilir. Bu verilerde bağımsız değişkenler kredi alan müşterinin özellikleri, bağımlı değişken

değeri ise kredinin geri ödenip ödenmediğidir. Bu verilere uygun olarak kurulan model, daha sonraki kredi taleplerinde müşteri özelliklerine göre verilecek olan kredinin geri ödenip ödenmeyeceğinin tahmininde kullanılmaktadır [39].

Tanımlayıcı modeller analiste daha önceden bir hipoteze sahip olmaksızın, veri kümesinin içinde ne tür ilişkiler olduğunu anlama imkanı sunar. Analizcinin çok geniş veri tabanlarındaki bilgileri incelemek, örüntüleri keşfetmek için doğru soruları sorup hipotezler geliştirmesi pratikte zor olduğundan, ilginç örüntüleri keşfetme inisiyatifi veri madenciliği programına bırakılır. Keşfedilen bilginin kalitesi ve zenginliği, uygulamanın kullanışlılığını ve gücünü oluşturur [40].

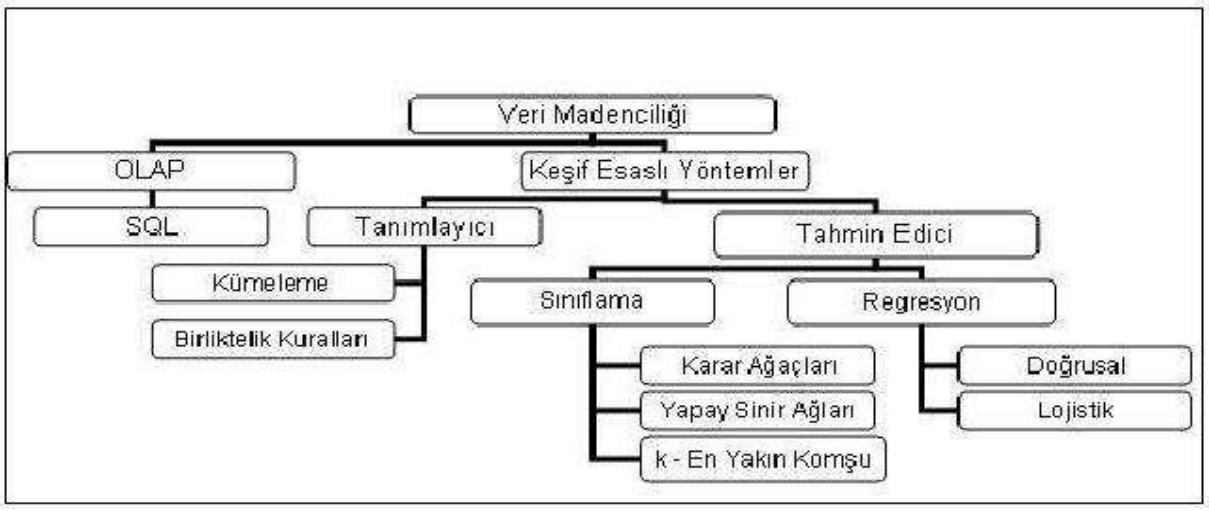
Veri madenciliği modelleri gördükleri işlevlere göre 3 ana başlık altında toplanabilir. Bunlar:

1. Sınıflama (Classification) ve Regresyon (Regression).

2. Kümeleme (Clustering).

3. Birliktelik Kuralları (Association Rules) ve Ardışık Zamanlı Örüntüler (Sequential Patterns).

Sınıflama ve regresyon modelleri tahmin edici, kümeleme ve birliktelik kuralları modelleri tanımlayıcı modellerdir. Şekil 5.2‘ de bu ilişkiler özetlenmiştir [31].



Şekil 5.2. Veri madenciliği modelleri [31]

5.4.1. Sınıflama ve regresyon algoritması

Sınıflama ve regresyon, önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin eden modelleri kurabilen iki veri analiz yöntemidir. Sınıflama kategorik değerleri tahmin ederken, regresyon süreklilik gösteren değerlerin tahmin edilmesinde kullanılır. Örneğin, bir sınıflama modeli banka kredi uygulamalarının güvenli veya riskli olmalarını kategorize etmek amacıyla kurulurken, regresyon modeli geliri ve mesleği verilen potansiyel müşterilerin bilgisayar ürünleri alırken yapacakları harcamaları tahmin etmek için kurulabilir [39].

Sınıflama ve regresyon modellerinde kullanılan başlıca yöntemler şunlardır [41]:

- Diskriminant analizi,
- Naive-Bayes,
- Karar ağaçları,
- Yapay sinir ağları,
- Kaba kümeler,
- Genetik algoritma,
- Regresyon analizi.

Diskriminant analizi

Diskriminant (ayırma) analizi, iki veya daha fazla sayıdaki grubun ayrımı ile ilgilenen çok değişkenli bir istatistik analiz tekniğidir. Diskriminant analizi bağımlı değişkenin nominal (metrik olmayan veya kategorik), bağımsız değişkenlerin ise metrik olduğu hallerde kullanılan en uygun tekniktir. İki gruplu bağımlı değişken söz konusu olduğunda analiz diskriminant analizi olarak ifade edilirken; grup sayısı üç veya daha fazla olduğunda analiz çoklu diskriminant analizi adını almaktadır. Diskriminant analizinde amaç, çok değişkenli problemin tek değişkenli biçime

dönüştürülmesidir. Yani tüm değişkenlerin uygun ağırlıklarla katılacağı tek bir değişkenin (fonksiyon) elde edilmesidir [36].

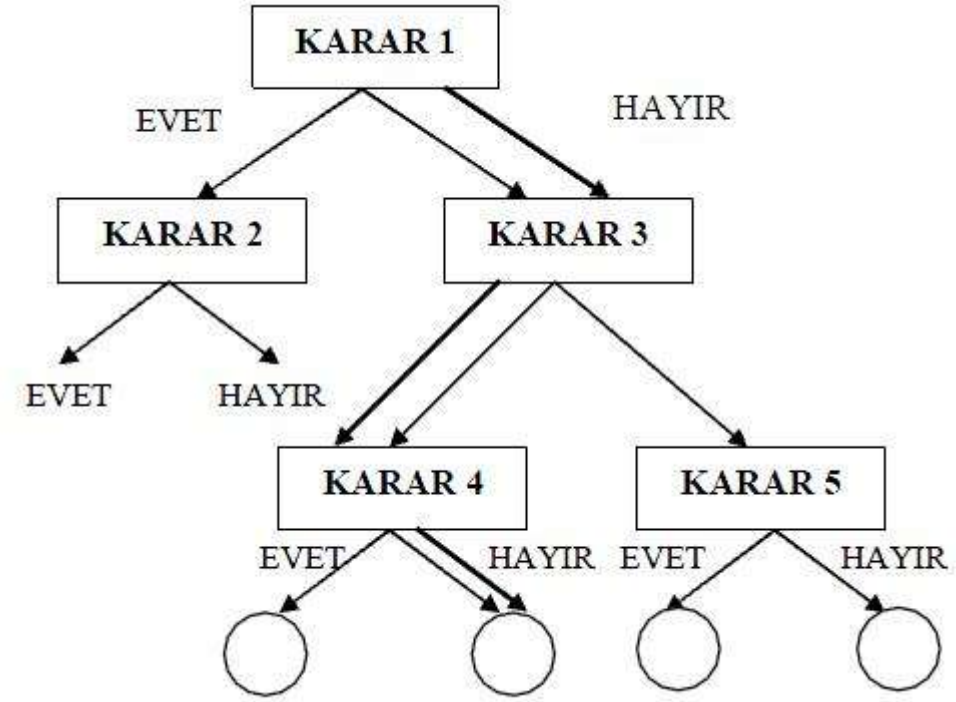
Naive-Bayes

Naive Bayes, hedef değişkenle bağımsız değişkenler arasındaki ilişkiyi analiz eden tahminci ve tanımlayıcı bir sınıflama algoritmasıdır. Naive Bayes, modelin öğrenilmesi esnasında, her çıktının öğrenme kümesinde kaç kere meydana geldiğini hesaplar. Bulunan bu değer, öncelikli olasılık olarak adlandırılır. Örneğin; bir banka kredi kartı başvurularını iyi ve kötü risk sınıflarında gruplandırmak istemektedir. İyi risk çıktısı toplam 5 vaka içinde 2 kere meydana geldiyse iyi risk için öncelikli olasılık 0,4' tür. Bu durum, "Kredi kartı için başvuran biri hakkında hiçbir şey bilinmiyorsa, bu kişi 0,4 olasılıkla iyi risk grubundadır" olarak yorumlanır. Naive Bayes, bağımsız ve bağımlı değişken kombinasyonunun meydana gelme sıklığını bulur. Bu sıklıklar öncelikli olasılıklarla birleştirilmek suretiyle tahminde kullanılır [41].

Karar ağaçları

Karar ağaçları (decision trees), sınıflandırma, kümeleme ve tahmin modellerinde kullanılan bir tahmin tekniğidir. Sorunla ilgili araştırma alanını alt gruplara ayırmak için kullanılır. Karar ağaçlarında kök ve her düğüm bir soruyla etiketlenir. Düğümlerden ayrılan dallar ise ilgili sorunun olası yanıtlarını belirtir. Her dal düğümü de söz konusu sorunun çözümüne yönelik bir tahmini temsil eder [39].

Karar ağacı olası soruna ait tüm eylemlerin yönlerini, eylemlerin yönlerine etkisi olabilecek tüm olası faktörleri ve tüm bu faktörlere dayanan her bir olası sonucu, verilere bağlı olarak değerlendiren, çizgi, kare, daire gibi geometrik semboller kullanımı yoluyla karar vericiye sorunu anlamada kolaylık sağlayan düzenleme biçiminde tanımlanabilir. Bu tanıma göre karar ağacının türlü eylem seçeneklerini, farklı olası etkenlerin ve eylemlerin sonuçlarını içerdiği söylenebilir. Şekil 5.3' de örnek bir karar ağacı yapısı gösterilmiştir [36].



Şekil 5.3. Basit bir karar ağacı yapısı

Yapay sinir ağları

İlk kez 1943' te ortaya çıktı ama bilgisayarlarda kullanımı 1980' lerde başladı. Yapay sinir ağları (artificial neural networks), beynin yapısından esinlenilmiş bir bilgi işleme sistemidir. Nöronlara benzeştirilmiş işlem öğeleri arasındaki ilişkilerle yapılandırılmıştır. İnsan beyni gibi yapay sinir ağı da birbirine bağlı birçok işlem biriminden oluşmuştur. Sonra, birçok düğüm (işlem birimi) ve arkla (iç bağlantılar) yönetilen bir grafik olarak yapılandırılır. Bu işlem birimleri birbirlerinden bağımsız işlev görürler ve yalnızca yerel veriyi (düğüme gelen girdi ve düğümden çıkan çıktı) kullanırlar. Bu özellik, sinir ağlarının dağıtık ya da paralel ortamlarda kullanımını kolaylaştırır. Sinir ağları, kaynak (girdi), çıktı ve iç (gizli) düğümlerle yönetilen bir grafik olarak görülebilir. Girdi düğümü girdi katmanında, çıktı düğümü ise çıktı katmanında bulunur. Gizli düğümler, bir ya da daha çok gizli katmanda bulunur. Veri madenciliğinde, çıktı düğümü tahmini belirler. Tek bir girdi düğümünün olduğu (ağacın kökü) karar ağaçlarından farklı olarak sinir ağlarında, her öznitelik değeri için bir girdi düğümü vardır. Sinir ağları karmaşık sorunları çözebilir, ayrıca temel

uygulamalardan öğrenebilir. Yani, soruna kötü bir çözüm bulunduysa, ağ bu soruna bir dahaki sefer daha iyi bir çözüm bulacak biçimde değiştirilir. Sinir ağları üç bölümden oluşur [42]:

- Sinir ağının veri yapısını tanımlayan sinir ağı grafiği.
- Öğrenmenin nasıl gerçekleşeceğini belirten öğrenme algoritması.
- Bilginin ağdan nasıl elde edileceğini belirleyen teknikler.

Bu tez çalışması içerisinde geliştirilen STS yazılımında YSA tekniği kullanıldığı için, YSA' ya bölüm 5.5' de ayrıntılı olarak değinilmiştir.

Kaba kümeler

Kaba küme teorisi 1970' li yıllarda Pawlak tarafından geliştirilmiştir. Kaba küme teorisinde bir yaklaştırma uzayı ve bir kümenin alt ve üst yaklaştırmaları vardır. Yaklaştırma uzayı, ilgilenilen alanı ayrı kategorilerde sınıflandırır. Alt yakınlaştırma belirli bir altkümeye ait olduğu kesin olarak bilinen nesnelerin tanımıdır. Üst yakınlaştırma ise alt kümeye ait olması olası nesnelerin tanımıdır. Alt ve üst sınırlar arasında tanımlanan herhangi bir nesne ise kaba küme olarak adlandırılır [41].

Genetik algoritma

Evrimsel hesaplama ilk olarak 1960' larda I. Rechenberg tarafından "Evrim Stratejileri (Evolutions Strategies)" isimli eserinde tanıtılmıştır. Onun fikri daha sonra başka araştırmacıların da ilgisini çekmiş ve teknik geliştirilmiştir. John Holland evrim sürecinin bir bilgisayar yardımıyla kullanılarak, bilgisayara anlayamadığı çözüm yöntemlerinin öğretilebileceğini düşünmüştür. Genetik algoritma (GA) böylece John Holland tarafından bu düşüncenin bir sonucu olarak bulunmuştur. 1992 yılında John Koza genetik algoritmayı kullanarak çeşitli görevleri yerine getiren programlar geliştirmiştir. Bu metoda genetik algoritma adını vermiştir [43].

Genetik algoritmalar çok değişkenli fonksiyonların optimizasyonu amacıyla kullanılan nümerik araştırma araçlarıdır. Olasılık kurallarına göre çalışan genetik

algoritmalar, yalnızca amaç fonksiyonuna ihtiyaç duyarlar. Geleneksel optimizasyon yöntemlerine göre farklılıkları olan genetik algoritmalar, parametre kümesini değil kodlanmış biçimlerini kullanırlar. Genetik algoritmalarda, başlangıç olarak bir çözüm seti oluşturulur ve bu çözümü geliştirmek için biyolojik evrimi esas alan bir süreç kullanılır. Bu sürecin sonunda da en iyi kromozoma ulaşmak amaçtır. Çözüm uzayının tamamı değil bir kısmı taranır. Etkin arama yapılarak çok daha kısa bir zamanda çözüme ulaşılır [36].

Regresyon analizi

Regresyon analizi, bir ya da daha fazla bağımsız değişken ile hedef değişken arasındaki ilişkiyi matematiksel olarak modelleyen bir yöntemdir. Veri madenciliğinde yaygın olarak kullanılan regresyon modellerinden doğrusal regresyonda tahmin edilecek olan hedef değişken sürekli değer alırken; lojistik regresyonda hedef değişken kesikli bir değer almaktadır. Doğrusal regresyonda hedef değişkenin değeri; lojistik regresyonda ise hedef değişkenin alabileceği değerlerden birinin gerçekleşme olasılığı tahmin edilmektedir [41].

5.4.2. Kümeleme

Kümeleme işlemi, heterojen yapıya sahip bir kitleyi daha homojen birkaç alt gruba ya da kümeye bölme işlemidir. Sınıflama ile kümelemeyi birbirinden ayıran en önemli fark, kümeleme işleminin sınıflama işleminde olduğu gibi önceden belirlenmiş bir takım sınıflara göre bölme yapmamasıdır. Sınıflamada her bir veri, önceden sınıflandırılmış bir takım sınıflar üzerinde yapılan bir eğitim neticesinde ortaya çıkan bir modele göre önceden belirlenmiş olan bir sınıfa atanmaktadır. Kümeleme işleminde ise önceden tanımlanmış sınıflar ya da örnek sınıflar bulunmamaktadır. Verilerin kümelenmesi işlemi, verilerin birbirlerine olan benzerliklerine göre yapılmaktadır. Oluşan sınıfların hangi anlamları taşıdığının belirlenmesi tamamen çözümlemeyi yapan kişiye kalmıştır. Kümeleme işlemi çoğunlukla bir başka veri madenciliği uygulaması için bir ilk işlem olarak kullanılır [31].

5.4.3. Birliktelik kuralları ve ardışık zamanlı örüntüler

Birliktelik analizi, bir veri kümesindeki kayıtlar arasındaki bağlantıları arayan denetimsiz veri madenciliği şeklidir. Birliktelik analizi çoğu zaman perakende sektöründe süpermarket müşterilerinin satın alma davranışlarını ortaya koymak için kullanıldığından "pazar sepeti analizi" olarak da adlandırılır. Ardışık analiz ise birbiriyle ilişkisi olan ancak birbirini izleyen dönemlerde gerçekleşen ilişkilerin tanımlanmasında kullanılır [41].

Birliktelik kurallarına ait örnekler aşağıda yer almaktadır:

- Müşteriler bira satın aldıklarında %75 olasılıkla çocuk bezi de satın alırlar.
- Düşük yağlı peynir ve yağsız süt alan müşteriler %85 olasılıkla diyet süt alırlar.

Aşağıda ardışık analize ait örnekler yer almaktadır:

- Çadır alan müşterilerin %10' u bir ay içerisinde sırt çantası almaktadır.
- A hissesi %15 artarsa üç gün içinde B hissesi %60 olasılıkla artacaktır.

