T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR GÜVENLİĞİNDE YAPAY BAĞIŞIKLIK SİSTEMİNİN KULLANILMASI

YÜKSEK LİSANS TEZİ

Bil.Müh. Cengiz SERTKAYA

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜH.

Tez Danışmanı : Doç. Dr. Feyzullah TEMURTAŞ

Ocak 2009

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR GÜVENLİĞİNDE YAPAY BAĞIŞIKLIK SİSTEMİNİN KULLANILMASI

YÜKSEK LİSANS TEZİ

Bilg.Müh. Cengiz SERTKAYA

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜH.

Bu tez 08 /01 /2009 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Doç.Dr. Feyzullah TEMURTAŞ	**Prof.Dr. Etem KÖKLÜKAYA**	**Yrd.Doç.Dr. Ali GÜLBAĞ**
Jüri Başkanı	**Üye**	**Üye**

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Danışman Hocam Sayın Doç. Dr. Feyzullah TEMURTAŞ'a, maddi ve manevi her türlü destekleriyle beni hiçbir zaman yalnız bırakmayan çok değerli aileme teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

A	: Uygun timus hücreleri barındıran populasyon
Ab	: Antibody hücresi
AFRL	: Hava Kuvvetleri Araştırma Laboratuarı (Air Force Research Projects Agency)
Ag	: Antijen hücresi
B	: Bağışıklık sistemindeki dedektör ajanları
C	: Veri seti elemanlarının permütasyon değeri
CLONALG	: Optimizasyon problemleri için önerilen algoritma.
C#	: Sistemin geliştirildiği programlama dili.
DARPA	: Savunma İleri Araştırma Projeleri Teşkilatı (Defence Advanced Research Projects Agency)
DARS	: Otonom robot sistemi
DoS	: Hizmet aksattırma saldırıları
ε	: Eşik değer
FTP	: Dosya Transfer Protokolü (File Tranfer Protokol)
GA	: Genetik Algoritma
ICMP	: İnternet Mesaj Kontrol Protokolü (Internet Control Message Protokol)
IDES	: Saldırı Tespiti Uzman Sistemi (Intrusion Detection Expert System)
IDEVAL	: İlk standart veri kümesi gövdesi (Intrusion Detection Evaluation)
IITP	: Rus bilim akademisi
IP	: İnternet Protokolü (Internet Protocol)
KDD	: Bilgi Keşfi ve Teslimatı (Knowledge Discovery and Delivery)
KDD'99	: Bilgi Keşfi ve Teslimatı yapılanmasının ürettiği veri setleri

L	: T hücrelerinin uzunluk ölçüsü
LB	: Problem alt limiti
MHC	: Yabancı doku analizinde kullanılan genetik bölgeler (Major Histokompatibilite kompleks)
M(k)	: K antikoru ile gösterilen çözümün toplam tamamlanma zamanı
N	: Anormallik tespitinde gözlem sayısı
P	: T hücrelerinin potansiyel repertuarı
PHAD	: Paket Başlığı Davranış Tespiti
POP3	: E-posta Kontrol Protokolü (The Post Office Protokol)
R2L	: Bilgisayar açıklarından yararlanarak sızmaya çalışma saldırısı.
S	: T hücre populasyonu
SES	: Saldırı Engelleme Sistemleri (Intrusion Prevention Systems)
SMTP	: Mail Protokolü (Simple Mail Transfer Protocol)
STS	: Saldırı Tespit Sistemleri (Intrusion Detection Systems)
T	: Timus hücresi
TCP	: Ağ İletişim Protokolü (Transmission Control Protokol)
TELNET	: Terminal Protokolü (Telecommunication Network)
TN/R	: Eğitim ve test aralıklarında anaormallik skoru
TTL	: Cevap alma limiti (Time to live)
U2R	: Yönetici yetkisine ulaşma çabasından doğan saldırı türü.
UDP	: Ağ İletişim Protokolü (User Datagram Protokol)
UNIX	: Bilgisayar İşletim Sistemi
YBS	: Yapay Bağışıklık Sistemi (Artificial Immune System)
YSA	: Yapay Sinir Ağları (Artificial Neural Network)

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Saldırı tespit sistemleri,yapay bağışıklık sistemi.

Bu çalışmada bilgi ve bilgisayar güvenliğini sağlamak için geliştirilen araçlardan birisi olan saldırı tespit sistemleri (STS) incelenmiş, STS geliştirmek için kullanılan yöntemler araştırılmıştır. Bu çalışmanın ana konusu olan yapay bağışıklık sistemi (YBS) üzerine detaylı bir inceleme yapılmış ve YBS kullanılarak zeki bir STS geliştirilmiştir.Geliştirilen sistemden elde edilen sonuçlar, diğer yaklaşımlardan elde edilen değerler ile karşılaştırılmış ,farklılıklar sunulmuştur.

Yapılan araştırma, inceleme ve değerlendirme çalışmalarından yola çıkarak bu tez kapsamında geliştirilen yazılım ve sunulan önerilerin, ülkemizde bilgi ve bilgisayar güvenliği konusunda yapılacak çalışmalara büyük katkılar sağlaması ve yeni ufuklar açması beklenmektedir.

IMMUNE BASE SYSTEM IN COMPUTER SECURITY

SUMMARY

Key Words: Intrusion detection systems, Artificial immune base system

In this thesis, intrusion detection systems (IDS) which are important tools for providing information and computer security were analyzed, the methods used in developing IDS were reviewed. A detailed analysis about the artificial immune base system, IDS was presented and this IDS were compared to the other classical IDS. In the light of those, differences between this methods were given.

According to the reviews, the evaluations and the experiences gained on IDSs during this study, it can be concluded that this study might contribute to improve the studies on information and computer security being done in near future and bring more awareness and perceptions to the security issues.

BÖLÜM 1. GİRİŞ

İnsanlığın gelişim süreçlerinde en önemli yeri alan "bilgi", bir konu hakkındaki belirsizliği azaltan, sürekli gelişen değişen ve yenilenen bir varlıktır. Ayrıca, saygınlık, ayrıcalık ve farkındalık oluşturduğundan yüksek değere sahiptir. Doğal olarak sahip olunan bu varlığın elektronik ortamlarda karşılaşabileceği tehdit ve tehlikelerden korunması gereklidir. Bilginin değeri, kullanılacağı yere göre değişim gösterir. İhtiyacımız olmayan bir bilgi bizim için önemli değil iken, başkaları için hayati önem taşıyabilir. Bu açıdan bilginin gizliği ve bütünlüğünün sağlanması, yararlı bilgilerin başkalarının veya saldırganların ellerine geçmemesi gereklidir[1].

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 başlanması 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. Ülkemizde de bu konuda hızlı bir gelişim olduğu dikkat çeken konulardan birisidir [1].

Günümüzde siyasi gücün de bir temsili olan bilişim teknolojilerinin gelişmişliği, bilgi ve bilgisayar güvenliği konusundaki ulusal yapılanmaların önemini ortaya koymaktadır. Artık fiziksel saldırılardan farklı olarak, yakın zamanlarda örneklerini gördüğümüz siber saldırılar devri başlamıştır. Ulusal olarak elektronik ortama verilen önem sadece yeni sistemler geliştirmek ve prosedürel işlere hız kazandırmak olarak

algılanırsa, güvenlik açıklarından kaynaklanacak sorunların çok büyük maddi kayıplara yol açabileceği göz önüne alınmalıdır. Elektronik ortama geçiş sürecinde, standartlarla belirlenmiş olan güvenlik prosedürleri dikkate alınarak, risk ve tehditler belirlenmeli, bilginin değeri ölçülmeli, saldırı senaryolarına yönelik çalışmalar yapılmalıdır. Bilgiyi korumak için belirlenen kriterlere göre güvenlik politikaları oluşturulmalı ve uygulanmalıdır[1].

Teknolojik önlemlerin geliştirilmesi sırasında bir varlık olarak bilgiyi, tehdit ve saldırılara karşı korumak için güvenlik duvarları, antivirüs yazılımları, saldırı tespit sistemleri (STS), saldırı engelleme sistemleri (SES) gibi araçlar geliştirilmiştir. Bu araçların belirlenen kural ve politikalara göre yapılandırılması, bilgi güvenliğimizi büyük ölçüde sağlayacaktır[2].

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[2].

İlk olarak 1980 yılında Anderson'un yaptığı çalışmalar sonucunda ortaya çıkan STS'ler, ardından yapılan birçok çalışma ile hızla gelişmesini devam ettirmiştir. 1988 yılında geliştirilen IDES (Saldırı Tespiti Uzman Sistemi), o yıla kadar yapılan birçok çalışmayı üzerinde barındırması açısından en önemli STS çalışmalarından biridir [2].

İstatistiksel yaklaşımların dışında yine o yıllarda, kural tabanlı, eşik değeri belirleme (threshold value), durum geçiş diyagramları (state transition diagrams), veri madenciliği gibi metotların da kullanıldığı bilinmektedir[3,4,5,6]. Ancak teknolojinin hızlı gelişimi ve saldırganların bu gelişimden faydalanarak eskiye oranla daha az bilgi ve tecrübe sahibi olmalarına rağmen daha etkili ve hızlı saldırılar

gerçekleştirmeleri, güvenlik boyutunu dinamikleştirmiş ve sürekliliği zorunlu kılmıştır. Bu nedenle, STS'lerin tarihsel gelişimi sürecinde zeki STS'lere ihtiyaç duyularak, yapay sinir ağları (YSA), yapay bağışıklık sistemi(YBS), bulanık mantık gibi zeki teknikler de kullanılmaya başlanmıştır. 1990'ların başında kullanılmaya başlanan ve ön plana çıkan zeki tekniklerin kullanımı ile STS'lerin başarı oranlarının arttığı gözlenmiştir. Özellikle, önceden bilinmeyen yeni saldırıların tespit edilebilmesi için kullanılan anormallik tespiti yaklaşımında, zeki tekniklerin kullanılması başarı oranının arttırılmasında en büyük etkenlerdendir[7]. Bu nedenle, bu tez kapsamında geliştirilen STS, klasik yaklaşımlar yerine zeki tekniklerinden birisi olan Yapay Bağışıklık Sistemi kullanılarak gerçekleştirilmiştir.

Günümüze kadar yapılan çalışmalar incelendiğinde, STS'lerde; veri toplama, özellik seçme, davranış modellerinin belirlenmesi ve sınıflandırılması, kural tabanlı sistemler için kuralların belirlenmesi, raporlama ve sonuç üretme aşamalarında güçlüklerle karşılaşılmaktadır. Çok sayıda işlemde karşılaşılan bu güçlüklerden de görülebileceği gibi aslında en büyük problem, STS'lerin tasarım ve uygulama aşamalarında kullanılabilecek veritabanlarının eksikliğinden kaynaklanır [8]. Uygulamalarda kullanılabilir bir veritabanının olması, gerçekleştirilmesi planlanan sistemin başarılı olup olmayacağı konusunda daha hızlı sonuç üretilebilmesi açısından önemlidir. Bununla birlikte, STS tasarımı sırasında araştırmacıların farklı veritabanları geliştirmesi sonucunda, başarı oranlarının objektif olarak değerlendirilememesinden dolayı, literatürde saldırı veritabanlarının oluşturulmasına yönelik çalışmalar yapılmıştır[8]. STS'lerin geliştirilmesi ve test edilmesi için kullanılabilecek saldırı veritabanlarının:

1. Geliştirilmesinin maliyetli ve zor olması,
2. Tamamen ayrı bir çalışma zamanı gerektirmesi,
3. Geliştirilse bile gerçeğe uygunluğunun kabul edilebilir olmasının sağlanması,
4. Farklı yaklaşım ve tekniklerle kullanılmaya elverişli ve kolay işlenebilir olmasının sağlanması,
5. Güncelliğinin korunması gibi problemlerle karşılaşılmaktadır [8].

Günümüze kadar yapılan STS çalışmalarının pek çoğunda önceden hazırlanmış veri kümelerinin kullanılmış olduğu tespit edilmiştir. Ancak STS'ler için geliştirilen, literatürdeki veritabanı uygulamaları incelendiğinde, günümüzde araştırmacıların uygulamalarında kullanılabilecekleri güncel bir veritabanına rastlanmamıştır[9]. DARPA'nın 1998 ve 1999'da STS'lerin başarılarının değerlendirilmesi için yaptığı çalışma ve DARPA'nın çalışmasının farklı bir versiyonu olan KDD'99 verilerinin oluşturulması, günümüzde hala kullanılan ve saldırı veritabanlarının nasıl hazırlanması gerektiğine örnek olan çalışmalardır[8]. Bu çalışmaların en büyük dezavantajı ise son on yıldır eklenen yeni saldırıları içermemesidir. Yukarıda sıralanan problemlerin ortadan kaldırılabilmesi için STS'lerin tasarımlarının artması, sistemlere yapılan saldırıların güncel veritabanlarında tutulması ve tasarlanan sistemlerin ise bu güncel verilerle test edilmesi önem arz etmektedir[10]. Bu tez çalışmasında, belirtilen bu problemlerin kısmen ortadan kaldırılabilmesine yönelik olarak STS tasarımı gerçekleştirilmiş, literatürdeki mevcut veritabanları kullanılarak test edilmiş ve STS'lerin geliştirilmesi ve başarıların test edilmesinde kullanılabilecek olan "Ulusal Saldırı Veri Kümesi" hazırlanması zorunluluğu ortaya konulmuş ve bunun için farklı önerilerde bulunulmuştur.