5.5. Yapay Sinir Ağları (YSA)

Genel anlamda YSA, beynin bir işlevini yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. Bir YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur. YSA'lar öğrenme algoritmaları ile öğrenme sürecinden geçtikten sonra, bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip olurlar. YSA'lar yapılarına göre farklı öğrenme yaklaşımları kullanırlar [44].

Yapay sinir ağlarının dayandığı ilk hesaplama modelinin temelleri 1940'ların başında araştırmalarına başlayan W.S. McCulloch ve W.A. Pitts'in, 1943 yılında yayınladıkları bir makaleyle atılmıştır. Daha sonra 1954 yılında B.G. Farley ve W.A. Clark tarafından bir ağ içerisinde uyarılara tepki veren, uyarılara adapte olabilen model oluşturulmuştur. 1960 yılı ise ilk neural bilgisayarın ortaya çıkış yılıdır. 1963

yılında basit modellerin ilk eksiklikleri fark edilmiş fakat başarılı sonuçların alınması 1970 ve 1980'lerde termodinamikteki teorik yapıların doğrusal olmayan ağların geliştirilmesinde kullanılmasına kadar gecikmiştir. 1985 yapay sinir ağlarının tanındığı ve yoğun araştırmaların başladığı yıl olmuştur [45].

5.5.1. Literatürdeki YSA tanımları

Yapay sinir ağının genel bir tanımı yapılması gerekirse, Yapay Sinir Ağı, insan beyninin çalışma ve düşünebilme yeteneğinden yola çıkılarak oluşturulmuş bir bilgi işlem teknolojisidir [46].

Yapay sinir ağının işleyiş özelliklerine dayanan ikinci tür tanımı ise ilk ticari yapay sinir ağının geliştiricisi olan Dr. Robert HECHT-NIELSEN'e ait bir tanımdır: "Yapay sinir ağı dışarıdan gelen girdilere dinamik olarak yanıt oluşturma yoluyla bilgi işleyen, birbiriyle bağlantılı basit elemanlardan oluşan bilgi işlem sistemidir. Bu tanıma yakın bir tanımda yapay sinir ağında çok tanınan Teuvo KOHONEN'e ait bir tanımdır:" Yapay sinir ağları paralel olarak bağlantılı ve çok sayıdaki basit elemanın, gerçek dünyanın nesneleriyle biyolojik sinir sisteminin benzeri yolla etkileşim kuran, hiyerarşik bir organizasyonudur [46].

5.5.2. YSA' nın üstünlükleri

Yapay sinir ağları modelleri, biyolojik sinir ağlarının çalışma biçimlerinden esinlenerek ortaya çıkarılmıştır. Yapay sinir ağları, biyolojik olmayan yapı taşlarının düzgün bir tasarımla birbirlerine yoğun olarak bağlanmalarından oluşmaktadırlar. Sinir sisteminin modellenmesi için yapılan çalışmalar sonucu oluşturulan yapay sinir ağları, biyolojik sinir sisteminin üstünlüklerine de sahiptir. Bu üstünlükler şu şekillerde özetlenebilir[46]:

Doğrusal Olmama: YSA' nın temel işlem elemanı olan hücre doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özelliği ile YSA, doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmuştur.

Paralellik: Alışılmış bilgi işlem yöntemlerinin çoğu seri işlemlerden oluşmaktadır. Bu da hız ve güvenilirlik sorunlarını beraberinde getirmektedir. Seri bir işlem gerçeklenirken herhangi bir birimin yavaş oluşu tüm sistemi doğruca yavaşlatırken, paralel bir sistemde yavaş bir birimin etkisi çok azdır. Nitekim seri bir bilgisayarın bir işlem elemanı beyine göre binlerce kez daha hızlı işlemesine rağmen, beynin toplam işlem hızı seri çalışan bir bilgisayara göre kıyaslanamayacak kadar yüksektir.

Gerçeklenme Kolaylığı: Yapay sinir ağlarında basit işlemler gerçekleyen türden hücrelerden oluşması ve bağlantıların düzgün olması, ağların gerçeklenmesi açısından büyük kolaylık sağlamaktadır.

Yerel Bilgi İşleme: Yapay sinir ağlarında her bir işlem birimi, çözülecek problemin tümü ile ilgilenmek yerine, sadece problemin gerekli parçası ile ilgilenmektedir ve problemin bir parçası işlemektedir. Hücrelerin çok basit işlem yapmalarına rağmen, sağlanan görev paylaşımı sayesinde, çok karmaşık problemler çözülebilmektedir.

Hata Toleransı: Sayısal bir bilgisayarda, herhangi bir işlem elemanını yerinden almak, onu etkisiz bir makineye dönüştürmektedir. Ancak yapay sinir ağlarında bir elemanda meydana gelebilecek hasar çok büyük önem teşkil etmez. Yapay sinir ağlarının paralel çalışması hız avantajı ile birlikte düşük hataya sebep olmaktadır. Seri bilgi işleyen bir sistemde herhangi bir birimin hatalı çalışması, hatta bozulmuş olması tüm sistemin hatalı çalışmasına veya bozulmasına sebep olacaktır. Paralel bilgi işleme yapan bir sistemde ise, sistemin ayrı ayrı işlem elemanlarında meydana gelecek olan hatalı çalışma veya hasar, sistemin performansında keskin bir düşüşe yol açmadan, performansın sadece hata birimlerinin bir oranınca düşmesine sebep olur. YSA, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuğundan paralel dağılmış bir yapıya sahiptir ve ağın sahip olduğu bilgi, ağdaki bütün bağlantılar üzerine dağılmış durumdadır. Bu nedenle, eğitilmiş bir YSA'nın bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ağın doğru bilgi üretmesini önemli ölçüde etkilemez ve hatayı tolere etme yetenekleri son derece yüksektir.

Öğrenebilirlik: Alışılagelmiş veri işleme yöntemlerinin çoğu programlama yolu ile hesaplamaya dayanmaktadır. Bu yöntemler ile tam tanımlı olmayan bu problemin çözümü yapılamaz. Bunun yanında, herhangi bir problemin çözümü için probleme yönelik bir algoritmanın geliştirilmesi gerekmektedir. Yapay sinir ağları problemleri verilen örneklerle çözer. YSA' nın arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. YSA' nın karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle YSA, ilgilendiği problemden aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

Genelleme: YSA, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de arzu edilen tepkiyi üretebilir. Örneğin, karakter tanıma amacıyla eğitilmiş bir YSA, bozuk karakter girişlerinde de doğru karakterleri verebilir ya da bir sistemin eğitilmiş YSA modeli, eğitim sürecinde verilmeyen giriş sinyalleri için de sistemle aynı davranışı gösterebilir.

Uyarlanabilirlik: YSA, ilgilendiği problemdeki değişikliklere göre ağırlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere göre tekrar eğitilebilir, değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir. Bu özelliği ile YSA, uyarlamalı örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılır.

Donanım ve Hız: YSA, paralel yapısı nedeniyle büyük ölçekli entegre devre teknolojisiyle gerçeklenebilir. Bu özellik, YSA' nın hızlı bilgi işleme yeteneğini artırır ve gerçek zamanlı uygulamalarda arzu edilir.

Analiz ve Tasarım Kolaylığı: YSA' nın temel işlem elemanı olan hücrenin yapısı ve modeli, bütün YSA yapılarında yaklaşık aynıdır. Dolayısıyla, YSA' nın farklı uygulama alanlarındaki yapıları da standart yapıdaki bu hücrelerden oluşacaktır. Bu nedenle, farklı uygulama alanlarında kullanılan YSA' ları benzer öğrenme algoritmalarını paylaşabilirler. Bu özellik, problemlerin YSA ile çözümünde önemli bir kolaylık getirecektir [46].

5.5.3. YSA' nın uygulama alanları

Son yıllarda YSA' ları, özellikle günümüze kadar çözümü güç ve karmaşık olan ya da ekonomik olmayan çok farklı alanlardaki problemlerin çözümüne uygulanmış ve genellikle başarılı sonuçlar alınabilmiştir. Yapay sinir ağları aşağıdaki özellikleri gösteren alanlarda kullanıma uygun bir araçtır [46] :

- Çok değişkenli problem uzayı,
- Probleme ilişkin değişkenler arasında karmaşık etkileşim,
- Çözüm uzayının bulunmaması, tek bir çözümün olması veya çok sayıda çözüm bulunması.

YSA uygulamaları temel olarak tahmin, sınıflandırma, veri yorumlama, veri filtreleme ve veri ilişkilendirme olmak üzere 5 sınıfa ayrıştırılabilir. Bunlar [47] ;

Tahmin (Prediction): Uygulanan giriş değerlerinin bazı çıkış değerleri bulunmaya çalışılır. Hava durumu tahmini, kanser riskini belirleme buna örnek olarak verilebilir.

Sınıflandırma (Classification): Giriş değerleri kullanılarak sınıflandırma yapılır. Arıza sınıflandırma, karakter tanıma, hastalık teşhis etme buna örnek olarak verilebilir.

Veri İlişkilendirme (Data Association): Bu, sınıflandırmaya benzer bir yaklaşım gibi görünse de buna ilave olarak hatalı olan verileri tanımlar. Örnek olarak taranan bir dokümandaki karakterleri algılamanın yanında tarayıcının düzgün olarak çalışmadığını da algılayabilir.

Veri Yorumlama (Data Conceptualization): Giriş verisindeki gruplar arasındaki ilişkileri analiz etme işlemidir. Bir veritabanındaki benzer verileri gruplandırma örneği verilebilir.

Veri Filtreleme (Data Filtering): Giriş sinyalleri içerisindeki uygun olmayan verilerin ayıklanmasını sağlar. Bir telefon sinyali içerisinde bulunan gürültüyü ayıklama, bir resimdeki istenmeyen parazitleri temizleme örnek olarak verilebilir.

YSA'nın birçok alanda uygulamaları vardır. Bunlardan bazıları [47];

- Havacılık ve uzay alanında yüksek performanslı oto-pilot geliştirme çalışmalarında, uçuş kontrol sistemlerinde ve simülasyon cihazlarında.
- Otomotiv sektöründe YSA'dan oto-rehberlik uygulamaları geliştirmede.
- Bankacılıkta, kredi risk analizlerinde, kontrol ve doküman okuma uygulamalarında.
- Savunma sanayinde hedef tanıma, hedef izleme, silah oryantasyonu, sayısal görüntü işleme, veri kodlama ve sıkıştırmada.
- Bilgisayar destekli görme, ses tanıma, sinyal işleme ve filtrelemede.
- Tıpta teşhis ve bioistatistiksel ilişkilerin aranmasında.
- Robotikte, yüzey modellemede, yol ve hız tahmininde.
- Coğrafi bilgi sistemlerinde hareketli cisimlerin izlenmesi, konum ve durumlarının tahmininde.
- CAD/CAM uygulamalarında yüzey interpolasyonu, yüzey izleme, yüzey modelleme uygulamalarında.
- Meteorolojide hava tahmin algoritmalarının geliştirilmesinde, yağmur yükü tahminlerinde kullanılır.

5.5.4. YSA' nın çalışması

Sinir ağı ile hesaplamalarda istenilen dönüşüm için, adım adım yürütülen bir yöntem gerekmez. Sinir ağı ilişkilendirmeyi yapan iç kuralları kendi üretir ve bu kuralları

sonuçları örneklerle karşılaştırarak düzenler. Deneme ve yanılma ile ağ kendi kendine işi nasıl yapması gerektiğini öğretir. YSA'larda bilgi saklama, verilen eğitim özelliğini kullanarak eğitim örnekleri ile yapılır. Sinirsel hesaplama, algoritmik programlamaya bir seçenek oluşturan, temel olarak yeni ve farklı bir bilgi işleme olayıdır. Uygulama imkânının olduğu her yerde, tamamen yeni bilgi işleme yetenekleri geliştirebilir [44].

Bir yapay sinir ağı girdi setindeki değişiklikleri değerlendirerek öğrenir ve buna bir çıktı üretir. Öğrenme işlemi benzer girdi setleri için aynı çıktıyı üretecek bir öğrenme algoritması ile gerçekleşir. Öğrenme setindeki girdilerin istatistiksel özelliklerinin çıkarılarak benzer girdilerin gruplandırılmasını sağlayan bir işlemdir [45].

Sinir yapılarına benzetilerek bulunan ağların eğitimi de, normal bir canlının eğitimine benzemektedir. Sınıfların birbirinden ayrılması işlemi (dolayısıyla kendini geliştirmesi), öğrenme algoritması tarafından örnek kümeden alınan bilginin adım adım işlenmesi ile gerçeklenir. YSA kullanılarak makinelere öğrenme genelleme yapma, sınıflandırma, tahmin yapma ve algılama gibi yetenekler kazandırılmıştır [44].

5.5.5. YSA’ nın eğitimi ve testi

Geleneksel bilgisayar uygulamalarının geliştirilmesinde karşılaşılan durum, bilgisayarın belli bilgisayar dilleri aracılığıyla ve kesin yazım algoritmalarına uygun ifadelerle programlanmasıdır. Bu oldukça zaman alan, uyumluluk konusunda zayıf, teknik personel gerektiren, çoğu zaman pahalı olan bir süreçtir. Oysa biyolojik temele dayalı yapay zeka teknolojilerinden biri olan yapay sinir ağlarının geliştirilmesinde programlama, yerini büyük ölçüde "eğitime" bırakmaktadır. Proses elemanlarının bağlantı ağırlık değerlerinin belirlenmesi işlemine “ağın eğitilmesi” denir. Yapay sinir ağının eğitilmesinde kullanılan girdi ve çıktı dizileri çiftinden oluşan verilerin tümüne "eğitim seti" adı verilir [44].

Yapay sinir ağı öğrenme sürecinde, gerçek hayattaki problem alanına ilişkin veri ve sonuçlardan, bir başka deyişle örneklerden yararlanır. Gerçek hayattaki problem

alanına ilişkin değişkenler yapay sinir ağının girdi dizisini, bu değişkenlerle elde edilmiş gerçek hayata ilişkin sonuçlar ise yapay sinir ağının ulaşması gereken hedef çıktıların dizisini oluşturur [47].

Öğrenme süresinde, seçilen öğrenme yaklaşımına göre ağırlıklar değiştirilir. Ağırlık değişimi, öğrenmeyi ifade eder. YSA' da ağırlık değişimi yok ise, öğrenme işlemi de durmuştur. Başlangıçta bu ağırlık değerleri rastgele atanır. YSA' lar kendilerine örnekler gösterildikçe, bu ağırlık değerlerini değiştirirler. Amaç, ağa gösterilen örnekler için doğru çıktıları üretecek ağırlık değerlerini bulmaktır. Ağın doğru ağırlık değerlerine ulaşması örneklerin temsil ettiği olay hakkında, genellemeler yapabilme yeteneğine kavuşması demektir. Bu genelleştirme özelliğine kavuşması işlemine, "ağın öğrenmesi" denir [44].

Yapay sinir ağının öğrenme sürecinde temel olarak üç adım bulunmaktadır [44].

- Çıktıları hesaplamak,
- Çıktıları hedef çıktılarla karşılaştırmak ve hatayı hesaplamak,
- Ağırlıkları değiştirerek süreci tekrarlamak.

Yapay sinir ağı öğrendikten sonra daha önce verilmeyen girişler verilip, sinir ağı çıkışıyla gerçek çıkışı yaklaşımı incelenir. Eğer yeni verilen örneklere de doğru yaklaşıyorsa sinir ağı işi öğrenmiş demektir. Sinir ağına verilen örnek sayısı optimum değerden fazla ise sinir ağı işi öğrenmemiş ezberlemiş demektir. Genelde eldeki örneklerin yüzde sekseni ağa verilip ağ eğitilir, daha sonra geri kalan yüzde yirmilik kısmı verilip ağın davranışı incelenir diğer bir deyişle ağ böylece test edilir [48].

Örneklerin toplanması: Ağın öğrenmesi istenilen olay için daha önce gerçekleşmiş örneklerin bulunması adımıdır. Ağın eğitilmesi için örnekler toplandığı gibi (eğitim seti) ağın test edilmesi için de örneklerin (test seti) toplanması gerekmektedir. Eğitim setindeki örnekler tek tek gösterilerek ağın olayı öğrenmesi sağlanır. Ağ olayı

öğrendikten sonra test setindeki örnekler gösterilerek ağın başarımı ölçülür. Ağın hiç görmediği örnekler karşısındaki başarısı iyi öğrenip öğrenmediğini ortaya koyar.

Ağın topolojik yapısının belirlenmesi: Öğrenilmesi istenen olay için oluşturulacak olan ağın topolojik yapısı belirlenir. Kaç tane girdi ünitesi, kaç tane ara katman, her ara katmanda kaç tane süreç eleman ve kaç tane çıktı eleman olması gerektiği bu adımda belirlenmektedir.

Öğrenme parametrelerinin belirlenmesi: Ağın öğrenme katsayısı, süreç elemanlarının toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler bu adımda belirlenmektedir.

Ağırlıkların başlangıç değerlerinin atanması: Süreç elemanlarını birbirlerine bağlayan ağırlık değerlerinin ve eşik değer ünitesinin ağırlıklarının başlangıç değerlerinin ataması yapılır. Başlangıç genellikle rastgele değerler atanır. Daha sonra ağ uygun değerleri öğrenme sırasında kendisi belirler.

Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi: Ağın öğrenmeye başlaması ve yukarıda anlatılan öğrenme kuralına uygun olarak ağırlıkları değiştirmesi için ağa örnekler (girdi/çıktı değerleri) belirli bir düzeneğe göre gösterilir.

Öğrenme sırasında ileri hesaplamaların yapılması: Sunulan girdi için ağın çıktı değerleri hesaplanır.

Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması: Ağın ürettiği hata değerleri bu adımda hesaplanır.