Bu tez çalışması 6 bölümden oluşmuştur. Tezin ikinci bölümünde, STS'ler hakkında bilgi verilmiş , bilgi ve bilgisayar güvenliği açısından, öneminden bahsedilmiştir

Üçüncü bölümde, bu tez kapsamında kullanılan Yapay Bağışıklık Sistemleri (YBS) ile ilgili temel bilgiler, kullanılan teknikler ve uygulama alanları hakkında bilgi verilmiştir.

Dördüncü bölümde, bir STS ve veri kümesi tasarımı için gerçekleştirilmesi gereken işlem adımları belirlenerek, anormallik tanıma yapabilen YBS tabanlı bir zeki STS önerisi yapılmıştır.

Beşinci bölümde, YBS ile oluşturulan sistem ve diğer yaklaşımlar için elde edilen sonuçlar hakkında yorumlar yapılmıştır.

Altıncı ve son bölümde, bu tez kapsamında elde edilen sonuçlardan kazanımlar sunulmuştur.

BÖLÜM 2. SALDIRI TESPİT SİSTEMLERİ (STS)

2.1. Giriş

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 [10].

Bilgiyi korurken, var olan sistemlerin sürekliliğinin sağlanması da hayati önem taşımaktadır. Sürekliliği korumak için, yapılan saldırılara karşı alınan önlemlerin güncelliğini koruması gerekmektedir. Bu güncellik de ancak değişen saldırı ve yöntemlerin bilinmesi ve var olan sisteme adapte edilmesi ile sağlanabilir. Yeni bir saldırının veya yöntemin tespitine geçmeden önce "saldırı"nın ne anlama geldiğini bilmek gerekir. Bu nedenle güvenlik konusunda yapılan çalışmalarda saldırının birçok tanımı yapılmıştır. Anderson'a göre bir saldırı, izin almadan bilgiye ulaşım, değiştirme, sistemi kullanılmaz veya güvenilmez hale getirmektir [11]. Anderson'un 1980'de yapmış olduğu bu tanım hala geçerliliğini koruyan en temel tanımlardan biridir. Günümüzde ise saldırı, "bilginin mahremiyetini, bütünlüğünü ve erişilebilirliğini tehlikeye atabilecek girişimlerin kümesi" olarak tanımlanmaktadır [12]. 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 [13].

En genel anlamıyla, saldırı tespiti işini yapmak için geliştirilen sistemlere "saldırı tespit sistemleri" denir. Ancak günümüze kadar yapılan araştırmalar ve çalışmalar incelendiğinde, saldırı tespit sistemlerinin tanımları olduğu görülmüştür. Yapılan bu tanımlara göre STS'ler;

Bilgisayar sistemlerine yapılan atakları ve kötüye kullanımları belirlemek için tasarlanmış sistemlerdir [14].

Tercihen gerçek zamanlı olarak, bilgisayar sistemlerinin yetkisiz ve kötüye kullanımı ve suistimalini tespit etmek için kullanılırlar [15].

Saldırıyı durdurma girişiminde bulunmayan ve olası güvenlik ihlali durumlarında, sistem güvenlik çalışanlarına uyarı mesajı (alarm) veren sistemlerdir [16].

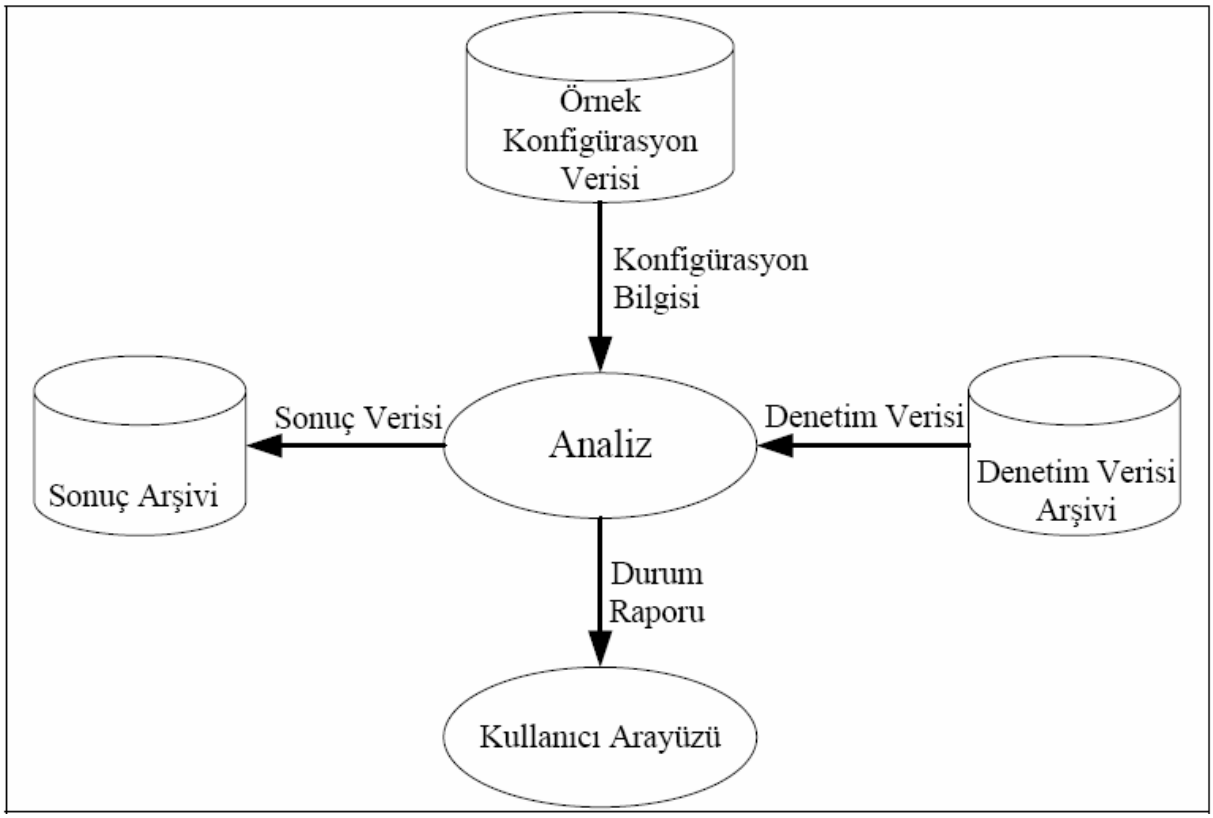
Bilgisayar sistemlerinin kaynaklarına veya verilerine yetkisiz erişimleri belirler [17].

Bilgisayar güvenliği alanındaki "hırsız alarm"larıdır [18].

Bilgisayar veya ağ sistemine yapılan yetkisiz erişimleri tespit etmek için kullanılan yazılım araçlarıdır. STS'ler kötü niyetli ağ trafiği ve bilgisayar kullanımını tespit etme yeteneğine sahiptir. Bir STS, olası güvenlik açıklarını belirleyebilmek için bilgisayar veya ağ içerisinde değişik alanlardan bilgileri toplar ve analiz eder. Güvenlik duvarının statik izleme kabiliyetini tamamlayan dinamik izleme elemanıdır[19].

Yukarıda sunulan tanımlardan yola çıkarak, STS'leri, bilginin elektronik ortamlarda taşınırken, işlenirken veya depolanırken başına gelebilecek tehdit ve tehlikelerin ortadan kaldırılması amacıyla, bilgiye yetkisiz erişim veya kötüye kullanım gibi girişimleri tespit edebilme ve bu tespiti sistem güvenliğinden sorumlu kişilere iletebilme özelliğine sahip yazılımsal ve/veya donanımsal güvenlik araçları olarak tanımlayabiliriz. Aynı zamanda STS'ler, ağ cihazlarını izleyerek anormal davranışları ve kötüye kullanımı tespit ederler[19].

Literatürdeki STS çalışmaları incelendiğinde, STS'lerin yapısı ile ilgili birçok gösterim yapıldığı görülmüştür. STS çalışmalarından biri olan NIDES'in geliştirilmesi sırasında çizilen örnek Şekil 2.1'de gösterilmiştir [9]. Kullanılan teknik ve yaklaşımlara göre şekillenen diğer gösterim şekillerinden ilerleyen bölümlerde bahsedilecektir.

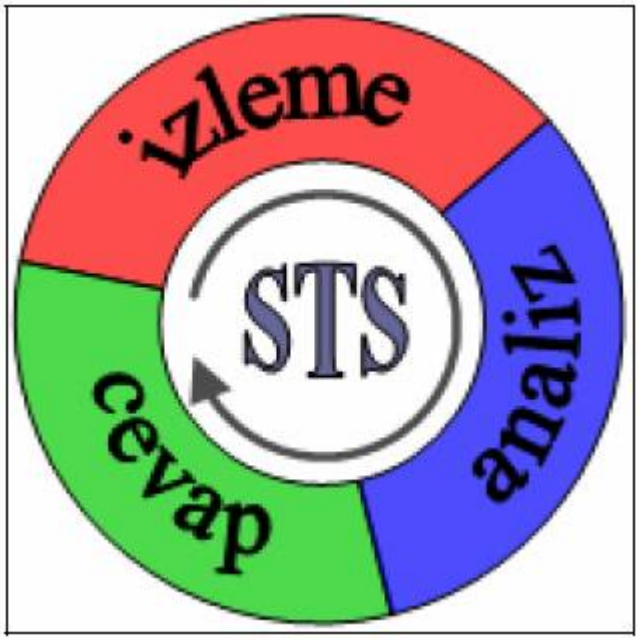


Şekil 2.1. STS'lerin temel yapısı [20]

Şekil 2.1'de sunucu tabanlı bir STS'nin temel yapısına bir örnek gösterilmiştir. Bu örnekte, bilgi kaynağı olarak denetim verileri kullanılmıştır. Denetim verisi arşivinden alınan denetim verileri, örnek konfigürasyon verileri ile karşılaştırılıp analiz edilerek, elde edilen sonuçlar, sonuç arşivine aktarılmaktadır. Aynı zamanda, olası saldırıların, kullanıcı tarafından görülebilmesi için kullanıcı arayüzüne de analiz sonuçlarını içeren durum raporu gönderilmektedir. İstenildiği takdirde, sonuç arşivi incelenebilir veya örnek konfigürasyon verileri güncellenebilir. STS'lerin yaptıkları iş temelde izleme, analiz ve cevap olmak üzere üç adımda özetlenebilir. STS'lerin

yapısal bileşenleri Şekil 2.2'de sembolik olarak gösterilmiştir. İzlemeyle başlayan saldırı tespit sürecini, analiz ve ardından sistem cevabı izlemektedir[20].

STS'ler, korunan sisteme ve kullanılan tekniklere bağlı olarak Şekil 2.2'de gösterilen üç işlevi gerçekleştirmelidir. Birbirini takip eden ve sürekli tekrarlanan bu işlemler ilk olarak izleme ile başlar. Korunacak sistem eğer bir bilgisayar ağı ise, bu bilgisayar ağında tanımlı kullanıcıların hareketleri ve ağdaki paketler, eğer bir sunucu ise bu sunucuya gelen giden veri paketleri izlenebilir. Sistemin izlenmesiyle edinilen bilgiler gerekli önişlemlerden geçirildikten sonra, kullanılan yaklaşımlara, tekniklere ve kurallara göre analiz edilir. Analiz sonucunda, izlenen hareketin veya paketin saldırı olup olmadığı tespit edilir ve sistem yöneticisine veya sorumlusuna bir cevap dönülür. STS'lerde cevap verme işlemi, genellikle bir saldırı ile karşılaşıldığında sistem yöneticisine alarm veya bilgi verme niteliğinde olabilmektedir[20].



Şekil 2.2. Bir STS'nin fonksiyonel olarak gösterilmesi [20]

2.2. Bilgi Güvenliğinde STS'lerin Önemi

İnternetin ve iletişim olanaklarının artmasıyla birlikte saldırganlar tarafından saldırılabilecek daha çok sistem ortaya çıkmıştır. Bu saldırıların büyük bir bölümü kullanılan sistemin kusurları veya eksiklerinden faydalanılarak yapılır. Bu tür saldırıları engellemenin iki yolu vardır; ilki tamamen güvenli bir sistem ve ortam oluşturmak, ikincisi ise saldırıları tespit edip gerekli önlemleri almaktır. Bunlardan ilki pratik açıdan mümkün olmamaktadır. Bunların gerekçeleri ise [21],

Kullanılan işletim sisteminde var olan açıkların genellikle ilk olarak saldırganlar tarafından fark edilmesi ve önlem alınana kadar bu açıkların kullanılabilmesi,

Veri iletiminde kullanılan protokollerin yapısında var olan bazı kuralların saldırı amaçlı kullanılabilmesi,

Kriptografik metotların ve anahtarlarının kırılabilmesi, kullanıcıların şifrelerini unutabilmesi veya kripto-sistemin kırılabilmesi gibi nedenlerle yüksek seviyede bir güvenlik sağlanamaması,

Dış ortama karşı güvenliği sağlanan sistemin, iç ortamlardan suistimal edilerek güvenliğin ortadan kaldırılması,

Güvenlik amacıyla kullanıcı yetkilerinin minimuma indirilmesi sonucu kullanıcı verimliliğinin düşmesi gibi nedenleri vardır[21].