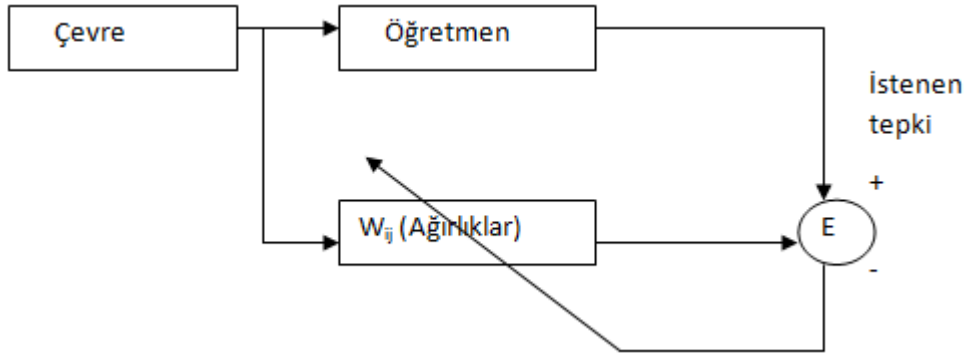
Ağırlıkların değiştirilmesi: Geri hesaplama yöntemi uygulanarak üretilen hatanın azalması için ağırlıkların değiştirilmesi yapılır.

İleri beslemeli sinir ağı öğrenmesi tamamlanıncaya, yani gerçekleşen ile beklenen çıktılar arasındaki hatalar kabul edilir düzeye ininceye kadar devam eder.

5.5.6. YSA' da öğrenme

YSA'da öğrenme danışmanlı ve danışmansız öğrenme olarak iki şekilde gerçekleştirilir.

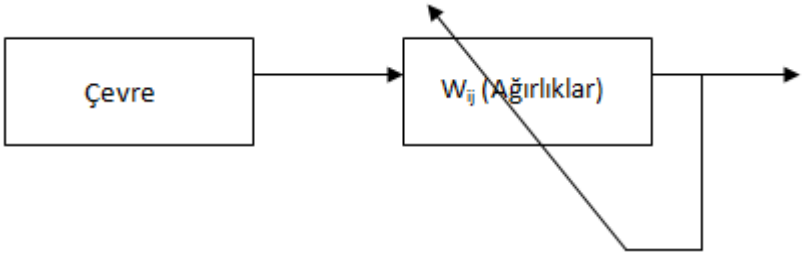
Danışmanlı Öğrenme: Bu tip öğrenmede, öğrenme aşamasında YSA'ya ne öğrenmesi gerektiği örnek bir çıkışla bildirilir. İstenilen veya arzu edilen çıkış ile gerçek çıkış (ağ çıkışı) arasındaki farka (hataya) göre, nöronlar arası bağlantıların ağırlığı, en uygun çıkışı elde etmek için bir öğrenme algoritmasıyla düzenlenir. Bu tip öğrenme yaklaşımında mutlaka bir danışmana veya YSA'ya ne öğrenmesi gerektiğini aktaracak bir yaklaşıma ihtiyaç vardır [47]. Şekil 5.4' de danışmanlı öğrenme modeli gösterilmiştir.



Şekil 5.4. Danışmanlı öğrenme

Geri Yayılım (BP), Levenberg-Marquardt (LM), Esnek Yayılım (RP), Delta-Bar-Delta (DBD) ve Hızlı Yayılım (QP) gibi algoritmalar danışmanlı öğrenme algoritmalarına örnek olarak verilebilir [6].

Danışmansız Öğrenme: Bu öğrenme algoritmalarında, istenilen çıkış değerinin bilinmesine gerek yoktur. Öğrenme süresince sadece giriş bilgileri ağa uygulanır. Uygulanan girişe ve bu giriş verileri arasındaki matematiksel ilişkilere göre bağlantı ağırlıkları ayarlanır. Aynı özellikleri gösteren desenlerde aynı çıkışlar, farklı çıkışlarda ise yeni çıkışlar oluşturulur [47]. Şekil 5.5' de danışmansız öğrenme modeli gösterilmiştir.



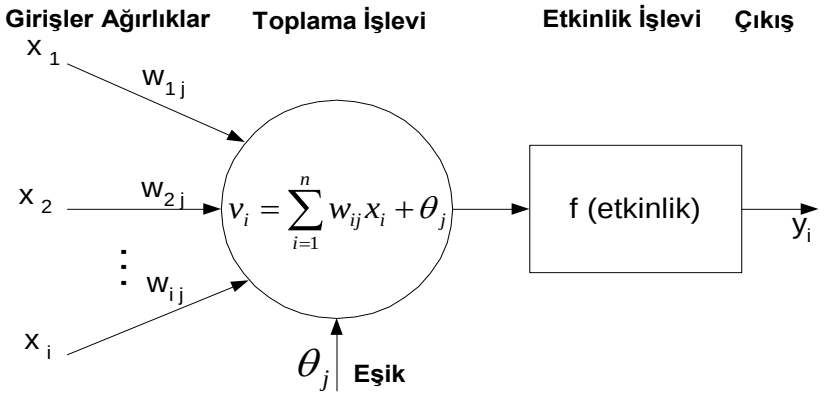
Şekil 5.5. Danışmansız öğrenme.

Grossberg tarafından geliştirilen ART (Adaptive Rezonance Theory) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı danışmansız öğrenmeye örnek olarak verilebilir [47].

5.5.7. Yapay bir sinir

Yapay sinir ağlarının temel birimi, düğüm olarak adlandırılan yapay bir sinirdir (Şekil 5.6.) [45].

Girişler: Girişler (x_1, $x_{2,\dots}$,x_n), çevreden aldığı bilgiyi sinire getirir. Girişler kendinden önceki sinirlerden veya dış dünyadan sinir ağına gelebilir.



Şekil 5.6. Yapay bir sinir.

Ağırlıklar: Ağırlıklar(w_1, $w_{2,\dots}$,w_n), yapay sinir tarafından alınan girişlerin sinir üzerindeki etkisini belirleyen uygun katsayılardır. Bir ağırlığın büyük olması, o girişin yapay sinire güçlü bağlanması ya da önemli olması; değerinin küçük olması ise zayıf bağlanması ya da önemsiz olması anlamına gelmektedir.

Toplama İşlevi: Toplama işlevi v_i , sinirde her bir ağırlığın ait olduğu girişlerle çarpımının toplamlarını eşik değeri Q_j ile toplayarak etkinlik işlevine gönderir. Bazı durumlarda toplama işlevi bu kadar basit bir işlem yerine, en az (min), en çok (max), çoğunluk veya birkaç normalleştirme algoritması gibi çok daha karmaşık olabilir.

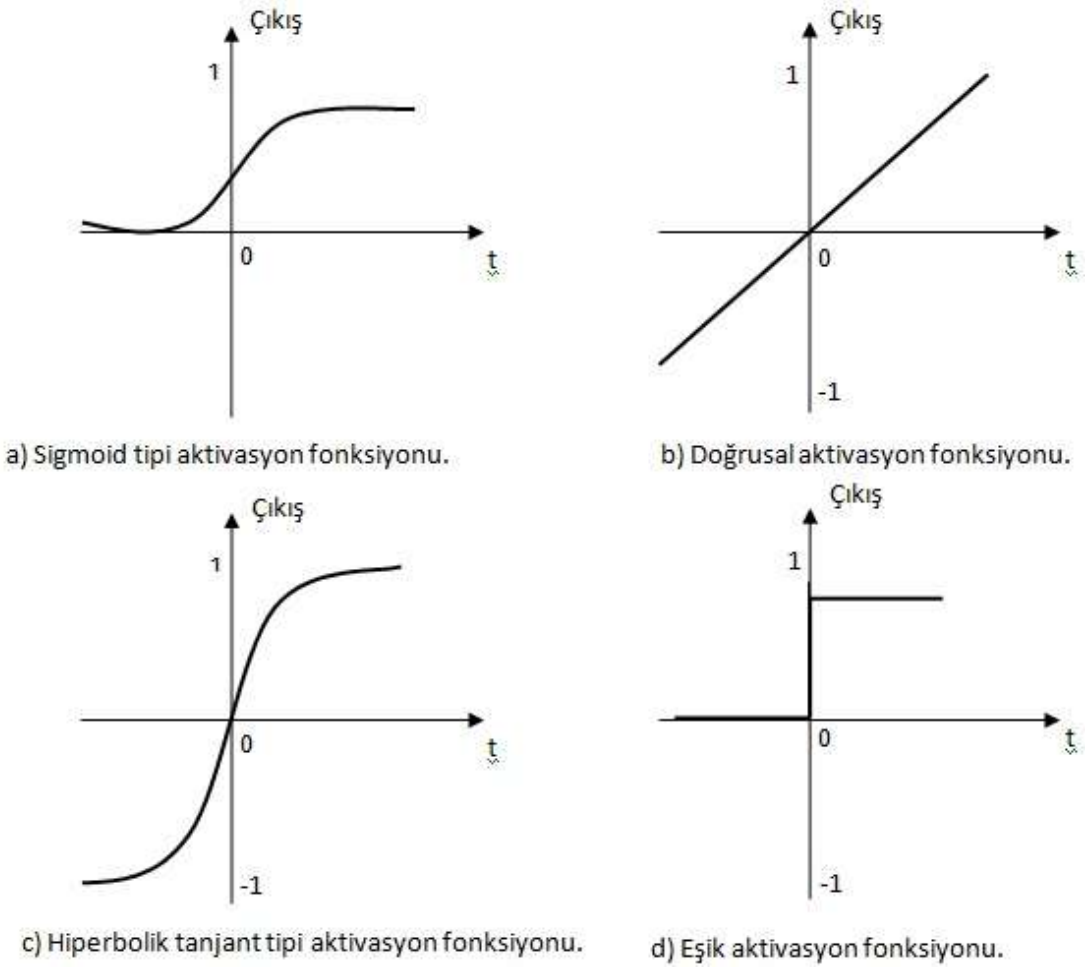
Etkinlik İşlevi: Etkinlik işlevinin kullanım amacı zaman söz konusu olduğunda toplama işlevinin çıkışının değişmesine izin vermektir. Örneğin;

$$y_i = \quad 1, \text{eğer } w_1 . x_{1+} w_2 . x_{2+\ldots\ldots\ldots+} w_N . x_N \geq T$$

$$0, \text{eğer } w_1 . x_{1+} w_2 . x_{2+\ldots\ldots\ldots+} w_N . x_N < T$$

Eşiklikteki *f(etkinlik)* aktivasyon fonksiyonudur. Şekil 5.7.a'da sigmoid transfer fonksiyonu görülmektedir. Lojistik fonksiyon olarak da adlandırılan bu fonksiyonun lineer olmasından dolayı türevi alınabilmektedir. Böylece geri yayılımlı ağlarda kullanmak mümkün olabilmektedir. Doğrusal aktivasyon fonksiyonunun çıkışı girişine eşittir. Sürekli çıkışlar gerektiği zaman çıkış katmanındaki aktivasyon fonksiyonunun lineer aktivasyon fonksiyonu olabildiğine dikkat edilmelidir (Şekil 5.7. b).

Hiperbolik tanjant aktivasyon fonksiyonu da lineer olmayan türevi alınabilir bir fonksiyondur. +1 ile -1 arasında çıkış değerleri üreten bu fonksiyon lojistik fonksiyona benzemektedir. Şekil 5.7 d'de eşik aktivasyon fonksiyonunun grafiği görülmektedir. Eşik aktivasyon fonksiyonu eğer net değeri sıfırdan küçükse sıfır, sıfırdan daha büyük bir değer ise net çıkışında +1 değeri verir. Şekil 5.7'de verilen fonksiyonlar en genel aktivasyon fonksiyonlarıdır. Yapay sinir ağında hangi aktivasyon fonksiyonunun kullanılacağı probleme bağlı olarak değişmektedir.



Şekil 5.7. Aktivasyon fonksiyonları.

Çıkış İşlevi: Çıkış yi = f (s), etkinlik işlevi sonucunun dış dünyaya veya diğer sinirlere gönderildiği yerdir. Bir sinirin bir tek çıkışı vardır. Sinirin bu çıkışı, kendinden sonra gelen herhangi bir sayıdaki diğer sinirlere giriş olabilir.

5.5.8. YSA mimarileri

İleri Beslemeli Ağlar

Reel değerli *n* boyutlu girdi özel vektörleri şu şekilde ifade edilir; *j* gizli katman siniri, i girdisini $w_{ij}(i=1,2,...,n,\ j=1,2,...)$ ağırlığına göre alır. j birimi x girdi işaretinin ve w_{ij} ağırlıklarının bir işlevini hesaplayıp, sonucu sonraki tüm komşu sinirlere iletir. İlk gizli katman gibi ikinci gizli katman sinirleri de ağırlıklarla önceki

katmana tam bağlıdır. Bu sinirler de girişlerin ve girişlerin ağırlıklarının bir işlevini hesaplayıp sonucu sonraki aşamaya aktarır. Bu işlem, çıkış katmanındaki sinirler tarafından da yapıldıktan sonra tamamlanır. Bu ağlar çok katmanlı ileri beslemeli ağlar olarak isimlendirilir [45].

Rosentblatt'ın Perceptronu ortaya atmasından sonraki yıllarda, çok katmanlı ileri beslemeli ağların birçok çeşidi önerilmiş ve bunlar üzerinde çalışılmıştır. Bu ağlar, geri yayılım algoritması ile uygulanabilir hale gelmiştir. Çok katmanlı ileri beslemeli ağlar, rastgele $g : R^m \to R^m (g(x) = z)$ yapılanmalarını gerçekleştirdikleri için oldukça yaygın bir kullanım alanı bulmuşlardır [45]. İleri beslemeli ağlara MLP, RBFN ve LVQ örnek verilebilir [47].

Geri Beslemeli Ağlar

Geri beslemeli ağ mimarileri, genellikle danışmansız öğrenme kurallarının uygulandığı ağlarda kullanılmaktadır. Geri beslemeli ağlarda isminden de anlaşılacağı gibi bir tür geri besleme işlemi vardır. Hopfield ağı, bu tür mimariye sahip bir yapay sinir ağıdır. Bu tür ağlarda bir sinirin çıkışı diğer her bir sinirin girişine bağlıdır. i-nci sinirin dış girdisi x_i, çıkışı y_i, j sinirinin çıkışı ile i-nci siniri arasındaki bağlantının ağırlığı ise w_{ij} ile gösterilmektedir. Ağın çalışması, aşağıdaki gösterilen eşitlik 5.1 ve eşitlik 5.2 ile tanımlanır [45].

$$\frac{du_i}{dt} = -u_i + \sum_{i=1}^{n} w_{ij} y_i + x_i \tag{5.1}$$

Burada $y_j = g(u_i)$ ve

$$g(x) = \frac{1}{2}\left(\tanh\left(\frac{x}{\alpha_0} \right) + 1 \right) \tag{5.2}$$

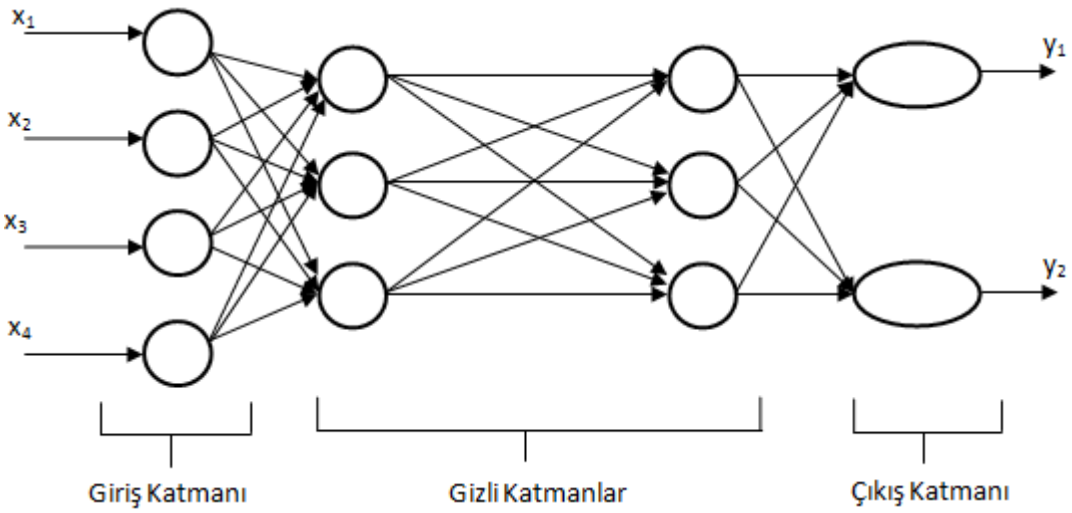
ile verilir. $\alpha_0 = 0$ olması durumunda katı sınırlayıcı transfer işleve karşılık gelir. α_0 değeri yeterince küçük ise, ağırlıkların simetrik yani tüm i ve j'ler için $w_{ij} = w_{ji}$ olması durumunda, Hopfield ağı

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} y_i y_j - \sum_{i=1}^{n} y_i x_i \tag{5.3}$$

şeklinde bir sistem enerji işlevini en küçükleyecek şekilde davranır ve bu enerji işlevinin bir yerel minimuma karşılık gelen kararlı duruma ulaşır (Eş. 5.3) [45].

5.5.9. Yapay sinir ağı katmanları

Yapay sinir ağları yapay sinir hücrelerinin birbirine bağlanmasıyla oluşan yapılardır. Yapay sinir ağları üç ana bölümde incelenir; giriş, ara ve çıkış katmanları (Şekil 5.8).