Sistemlerini korumak isteyenler, genelde saldırı gelene kadar bekleme pozisyonunda kalmak, saldırı geldiğinde ise olabildiğince hızlı tespit etmek isterler. Bu ise STS'nin yaptığı iştir [21]. Bir saldırının hangi adresten veya hangi porttan geldiğini bilmeden engel olmak mümkün değildir. STS'ler saldırıları tespit ederken bu bilgileri de elde ederler. STS'ler, detaylı olarak topladığı ve depoladığı bilgilerden yararlanarak, saldırıları olabildiğince erken tespit etme özelliğine sahiptir. Yine aynı bilgilerin incelenmesi ile daha önce hiç karşılaşılmamış bir saldırıyı da tespit edebilir. STS'leri

de cazip hale getiren bu özelliktir. Bu nedenlerden dolayı, tamamen güvenli bir sistem oluşturmanın mümkün olmadığı görülse de, üst düzey güvenliğe sahip bir sistem oluşturmak mümkündür. Bunun için maliyet de göz önünde bulundurularak gerekli olan tüm güvenlik araçlarından faydalanılmalıdır. Ancak saldırganların, kapatılan açıkların ardından, saldıracak yeni açıklar bulabilmesinin ve önceden tahmin edilemeyen yollara başvurmasının, tamamen güvenli bir ortamın oluşturulamamasına neden olduğu unutulmamalıdır. STS'lerin bu konudaki önemi, kullanılan yeni teknikler sayesinde önceden bilinmeyen saldırıların da tespit edilebilmesini sağlamasıdır[10].

2.3. STS Bilgi Kaynağı

STS'ler, bilgi kaynaklarını analiz ve karşılaştırma yapmak için kullanırlar. Bilgi kaynakları, bilgisayar veya ağ paketlerinin dinlenmesinden elde edilebildiği gibi, kullanıcı profillerinin davranış modellerinden de elde edilebilir. Bilginin nasıl ve nereden toplanacağı, geliştirilecek olan STS'nin amaçlarına göre değişir. Bunlar; denetleme izi, ağ paketleri, uygulama kayıt dosyalarıdır. Bunlar aşağıda kısaca açıklanmıştır[10].

Denetim (hesap, günlük) izi; bilgi ve iletişim güvenliği için, sistem aktivitelerinin kronolojik olarak sıralanmış şeklidir. Denetleme izleri, sistemde gözlenen olayların sıralaması bozulduğunda veya olaylarda değişiklikler meydana geldiğinde, yeniden yapılandırma ve test etmeyi mümkün kılmak için kullanılır. Kullanıcı tanımlama sistemleri ve veritabanı yönetim sistemlerinin çoğu denetleme izi bileşeni içerir. STS'ler bilgi kaynağı olarak denetleme izini, daha çok sistemde tanımlanmış olan kullanıcıların veya grupların hareket profillerini çıkarmak için kullanır. Kullanıcıların günlük yaptıkları işler ve bu işlere yönelik sistemdeki yetkileri göz önünde bulundurularak, bir kullanıcının gün içindeki hareketleri, o kullanıcının profilini oluşturur. Uzun süren bir gözlem ve inceleme ile bu profilin doğruluğunun kanıtlanması gerekir. Eğer kullanıcının profili belirlenmişse, STS'ler belirlenen profil dışına çıkılan hareketleri saldırı olarak algılarlar. Bu nedenle, yanlış alarm oranlarını azaltmak için kullanıcı profillerinin güncellenebilir olması gerekir[10].

Ağ paketleri; koklayıcılar (sniffers) tarafından trafiğin dinlenmesiyle elde edilir. Bu günlükler daha çok hizmet aksattırma (engelleme) saldırılarını tespit etmekte kullanılır. Ağ paketlerinden elde edilen bilgiler sayesinde, sunucu tabanlı STS'lerden farklı olarak, ağ katmanında gerçekleşen atak olaylarını tespit etmek de mümkündür. Ancak bunun için atak tespit sisteminin ağ yapılandırması sırasında en uygun yere konumlandırılmış olması gerekir[10].

Uygulama kayıt dosyaları; uygulama katmanında gerçekleşen atakları tespit etmekte kullanılırlar. Bu veri kaynağı, diğer iki kaynaktan daha kolay elde edilmesiyle birlikte sağladığı atak tespit oranı sınırlıdır. Bilgi kaynağı olarak uygulama kayıt dosyalarını kullanan bir STS, web tabanlı saldırıları tespit etmek için geliştirilen bir yazılım olan WebWatcher'ın uygulama log dosyalarını kullanabilir[10].

2.4. STS Modelleri

Bilgisayarlara yapılan saldırıları önlemek için kullanılan STS 'ler belirlenen bir politika çerçevesinde yalıtım sağlayan bileşenlerdir. STS'ler bilgisayar sistemlerinde veya bilgisayar ağlarında oluşan olayları otomatik olarak belirleyerek güvenlik sorunları oluşturabilecek durumları analiz etmeli ve sonuçları bildirmelidir.

STS'ler analiz yaklaşımlarına göre kural-temelli (imza-temelli) ve anormallik-temelli (davranış-temelli) olmak üzere ikiye ayrılmaktadır[10].

2.4.1. İmza temelli yaklaşım : SNORT

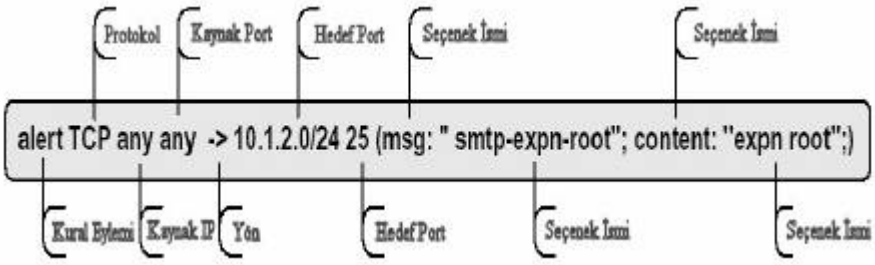
Kötüye kullanım detektörleri sistem etkinliğini analiz eder, olayları veya bilinen bir saldırıyı tanımlayan olayların önceden tanımlanmış modeliyle benzeşen olaylar dizisini arar. Bilinen saldırıları modelleyen şablona imza adı verildiğinden dolayı bu yönteme "imza-temelli tespit" de denir [22,23,24].

Snort, IP ağları üzerinde kötüye kullanım tespiti ve gerçek-zamanlı trafik analizi yapabilen yakın zaman önce geliştirilmiş bir ağ-temelli saldırı tespit sistemidir

[25,26]. Kaynak kodu ile birlikte dağıtılan ve kısa sürede son derece popüler olan Snort yazılımının artan işlevselliği ve becerileri nedeniyle pek çok firma Snort temelli ticari STS çözümleri geliştirmekte ve satmaktadır. Snort günümüzde çok sayıda büyük kuruluş tarafından tercih edilen bir STS haline gelmiştir. Snort, şablon eşleme tekniğine dayanır ve içerik analizi yapar. Önceden tanımlanmış kötüye kullanım kurallarına göre alarm verir. Snort kural tabanlıdır ve kullanılan dil yeni kurallar tanımlamaya elverişlidir. Bu sayede kullanıcılar, var olan kuralları kendilerine göre düzenleyip kendi kurallarını ekleyebilmektedirler. Gerekli tüm özelleştirmeler düz metin dosyaları üzerinde yapılmaktadır[10].

Snort kural tanımlama dilinde her bir kural iki kısımdan oluşur: Kural başlığı ve Kural seçenekleri.

Kural başlığı beş bölümdür; kural tepkisi (saldırı tespit edildiğinde verilecek tepki), uçlar arasındaki kaynak ve hedef bilgisi (protokole özgü kaynak ve hedef IP adresleri ve port numaraları), trafik akış yönü bilgisi ve protokol türü (TCP, UDP veya ICMP).



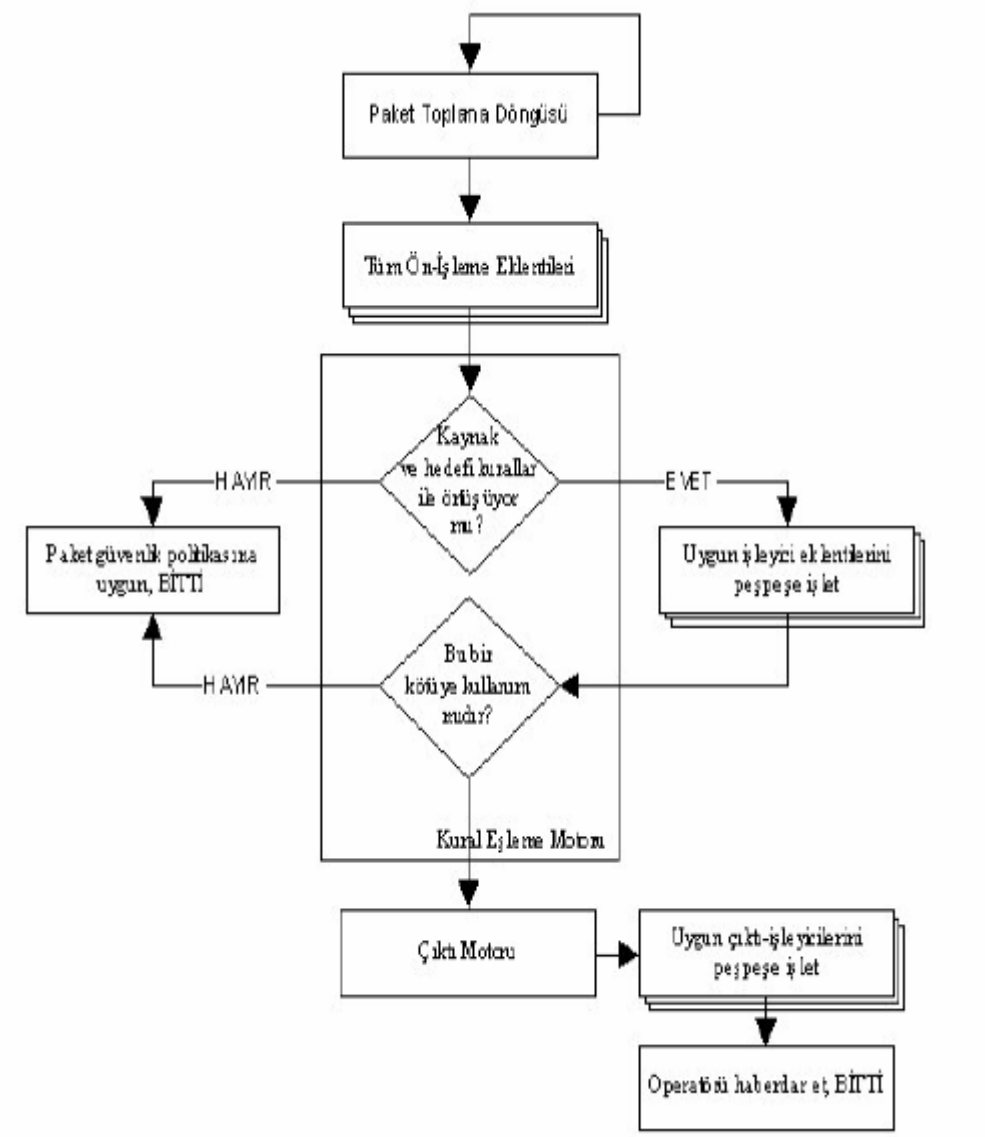
Şekil 2.3. Snort Kural Yapısı [27]