Şekil 5.8. Yapay sinir ağı katmanları.

Giriş Katmanı: *Yapay* sinir ağına dış dünyadan girdilerin geldiği katmandır. Bu katmanda dış dünyadan gelecek giriş sayısı kadar nöron bulunmasına rağmen genelde girdiler herhangi bir işleme uğramadan alt katmanlara iletilmektedir.

Ara Katmanı: Giriş katmanından çıkan bilgiler bu katmana gelir. Ara katman sayısı ağdan ağa değişebilir. Bazı yapay sinir ağlarında ara katman bulunmadığı gibi bazı

yapay sinir ağlarında ise birden fazla ara katman bulunmaktadır. Ara katmanlardaki nöron sayıları giriş ve çıkış sayısından bağımsızdır. Birden fazla ara katman olan ağlarda ara katmanların kendi aralarındaki nöron sayıları da farklı olabilir. Ara katmanların ve bu katmanlardaki nöronların sayısının artması hesaplama karmaşıklığını ve süresini arttırmasına rağmen yapay sinir ağının daha karmaşık problemlerin çözümünde de kullanılabilmesini sağlar.

Çıkış Katmanı: Ara katmanlardan gelen bilgileri işleyerek ağın girdi katmanından gelen verilere karşılık olan çıktıları üreten katmandır. Bu katmanda üretilen çıktılar dış dünyaya gönderilir. Geri beslemeli ağlarda bu katmanda üretilen çıktı kullanılarak ağın yeni ağırlık değerleri hesaplanır [49].

5.5.10. YSA'nın STS'lerde kullanılması

Yapay sinir ağları (YSA), saldırı tespit sistemlerinde 1990'ların başında kullanılmaya başlanan zeki yaklaşım yöntemlerinden biridir. YSA' lar, anormallik tespiti yapan STS' ler için istatistiksel yöntemlere alternatif olarak önerilmiştir [6].

YSA' lar, giriş ve çıkış vektörleri arasında ilişki kurarak kendi algoritmalarını uygularlar ve genelleştirerek yeni giriş/çıkış ilişkilerini ortaya çıkarırlar. Bu yaklaşım, sistemdeki kullanıcıların davranışlarının öğrenilmesiyle gerçekleşir. Spesifik bir kullanıcı için önceki komutlardan yola çıkarak yeni komutu tahmin eder [6].

YSA' ların STS' lerde kullanımı, YSA' nın normal sistem davranış izleriyle eğitilmesi ile başlar. Normal veya anormal olarak sınıflandırılan olay akışları yapay sinir ağına verilir. Toplanan veriler ile sistemin davranışına bağlı olarak öğrenme değişimi yapılabilir. Yani eğitim, izin veriliyorsa sürekli hale getirilebilir. Buradaki yaklaşım, kullanıcının n adet hareket veya komutundan, sonraki hareket veya komutunun tahminen eğitilmesidir. Bu tahminin yapılması için öncelikle eğitim veri seti oluşturulmalıdır, daha sonra da eğitim tamamlanmalıdır[6].

YSA' lar, anormallik tespitinde kullanıldığı gibi kötüye kullanım tespitinde de kullanılan bir tekniktir. Ağ ataklarının sürekli değişen yapısı, ağ trafiğini geniş çapta

analiz edebilen ve kural tabanlı sistemlerden daha esnek bir savunma sistemi ihtiyacını doğurmuştur. YSA tabanlı kötüye kullanım tespiti yapan sistemlerle, kural tabanlı sistemlerde var olan bu tür problemlere çözüm sağlanabilmektedir [6].

YSA' lar, kötüye kullanım tespitinde iki farklı şekilde kullanılmaktadır. İlk yaklaşımda yapay sinir ağları, zaten var olan bir uzman sistemin bir parçası olarak kullanılmaktadır. Bu yaklaşımda, saldırı tespitinin daha etkin yapılması amacıyla gelen verilerin filtrelenmesi ve uzman sisteme gönderilmesinde YSA' lardan faydalanılır. İkinci yaklaşımda ise, YSA' lar tek başına bir sistem olarak kötüye kullanımı tespit etmekte kullanılır. Bu yaklaşımda YSA, ağ akış bilgilerinden, kötüye kullanım tespitinin analizinde kullanılır [6].

YSA' lar, STS' lerde karşılaşılan pek çok problemlere esnek çözümler sunabildikleri STS tasarımlarında önemli çözümlerin başında gelmektedir. YSA' nın çözüm sunduğu temel iki problem, veri redaksiyonu ve sınıflandırmadır. Veri redaksiyonu (azaltma), işleme zamanını, iletişim yükünü ve depolama gereksinimlerini azaltmak amacıyla veri koleksiyonunu analiz ederek en önemli girişleri tanımlama işidir. Sınıflandırma ise saldırganları ve atak yapanları tanımlama işlemidir. YSA' lar pek çok STS' lerde bu iki önemli problemi çözmek için kullanılmışlardır [6].

STS' lerde karşılaşılan diğer problemler, istatistiksel yayılımı doğrulama ihtiyacı, tespit ölçütlerinin değerlendirilmesinin zorluğu, algoritma geliştirmenin yüksek maliyeti ve ölçeklemede zorluk olarak sıralandırılabilir. Bu problemler aşağıda açıklanmıştır [6];

İstatistiksel yayılımı doğrulama ihtiyacı: İstatistiksel metotlar kullanıcı davranışlarının yayılımı hakkında varsayımlara dayanır. Bu varsayımlar doğru olmayabilir ve yanlış alarm oranının artmasına neden olabilir. YSA' lar veri yayılımında bu tür varsayımları ortadan kaldıran çözümler sunarlar.

Tespit ölçütlerinin değerlendirilmesinin zorluğu: İstatistiksel metotlarda belirlenen ölçütler deneyim ve gözlemler ile elde edilir. Ancak bu belirleme sırasında bir ölçütün ne kadar önemli olduğu kesin olarak tespit edilemez. Genelde etkili olmadığı

için kullanılmayan bir ölçüt, özel bir çözüm için önemli olabilir. Yapay sinir ağları, çeşitli ölçüt kümelerinin ne kadar etkili olduğunu değerlendirmede de çözümler sunabilir.

Algoritma geliştirmenin yüksek maliyeti: Yeni bir istatistiksel algoritma önermenin ve yeni bir yazılım geliştirmenin süresi önemlidir. Algoritmanın tekrar yapılandırılması ve yazılım uygulamasının tekrar oluşturulması maliyetli bir iştir. YSA simülatörleri ise modifiye işlemi için daha kolaydır, daha az çaba ve zamanda kullanılan yöntem değiştirilebilir.

Ölçeklemede zorluk: STS' lerin geniş topluluklarda kullanılması ile binlerce kullanıcıya göre çalışmak gibi yeni problemlerle karşı karşıya gelinmesine sebep olmuştur. Bu problemi çözmek için, güvenlik yöneticisi tarafından kullanıcıların iş tanımları veya sorumluluklara göre gruplandırılması gerekebilir. YSA kullanılarak, var olan davranışlarına göre sınıflandırılma yapılması, grupları izlemeyi daha etkili hale getirecektir.

YSA' nın STS' lerde kullanılmasının avantajları [6];

- Kötüye kullanım ataklarının karakteristiğini öğrenmesi ve daha önce ağda kaydedilen örneklere benzemeyenleri ayırt edebilmesi,
- Hızlı sonuç üretmesi sayesinde gerçek zamanlı uygulamalar için kullanışlı olması,
- Gürültülü verilerle de çalışabildiklerinden, gürültülü verilerin sonuçları diğer yöntemlere göre daha az bozması,
- Farklı sistemlerle beraber çalışabilmeleri,
- Genel olarak çözüm sağlayabilmeleri yanında kısmi olarak da STS'lerin tasarımında kullanılabilmeleri,
- Az örneklerde sistemin veya atağın genel davranışını öğrenebilme yetenekleri olarak sıralanabilir.

Tüm bu avantajların yanı sıra, en büyük problem yapay sinir ağlarının eğitilmesidir.

Yapay sinir ağı, etkin şekilde çalışması için iyi eğitilmelidir ve geniş bir veri seti kullanılmalıdır.

6. VERİ MADENCİLİĞİ İLE SALDIRI TESPİT UYGULAMA YAZILIMI

Ağ güvenliğinin devamlı olarak sağlanmasının yolu yetkisiz erişimleri gerçek zamanlı olarak tespit etmektir. Etkili bir saldırı tespit sistemi, atak ve şüpheli ağ erişimlerini etkin bir şekilde tespit ederek ağ güvenliğinin bir bacağını oluşturur. Ama bu istenmeyen trafiğin sadece saptanması yeterli değildir. Kullanılan saldırı tespit uygulamasından beklenen bir diğer özellik, belirlenecek bu tür istenmeyen bağlantılara anında yanıt verebilmeli ve ağ kaynaklarına yetkisiz erişimi engellemelidir.

Yapılan saldırı tespit sistemi uygulaması; gerçek zamanlı bir saldırı tespit sistemi olup brute force tipi saldırıları IP tabanlı tespit eder. Tespit edilen IP adreslerinin sisteme girişi engellenir ve e-posta ile sistem yöneticisine bildirilir. Ayrıca saldırı tespit sistemi uygulaması; sisteme yapılan her türlü erişimin kullanıcı adı, IP adresi, tarihi ve saati bilgilerini de yöneticiye sunmaktadır.

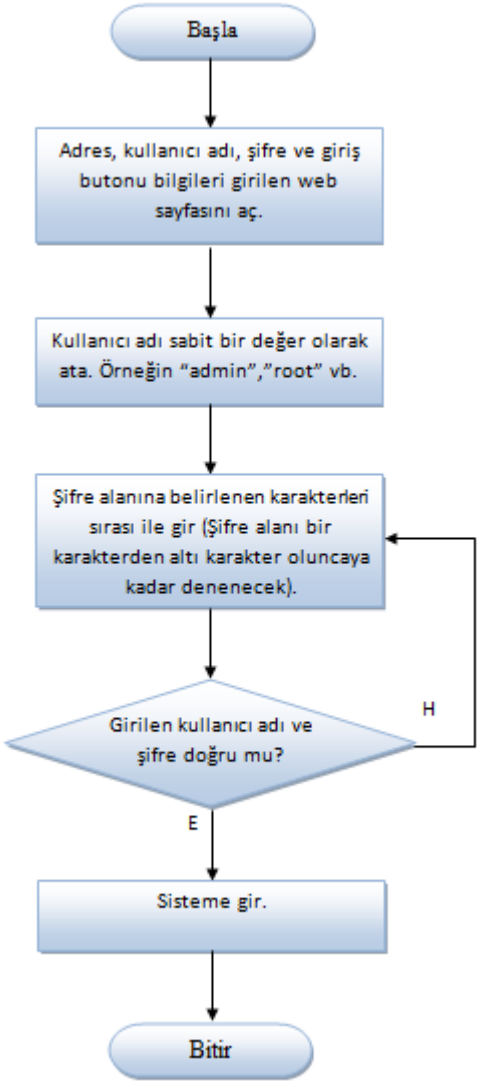
Saldırı tespiti için yapılan yazılımda kullanılan model sınıflamadır. Kullanılan yöntem ise YSA' dır. Bu yöntem aracılığı ile sisteme giriş yapmaya çalışan kullanıcıların saldırı yapıp yapmadıkları tespit edilmeye çalışılmıştır.

Uygulama için iki ayrı yazılım geliştirilmiştir. Bunlar; Brute Force saldırı yazılımı ve web tabanlı saldırı tespit yazılımlarıdır.

6.1. Brute Force Saldırı Yazılımı

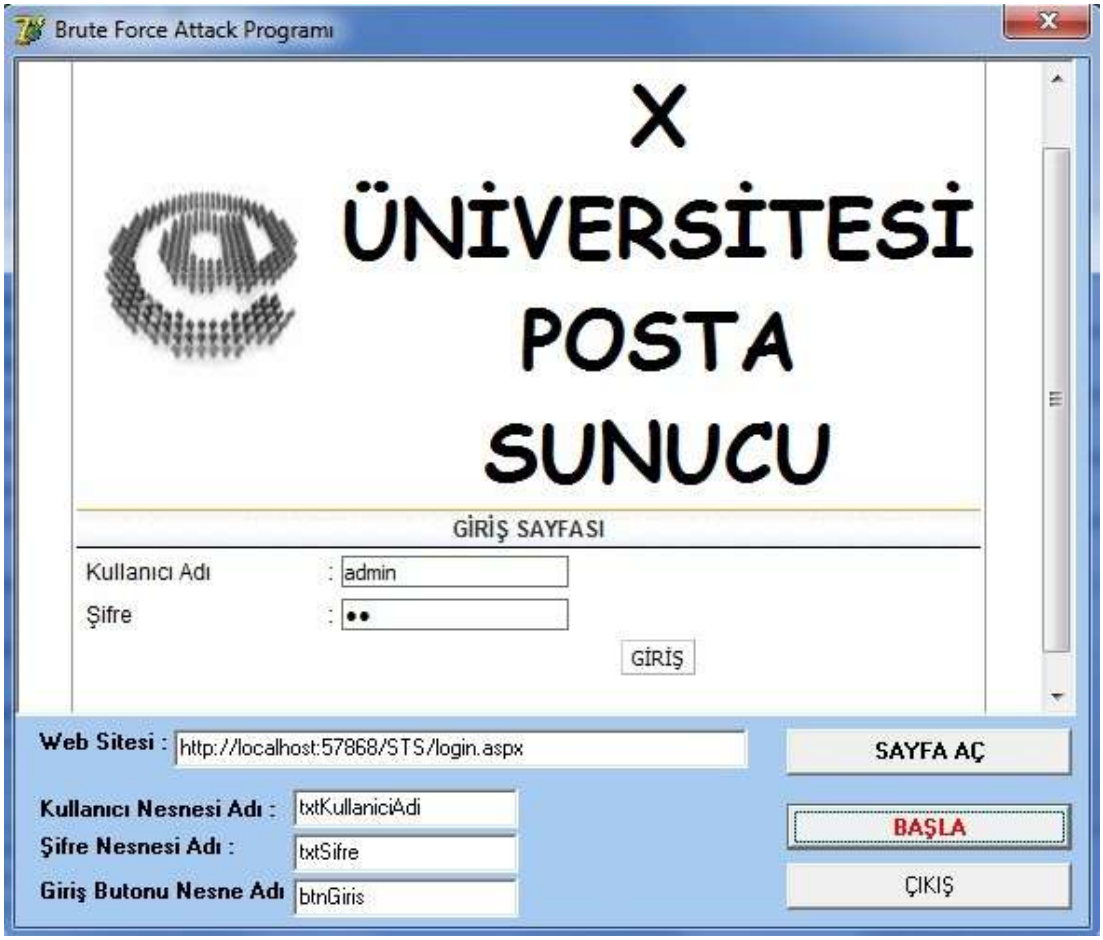
Yazılım, Borland Delphi 7.0 programlama dili kullanılarak yazılmıştır. Kullanıcı girişli web sitelerine Brute Force saldırı yapmaktadır.

Yazılımda kullanılan akış şeması Şekil 6.1' de gösterilmiştir.



Şekil 6.1. Brute Force atak yazılımı akış şeması

Yazılımın ekran görüntüsü Resim 6.1' de gösterilmiştir.