Kural seçenekleri, belirtilen kötüye kullanım işleminin gerçekleşip gerçekleşmediğine karar vermede kullanılan çeşitli koşullardan oluşur. Örnek bir Snort kuralı Şekil-1'de verilmektedir. Her kuralın ilk alanı eylemdir. Şekil-2.3.'deki kuralda seçilen eylem 'alert'tir. Bunun anlamı kuralda belirtilen kriterle eşleşen bir giriş geldiğinde, bir alarm oluşturulacağıdır. Sonraki alan protokol bilgisini göstermektedir. Örnek kuraldaki protokol TCP'dir. Üçüncü ve dördüncü alanlar

kaynak adreslerden oluşur; ilk kısım IP adresi, ikinci kısım kaynak port numarasıdır. Eğer bu alanda “any any” şeklinde değerler bulunuyorsa bu, paketlerin herhangi bir IP adresinden ve herhangi bir TCP portundan gelebileceğini gösterir. Beşinci alan bilgi akış yönünü göstermektedir. Altıncı ve yedinci alanlar hedef adreslerden oluşur. Örnek kuraldaki hedef IP adresi 10.1.2.0/24 olarak verilmiştir ve ilgili bir ağdaki bütün IP adreslerini eşler. Bu örnekte TCP hedef portu 25 olarak ayarlanmıştır. 25 numaralı port, Simple Mail Transfer Protocol (SMTP) için kullanılmaktadır [27]. Hedef adresi takiben parantez içinde seçenekler listesi bulunmaktadır. Her seçenek bir seçenek ismi, varsa seçenek değeri ve seçeneğin bitişini gösteren bir noktalı virgülden oluşur. Şekil-2.3.’de gösterilen kuralda ilk seçenek ‘msg’dir ve eylem mesajını belirtmek için kullanılmıştır. İkinci seçenek olan ‘content’, bir şablon eşleşme kriterini belirtmektedir. Örnekte girişin veri alanı kısmında ‘expn root’ karakter dizisi aratılmaktadır. TCP veri alanında bu karakter dizisine rastlandığı zaman, koşul gerçeklenmiş olur. Bu kriterlerden birinin bile sağlanmaması halinde alarm üretilmez. Snort’un veri yakalama motoru, Lawrence Berkeley National Laboratuarında geliştirilen libpcap paket yakalama kütüphanesini [28] kullanır. Sistem on taneden fazla UNIX türevini ve MS-Windows’u desteklemektedir. Ayrıca libpcap kütüphanesini kullanmasından ötürü, farklı ağ ortamlarında çalışabilmektedir. Snort’un başlıca beş bileşeni vardır [29]:

1. Paket yakalama / ayıklama (dekoder) motoru
2. Önişlemci eklenti-yazılımları (plug-ins)
3. Tespit motoru
4. Kayıt ve Alarm Sistemi
5. Çıkış eklenti-yazılımları

Snort’un paketleri işlemesi Şekil-2.4.’de gösterilmektedir [29].



Şekil-2.4. Snort'un Paketleri İşleme Döngüsü [29]

2.4.2. İstatistiksel yaklaşımla anormallik tespiti : PHAD

Bu çalışmada istatistiksel temelli bir saldırı tespit sistemi olan anormallik tespit yaklaşımlarından Paket Başlığı Davranış Tespiti (PHAD) üzerinde durulmuş ve modelin deneysel olarak iyileştirilmesi sağlanarak Snort'a önişlemci olarak eklenmiştir[30].

PHAD diğer ağ temelli davranış tespit sistemlerinden iki yönüyle farklıdır. Bu farklardan ilki PHAD'ın kullanıcı davranışlarından ziyade protokolleri

modellemesidir. Çünkü birçok saldırı protokol uygulamalarındaki açıklardan faydalanır ve ancak sıra dışı girdi ve çıktıların tespit edilmesi ile anlaşılabilir. Diğer fark da PHAD'ın ağ istatistiklerinin kısa süre içerisindeki hızlı değişiminden uyarılan zaman-temelli bir model kullanmasıdır. Bu modellerin temel özellikleri şöyle özetlenebilir [30]:

Protokol modeli: Davranış tespiti sistemlerinin çoğu, yetkili ve yetkisiz kullanıcıları ayırt etmek için tasarlanmıştır. Örneğin yetkili bir kullanıcı ağ topolojisini bildiği için port taramasının yaptığı gibi var olmayan sunuculara ve hizmetlere bağlanmaya çalışmaz. Ayrıca şifre isteyen sunucular (TelNet, FTP, POP3, ...) kaynak IP adresleri ile tanımlı yetkili istemcilerden ve/veya günün belirli zamanlarında gelen isteklerden normal davranışı anlar. O halde bu hizmetlere erişmeye çalışan farklı kaynak adresleri için yetkisiz erişim uyarısı verilebilir. Bu tür STS'ler kullanıcı modellemeye dayanmaktadır. Diğer bir yaklaşım ise PHAD'ın da benimsediği protokol modellemesidir. Bilindiği gibi birçok saldırı, protokollerin uygulamalarındaki açıklarından yararlanır. Örneğin bu tip saldırılar sendmail, imap ve named protokollerin hatalı uygulamalarını kullanabilirler. Teardrop ve ping of death saldırıları, IP Protokolünün hatalı uygulamalarını deşerler. Bu saldırılar sırasında ağdaki etkinlik bir protokol anormalliğini işaret edebilir. Protokol anormalliklerindeki diğer bir etmen saldıran kodun hatalarından gelir. Aynı sunucu veya istemciyi yazan programcının protokolün tüm ayrıntılarını doğru uygulayamaması gibi saldırgan da her şeyi doğru yapamaz. Saldırganın, TTL, başlık uzunluğu, doğrulama biti, parçalanma göstergesi gibi IP başlık alanlarını doldururken yaptığı çeşitli hatalar veya alışılmamış uygulamaları ağda olağan dışılıklara yol açabilir[30].

Zaman-temelli model: Birçok ağ olayı kendine benzerdir ve değişik periyotlarda kendini tekrarlayan bir yapıdadır. Ağ olayları birbirinden bağımsız değildir. Tersine uzun vadede bir bağımlılık vardır [30].

Zaman-temelli modeli anormallik tespitine uygulamak için eğitim ve test aralıklarında "tn/r" ile bir anormallik skoru hesaplanır; burada n (her bir alan için

uygun türden paketlerin sayısı) ve r (normal değerlerin sayısı) eğitim aralığı boyunca sayılır ve t en son anormalliğin görüldüğü zamandan bu yana geçen süredir [30].

Bu modelde eğitim aşamasında normal olan değerler bulunarak test sırasında normalden sapmalar belirlenir. Örneğin şu eğitim ve test verileri için:

Eğitim Safhası (zaman 0-19):00000000000000001111
Test Safhası (zaman 20-24):01223. Eğitim sırasında izin verilen değerler kümesi kayıt edilir{0,1}; bu kümenin eleman sayısı, r = 2, ve gözlem sayısı, n = 20'dir. Eğer gözlemler 0 ile başlayan birim aralıklarla yapılırsa eğitim sırasında görülen en son değer olan "1", 16 zamanında gerçekleşir ve zaman değeri test aşamasında kullanılmak üzere tutulur. Test safhasındaki 22, 23, ve 24. zamanlardaki "2", "2", ve "3" anormalliktir çünkü bunlar eğitim setinde bulunmamaktadır. Görülen ilk "2"nin anormallik skoru tn/r = (22-16)*20/2 = 60 olarak hesaplanır[30].

İkinci görülen "2"nin anormallik skoru (23-22)*20/2 = 10 olarak hesaplanır. "3"ün anormallik skoru (24- 23)*20/2 = 10 olarak hesaplanır. "0" ve "1" in anormallik skorları 0'dır çünkü bunlar eğitim safhasında en az bir defa görülmüştür. Bu örnekteki hesaplar tek bir değer baz alınarak yapılmıştır. Birden fazla anormallik özelliğine sahip bir örnek(paket) için anormallik skoru Σ tn/r dir ve burada toplam, anormal özelliklerin üzerinde hesaplanır[30].

PHAD'ın anormallik tespitinde kullandığı özellikler: PHAD, ağ paketlerini tespit etmek için kullanılan bir zaman-temelli protokoldür. Her paket için bir skor hesaplar ve gelen ve giden trafik arasında ayrım yapmaz. Paket başlığındaki ilk 4 bayt alanlarına karşılık gelen 33 özelliği modeller. Bir bayttan küçük olan alanlar (TCP bayrakları gibi) bir bayt içinde birleştirilir. 4 bayttan büyük olan alanlar (6 baytlık Ethernet adresleri gibi) bölünür. Özellikler şunlardır[30]:

1. Ethernet Başlığı (bütün paketlerde yer alır)
2. IP Başlığı
3. TCP Başlığı
4. UDP Başlığı
5. ICMP Başlığı

PHAD, anormal özelliklerin üzerinde $\Sigma\ tn\ /\ r$ kullanarak bir anormallik skoru hesaplar[30].

2.5. STS'lerde Kullanılan Teknikler

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 [19]. STS'lerde kullanılan tekniklerden bazıları aşağıda açıklanmış ve Tablo 2.1'de sunulmuştur.

Veri Madenciliği: Veritabanındaki saklı olayları ortaya çıkarmak için yapılan bilgi açılımıdır. Paternleri ve veriler arasındaki ilişkileri bularak kural çıkarmak için kullanılır. Bu şekilde, hesap izlerini kullanarak normal kullanıcı aktiviteleri tanımlanır [3].

Kural Tabanlı (Rule Based) Sistemler: Sistem trafiğini inceleyip kurallar oluşturur ve saldırı tespiti sırasında belirlenen kurallara göre davranışlar sınıflandırılır [4].

Açıklayıcı 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 [5]. 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 [21].

Eşik Değeri Tespiti: Bu model oluşturulurken spesifik olayların tekrarlama sayısı ve spesifik zaman periyodu dikkate alınır. Karşılaşılan en büyük sorun; eşik değerinin belirlenmesi ve spesifik olaylar için pencere boyutunun belirlenmesidir. Örnek olarak; yanlış girişler, giriş/çıkış hata sayısı veya silme sayıları verilebilir. Tek başına pek güçlü değilse de büyük STS'lerde alt bileşen olarak kullanılır[21].

Durum Geçiş Analizi: 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].

Uzman Sistemler: Belirli bir alanda sadece o alan ile ilgili bilgilerle donatılmış ve problemlere alanda uzman bir kişinin getirdiği şekilde çözümler getirebilen bilgisayar programları olarak tarif edilebilir. Sızma belirleme sistemlerinin ilkleri kural-tabanlı (rule based) uzman sistemlerdir [18].

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 [18].

STS'lerde kullanılan teknikler, literatür gözden geçirilerek Tablo 2.1'de özetlenmiştir.

Tablo 2.1. STS'lerde kullanılan tekniklerin karşılaştırılması [26]

Teknik	Tespit Yaklaşımı	Kullanılan Bilgi Kaynakları	Bilinen Ataklar	Bilinmeyen Ataklar	Performans
İstatistiksel	Anormallik	Denetim verisi, kullanıcı profili	Evet	Evet	Orta
Veri Madenciliği	Anormallik	Denetim verisi, bilgi tabanı	Evet	Evet	Orta
Durum Geçiş Analizi	Kötüye Kullanım	Denetim kayıtları, bilinen atak örneklerinin durum geçiş diyagramları	Evet	Hayır	Yüksek
Dosya Kontrol	Anormallik	Sistem dosyaları	Evet	Evet	Yüksek
Örüntü Eşleme	Kötüye Kullanım	Denetim kayıtları, saldırı imzaları	Evet	Hayır	Yüksek
Protokol Analizi	Anormallik	Denetim kayıtları, bir protokolün normal kullanım modeli	Evet	Evet	Düşük
Tuş Vuruşu İzleme	Kötüye Kullanım	Bilinen saldırıların bilgi tabanı, Tuş vuruşları	Evet	Hayır	Yüksek

2.6. Saldırı Tipleri

Bilgisayar sistemlerine yapılan saldırılar birçok araştırmacı tarafından çeşitli şekillerde gruplandırılmıştır [31, 32]. Saldırganların sürekli kendilerini yenilemeleri ve bilgisayar sistemlerinde var olan açıkları tespit etmeleri nedeniyle saldırı tiplerinin çeşitliliği çok artmıştır. Bu çalışmada STS'lerin gelişimi sırasında önemli bir yeri olan DARPA veri kümelerinin oluşturulması sırasında belirlenen ve hala geçerliliğini koruyan saldırı tipleri esas alınmıştır. Lincoln Laboratuarlarında yapılan bu çalışmada, saldırılar bilgisayar sistemine yapılan atak türlerinin kullandıkları yöntemlere göre dört gruba ayrılmış ve DoS, U2R, R2L ve Probing olarak adlandırılmışlardır [33]. Bu saldırılar, aşağıda verilen alt başlıklarda açıklanmıştır.

2.6.1. Hizmet aksattırma saldırıları

Hizmet aksattırma (DoS) saldırıları, saldırılan sistemlerin hizmetlerini engellemek amacıyla yapılırlar. Genellikle hizmetin verilmesi engellenmek istenildiğinde sisteme cevap verebileceğinden çok istek gönderilmesi ile gerçekleştirilir. DARPA veritabanında isimlendirildiği haliyle, en çok bilinen DoS saldırı tipleri, SYN flood, Smurf, UDPstorm, Pingflood, Neptune, Mailbomb gibi saldırılardır [34].

2.6.2. U2R saldırıları

U2R saldırıları, kullanıcıların normal yetkilere sahip olan kendi hesaplarından oturum açtıktan sonra yönetici yetkisine ulaşmaya çalışmasıdır. Bu şekilde sistem üzerinde istedikleri bilgilere erişebilirler. DARPA veritabanında isimlendirildiği haliyle, en çok bilinen U2R saldırı tipleri Eject, Ffbconfig, Fdformat, Loadmodule, Perl gibi saldırılardır [34].

2.6.3. R2L saldırıları

Bu saldırı tipinde saldırgan saldırdığı makineye ağ üzerinden paketler yollayarak açıklardan yararlanmaya çalışır. Bu konuda birçok araç olması ve bu araçlara kolay erişilebilir olması nedeniyle, sistemde var olan açıklar saldırgandan önce belirlenip kapatılmamışsa oldukça etkili ve kolay bir saldırı yöntemidir. DARPA veritabanında isimlendirildiği haliyle, en çok bilinen R2L saldırı tipleri Dictionary, Guest, Imap, Named, Sendmail gibi saldırılardır [34].

2.6.4. Probing saldırıları

Probing saldırısı, ağı veya bilgisayarı tarayarak zayıflıkları tespit etmek ve sistem yapısı ile ilgili genel bir bilgiye ulaşmak için yapılmaktadır. Sistem hakkında detaylı bilgi edinildikten sonra nasıl bir saldırı yapılması gerektiği belirlenir. Probing saldırısı için kullanılan araçlar aynı zamanda güvenlik uzmanları tarafından sistemin güvenliğinin test edilmesi için de kullanılan araçlardandır. DARPA veritabanında

isimlendirildiği haliyle, en çok bilinen Probe saldırı tipleri, Ipsweep, Mscan, Nmap, Saint, Satan gibi saldırılardır [34].

2.7. STS'lerin Başarı Kriterleri

Saldırı tespit sistemleri, hızlı gelişimi ve sağladığı güvenlik desteği nedeniyle büyük ağlarda standart bir araç haline gelmiştir. Her geçen gün yenileri eklenen STS'lerin, diğerlerine oranla ne kadar başarılı olduğu ya da gereksinimlere ne kadar cevap verdiği sorusunu cevaplayacak net bir cevap yoktur. STS'lerin test edilmesi için günümüze kadar birçok çalışma yapılmıştır, ancak bu çalışmalar genellikle bazı STS'lerin karşılaştırılmasından ibarettir. Bu nedenle saldırı tespit sistemlerinin başarılarını ölçmek için genel bazı kurallar ve hesaplanabilir değişkenler belirlenmiştir. Mell ve arkadaşlarının yaptığı bir çalışmada STS'lerin başarı kriterleri belirlenmiştir [35]. Bu kriterler, ilgili kaynak temel alınarak sırasıyla aşağıda alt başlıklarda açıklanmıştır.

2.7.1. Kapsam

Bu ölçüt, bir STS'nin ideal koşullarda hangi atakları tespit edebildiği olarak tanımlanır. Kötüye kullanım tespitine dayalı STS'lerde kapsam, imza sayısı ve bu imzaların standart isim düzeni ile haritalanmasını içerir. Anormallik tespiti yapan STS'lerde, hangi atakların özel metodolojilerle tespit edilebildiği bilinen ataklar kümesinin dışında kaldığına karar vermek gerekir. Ataklarda değişiklikler yapılarak türetilen ataklar bu ölçütü zorlaştırır[35].

2.7.2. Yanlış alarm olasılığı (probability of false alarms)

Bu ölçüt, verilen çevrede belli bir zaman aralığında STS tarafından üretilen yanlış pozitif oranını tanımlar. 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[35].

2.7.3. Tespit etme olasılığı (probability of detection)

Bu ölçüt, verilen çevrede belli bir zaman aralığında STS tarafından doğru tespit edilebilen atak oranını tanımlar. Bir STS'nin var olan saldırıların kaçını yakaladığı önemli bir parametredir. Fakat STS'ler tespit etme oranını artırırken, yanlış alarm oranını da çok yükseltmemelidir[35].

2.7.4. Daha önce görülmemiş atakları tespit edebilme

Bu ölçüt, bir STS'nin daha önce hiç karşılaşmadığı bir atağı tespit etmede ne kadar iyi olduğunu gösterir. Bu sadece anormallik tespitine dayalı sistemler için geçerlidir. İmza temelli sistemler yeni gelen hiçbir saldırıyı tanıyamazlar ve birçok ticari STS imza tanıma yöntemine göre geliştirilmiştir[35].

2.7.5. Bir atağı tanımlayabilme

Saldırının varlığını tespit etmek ve saldırının tipini söyleyebilmek iki ayrı kavramdır. STS'lerin ilk odaklandığı konu doğal olarak saldırıların varlığını tespit etmektir. Genelde saldırı tipi, saldırı olduktan sonra ağ yöneticisi tarafından çeşitli kayıtlar incelenerek ortaya çıkartılır. Fakat yine de çok genel sınıflandırmalar yapabilen STS'ler mevcuttur[35].

2.8. STS Veri Kümesi Tasarımı

Bir STS tasarlanırken ele alınması gereken en önemli hususlardan birisi kullanılacak olan veri kümesidir. STS veri kümesi, geliştirilecek olan STS'nin eğitim ve test aşamalarında saldırıyı tanımlamak için gereken ve içerisinde saldırı verileri içeren ağ

paketleri veya günlük kayıtlardan elde edilen veriler bütünüdür. STS çalışmalarının bazılarında, uygulama geliştiriciler, veri kümelerini kendileri oluşturmuşlardır. Ancak bu oldukça zahmetli, bir o kadar da maliyetli bir çalışmadır. STS geliştiricilerinin, sınırlı olan işgücü ve zamanı, yeni ve farklı teknikler kullanmak yerine, veri kümesi oluşturmaya harcaması yüklerini oldukça artırmıştır. Bunun yanında, geliştirilen sistem kendilerinin tanımladığı veri kümesi üzerinde yüksek başarı oranları gösterse de, bu sonuçlar gerçeklikten uzaktır. Hem STS'lerin tasarım ve uygulama çalışmalarının hızlandırılması hem de standart ve objektif bir test ortamı oluşturulabilmesi için güncel ve standart hale gelmiş STS veri kümelerine ihtiyaç duyulacağı aşikardır.

Literatüre bakıldığında, bu ihtiyaca yönelik bazı çalışmalar yapılmıştır [5,7,33]. Ancak 1998–2000 yılları ile sınırlı kalan bu çalışmaların, o yıllarda çok değerli sonuçlara ve değerlendirmelere imza atmış olsalar da güncelliklerini kaybettikleri görülmektedir. Bu nedenle STS'ler için; yeni, gerçeğe uygun ve güncel saldırıları da kapsayan veri kümelerinin oluşturulması gerekmektedir.

Geliştirilen sistemin detaylarına geçmeden önce bu konuda daha önce yapılmış ve halen kullanılmaya devam eden IDEVAL ve KDD'99 veri kümeleri hakkında bilgiler verilmiştir.

2.8.1. IDEVAL (Intrusion detection evaluation)

Lincoln Laboratuarları, Bilgi Sistemleri Teknolojisi (Information Systems Technology - IST) grubunun, DARPA (Defence Advanced Research Projects Agency) ve AFRL (Hava Kuvvetleri Araştırma Laboratuarı - Air Force Research Projects Agency) desteğiyle yürüttüğü çalışmaların sonucunda, saldırı tespit sistemlerinin değerlendirilmesi ve karşılaştırılması için IDEVAL (Intrusion Detection Evaluation) adı verilen ilk standart veri kümesi gövdesi oluşturulmuştur[36].

Bu çalışmada, Hava Kuvvetleri Araştırma Laboratuarı ile birlikte saldırı tespit sistemlerinin, ilk düzenli, tekrarlanabilir ve önemli-istatistiksel ölçütleri belirlenmiştir. Bu ölçütler, test altındaki her sistem için tespit olasılığı ve yanlış alarm olasılığını kapsamaktadır. Bu ölçütler, mevcut saldırı tespit sistemleri tasarımına, yeni yapılacak araştırma çalışmalarına yön verme ve objektif bir kalibrasyon sağlama gibi önemli hususları desteklemektedir[36].

2.8.1.1. IDEVAL veri kümeleri

Lincoln Laboratuarlarında gerçekleştirilen, saldırı tespit sistemlerini değerlendirme çalışmaları 1998-2000 yıllarında yürütülmüştür. Oluşturulan gerçek zamanlı olmayan (off-line) veri kümeleri, geniş atak ve ağ trafiği örnekleri ile araştırmacılar tarafından kullanılmaya uygun hale getirilmiştir[33].

DARPA saldırı tespit değerlendirme çalışmaları 1998 ve 1999 DARPA IDEVAL veri kümeleri olmak üzere iki veri kümesi elde edilmiştir [36, 37].

Bu veri kümelerinin haricinde 2000 yılında özel senaryolar için 3 farklı veri kümesi daha oluşturulmuş ve bu veri kümeleri araştırmacıların kullanımına sunulmuştur [38]. Bu tez çalışmasında, günümüzde halen kullanılmaya devam eden ve ağ trafiğinin dinlenmesi ile oluşturulan,DARPA veri kümelerinden meydana gelen KDD ’99 veri kümeleri kullanılmıştır.

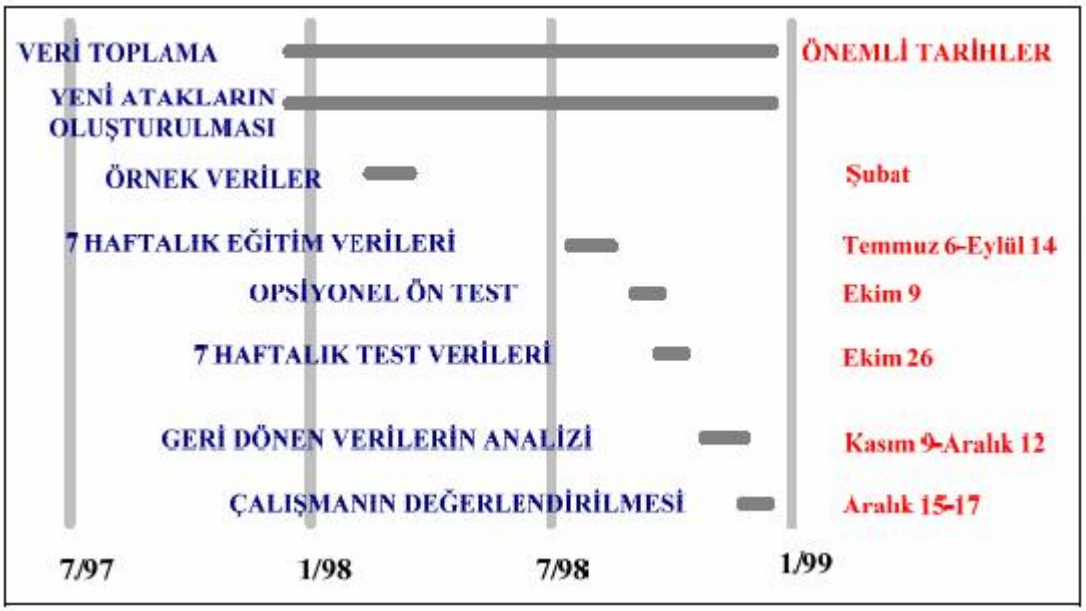
DARPA 2000 veri kümeleri sadece özel senaryoları içerdiğinden ve bu çalışmanın amacına uygun olmadığından kullanılmamıştır.

IDEVAL 1998 ve 1999 veri kümeleri, gerçek zamanlı ve gerçek zamanlı olmayan değerlendirme olmak üzere iki kısımdan oluşmaktadır. Lincoln Laboratuarlarında oluşturulan veri kümeleri, STS’lerin gerçek zamanlı olmayan (off-line) değerlendirilmesinde kullanılır. IDEVAL 1998 ve 1999 değerlendirmesinin gerçek zamanlı kısmı hakkında ayrıntılı bilgi AFRL’den elde edilebilir[33].

2.8.1.2. 1998 DARPA

1998 DARPA veri kümesi, 1998 IDEVAL veri kümesinin gerçek zamanlı olmayan değerlendirme kısmı için geliştirilmiştir [36].

1998 veri kümesinin oluşturulma aşamalarını gösteren zaman süreçleri Şekil 2.5'de gösterilmiştir [36].



Şekil 2.5. 1998 DARPA veri kümesinin oluşturulma süreçleri [36]

Şubat 1998'de 10 dakikalık örnek bir veri kümesi oluşturulmuştur. Bu veri kümesi STS'lerin nasıl skorlandığını göstermek amacıyla, sadece bazı örnek saldırı tiplerini içerir. Bunlar; guess, ping-sweep, port-scan, phf, rlogin, rsh ve rcp'dir[36]. Oluşturulan örnek veri kümesi EK-1'de sunulmuştur.

Örnek veri kümesinden sonra, Mayıs 1998'de eğitim verilerinin alt kümesi olan 4 saatlik veri kümesi oluşturulmuştur. Bu veri kümesinin amacı, bazı ilk verilerin oluşturularak, verinin doğru okunabildiğini ve değerlendirme için yeterli bilginin sağlanabildiğini gösterebilmektir. Oluşturulan veri kümesi, gerçek eğitim veri kümesinin ilk 4 saati değildir ve ilk hafta elde edilen verilere benzer olmasına

rağmen özdeşi de değildir. 4 saatlik veri kümesi, örnek 10 dakikalık veri kümesindeki saldırıların haricinde 2 farklı saldırı içerir, bunlar, EK-2'de yer alan dict ve eject'dir[36].