Resim 6.1. Brute Force saldırı yazılımı ekran görüntüsü

Yazılımda web sitesi adresi, kullanıcı adının yazılacağı nesne adı, şifrenin yazıldığı nesne adı ve sisteme giriş için tıklanacak butonun nesne adı parametre olarak girilir. Parametreler girildikten sonra "SAYFA AÇ" butonuna tıklanarak, adresi girilen web sayfası açılır. Yazılımda parametre olarak kullanılan nesne adlarını tespit edebilmek için saldırı yapılacak web sitesi herhangi bir web tarayıcıda açılır. Açılan web sitesi üzerinde sağ tıklanıp "Kaynağı görüntüle" seçilerek web sayfasının kaynak kodlarına erişilir. Buradan kullanıcı adı, şifre ve giriş butonunun nesne adları alınarak yazılıma parametre olarak girilir (Resim 6.2).

```
    <td colspan="2" style="width: 40%" align="left">
        Kullanıcı Adı</td>
    <td style="width: 1%" align="center">
        :</td>
    <td colspan="3" style="width: 57%" align="left">
        <input name="txtKullaniciAdi" type="text" maxlength="30" id="txtKullaniciAdi" class="frmInput" />
        <span id="RequiredFieldValidator1" style="color:Red;visibility:hidden;">*</span></td>
    <td style="width: 1%">
    </td>
</tr>
<tr>
    <td style="width: 1%">
    </td>
    <td colspan="2" style="width: 40%" align="left">
        Şifre</td>
    <td style="width: 1%">
        :</td>
    <td colspan="3" style="width: 57%" align="left">
        <input name="txtSifre" type="password" maxlength="15" id="txtSifre" class="frmInput" />
        <span id="RequiredFieldValidator2" style="color:Red;visibility:hidden;">*</span></td>
    <td style="width: 1%">
    </td>
</tr>
<tr>
    <td style="width: 1%">
    </td>
    <td colspan="2" style="width: 40%" align="left">
        </td>
    <td style="width: 1%">
        </td>
    <td colspan="3" style="width: 57%" align="center">
        <input type="submit" name="btnGiris" value="GİRİŞ" onclick="javascript:WebForm_DoPostBackWithOptions(new
ions(&quot;btnGiris&quot;, &quot;&quot;), true, &quot;&quot;, &quot;&quot;, false, false))" id="btnGiris"
```

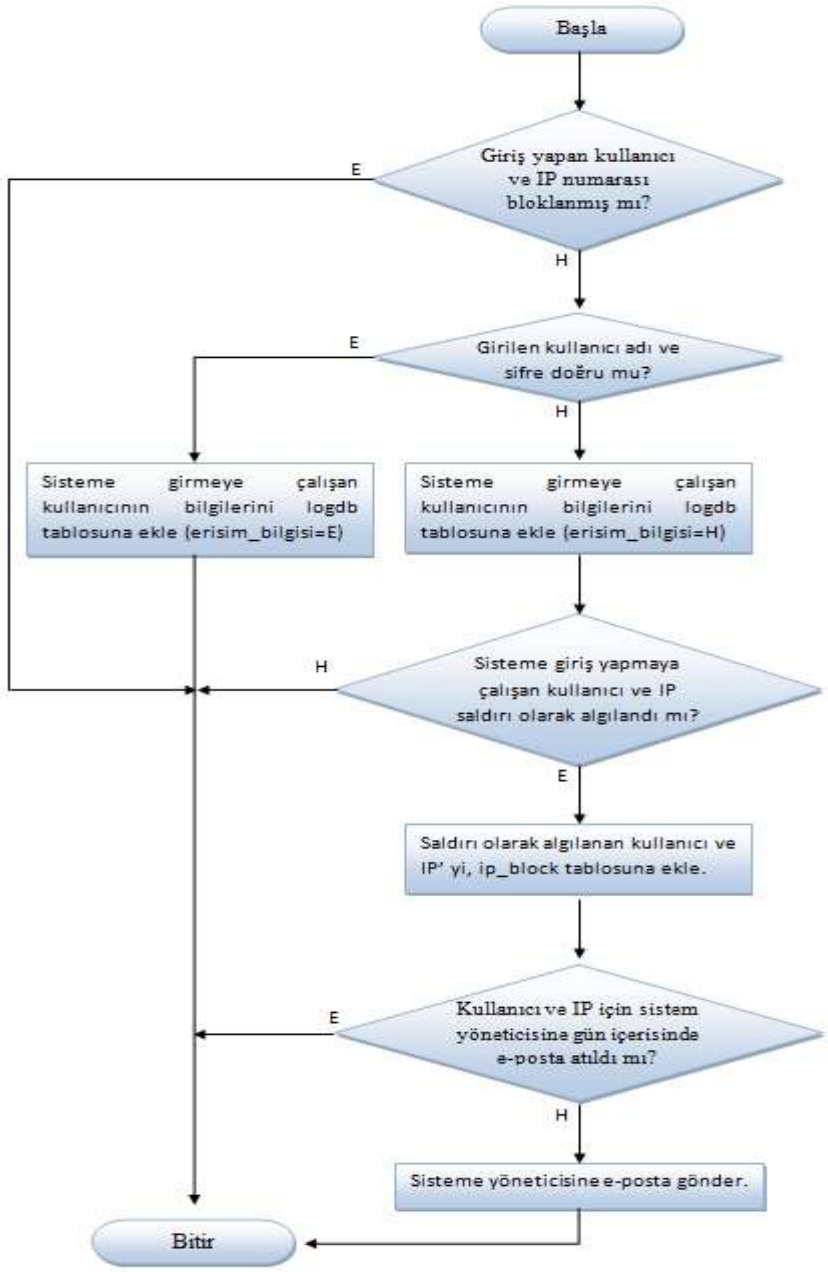
Resim 6.2. Web sayfası kaynak kodları

“BAŞLA” butonuna basıldığı anda yazılım içerisinde açılan web sitesine Brute Force saldırı başlar. Bu saldırıyı yaparken kullanıcı adı sabit bir değer olarak girilir (Örnek kullanıcı adları: admin, administrator, yonetici, root vb.). Yazılım tarafından şifre alanı bir haneden altı haneye kadar denenmeye başlanır. Bu şekilde doğru kullanıcı adı ve şifre yakalanarak sisteme giriş yapılması amaçlanmaktadır.

6.2. Saldırı Tespit Sistemi Yazılımı

Uygulama, Visual Studio 2005 yazılım geliştirme ortamında hazırlanmış web yazılımından ve Microsoft Office 2007 Access ortamında hazırlanmış veritabanı dosyasından oluşmaktadır.

Yazılım Şekil 6.2' de gösterilen akış şemasına göre tasarlanmıştır.

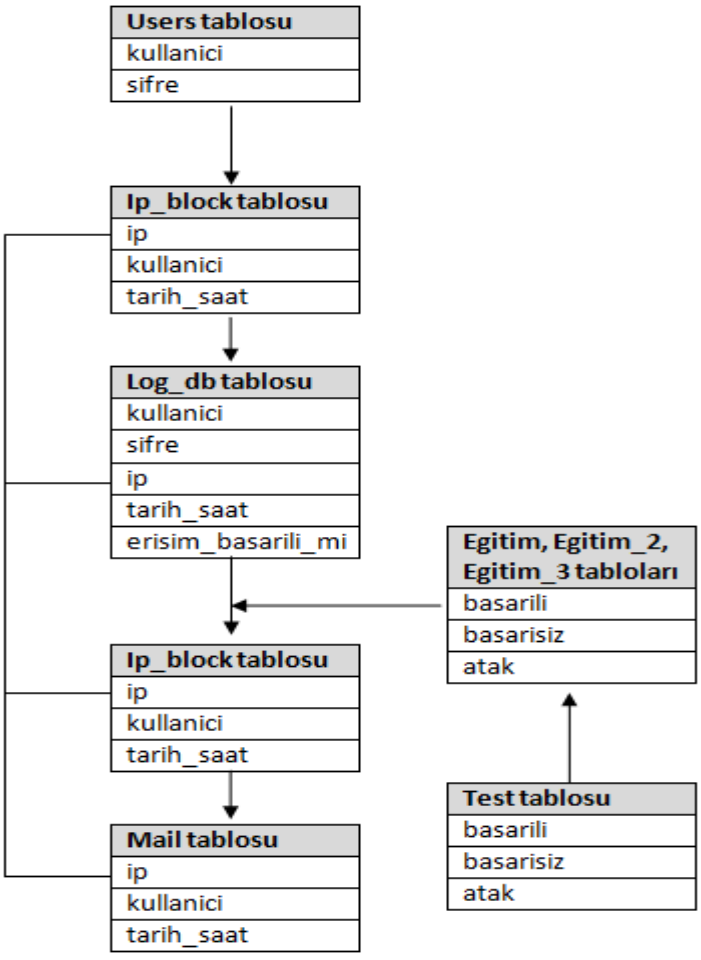


Şekil 6.2. Saldırı tespit yazılımı akış şeması

Veritabanında sekiz adet tablo vardır. Bunlar:

- users tablosu,
- logdb tablosu,
- ip_block,
- mail,
- egitim, egitim_2, egitim_3, test tablosudur.

Geliştirilen yazılımda kullanılan tablolar arasındaki ilişkiler, Şekil 6.2.'de gösterilen yazılım akış şemasındaki işlem sırasına göre Şekil 6.3.' de gösterilmiştir.



Şekil 6.3. Tablolar arası ilişkiler

Users tablosu, “kullanici” ve “sifre” olmak üzere iki alandan oluşmaktadır. Sisteme giriş için tanımlı olan kullanıcıların, kullanıcı adları ve şifreleri bu tabloda tutulur. Kullanıcılara ait şifreler, MD5 şifreleme algoritması ile şifrelenerek kaydedilmiştir. Örnek bir users tablosu Resim 6.3’ te gösterilmiştir.

users

kullanici	sifre
admin	77C9E808BFA89577F5D4C04E98303CF5
elmas	CE0BFD15059B68D67688884D7A3D3E8C
yonetici	61109B13DC74492B349605434EFAFA3F

Resim 6.3. Users tablosu

Logdb tablosu, “kullanici”, “ip”, “tarih_saat”, “erisim_basarili_mi” ve “sifre” alanlarından oluşmaktadır. Sisteme giriş yapmaya çalışan kullanıcıların hangi kullanıcı adı, şifre, ip ve tarih saatte sisteme giriş yaptıkları veya yapmaya çalıştıklarının kayıt bilgisini tutar. Bu tablodaki “erisim_basarili_mi” alanı değer olarak “E” ve “H” tutar. Sisteme başarılı bir şekilde giriş yapan kullanıcılar için “E”, sisteme giriş yapamayan kullanıcılar için “H” bilgisini tutar. Böylelikle sisteme giriş yapabilen ve yapamayan kullanıcılar ayırt edilir. Örnek bir logdb tablosu Resim 6.4’ te gösterilmiştir.

logdb

kullanici	ip	tarih_saat	erisim_basarili_mi	sifre
admin	78.180.12.74	09.12.2009 13:51:49	H	******
admin	78.180.12.74	09.12.2009 13:51:49	H	******
admin	78.180.12.74	09.12.2009 13:51:49	H	******
admin	78.180.12.74	09.12.2009 13:51:49	H	******
admin	78.180.12.74	09.12.2009 13:51:49	H	******
admin	78.180.12.74	09.12.2009 13:51:49	H	******
admin	78.180.12.74	09.12.2009 13:51:49	H	******
admin	78.180.12.74	09.12.2009 13:51:49	H	******
admin	78.180.12.74	09.12.2009 13:51:49	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******
admin	78.180.12.74	09.12.2009 13:51:50	H	******

Resim 6.4. Logdb tablosu

Ip_block tablosu, “ip”, “kullanici” ve “tarih_saat” olmak üzere üç alandan oluşmaktadır. Saldırı yapıldığı tespit edilen veya yetkili sistem kullanıcısının, sisteme girişe izin vermediği kullanıcı ve IP’ lerin tutulduğu tablodur. Örnek bir ip_block tablosu Resim 6.5’ te gösterilmiştir.

ip_block

ip	kullanici	tarih_saat
192.168.1.21	admin	29.12.2008 13:01:24
10.20.3.31	admin	13.11.2008 14:05:42

Resim 6.5. Ip_block tablosu

Mail tablosu, “ip”, “kullanici” ve “tarih_saat” olmak üzere üç alandan oluşmaktadır. Saldırı yapıldığı tespit edilen kullanıcı ve IP adresleri önceden belirlenmiş olan e-posta adresine otomatik olarak bildirilir. Hangi kullanıcı ve IP için hangi tarih ve saatte e-posta atıldığı bu tabloda tutulur. Örnek bir mail tablosu Resim 6.6’ da gösterilmiştir.

mail

ip	kullanici	tarih_saat
192.168.1.21	admin	29.12.2008 13:01:24
10.20.3.31	admin	13.11.2008 14:05:42

Resim 6.6. Mail tablosu

Eğitim (egitim, egitim_2, egitim_3) ve test tabloları “basarili”, “basarisiz” ve “atak” olmak üzere üç alandan oluşmaktadır. Geliştirilen yazılım için üç adet eğitim tablosu hazırlanmıştır. Bu sayede YSA’ nın farklı eğitim tabloları kullanılarak eğitilmesi ve eğitim sonuçlarının karşılaştırılması yapılmıştır. Eğitim tabloları yapı olarak aynıdır. Ancak içerdikleri kayıtlar ve kayıt sayıları farklıdır.

Eğitim tabloları kayıtları, sistemin log tablosuna (lodgb) bakılarak yetkili sistem kullanıcısı veya uzman kişiler tarafından oluşturulur. Sistemin log kayıtları belirli periyotlarda (1 dakikalık, 1 saatlik, 1 günlük, 1 haftalık, 1 aylık log bilgileri gibi.) analiz yapılarak, saldırı olarak kabul edilen ve saldırı olarak kabul edilmeyen

durumlar belirlenip gerekli olan eğitim tabloları hazırlanır. Bu işlem yapılırken sırası ile aşağıdaki adımlar izlenmiştir:

- logdb tablosundaki kayıtlar içerisinden her bir eğitim tablosu için birer haftalık üç kayıt bloğu seçilmiştir.
- Seçilen log kayıtları tarih saat bilgisinden yararlanılarak IP, kullanıcı adı, dakikadaki başarılı ve başarısız giriş sayılarına göre gruplanmıştır.
- Gruplama sonucunda çıkan sonuçlar içerisinde bir dakika içerisinde aynı başarılı ve başarısız giriş sayısına sahip kayıtlar tek kayıt gibi düşünülmüştür.
- Elde edilen başarılı ve başarısız giriş sayıları yetkili kullanıcı tarafından sistem durumuna göre atak ve atak değil olarak belirlenerek eğitim tabloları hazırlanmıştır.

Örnek bir eğitim Resim 6.7' de gösterilmiştir.

egitim

basarili	basarisiz	atak
2	21	H
2	22	H
2	23	H
2	24	H
2	25	H
2	26	H
2	27	H
2	28	H
2	29	H
2	30	E
0	31	E
0	32	E
0	33	E
0	34	E

Resim 6.7. Egitim tablosu

Hazırlanan eğitim tabloları kullanılarak eğitilmiş YSA' yı test edebilmek için, test tablosu hazırlanmıştır. Test tablosu içerisinde sistem tarafından saldırı olarak kabul

edilecek ve edilmeyecek durumları belirten kayıtlar bulunur. Bu tablo içerisindeki değerlerin doğruluğu çok önemlidir. Aksi durumda hatalı kayıtlarla test yapılacağından sistemin çalışması doğru bir şekilde analiz edilemez. Örnek bir test Resim 6.8' de gösterilmiştir.

test

basarili	basarisiz	atak
5	25	H
5	30	E
3	45	E
1	2	H
10	0	H
1	0	H
15	5	H
3	33	E
2	2	H
2	42	E
6	101	E

Resim 6.8. Test tablosu

Saldırı Tespit Sistemi yazılımında amaç; hazırlanan web yazılımına bağlanmaya çalışan kullanıcıların hangi IP' lerden bağlandıklarını tespit edip hazırlanan "egitim" tablolarını kullanarak eğitilen YSA' nın bu girişlerin bir saldırı olup olmadığına karar vermesini sağlamaktır.

Hazırlanan saldırı tespit yazılımı kullanıcı giriş ekranı Resim 6.9' da gösterilmiştir.

Resim 6.9. Giriş ekranı

Kullanıcıların sisteme girebilmeleri için kullanıcı adı ve şifrelerini girmeleri gerekmektedir. Kullanıcı adı ve şifre boş geçilemez. Boş geçilmek istendiğinde Resim 6.10' daki ekranla karşılaşılmaktadır.

Resim 6.10. Uyarı mesajı (kullanıcı adı ve şifre boş geçilemez)

Sisteme giriş yapmaya çalışan kullanıcının kullanıcı adı ve şifresi, veritabanında kayıtlı olan geçerli bir kullanıcıya ait değil ise Resim 6.11' deki ekranla karşılaşılmaktadır. Burada sisteme giriş yapmaya çalışan kullanıcının kayıt (log) bilgileri veritabanında "logdb" tablosuna kaydedilmektedir. Veritabanına kaydedilen bilgiler; kullanıcı adı, şifre, kullanıcı ip, sistem tarih saati ve sisteme erişim bilgisidir.

Resim 6.11. Uyarı mesajı (hatalı kullanıcı adı veya şifre)

Veri toplama işlemi bu aşamada gerçekleşmektedir. Bu işlem basamağı sırasında toplanan veriler daha sonra saldırı tespitinde kullanılmak üzere ilgili tabloya kayıt edilir.

Sisteme giriş yapmaya çalışan kullanıcı, giriş sayfasından değil de adres çubuğuna direk adresi yazarak ulaşmaya çalışırsa, sisteme giriş yapmadığı için hata mesajı veren bir sayfaya yönlendirilir. Resim 6.12' de bu ekran görünmektedir.

Resim 6.12. Hata ekranı (adresi yazarak direk ulaşmak istenirse)

Kullanıcı, giriş sayfasında kullanıcı adı ve şifreyi doğru girerek sisteme girebilmektedir. Sisteme giriş yapan kullanıcı Resim 6.13' deki gösterilen karşılama ekranı ile karşılaşacaktır.

Resim 6.13. Karşılama ekranı

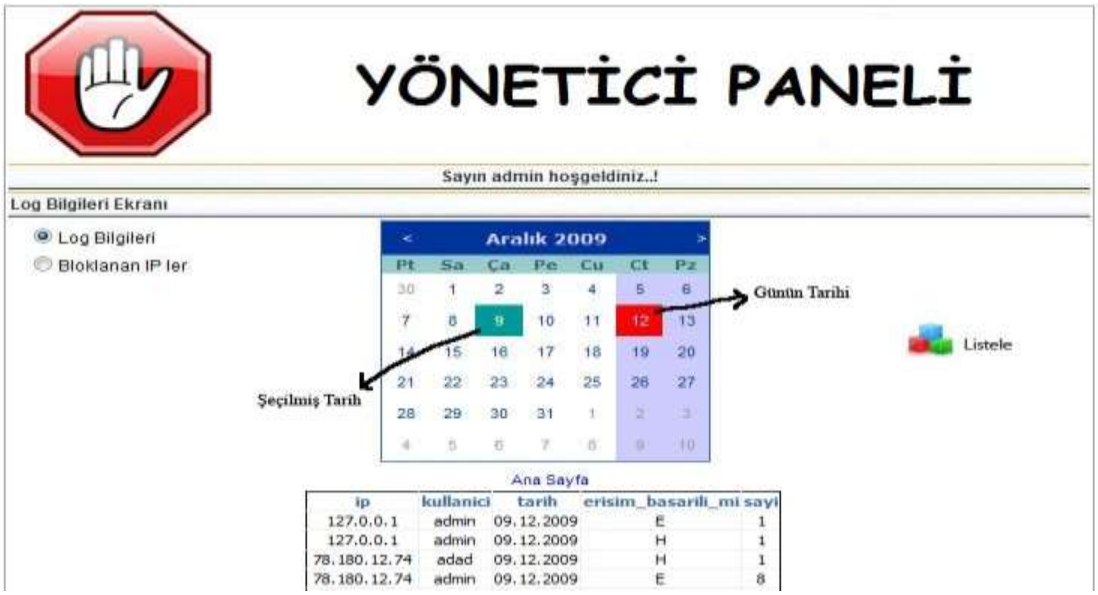
Karşılama ekranı üzerinden "Yönetim Paneli" butonunu tıklayan kullanıcının karşısına Resim 6.14' de gösterilen menü ekranı gelecektir.