Eğitim veri kümesi elde edildikten sonra, 2 hafta boyunca üretilen verilerden test veri kümesi oluşturulmuştur. 1998 test veri kümesinde üretilen atak sayısı 38 tanedir. Bu ataklar Bölüm 2.5'de belirtilen saldırı tiplerine göre 4 ana gruba ayrılmıştır. Atakların hangi atak tipine ait olduğunun gösterildiği Tablo 2.2'de, 38 ataktan 14 tanesi sadece test veri kümesinde bulunmaktadır. Tablo 2.2'de sadece test veri kümesinde bulunan bu ataklar altı çizili olarak gösterilmiştir[36].

Tablo 2.2. Test veri kümesinde yer alan ataklar [36]

	Solaris Sunucu	SunOS	Linux	Cisco Yönlendirici
DoS	back neptune ping of death smurf syslog land apache2 mailbomb process table UDP storm	back neptune ping of death smurf land apache2 mailbomb process table UDP storm	back neptune ping of death smurf teardrop land apache2 mailbomb process table UDP storm	snmp getattack
R2U	Dictionary ftp-write guest phf ftp-write httptunnel xlock xsnoop	dictionary ftp-write guest phf httptunnel xlock xsnoop	dictionary ftp-write guest imap phf httptunnel named sendmail xlock xsnoop	
U2R	eject ffbconfig fdformat ps	loadmodule ps	perl xterm	
PROBE	ip sweep nmap port sweep satan mscan saint	ips weep nmap port sweep satan mscan saint	ip sweep nmap port sweep satan mscan saint	ip sweep nmap port sweep satan mscan saint

2.8.1.3. 1999 DARPA

DARPA 1999 veri kümesi, 1998'de geliştirilen veri kümesinin değiştirilmesi ile oluşturulmuştur. Yeni ve farklı bazı atakların da veri kümesine dahil olması için böyle bir değişikliğe ihtiyaç duyulmuştur[37].

Bu değişikliklerin bazıları şu şekildedir;

1. Eğitim verilerinde atağın hiç olmadığı günler sağlamak,
2. Eğitim ve test verilerinde daha az örtüşen ataklar gibi sıralanabilir.

Geliştirilen DARPA veri kümelerinin asıl amacı olan, STS'lerin değerlendirilmesi işlemleri, toplanan ağ trafiği ve günlük kayıtlar (audit data) kullanılarak, gerçek zamanlı olmayan değerlendirme aşamasında gerçekleştirilmiştir[37].

Gerçek zamanlı test için STS'ler AFRL'ye gönderilmiştir. Bu sistemler AFRL ağ test yatağına eklenmiş ve gerçek zamanlı olarak, normal aktiviteler arasında atak durumları belirlenmeye çalışılmıştır[37].

2.8.2. KDD'99

KDD'99 veri kümesi 1999 yılında DARPA veri kümesinin bazı önişlemlerden geçirilmesi ile elde edilmiş 41 özellikten oluşur. Bu veri kümesinin tasarlanmasının amacı, son yıllarda farklı tekniklerle gerçekleştirilmek istenen STS'ler için eğitim ve test işlemlerinde kolaylık sağlamaktır. Her ne kadar STS'ler için veri kümesi problemi DARPA ile çözülse de uygulanan bu yeni tekniklerde kullanabilmek için çok fazla önişlem gerektirir. KDD'99 veri kümesi ile eğitim ve test sonuçlarının daha hızlı alınabilmesi ise araştırmacılar için büyük bir avantajdır[5].

KDD'99 veri kümelerinde, 9 temel ve 32 adet türetilmiş olmak üzere toplamda 41 tane özellikten oluşan bir özellik haritası çıkarılmıştır. Bu 41 özellik 3 temel kategoriye ayrılarak ifade edilmiştir[5].

1. İçerik özellikleri (content features)
2. Sunucu tabanlı trafik özellikleri (host-based traffic features)
3. Zamana bağlı trafik özellikleri (time-based traffic features)

Tablo 2.3, Tablo 2.4 ve Tablo 2.5'de, sırasıyla bu 3 kategori ve kategoriler içerisindeki veri özellikleri gösterilmiştir.

Tablo 2.3. İçerik özellikleri [5]

Özellik adı	**Tanım**	**Tip**
duration	Bağlantı uzunluğu	sürekli
protocol_type	Protokol tipi	ayrık
service	Servis tipi	ayrık
src_bytes	Kaynaktan hedefe veri	sürekli
dst_bytes	Veri byte sayısı	sürekli
flag	Bayrak	ayrık
land	Kaynak ve hedef IP aynı ise 1 değilse 0	ayrık
Wrong_fragment	Yanlış parçalama	sürekli
urgent	Acil paket sayısı	sürekli

İçerik özellikleri, sadece TCP bağlantılarından alınan temel özelliklerdir. Bu özelikleri elde etmek, ağ trafiği verileri üzerinde önişlem yapılması gerekmediğinden, diğer kategorilere göre daha kolaydır[5].

Tablo 2.4. Sunucu tabanlı trafik özellikleri [5]

Özellik adı	Tanım	Tip
hot	"hot" göstergesi	sürekli
num_failed_logins	Hatalı giriş sayısı	sürekli
Logged_in	Giriş başarılı ise 1 değilse 0	ayrık
num_compromised	Gizliliğin ihlal edilme sayısı	sürekli
root_shell	"Root Shell" elde edildiyse 1 değilse 0	ayrık
su_attempted	"Su Root" komutu girildiyse 1 değilse 0	ayrık
num_root	"Root" erişim sayısı	sürekli
num_file_creations	Dosya oluşturma işlemleri sayısı	sürekli
num_shells	Shell promptlarının sayısı	sürekli
num_access_files	Kontrol dosyalarına erişim işlemleri sayısı	sürekli
num_outbound_cmds	ftp oturumunda giden komut sayısı	sürekli
is_hot_login	Giriş "hot" listesindeyse 1 değilse 0	ayrık
is_guest_login	Giriş "guest" ise 1 değilse 0	ayrık

Sunucu tabanlı trafik özellikleri, etki alanı (domain) bilgisi ile ortaya çıkan bağlantı içerik özellikleridir.

Zamana bağlı trafik özellikleri, "aynı sunucu" ve "aynı servis" özellikleri kullanılarak çıkarılan özelliklere verilen isimdir. "Aynı sunucu" özellikleri, son iki saniye içerisinde aynı sunucuya yapılan bağlantıların gözden geçirilmesi ile elde edilir. Benzer olarak "aynı servis" özellikleri son iki saniye içerisinde aynı servise yapılan bağlantıların gözden geçirilmesi ile elde edilir[5].

Tablo 2.5. Zamana bağlı trafik özellikleri [5]

Özellik adı	Tanım	Tip
count	Aynı sunucuya önceki iki bağlantıyla aynı bağlantıların sayısı	sürekli
serror_rate	"SYN" hata bağlantılarının yüzdesi	sürekli
rerror_rate	"REJ" hata bağlantılarının yüzdesi	sürekli
same_srv_rate	Aynı servise bağlantıların yüzdesi	sürekli
diff_srv_rate	Farklı servislere bağlantıların yüzdesi	sürekli
srv_count	Aynı servise önceki iki bağlantıyla aynı bağlantıların sayısı	sürekli
srv_serror_rate	"SYN" hata bağlantılarının yüzdesi	sürekli
srv_rerror_rate	"REJ" hata bağlantılarının yüzdesi	sürekli
srv_diff_host_rate	Farklı servislere bağlantıların yüzdesi	sürekli

DARPA veri kümelerinin, belirli önişlemlerden geçirilmesi ile oluşturulan KDD'99 veri kümesi toplamda 38 farklı atak içerir. Bunlardan 24 tanesi eğitim veri kümesinde iken, test veri kümesi, eğitim veri kümesinde bulunmayan 14 farklı atak tipini daha içerir[5].

KDD'99 eğitim veri kümesinde bulunan 24 atak ve bu atakların ait oldukları saldırı tipleri ve saldırıları veri kümesinde bulunan örnek sayıları Tablo 2.6'de gösterilmiştir.

Sadece test veri kümesinde yer alan ve etiketlenmiş KDD'99 dosyasından alınan 14 farklı atağa ait saldırı tipi ve örnek sayıları Tablo 2.7'da gösterilmiştir.

Tablo 2.6. KDD'99 veri kümesinin %10'luk kısmından alınan saldırı örneklerinin sayıları [5]

Atak	Örnek sayısı	Kategori
smurf.	280790	dos
neptune.	107201	dos
back.	2203	dos
teardrop.	979	dos
pod.	264	dos
land.	21	dos
normal.	97277	normal
satan.	1589	probe
ipsweep.	1247	probe
portsweep.	1040	probe
nmap.	231	probe
warezclient.	1020	r21
guess_passwd.	53	r21
warezmaster.	20	r21
imap.	12	r21
ftp_write.	8	r21
multihop.	7	r21
phf.	4	r21
spy	2	r21
buffer_overflow.	30	u2r
rootkit.	10	u2r
loadmodule.	9	u2r
perl.	3	u2r

Tablo 2.7. Eğitim kümesinde yer almayan ataklar [5]

Atak	Örnek sayısı	Kategori
apache	794	dos
mailbomb	5000	dos
processtable	759	dos
udpstorm	2	dos
mscan	1053	probe
saint	736	probe
httptunnel.	138	r21
named	17	r21
sendmail	17	r21
snmpgetattack	1040	r21
xlock	9	r21
xsnoop	4	r21
ps	16	u2r
xterm	13	u2r

2.9. Geliştirilen STS Modelleri

STS sınıflandırıcı öğrenme teknikleri konusu üzerine düzenlenen KDD'99 organizasyonunda bilgisayar ağlarından bilgisayarlara yapılan atakların sınıflandırılması için,doğru ve yanlış karar mekanizmalarını içeren geliştirilmiş öğrenme modelleri birbirleriyle yarışmışlardır.Bu organizasyonda kullanılan eğitim ve test veri setleri için KDD'99 veri setleri kullanılmıştır. Başarılı olarak görülen ilk 3 model şu şekildedir[39];

Birincilik ödülünü Avustralya Yapay Zeka Araştırma Enstitusü 'den Dr.Bernhard Pfahringer kazanmıştır. En yüksek performansın elde edildiği bu modelde yapay zeka teknikleri kullanılmıştır[39].

İkincilik ödülünü Kernel Miner veri madenciliği aracı kullanılarak geliştirilen Itzhak Levin'in modeli kazanmıştır[39].

Üçüncülük ödülünü Rus bilim akademisinden (IITP) Vladimir Miheev, Alexei Vopilov, ve Ivan Shabalin 'in geliştirdiği karar ağacı modeli kazanmıştır.Bu modelde karar ağacı oluşturmak için öncelikle diğer yöntemlerdeki gibi toplam veri seti eğitim ve test veri setlerine ayrılır.Eğitim ile bu ağacın hiyerarşik yapısı oluşturulmaya ve karmaşıklığı belirlenmeye çalışılır.Test numunesi ise bu ağaçta yer alacak olan bir alt ağacı seçer olmalıdır[39].

Yapılan bu çalışmalarda saldırı türleri kategoriler altında gruplanmıştır.Saldırı grupları ile ilgili bilgiler Tablo 2.8., Tablo 2.9. ve Tablo 2.10. da gösterilmiştir[39].

Tablo 2.8. Saldırı sınıfları [39]

0	normal
1	probe
2	denial of service (DOS)
3	user-to-root (U2R)
4	remote-to-local (R2L)

Tablo 2.9. Eğitim veri setinde yer alan saldırı türlerinin sınıfları [39]

Sınıf	Saldırı Türü
0	normal
2	back
3	buffer_overflow
4	ftp_write
4	guess_passwd
4	imap
1	ipsweep
2	land
3	loadmodule
4	multihop
2	neptune
1	nmap
3	perl
4	phf
2	pod
1	portsweep
3	rootkit
1	satan
2	smurf
4	spy
2	teardrop
4	warezclient
4	warezmaster

Tablo 2.10. Test veri setinde yer alan saldırı türlerinin grupları [39]

Grup	Saldırı Türü
0	normal
2	apache2.
2	back.
3	buffer_overflow.
4	ftp_write.
4	guess_passwd.
4	httptunnel.
3	httptunnel.
4	imap.
1	ipsweep.
2	land.
3	loadmodule.
2	mailbomb.
1	mscan.
4	multihop.
4	named.
2	neptune.
1	nmap.
3	perl.
4	phf.
2	pod.
1	portsweep.
2	processtable.
3	ps.
3	rootkit.
1	saint.
2	satan.
4	sendmail.
2	smurf.
4	snmpgetattack.
4	snmpguess.
3	sqlattack.
2	teardrop.
2	udpstorm.
2	warezmaster.
4	worm.
4	xlock.
4	xsnoop.
3	xterm.