Resim 6.14. Menü ekranı

Menü ekranı üzerinden "Log Bilgileri Ekranı" nı tıklayan kullanıcının karşısına Resim 6.15' de gösterilen log bilgileri ekranı gelecektir.

Resim 6.15. Log bilgileri ekranı

Bu ekranda; sisteme giriş yapmaya çalışan kullanıcıların log bilgileri ve sisteme girişine izin verilmeyen kullanıcı ve IP' ler liste olarak alınabilmektedir. Ekran üzerindeki tarih nesnesinde kırmızı ile işaretli olan tarih günün tarihi, koyu yeşil ile işaretli olan tarih log bilgisi alınmak istenilen günün tarihidir. "Log Bilgileri" seçeneği ve logların alınacağı tarih seçilip (resimdeki koyu yeşil ile gösterilen tarih), "Listele" butonuna basıldığında, ekranda seçilen tarihte sisteme giriş yapmaya çalışan kullanıcıların log bilgileri listelenir (Resim 6.16). Bu listedeki log bilgileri veritabanındaki "logdb" tablosunda tutulur. Bu tabloda tutulan kayıtlarda tarih saat bilgisi mevcut iken; ekranda sadece tarih bilgisi gösterilmiştir.



Resim 6.16. Seçilen tarihe ait log bilgileri

Sistem tarafından saldırı yaptığı kabul edilerek bloklanan kullanıcı ve IP' leri listelemek için "Bloklanan IP ler" seçeneği seçilip "Listele" butonuna tıklanır (Resim 6.17). Bu ekranda listeleme yaparken tarih seçmeye gerek yoktur. Bloklanan tüm IP' ler gösterilmektedir. "ip_block" tablosunda bloklanan IP' lere ait tarih saat bilgisi mevcut iken; ekranda sadece tarih bilgisi gösterilmiştir.

Resim 6.17. Bloklanan kullanıcı ve IP' ler

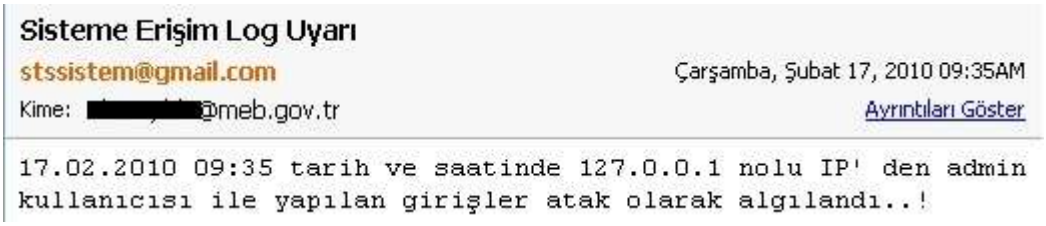
Bloklanmış bir kullanıcı ve IP' yi blok listesinden çıkarmak için IP' nin solundaki "Blok Kaldır" butonuna tıklanarak, seçilen kullanıcı ve IP, blok listesinden çıkarılır (Resim 6.18).

Resim 6.18. Kullanıcı ve IP adresini blok listesinden çıkarmak

Blok listesine alınan kullanıcı ve IP' den bir daha sisteme giriş yapılmasına müsaade edilmez. Bloklanan kullanıcı ve IP' den sisteme girilmeye çalışıldığında Resim 6.19' daki ekranla karşılaşılır.

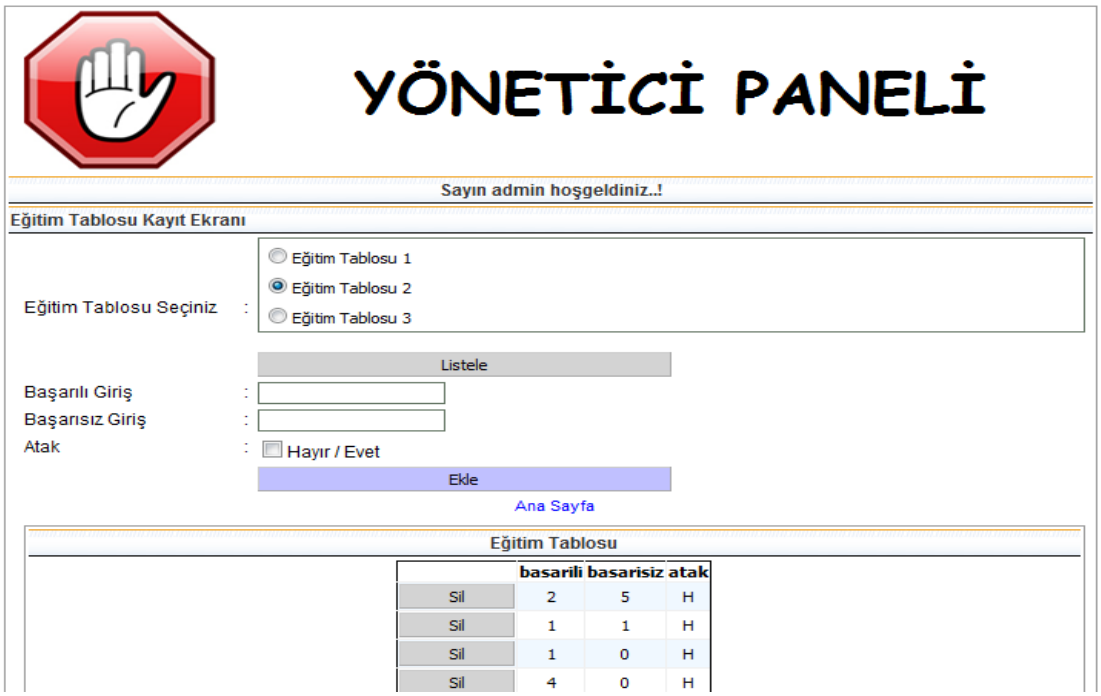
Resim 6.19. Uyarı mesajı (bloklanan kullanıcı ve IP)

Yazılımda kurgulanan modele göre sistem; saldırı olarak kabul edilen bir durumla karşılaştığında, önceden belirlenmiş e-posta adresine otomatik uyarı e-postası gönderir (Resim 6.20). Böylece yetkili kişinin durumdan haberdar olması sağlanmaktadır.



Resim 6.20. Yönetici bilgilendirme e-posta

Yazılımın menü ekranından "Eğitim Tablosu Ekranı" seçildiğinde Resim 6.21' deki ekran gelmektedir.



Resim 6.21. Eğitim tablosu ekranı

Bu ekran üzerinde YSA ağını eğitmek için kullandığımız eğitim tabloları ile ilgili işlemler yapılmaktadır. Eğitim tablolarına yeni kayıtlar ekleme veya eğitim tablolarında var olan kayıtları silme işlemi bu ekran üzerinde yapılmaktadır.

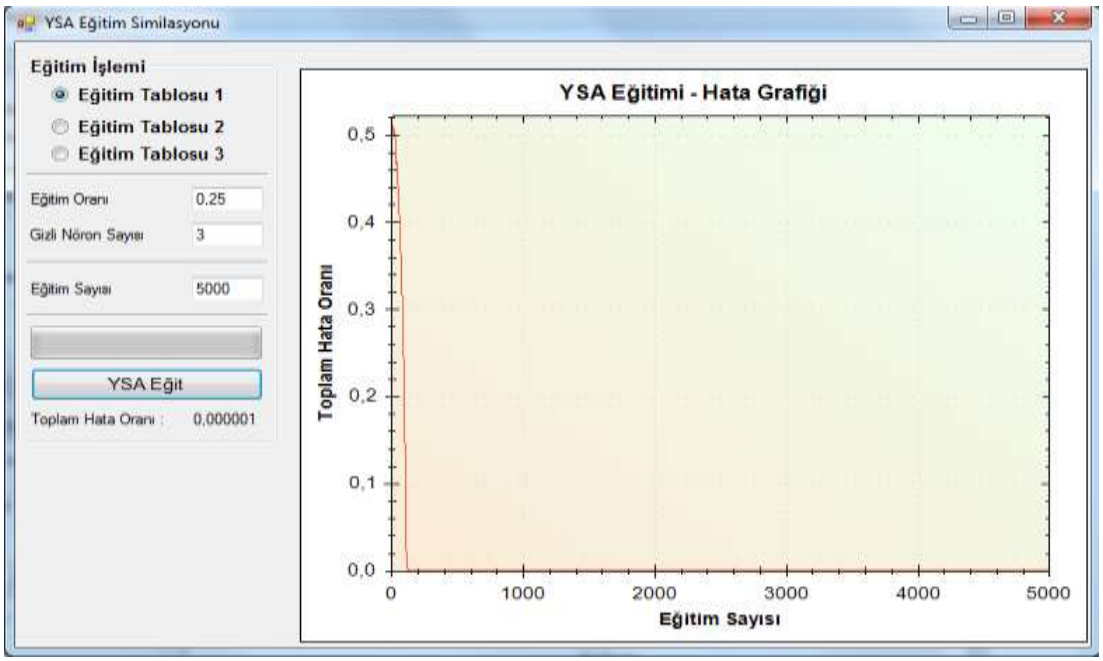
Resim 6.14' de verilen menü ekranından "YSA Eğitim Ekranı" seçildiğinde Resim 6.22' deki ekran ile karşılaşılmaktadır.

Resim 6.22. YSA eğitim ve test ekranı

Ekran üzerinde önceden hazırlanmış eğitim tabloları kullanılarak YSA eğitimi yapılmaktadır. Ekranda YSA eğitiminde kullanılacak tablo seçildikten sonra "YSA Eğit" butonuna basılarak işlem gerçekleştirilmektedir. Bu işlemi yapabilmek için Visual Studio 2005 programı içerisinde YSA için hazırlanmış olan "NeuronDotNet.Core" sınıfı kullanılmıştır. Geliştirilen yazılımda NeuronDotNet.Core sınıfına ait şu özellikler kullanılmıştır:

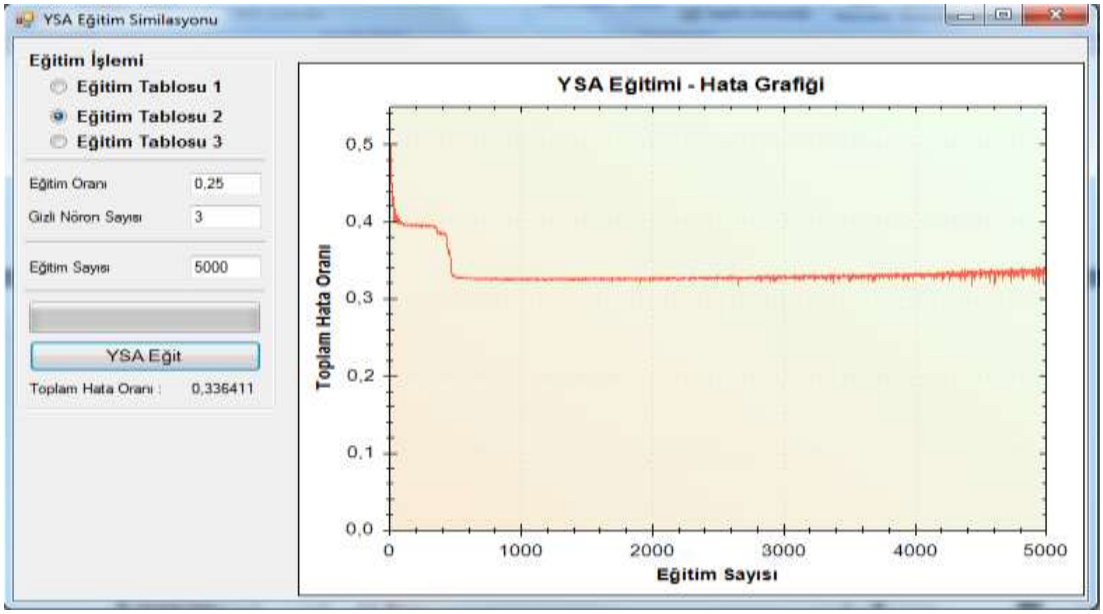
- TanhLayer : Giriş katmanını tanımlama ve tipini belirleme,
- SigmoidLayer : Gizli katman ve çıkış katmanını tanımlama ve tipini belirleme,
- BackpropagationConnector : Katmanlar arası bağlantıyı oluşturma ve tipini belirleme ,
- BackpropagationNetwork : YSA oluşturma ve tipini belirleme,
- LearningRateFunctions.HyperbolicFunction : YSA eğitim algoritmasını belirleme,
- TrainingSet : Eğitim setlerini tanımlama.

"Eğitim Tablosu 1" seçilerek eğitilmiş YSA' nın eğitim simülasyon grafiği ekran görüntüsü Resim 6.23' te gösterilmiştir.



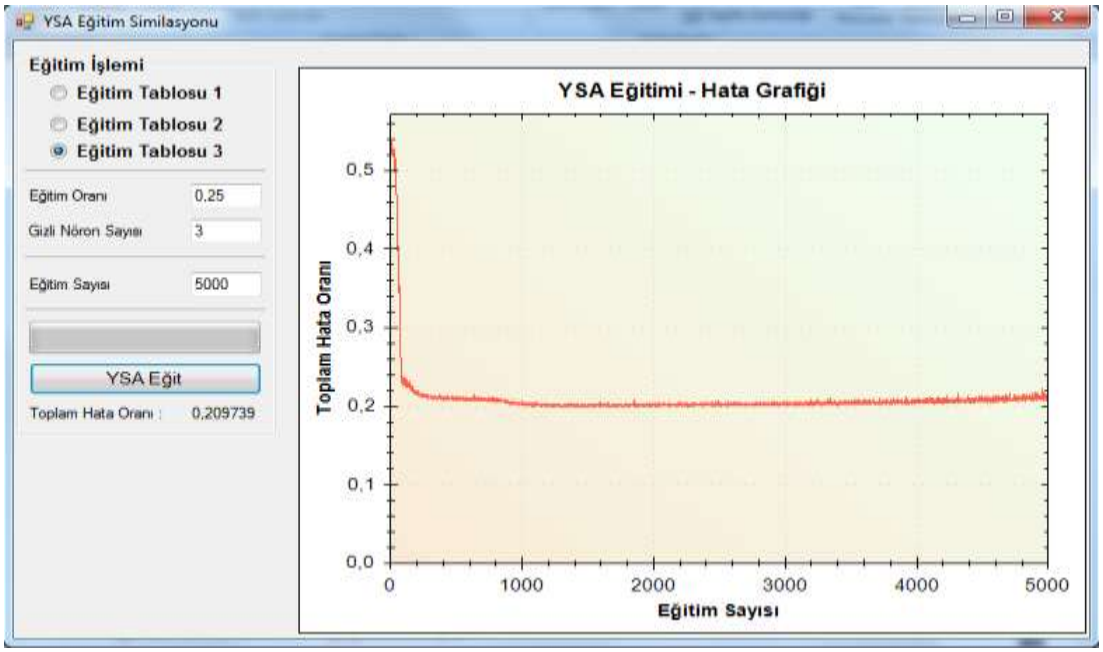
Resim 6.23. Eğitim tablosu - 1 YSA eğitim simülasyon görüntüsü

"Eğitim Tablosu 2" seçilerek eğitilmiş YSA' nın eğitim simülasyon grafiği ekran görüntüsü Resim 6.24' te gösterilmiştir.