Tablo 2.11. Eğitim ve test setlerinde yer alan saldırı gruplarının yüzde oranları [39]

Saldırı Grupları	Eğitim	Test
0	19.69%	19.48%
1	0.83%	1.34%
2	79.24%	73.90%
3	0.01%	0.07%
4	0.23%	5.20%

Tablo 2.12. En yüksek başarı oranını gerçekleştiren çalışmanın sonuçları [39]

Sınıf	0	1	2	3	4	başarı %
0	60262	243	78	4	6	99,5
1	511	3471	184	0	0	83,3
2	5299	1328	223226	0	0	97,1
3	168	20	0	30	10	13,2
4	14527	294	0	8	1360	8,40

BÖLÜM 3. YAPAY BAĞIŞIKLIK SİSTEMLERİ

3.1. Giriş

Yapay bağışıklık sistemi (YBS), 1986 yılında yayınlanan "bağışıklık sistemi, adaptasyonu ve makine öğrenmesi (The immune system, adaptation and machine learning)" adlı makale ile başlamıştır. İnsan vücudunun mikropları tanıyabilme ve ani olarak onları tahrip edebilmeleri doğal bağışıklık sistemi olarak bilinir. Bu yaklaşımda ise amaç, lenfosit aktiviteleri, doğal antikor üretimi, ön bağışıklık, dağarcık seleksiyonu, tolerans, hafıza ve bağışıklık sisteminin gelişimine benzer yaklaşımları gerçekleştirmektir[40]. Bağışıklık sistemleri tanıtılmadan önce bu sistemin olmazsa olmaz bazı terimlerinin tanıtılması yararlı olacaktır.

Antibody (Eğitim verileri) : Bağışık sistemi tarafından önceki bilgilere dayanılarak daha önceden tanımı yapılan Antijen (hastalık) belirtilerini içeren vektörlerdir. Belirtiler tanımına karşılık yapay bağışıklık sisteminde özellikler karşılık gelmektedir. Antibodyler, antijenler hakkında bilgiye sahiptirler. Benzer şekilde daha önceden görmediği yeni antijen hücrelerini de önceki tanımlamalardan, bıraktıkları izler ve benzerliklerini karşılaştırarak tanımaya çalışırlar. Öğrenme yetenekleri işte bu özelliklerinden gelmektedir. Bu tanımlamayı çok hızlı bir şekilde gerçekleştirirler. Eğitim sonucu elde ettiği karşılaştırma verileri antibody populasyonunu oluşturmaktadır[40].

Antijen (Test verileri) : Tanımlaması yapılacak yeni hücrelerdir. Test verileri başlangıçta antijen olarak tanımlanır. Antibody karşılaştırması sonunda sınıflandırılırlar. İnsan vücudunun ve benzer şekilde STS sistemlerinin zararlı olduğunu tespit ettikleri antijenlere olan tepkisi mümkün ise yok etmek veya değilse karantinaya alıp yayılmalarını önlemektir[40].

Affinity (Sağlıklı olma) ölçüsü : YBS sistemleri Antibodyleri kullanarak , birkaç özelliğini değiştirerek yeni antibody tanımlamaları oluşturmaya çalışır.Bu işlem YBS sistemine kendi sınıf tanım aralığını detaylandırma ve genişletme imkanı verir ve bu sayede ileride karşısına çıkabilecek yeni antijenleri de tanıyabilir.Tabiki bu işlem oluşturulan antibody'i olması gereken mevcut sınıfın dışına çıkarıp diğer sınıflara olan benzerliğini arttırabilir, bu durumda YBS sistemi yanlış bir üretim olduğunu ve karar verme mekanizmasını negatif yönde etkileyeceğini bilir.Yeni üretilen antibody yok edilir.Bu üretilen antibody'nin sağlıklı üretim olup olmadığını anlama ölçüsü Affinity (Sağlıklı olma) olarak tanımlanır[40].

Eşik değer : Antibody üretiminin ardından oluşan son antibody populasyonu artık antijenleri sınıflandırmak için hazırdır.YBS sistemi sahip olduğu antibody tanımlarından yola çıkarak antijenlerin hangi sınıfa yakın olduğunu anlamalıdır.Bunun için eşik değer bilgisi kullanılır.Her özellik ayrı ayrı karşılaştırılır ,özelliğin benzerliği eşik değer sınırları içerisinde ise bu özelliğe göre antijenin bu sınıfa benzerliği vardır yorumu yapılır ve % benzerlik değeri alır.Bu % lik benzerlik değerleri toplanarak ortalaması alınır elde edilen değer antijenin bu antibody sınıfına ne kadar benzer olduğu belirler.Sınıflandırma yapmak için en temel unsurlardan biridir[40].

Literatürde insan bağışıklık sistemine dayalı birçok STS çalışması sunulmuştur. Bu çalışmalardan bazıları doğal bağışıklık sistemi karakteristiklerinden esinlenerek, bilgisayar sistemlerinde anormallik tespiti için sistemler önermişlerdir. Doğal bağışıklık sisteminin, bağışıklığı tanımladığını düşündükleri 4 önemli özelliğini kullanmışlardır. Bunlar; farklılık (diversity), doğal dağıtık yapısı (distributed nature), hata toleransı (error tolerance) ve doğal dinamik yapısıdır (dynamic nature) [41].

YBS'nin yapısı, katmanlı bir yaklaşım olarak düşünülebilir. Her sistemin temeli uygulama alanıdır. Bu alan için sistemin bileşenlerinin uygun bir temsiline karar verildikten sonra bir ya da daha fazla affinite ölçüleri sistemin elemanları arasındaki etkileşimleri ölçmek için kullanılır. Birçok olası afinite ölçüleri Hamming ya da Öklit uzaklıkları gibi yöntemler kullanılabilir. Bir sonraki katman sistemin davranışını (dinamiğini) yöneten işlemler ya da algoritmaları içerir. Bu da bize

cevabı verir. Buna göre yapı, şekil-uzayı olarak adlandırılan bağışık hücreler ve moleküllerin genel soyut modelini çıkarmakla başlar[42].

3.2. Bağışık Hücre, Molekül ve Bunların Etkileşiminin Soyut Modelleri

Bir antijenin tanınabilmesi için moleküllerin (antijen ya da antikor) yüzeylerindeki belli bölgelerde birbirleriyle tümleyen olarak bağlanmaları gerekmektedir. Bu yüzden moleküller arasında geniş tümleyen bölgelerine ihtiyaç vardır. Temsil olarak vektörler kullanılabilir. Antikor = Ab_1, Ab_2, ..., Ab_L, Antijen = Ag_1, Ag_2, ..., Ag_L gibi. Gerçek değerli şekil uzayı, tam sayı şekil uzayı, Hamming şekil uzayı ve sembolik şekil uzayı da kullanılabilir. Hücreler arasındaki etkileşim de afinite ile açıklanır. Afinite çeşitli uzaklık ölçülerine bağlıdır. Bunun için Öklit, Hamming, Manhattan vb. uzaklıklar kullanılabilir[42].

3.3. Algoritmalar ve İşlemler

İki temel bileşen (kemik iliği ve timus) ve iki ayrı teori (klonal seçim ve bağışık ağ) bağışıklık sistemini modellemek için kullanılır [42].

Kemik iliği modeli: Hücreler ve moleküllerin repertuarını üretmede kullanılır.

Timus modeli: Öz/öz olmayan ayrımı yapmaya yetenekli hücre ve moleküllerin repertuarını üretmede kullanılır.

Klonal seçim algoritmaları: Bağışıklık sisteminin bileşenlerinin harici çevre ve antijenlerle nasıl etkileşim yaptığını kontrol etmede kullanılır

Bağışık ağ modelleri: Yapılarını, dinamiğini ve meta dinamiğini de içeren bağışık ağların benzetiminde kullanılır.

3.3.1. Kemik iliği modelleri

Gen kütüphanesi kemik iliğinden antikorları üretmek için kullanılır. Bu fikri reseptörleri temsil eden nitelik dizgelerini üretmek için kullanırız. Antikorlar gen kütüphanelerinden gelişigüzel birbirine bağlanma ile oluşur[42].

3.3.2. Timus modelleri

T hücreleri kemik iliğinde üretilir ve timusa göçer ve orada immünokompetent hücrelere farklılaşır (pozitif seçim) ve diğerleri öz peptit/MHC kompleksleriyle güçlü bir tanıma yaptığından repertuardan temizlenir (negatif seçim). Bu timik pozitif ve negatif seçim timustan ayrılan ve perifere giden T hücre populasyonunun öz peptitleri tanıyan hücreleri ihtiva etmediğini ve aynı zamanda bir öz MHC molekülü tarafından sunulan bir peptit ile uyarılmaya hazır olduğunu garanti eder. Pozitif seçim algoritmaları, şu şekilde özetlenebilir[42]:

Başlangıç: Olgunlaşmamış T hücrelerinin potansiyel repertuarını, P, üret. Tüm molekülleri aynı uzunluklu, L, ikili dizgeler ile temsil edildiği varsayıldığında 2^L ayrı hücre üretilir.

Afinite değerlendirmesi: P'deki tüm elemanların öz hücre kümesi S'deki tüm elemanlar ile afinitelerini belirle.

Uygun repertuarın üretimi: P'nin bir elemanı ile MHC'nin en az bir elemanı arasındaki afinite verilen çapraz reaktif eşiği ε'den büyükse ya da buna eşitse o zaman T hücresi bu MHC'yi tanır ve pozitif olarak seçilir ve sisteme tanıtılır (uygun repertuar A'ya); değilse T hücresi yok edilir.

T hücrelerinin negatif seçimi reseptörleri öz MHC tarafından sunulan öz peptitlere bağlanma kabiliyeti olan T hücrelerinin yok edilmesinden sorumludur. Bu işlem timustan ayrılan T hücrelerinin herhangi bir öz hücreyi ya da molekülü

tanımayacağını garanti eder. Bu işlemlerden esinlenerek geliştirilen negatif seçim algoritması, [43] şu şekilde özetlenebilir:

Başlangıç: Gelişigüzel dizgeler üret ve bunları olgunlaşmamış T hücrelerinin bir P kümesine yerleştir. Tüm moleküllerin (reseptör, öz peptitler) aynı uzunlukta, L, ikili dizgeler olarak temsil edildiğini varsay.

Afinite değerlendirmesi: P'deki tüm T hücrelerinin S'deki tüm elemanlarla afinitesini belirle.

Uygun repertuarın üretimi: Olgunlaşmamış bir T hücresinin (P'nin elemanı) en az bir öz peptitle afinitesi verilen çapraz reaktif eşiğinden, ε, büyükse ya da buna eşitse o zaman T hücresi bu öz peptidi tanır ve yok edilmelidir (negatif seçim); değilse T hücresi uygun repertuar A'ya tanıtılır.

Negatif seçim iki aşamaya bölünmüştür. Yukarıda anlatılan algılama safhasıdır. Diğer safha ise gözetleme safhasıdır. Bu safhada korunmuş dizgeler kümesi, S* uygun repertuar A'nın elemanları ile karşılaştırılır. S* kümesi S'in kendisi olabilir ya da tamamen yeni bir küme olabilir ya da, S'nin elemanlarından oluşmuş olabilir. Bir tanıma olursa o zaman bir öz olmayan örüntü tespit edilmiştir[43].

3.3.3. Klonal seçim algoritmaları

Klonal seçim prensibi, bağışıklık sisteminin bir antijenik uyarıma karşı bağışıklık cevabının temel özelliklerini tanımlamak amacıyla kullanılır. Bu prensip sadece antijenleri tanıyan hücrelerin çoğaldığı yani tanımayanlara göre seçildiği fikrini vurgular. Bu seçilmiş hücreler, afinite olgunlaşma işlemine mazur kalırlar ve bu işlem seçilmiş hücrelerin antijenlere benzerliğini geliştirir. Burada, bağışıklıkla ilgili dikkate alınan temel olaylar aşağıda verilmiştir[43]:

1. Dağarcıktan fonksiyonel olarak ayırt edilmiş olan hafıza hücrelerinin varlığının sağlanması
2. En fazla uyarılmış hücrelerin seçimi ve klonlaşması
3. Uyarılmamış hücrelerin ölümü
4. Daha yüksek afiniteli klonların afinite olgunlaşması ve tekrar seçimi
5. Farklılaşmanın üretilmesi ve sağlanması
6. Hücre afinitesi ile orantılı olarak hipermutasyon işleminin uygulanması

Algoritma [44] Şekil 3.1'de özetlenmiştir:

```
Gelişigüzel bir populasyon (P) üret

For Antijendeki her örüntü için
    Her P ile afiniteyi belirle
    P ile en yüksek n afiniteliyi seç
        Antijen afinitesi ile doğru orantılı olarak klonlama ve mutasyon yap
    P'ye yeni mutantlar ekle
endFor

M nin bölümünü oluşturmak için en yüksek afiniteli P'yi seç
n tanesini yeni oluşturulmuşlarla yer değiştir

Sonlandırma kriterine kadar
```

Şekil 3.1. Klonal seçim algoritması [44]

3.3.4. Bağışık ağ modelleri

Diferansiyel denklemler temelli sürekli ağ modelleri başarılı bir şekilde özerk hareket, optimizasyon ve otomatik kontrol gibi kompleks problemlere uygulanmıştır. Bunlar aynı zamanda fark denklemleri temelli ayrık ağ modellerine de ilham olmuştur. De Castro ve Von Zuben [45]'in önerdiği model Şekil 3.2'de özetlenmiştir.