Resim 6.24. Eğitim tablosu - 2 YSA eğitim simülasyon görüntüsü

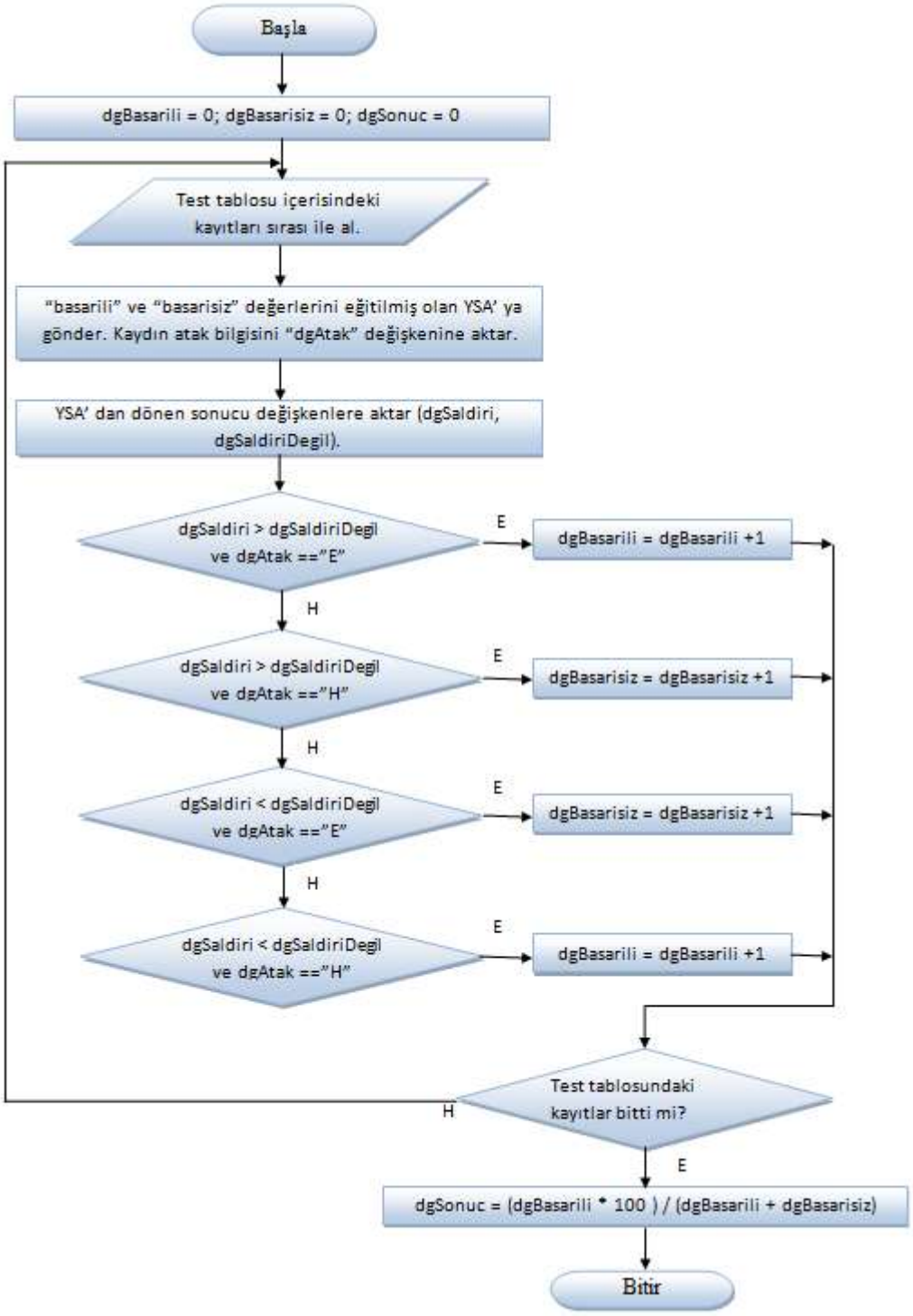
"Eğitim Tablosu 3" seçilerek eğitilmiş YSA' nın eğitim simülasyon grafiği ekran görüntüsü Resim 6.25' te gösterilmiştir.



Resim 6.25. Eğitim tablosu - 3 YSA eğitim simülasyon görüntüsü

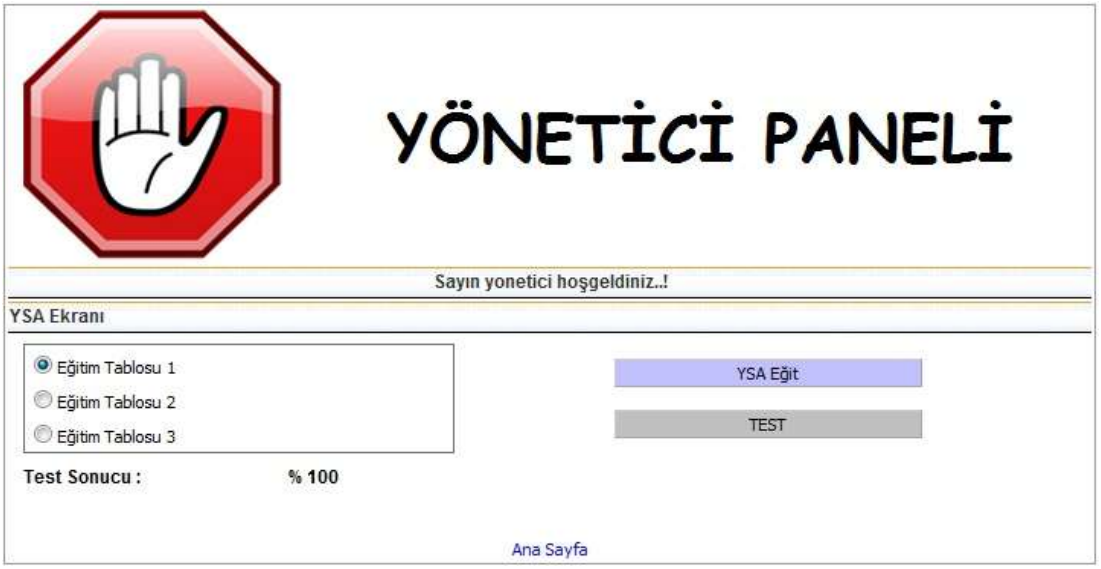
Üç adet eğitim tablosu ayrı ayrı kullanılarak YSA eğitilmiştir. Eğitim sonuçlarına göre en az hata oranı, "Eğitim Tablosu 1" seçilerek yapılmış eğitim sonucunda ortaya çıkmıştır. Eğitim tablosu 1' e göre yapılan eğitim sonucunda hata oranı sıfıra çok yakın bir değerdir (Resim 6.22). Hata oranları karşılaştırıldığında "egitim" tablosu içerisinde bulunan kayıtların diğer eğitim tabloları içerisinde bulunan kayıtlara göre daha tutarlı olduğu gözlemlenmiştir.

Eğitilmiş YSA, "TEST" butonuna tıklanarak test tablosundaki veriye göre test edilebilmektedir. Test işlemi akış şeması Şekil 6.4' te gösterilmiştir.



Şekil 6.4. YSA test işlemi akış şeması

Resim 6.26’ da “Eğitim Tablosu 1” kullanılarak eğitilmiş YSA’ nın test sonucu gösterilmektedir. Test sonucu oranı %100 çıkmıştır. Resim 6.23’ teki simülasyon görüntüsündeki hata oranı da bu sonucu desteklemektedir.

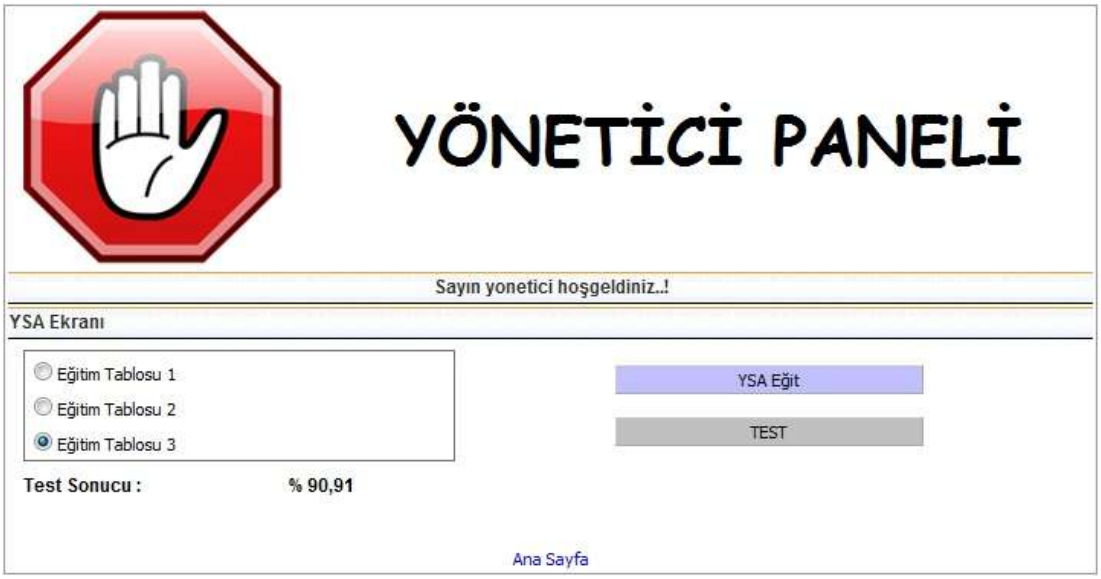


Resim 6.26. Eğitim tablosu – 1 kullanılarak eğitilen YSA’ nın test sonucu

Resim 6.27’ de “Eğitim Tablosu 2” kullanılarak eğitilmiş YSA’ nın test sonucu gösterilmektedir. Test sonucu oranı %81.82 çıkmıştır.

Resim 6.27. Eğitim tablosu – 2 kullanılarak eğitilen YSA’ nın test sonucu

Resim 6.28’ de “Eğitim Tablosu 3” kullanılarak eğitilmiş YSA’ nın test sonucu gösterilmektedir. Test sonucu oranı %90.91 çıkmıştır.



Resim 6.28. Eğitim tablosu – 3 kullanılarak eğitilen YSA’ nın test sonucu

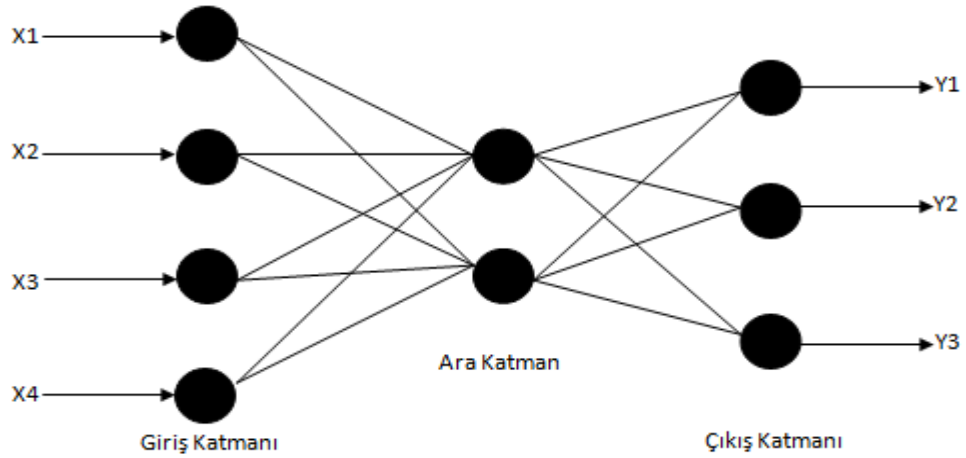
Test sonuçlarına göre en iyi sonuç eğitim tablosu 1’ e göre eğitilmiş YSA’ dan elde edilmiştir (%100). Bu sonuca göre eğitim tablosu 1 içersindeki kayıtların diğer eğitim tablolarında bulunan kayıtlara göre daha tutarlı olduğu ve başarılı sonuç verdiği anlaşılmaktadır

Bu tez çalışmasında, ileri beslemeli ağ yapılarından olan MLP kullanılmıştır. MLP tercih edilmesinin nedeni, bilinen en eski YSA modellerinden olması ve sınıflandırma problemlerinde başarılı sonuçlar üretmesidir. Bunların yanında, farklı öğrenme algoritmaları ile kullanıma uygun olması MLP’nin sağladığı diğer bir avantajdır. Şekil 6.5’ de örnek olarak gösterilen MLP modeli, bir giriş, bir veya daha fazla ara katman ve bir de çıkış katmanı içerir. Bir katmandaki bütün nöronlar bir sonraki katmandaki bütün nöronlara bağlıdır. Giriş katındaki nöronlar tampon gibi davranırlar ve giriş sinyalini ara katmandaki nöronlara dağıtırlar. Ara katmandaki her bir nöronun çıkışı, kendine gelen bütün giriş sinyallerini takip eden bağlantı ağırlıkları ile çarpımlarının toplanması ile elde edilir. Elde edilen bu toplam, çıkışın

toplam bir fonksiyonu olarak hesaplanabilir. MLP'de giriş katmanı hariç bir nöronun çıkışı aşağıda gösterilen eşitlik ile hesaplanır (Eş. 6.1).

$$y_k = f(\sum_k w_k x) \tag{6.1}$$

Burada, *w* nöronlar arasındaki ağırlık değerini, *f* ise kullanılan transfer fonksiyonunu gösterir [6].

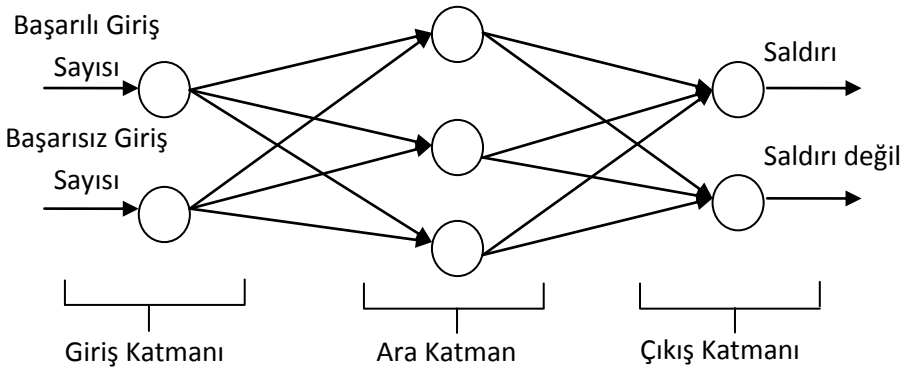


Şekil 6.5. Örnek bir MLP yapısı (4 giriş, 1 ara katman, 3 çıkış)

MLP'de giriş ve çıkış sayıları, uygulanacak olan probleme göre belirlenirken, ara katman sayısı ile ara katman nöron sayıları "deneme yanılma" yolu ile bulunur [47].

Bu tez çalışmasında YSA modelini eğitmek için LM öğrenme algoritması kullanılmıştır. Bu algoritma, Gauss-Newton ve En Dik İniş (Steepest Descent) algoritmalarının en iyi özelliklerinden oluşur ve bu iki metodun kısıtlamalarını ortadan kaldırır. Genel olarak bu metot yavaş yakınsama probleminden etkilenmez [47].

YSA' nı eğitirken girilen parametre değerleri; giriş nöron sayısı 2 (başarılı ve başarısız giriş sayısı), gizli katman sayısı 1, gizli katmandaki nöron sayısı 3 ve çıkış nöron sayısı 2 (saldırı ve saldırı değil oranları)' dir. YSA eğitimi 5000 adımda yapılacak şekilde belirlenmiştir. Geliştirilen yazılımda kullanılan YSA' nın topolojik yapısı Şekil 6.6' da gösterilmiştir.



Şekil 6.6. Yazılımda kullanılan YSA topolojik yapısı

6.3. Yazılımın Veri Madenciliği Süreçleri Açısından Değerlendirilmesi

1. *Problemin tanımlanması :* Geliştirilen STS çalışmasında tanımlanan problem; sistemlerde kullanıcı hesaplarını ve şifreleri kırmak için tahminlere dayalı deneme yanılma yöntemini kullanan bir DoS saldırısı türüdür. Amaç; kullanıcının şifresini ele geçirmek ve sisteme o kullanıcı üzerinden sızmak olan brute force saldırılarını gerçek zamanlı olarak engelleyip sistem yöneticisine haber veren bir STS tasarlamaktır.

2. *Verilerin toplanması :* Geliştirilen STS yazılım projesinde kullanılan YSA modelini eğitmek için gerekli verilerin toplanması sistem yöneticisi tarafından yapılmıştır. Sistemin log kayıtları belirli periyotlarda (1 dakikalık, 1 saatlik, 1 günlük, 1 haftalık, 1 aylık log bilgileri gibi.) analiz yapılarak, saldırı olarak kabul edilen ve saldırı olarak kabul edilmeyen durumlar belirlenip gerekli olan eğitim tabloları hazırlanır.

3. *Modelin kurulması :* STS' de kullanılan YSA' nın tasarlanmasında ileri beslemeli ağ yapılarından olan MLP kullanılmıştır. Öğrenme algoritması olarak LM algoritması kullanılmıştır. Ağın topolojisi 1 giriş, 1 ara katman ve 1 çıkış katmanından oluşmaktadır. Giriş katmanında başarılı giriş sayısı ve başarısız giriş sayısı olarak iki giriş değeri vardır. Ara katmanda 3 nöron vardır. Çıkış katmanında saldırı ve saldırı değil olarak iki çıkış değeri vardır (Şekil 6.6.). YSA ağı hazırlanmış eğitim tablosu kullanılarak eğitilir. YSA' nı eğitmek için Visual

Studio 2005 yazılım geliştirme programı içerisindeki hazır sınıflardan olan "NeuronDotNet.Core" kullanılmıştır.

4. *Modelin değerlendirilmesi :* YSA, hazırlanmış olan eğitim tabloları kullanılarak eğitilmiş ve hata oranlarını gösteren simülasyon grafikleri oluşturulmuştur (Resim 6.23, 6.24, 6.25). Hazırlanan eğitim tablolarına göre eğitilen YSA' da en az hata "Eğitim Tablosu 1" kullanılarak eğitilmiş YSA' da çıkmıştır.

   Eğitilen YSA, yazılım projesi içinde hazırlanan test ekranı kullanılarak test edilmiştir (Resim 6.26, 6.27, 6.28). Test sonuçlarına göre en başarılı sonuç "Eğitim Tablosu 1" kullanılarak eğitilmiş YSA' da çıkmıştır. Başarı oranı %100 olarak hesaplanmıştır.

5. *Modelin Kullanılması :* Eğitilmiş olan YSA, saldırı tespit işlemini yapabilmesi için yazılıma entegre edilmiştir. Saldırı tespiti yapıp yapmadığını test edebilmek için geliştirilen brute force yazılımı ile geliştirilen STS yazılımına saldırı düzenlenmiştir. Yapılan saldırı işlemi sonucunda geliştirilen STS' nin brute force saldırıyı algılayıp; saldırı yapılan kullanıcı ve IP adresini engellemiş ve yetkili kullanıcıya e-posta yolu ile bildirimde bulunmuştur.

6. *Modelin izlenmesi :* Tasarlanan YSA, saldırı tespit sistemi içinde izlenerek zaman içerisinde oluşabilecek durumlara göre yeniden eğitilebilir.

7. SONUÇLAR VE DEĞERLENDİRME

Günümüzde İnternet kullanıcılığı daha da yaygınlaşmış, eğitimden, sağlığa, alışverişten, bankacılık işlemlerine kadar kullanıcılarının her geçen gün arttığı büyük bir ağa dönüşmüştür. Bu büyük ağda dolaşan önemli verilerimiz karşımıza bilgi güvenliği kavramını çıkarmıştır.

Bilgi çağında, bilginin kendisi kadar güvenliği de önemli bir konu haline gelmiştir. Çünkü artık saldırılar hem çeşitli sebeplerle bilinçli kişi ve kişilerce yapılırken hem de bilinçsiz kullanıcılar tarafından istenmeden yapılmakta ve sistemlere zarar vermektedir. Üstelik bilinçli kullanıcılara karşı önlem almak mümkün iken, kurum ve kuruluşların içerisinde yer alan bilinçsiz kullanıcılar için çoğu zaman daha zor olmaktadır.

Veri tabanlarında önemli veriler bulunan kurum ve kuruluşlar ağlarını çok katmanlı güvenlik yöntemleriyle korumaya çalışmaktadırlar. Bunlardan biri olan saldırı tespit sistemleri büyük ağlara sahip kurum ve kuruluşlar için oldukça önemlidir. Ancak Saldırı Tespit Sistemlerini doğru yapılandırmalı ve takip etmeleri gerekmektedir.

Güvenli ve sorunsuz bir ağ için sistem yöneticileri, kurulmuş bir saldırı tespit sistemini sadece yönetmekle sınırlı kalmamalıdır. Saldırı tespit sistemi basit konfigürasyonlarla kendi ağlarına uyarlanıp daha verimli sonuçlar alınabilir. Son zamanlarda pazarlamadan banka ve sigortacılık işlemlerine kadar adını sıkça duyduğumuz veri madenciliği teknikleri ile saldırı tespit edilip, mevcut saldırı tespit sistemleri yeniden konfigüre edilebilir.

Bu tez çalışmasında veri madenciliği modellerinden tahmin edici model içerisinde bulunan sınıflama modeline ait Yapay Sinir Ağı tekniği kullanılarak, DoS ataklarından biri olan Brute Force tipi saldırıları gerçek zamanlı veriler yardımı ile kullanıcı bazlı ve IP tabanlı tespit eden saldırı tespit sistemi uygulaması geliştirilmiştir. Tespit edilen kullanıcı ve IP adreslerinin sisteme girişi engellenir ve e-posta ile sistem yöneticisine bildirilir. Ayrıca saldırı tespit sistemi uygulaması; kullanıcı adı, IP adresi, tarih ve saat gibi bilgileri sistem yöneticisine sunmaktadır.

Geliştirilen saldırı tespit sistemi yazılımı doğru kullanıcı adı ve şifre ile girildiğinde; sistem yöneticisine seçilen tarihe ait hatalı girişleri, saldırı yaptığı kabul edilerek engellenen IP listesini, YSA eğitimi ekranını ve eğitilen YSA test ekranını gösterir.

Aynı kullanıcı adı ve IP' den bir dakika içerisinde yapılan başarılı ve başarısız girişlerin sayısı tespit edilerek, sistem yöneticisi tarafından eğitilmiş olan YSA' ya parametre olarak gönderilir. YSA' dan dönen saldırı ve saldırı değil değerleri karşılaştırılarak (saldırı>saldırı değil), yapılan bağlantının saldırı olup olmadığına karar verilir. Saldırı yapıldığına karar verilen kullanıcı ve IP, blok tablosuna eklenir ve belirlenen e-posta adresine yazılım tarafından yapılan işlemle ilgili olarak e-posta atılır. Blok tablosuna eklenen kullanıcı bazlı IP' den yapılan bağlantılar artık geçersiz kabul edilir ve sisteme girişine izin verilmez.

Yapılan STS çalışmasının değerlendirilmesi:

- YSA, zeki STS çalışmalarında kullanılabilir.
- YSA' yı eğitmek ve test etmek için gerekli olan verinin hazırlanması hem zaman ve hem de ciddi bir analiz gerektirmektedir (log bilgilerinin toplanması, sınıflandırılması, elde edilen verinin değerlendirilmesi vb.).
- Her sistem yapısına göre farklılıklar içerebilir (Kullanıcı sayısı, ağ trafiği, uygulama sunucularının ve veritabanlarının cevap süresi gibi.). Burada STS tasarımında bulunacak kişinin bu ve benzeri durumları göz önünde bulundurması gerekir.
- YSA' nın STS içerisinde başarı oranı eğitim ve test tablolarının doğru bir şekilde hazırlanmasına bağlıdır.
- YSA' nın başarı oranı eğitim tablolarındaki kayıt sayılarından ziyade, kayıtların doğru analiz edilerek oluşturulmasına bağlıdır. YSA eğitimi için kayıt sayısı önemli olmakla beraber, asıl önemli olan tablolar içerisindeki kayıtların doğruluğudur.

- YSA yapısının kurulması ve test edilmesi deneme yanılma yöntemi ile olduğundan zaman almaktadır. Sistemin istenilen doğruluk oranına ulaşıncaya kadar işlemlerin tekrar edilmesi gerekmektedir.

- YSA' nın yapısına (ara katman sayısı, nöron sayısı ve fonksiyon tipi) bağlı olarak sonuçlar değişebilir.

- Hazırlanan eğitim tabloları ile eğitilmiş olan YSA kullanılarak eğitim tablosu kayıtlarında bulunmayan saldırıları başarılı bir şekilde tespit etmiştir.

- Hazırlanan YSA yapısı kolaylıkla gerçek zamanlı uygulamalara adapte edilebilir.

Yapılan STS çalışmasının sınırlılıkları:

- STS' lerin geliştirilmesi için gerekli olan verinin toplanması (log bilgileri) zaman almaktadır.

- Toplanan verilerin analiz edilerek ayrıştırılması bilgi ve uzmanlık gerektirmektedir.

- Hazırlanan veriye uygun yapının kurulması, test edilmesi ve değerlendirilmesi hem zaman hem de uzmanlık gerektirmektedir.

- Her sistemin kendine has bir yapısı olacağından veri toplama ve analiz işlemlerinin tekrarlanması gerekmektedir.

- STS' ler için mevcut veri tabanları güncel değildir.

Kurum ve kuruluşların kişisel bilgisayar kullanıcılarının Saldırı Tespit Sistemi kurması ve bunu düzenli olarak takip etmesi gerekmektedir. Bunun bir ihtiyaç ve güvenlik duvarı kadar önemli bir uygulama olduğunu tüm sistem yöneticilerinin bilmesi gerekmektedir. Çalışma sistem yöneticilerinin; saldırı tespit sistemlerini veri madenciliği tekniğini kullanarak ihtiyaçlarına göre yapılandırabilmeleri için bir fikir verecektir.

KAYNAKLAR

1. Vahaplar, A., İnceoğlu, M. M., "Veri Madenciliği ve Elektronik Ticaret", ***VII. Türkiye'de İnternet Konferansı*** , İstanbul, 1 (2008).

2. Kalıkov, A., "Veri madenciliği ve bir e-ticaret uygulaması", Yüksek Lisans Tezi, ***Gazi Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 1 (2006).

3. Sağıroğlu, Ş., Alkan, M., "Her yönüyle elektronik imza (e-imza)"***, Grafiker Yayınları***, Ankara, 1-100 (2005***)***.

4. Pei, J., Upadhyaya, S.J., Farooq, F., Govindaraju, V., "Data mining for intrusion detection: techniques, applications and systems," ***20th International Conference on Data Engineering (ICDE'04)***, 1063-6382 (2004).

5. Lunt, T. F., "Automated audit trail analysis and intrusion detection: A survey", ***11th National Computer Security Conference,*** Baltimore, 65-73 (1988).

6. Güven, E. N., "Zeki saldiri tespit sistemlerinin incelenmesi, tasarimi ve gerçekleştirilmesi", Yüksek Lisans Tezi, ***Gazi Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 2-20 (2007).

7. Endorf, C., Schultz, E., Mellander, J., "Intrusion detection & prevention", Jenn Tust, Jody McKenzie, Elizabeth Seymour, ***McGraw-Hill***, California, 10-150 (2004).

8. Akpınar, H., "Veri tabanlarında bilgi keşfi ve veri madenciliği", ***İstanbul Üniversitesi İşletme Fakültesi Dergisi***, 29 : 1 (2000).

9. Canbek, G., Sağıroğlu, Ş., "Bilgisayar sistemlerinde yapılan saldırılar ve türleri: bir inceleme", ***Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi,*** 23 (1-2): 1-12 (2007).

10. Can, E., "Gerçek zamanlı veriler yardımı ile karar veren bir bilgisayar ağı saldırı tespit sisteminin tasarlanması ve gerçeklenmesi", Yüksek Lisans Tezi, ***Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 4-5 (2007).

11. Karabağ, R., Takçı, H., Akyüz, T., "Kullanıcı davranış analizi ile nüfuz tespit modeli (KDA-NTM)", ***Gebze İleri Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümü***, Gebze, 1-5 (2006).

12. Hussain, A., Heidemann, J., Papadopoulos, C., "Distingushing between singel and multisource atttacks using signal processing", ***Computer Networks***, 46 (2004).

13. İnternet : "Firewall nedir?" http://www.wifi-turk.com/makale-21-firewall-nedir.html (2008).

14. İnternet : "Örün (WEB) Güvenliği" http://www3.itu.edu.tr/~orencik/AgGuvenligi2007Sunumlari/balkanay_web_sunum.pdf (2007).

15. Şahin, O., "Sınır Güvenliği", ***Tübitak-UEKAE Ağ Güvenliği Grubu Eğitim Raporu***, Ankara, 261-344 (2008).

16. Denning, D. E., "An intrusion detection model", ***IEEE Transactions on Software Engineering***, 13(2): 118–131 (1987).

17. İnternet : İTÜ / Bilgi İşlem Dairesi Başkanlığı "Destek; IDS türleri" http://www.bidb.itu.edu.tr/?d=493 (2008).

18. İnternet : "Saldırı tespit sistemi alınırken dikkat edilecek hususlar" http://www.olympos.org:81/article/articleview/275/1/2/saldiri_tespit_sistemi_ids_alirkendikkat_edilecek_hususlar (2008).

19. Mell, P., Hu, V., Lippmann, R., Haines, J., Zissman, M., "An overview of issues in testing intrusion detection systems", **NIST Technical Report**, 3-4 (2003).

20. İnternet : "Veri madenciliği ile saldırı tespiti" http://www.teknoturk.org/docking/yazilar/tt000117-yazi.htm (2008).

21. Takçı, H., Akyüz, T., Soğukpınar, İ., "Web atakları için metin tabanlı anormallik tespiti", ***Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi,*** 22 (2) : 247-253 (2007).

22. Axelsson, S., "Intrusion detection systems: A survey and taxonomy", Technical Report 99-15, ***Dept. of Computer Eng., Chalmers University of Technology***, Göteborg, Sweden, 1-23 (2000).

23. Han, J., Kamber, M., "Data mining: concepts and techniques", ***CA: Morgan Kaufmann***, San Francisco, 312 (2000).

24. Mitra, S., Acharya, T., "Data mining : multimedia, soft computing , and bioinformatics", ***John Wiley & Sons Publisher,*** (2003).

25. Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., Zanasi, A., "Discovering data mining: from concept to implementation", ***Prentice Hall***, Upper Saddle River, NJ, 517 (1998).

26. Jacobs, P., "Data mining: what general managers need to know", ***Harvard Management Update,*** 4 (10) : 8 (1999).

27. Davis, B., "Data mining transformed", ***Information Week***, 751: 86 (1999).

28. DuMouchel, W., "Bayesian data mining in large frequency tables, with an application to the FDA spontaneous", ***American Statistician***, 53 (3): 177 (1999).

29. Hand, D. J., "Data mining: statistics and more ?", ***The American Statistician***, 52:112-118 (1998).

30. Alataş, B., Akın, E.,; "Veri madenciliğinde yeni yaklaşımlar", ***Ya/Em-2004-Yöneylem Araştırması/Endüstri Mühendisliği XXIV Ulusal Kongresi***, Gaziantep-Adana , (2004).

31. Dolgun, M., Ö., "Büyük alışveriş merkezleri için veri madenciliği uygulamaları", Yüksek Lisans Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 23-27 (2006).

32. İnternet : "Cross industry standard process for data mining" http://www.crispdm.org/Process/index.htm (2004).

33. İnternet : "Perakende Sektöründe Veri Madenciliği " http://www.spss.com.tr/pdfs/SPSSPerakendeRaporu_web2010.pdf (2010).

34. Kasap, E., "Sigortacılık sektöründe müşteri ilişkileri yönetimi yaklaşımıyla veri madenciliği teknikleri ve bir uygulama", Yüksek Lisans Tezi, ***Marmara Üniversitesi,*** İstanbul, 35-37 (2007).

35. İnternet : "Veri madenciliği veya bilgi keşfi" http://www.bilgiyonetimi.org/Cm/Pages/MklGos.Php?Nt=538 (2005).

36. Şimşek, U. T., "Veri madenciliği ve müşteri ilişkileri yönetiminde (CRM) bir uygulama", Doktora Tezi, ***İstanbul Üniversitesi Sosyal Bilimler Enstitüsü***, İstanbul, 25-46 (2006).

37. Aydoğan, F., "E-ticarette veri madenciliği yaklaşımlarıyla müşteriye hizmet sunan akıllı modüllerin tasarımı ve gerçekleştirimi", Yüksek Lisans Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü,*** Ankara, 12-16 (2003).

38. Velickov, S., Solomatine, D., "Predictive data mining: practical examples, artificial intelligence in civil engineering", Almanya, 1-17 (2000).

39. Altıntaş, T., "Veri madenciliği metotlarindan olan kümeleme algoritmalarinin uygulamali etkinlik analizi", Yüksek Lisans Tezi, **Sakarya *Üniversitesi Fen Bilimleri Enstitüsü***, Sakarya, 13-14 (2006).

40. Özçınar, H., "KPSS sonuçlarının veri madenciliği yöntemleriyle tahmin edilmesi", Yüksek Lisans Tezi, ***Pamukkale Üniversitesi Fen Bilimleri Enstitüsü***, Denizli, 11 (2006).

41. Akbulut, S., "Veri madenciliği teknikleri ile bir kozmetik markanın ayrılan müşteri analizi ve müşteri segmentasyonu", Yüksek Lisans Tezi, ***Gazi Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 20-25 (2006).

42. Sıramkaya, E.,"Veri madenciliğinde bulanık mantık uygulaması", Yüksek Lisans Tezi, ***Selçuk Üniversitesi Fen Bilimleri Enstitüsü***, Konya, 31 (2005).

43. İnternet : Marmara Üniversitesi Teknik Eğitim Fakültesi Makine Bölümü "Genetik Algoritma ve Uygulama Alanları", http://www.mmo.org.tr/muhendismakina/arsiv/2001/ekim/Genetik_Algoritma.htm (2005).

44. Saraç, T., "Yapay Sinir Ağları Seminer Projesi", Gazi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü Ana Bilim Dalı, Ankara, 9-20 (2004).

45. Elmas, Ç., "Yapay Sinir Ağları (Kuram, Mimari, Eğitim, Uygulama)", ***Seçkin Yayınları***, Ankara, 192 (2003).

46. Öztemel, E., "Yapay Sinir Ağları", ***Papatya Yayınları***, İstanbul, 25-27 (2003).

47. Sağıroğlu, Ş., Beşdok, E., Erler, M., "Mühendislikte Yapay Zeka Uygulamaları-I Yapay Sinir Ağları", ***Ufuk Kitabevi,*** Kayseri, 10-100 (2003).

48. Bayır, R., "Yapay Zeka Teknikleri Kullanarak Marş Motorlarında Hata Teşhisi", Doktora Tezi, ***Gazi Üniversitesi Fen Bilimleri Enstitüsü Elektronik Bilgisayar Eğitimi Anabilim Dalı***, Ankara, 145 (2005).

49. İnternet : "Yapay Sinir Ağlarının Katmanları", http://www.ahmetkakici.com/yapay-sinir-aglari/yapay-sinir-aglarinin-katmanlari/ (2008).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : YILDIZ, Elmas
Uyruğu : T.C.
Doğum tarihi : 08.11.1982
Doğum yeri : Kayseri
Medeni hali : Evli
Telefon (İş) : 0 (312) 413 11 85
e-mail : elmasyildiz@meb.gov.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lisans	Gazi Üniversitesi /Elek. ve Bilg. Eğt. Böl.	2004
Lise	Sami Yangın Anadolu Ticaret Meslek Lisesi	2000

İş Deneyimi

Yıl	Yer	Görev
2006-…..	M.E.B. EĞİTEK Bilişim Hizmetleri Dairesi	Bilişim Tek. Öğrt.
2004-2006	Simav Anadolu Kız Meslek Lisesi	Bilişim Tek. Öğrt.

Yabancı Dil

İngilizce

Yayınlar

1. Yıldız, E., Arıcı, N.," Gerçek Zamanlı Bir Saldırı Tespit Sistemi Tasarımı Ve Gerçekleştirimi", ***e-Journal of New World Sciences Academy***, (Kabul Edildi, 2010).

Hobiler

Bilişim Teknolojileri, Ağ Yönetimi, Kitap Okuma, Sinema