1. Başlangıç: Ağ antikorlarının gelişigüzel bir başlangıç populasyonunu üret
2. Antijenik temsil: Her antijenik örüntü için do:
    2.1. Klonal seçim ve genişletme: Her ağ elemanı için sunulan antijen için afiniteyi hesapla. Yüksek afiniteli birkaç elemanı seç ve afiniteleri ile doğru orantılı olarak bunları tekrar üret (klonla)
    2.2.Afinite olgunlaşması: Afinitelerle ters orantılı olarak her klona mutasyon uygula. En iyi afiniteli birkaç klonu tekrar seç ve bunları klonal bellek setine yerleştir
    2.3. Metadinamik: Antijenle afinitesi verilen eşikten düşük olan hafıza klonlarını yok et
    2.4.Klonal etkileşim: Klonal hafıza setinin tüm elemanları arasında ağ etkileşimini (afinite) belirle
    2.5. Klonal baskı: Birbirleriyle afinitesi verilen eşikten az olan hafıza klonlarını elime et
    2.6. Ağ yapımı: Kalan klonal hafızanın klonlarını tüm ağ antikorları ile dahil et
3. Ağ etkileşimi: Her ağ antikor çifti arasındaki benzerliği belirle
4. Ağ baskılama: Afinitesi verilen eşikten az olan ağ antikorlarını yok et
5. Çeşitlilik: Ağa yeni gelişigüzel üretilmiş antikorlar ilave et
6. Çevrim: İkiden beşinciye kadarki aşamaları verilen iterasyon sayısına kadar tekrar et.

Şekil 3.2. De Castro ve Von Zuben'in ağ modeli [45]

3.4. Yapay Bağışıklık Sistemlerinin Uygulama Alanları

Bilgisayarları virüslerden ve yetkisiz kullanıcılardan korumak model tanıma araştırmaları için geniş bir araştırma alanıdır. Bu alandaki problemler için negatif ve klonal seçim mekanizmaları kullanılmaktadır. Negatif seçim; hata denetimi, anormal durum tespiti, bilgisayar ve ağ güvenliği problemleri için kullanışlı iken, klonal seçim optimizasyon problemleri ve öğrenme becerilerinden dolayı negatif seçim ile beraber kullanılmaktadır[46].

Forrest ve diğ.'de r-ardışık bit kuralı ve bir negatif seçim algoritması kullanılarak "self" ve "nonself" ayrımına dayanan bir bilgisayar güvenliği sistemi incelenmiştir. Sistem negatif seçim algoritması mantığına göre çalışmaktadır. Önce bir algılayıcılar kümesi oluşturulmakta, sonra korunan veriler oluşturulan algılayıcılar ile karşılaştırılarak izlenmektedir. Eğer iki dizideki ortak ardışık bitlerinin sayısı bir *r* sayısından büyük veya eşit ise, bir eşleşmeden bahsedilebilir. Çalışmada ayrıca, iki rastgele dizi arasında bir eşleşme olma olasılığını ve sistemin farklı konfigürasyonları için değişiklik-tespit olasılığını tahmin etmek üzere formüller verilmiştir. Elde edilen sonuçlara göre, korunacak dizilerin sayısı arttıkça, kullanılan algılayıcıların boyutunun artmasına gerek yoktur. Tespit olasılığı, bağımsız tespit

algoritmalarının sayısı ile üstel olarak artmaktadır. Algılayıcı oluşturmanın maliyeti "self" kümesinin büyüklüğü ile üstel artmaktadır[46].

Bir yapay bağışıklık sisteminin dağıtılmış, sağlam (robust), dinamik, çeşitlendirilmiş, adapte edilebilen bir sistem olması özelliklerini kullanarak, bilgisayar ağı güvenliği konusunda yapılmış çalışmalar bulunmaktadır. Bu yapay bağışıklık sistemlerinde, bağışıklık sisteminin kullanışlı özelliklerinin hepsini içeren temel bir tip algılayıcı tanımlanmıştır. Algılayıcılar ikilik bir Hamming şekil-uzayında bit-dizileri ile gösterilmiştir. Tespit olayı, iki dizi arasındaki r-ardışık bit'in eşleşmesi süreci ile sağlanır. Negatif seçim ile birlikte yalın algılayıcıların hafıza algılayıcılarına olgunlaştırılması sistemin öğrenme kısmının sorumluluğudur. Kurulan yapay bağışıklık sistemi, bir sınıflandırıcı sistemin (classifier system) çoğu önemli özelliği ile örtüşmektedir. Önerilen YBS, her birinde 100 algılayıcının bulunduğu, 50 bilgisayardan oluşan bir ağ sisteminde; sekiz tane normal dışı olayın tamamını tespit etmiştir. Bu alandaki bazı sistemler, bir ayda, milyonlarca yanlış alarm verirken; önerilen sistem günde ortalama iki yanlış alarm vermiştir[47,48].

Somayaji ve diğ.'de, biyolojik bağışıklık sistemine dayanarak bir bilgisayar bağışıklık sisteminin geliştirilmesi sürecini geniş bir şekilde anlatmışlardır. Bağışıklık sisteminin kullanılabilir prensiplerini ve uygulama için mümkün olan yapıyı ortaya koymuşlardır[49].

Yapay bağışıklık sisteminin uygulama alanlarından biri de optimizasyon problemleridir. De Castro ve Von Zuben optimizasyon problemleri için kendilerinin önerdiği CLONALG algoritmasını kullanmışlardır. CLONALG algoritmasını 30 şehirli bir gezgin satıcı problemine uygulamışlardır. Problem için çalışmada bir tam sayılı şekil uzayı (shape space) kullanılmıştır. L uzunluğundaki tamsayı değerli vektörler, C={1,2,...,L} elemanlarının permütasyonlarından oluşur ve bu vektörler mümkün turları belirtirler. Tamsayı vektörünün her bir bileşeni bir şehri gösterir. Her bir turun toplam uzunluğu, bir tura karşılık gelen vektörün benzerlik ölçüsünü verir[50,51].

Mutasyon basitçe, turları belirten antikorlar içindeki şehir çiftlerinin yerlerini değiştirmek ile sağlanır. Çalışmada, Moscato ve Fontanari tarafından ele alınan problem YBS ile çözülmüştür. Popülasyon büyüklüğü (antikor popülasyonu) 300 bireydir. Her 20 nesilde bir, antikorların en kötü %20'lik kısmı yenileri ile yer değiştirir. Algoritma 300 adım sonra, optimal sonuca ulaşmıştır[17].

Costa ve diğ.'de, paralel makinalarda toplam tamamlanma zamanının (makespan) enazlanması problemi üzerinde durmuşlardır. Çalışmada CLONALG algoritması üzerine kurulan bir yapay bağışıklık sistemi modeli ile bazı sezgiseller karşılaştırılmıştır. Karşılaştırılan sezgiseller; LPT, Multifit, Lokal Arama ve Tavlama Benzetimidir. Problem için her olurlu çözüm, örneğin tam bir çizelge, sabit *n* büyüklüğünde bir dizi olarak kodlanmıştır. Dizi üzerindeki her pozisyon bir süreç ile ilişkilidir. Her *i* pozisyonunun değeri işlemin yerleştirileceği makineyi belirtir[52].

Popülasyonun her bir antikoru (çözümü) için bir benzerlik (affinity) değeri vardır. Aşağıdaki denklemde gösterilen bu benzerlik değeri, çözümün kalitesini yansıtır[52]:

LB Benzerlik(k) = (1+M(k) - LB)

Burda; M(k); k antikoru ile gösterilen çözümün toplam tamamlanma zamanını (makespan) gösterir. LB; problemin alt limitini belirtir. Bu alt limit, tüm işlem zamanları toplamının işlemci sayısına oranı ile bulunur[52].

Denklemdeki payda; M(k)'nın LB'ye yakın olduğu durumlarda benzerliğin daha yüksek olmasını dolayısıyla çözümün iyileşmesini sağlar. Durdurma kriteri, en iyi çözüm üzerinde ilerleme sağlamadan geçen belirli bir nesil sayısıdır, ayrıca bir zaman sınırı da verilmiştir[52].

Algoritma, 390 örnek problem üzerinde test edilmiştir, her bir işin işlem süresi [1,k] aralığında uniform dağılımdan seçilmiştir. Algoritma diğer yöntemlere göre daha iyi sonuçlar vermiştir, özellikle uzun işlem süreli ve az sayıda makinenin olduğu problemlerde algoritma oldukça etkilidir. Yazarlar, iyi performansın nedeninin

bağışıklık sisteminin sunduğu yüksek çeşitlilikten kaynaklandığını belirtmişlerdir[52].

Atölye tipi çizelgeleme problemlerine sağlam (robust) çözümler bulmak amacıyla Jensen ve Hansen tarafından bir çalışma yapılmıştır. Gerçek bir sistem için optimal çizelgeler yerine, değişen şartlara göre üzerinde kolayca değişiklik yapılabilecek çizelgelerin bulunmasının önemine dikkat çeken yazarlar, bu amaca yönelik bir yapay bağışıklık sistemi geliştirmişlerdir. Çalışmada her biri bir miktar genetik dizi içeren kütüphaneler kurulmuştur, her dizi bir atölye tipi problem kümesi çözümünün bir parçasıdır. Atölye tipi probleme çözüm, her kütüphaneden dizileri seçerek (bu dizi bir antikordur) ve seçilen dizinin kodu çözülerek bulunabilir. Her işin başlama tarihleri değiştirilerek bir antijen kümesi elde edilir[53].

Bu antijenler, çeşitli hatalar veya duraksamalar nedeniyle mevcut planlardan farklı olarak ortaya çıkan çizelgelere karşılık gelir. Çalışmada, bir sağlamlık ölçütü tanımlanmıştır. Bu ölçüte göre yapılan değerlendirmeler göstermiştir ki, sağlam çözümler mevcuttur ve bu çözümler YBS ile bulunabilir[53].

Hart ve diğ., her işin belirli başlama ve bitiş tarihlerinin olduğu atölye tipi çizelgeleme problemlerinde maksimum gecikmeyi, enazlamak için yapay bağışıklık sistemi modeli kullanmışlardır. Model iki aşamalı çalışmaktadır. Sistemin birinci aşamasında, fabrikada en sık kullanılan ortak iş çizelgeleri modellerini tespit etmek için genetik algoritma(GA) ile birleştirilmiş bağışıklık sistemi yaklaşımı kullanılmaktadır. İkinci aşamada, tespit edilen modelleri kullanarak yeni çizelgeler üretmek için doğal bağışıklık sistemlerinin kombinatorik özellikleri modellenmiştir. Sonuçlar, geniş çaplı bir araştırma prosedürü kullanan bir model ile karşılaştırılmıştır[54].

Önerilen algoritma oldukça başarılı sonuçlar vermiştir, şöyle ki, daha önce ortaya çıkan herhangi bir duruma karşılık gelen çizelgeler kolaylıkla tekrar oluşturulabilmektedir[54].

Mori ve diğ., bir yarı iletken üretim hattını kontrol etmek için genel bir otonom dağıtılmış sistem tanımlamışlardır. Çalışmalarında, üretim hattının kontrolü bir ajanlar (agents) kümesi (detector, mediator, inhibitor ve restoration ajanları) ile yapılmaktadır. Her bir ajan üretim hattı ve diğer ajanlarla ilişki içindedir. Bu ilişki omurgalı bağışıklık sistemindeki ilişkiye dayanmaktadır[55].

Örneğin, dedektör ajanlar, bağışıklık sistemindeki B hücrelerine karşılık gelir ve sistemdeki belirli aksaklıkları tespit etmek için kullanılır. Sistem pratikte denenmemesine rağmen, çalışmada sistemin gerçek zamanlı karar vermede ve değişen çevreye uyum sağlamada başarılı olacağı iddia edilmiştir[55].

Dasgupta, ve Forrest, alet hatası tespiti için bir yapay bağışıklık algoritması geliştirmişlerdir. Metod; bağışıklık sisteminin self (vücut elemanları) ve nonself (yabancı elemanlar) hücreleri birbirinden ayırmayı sağlayan negatif-seçim mekanizmasından ilham almıştır[56].

Bu uygulamada ‘self’, normal kesme operasyonu değerlerini, “nonself” ise izin verilen kesme kuvveti farklılığının ötesinde herhangi bir sapmayı belirtir[56].

Önerilen algoritma, torna operasyonları için bir simülasyon çalışması ile gösterilmiş ve kalem ucunun bozulması durumunda algoritmanın bunu tespit etme performansı belirlenmiştir. Algoritma, tüm test durumları için kalem ucundaki bozulmaları tespit etmiştir[56].

Lee ve diğ., yapay bağışıklık sistemini dağıtılmış, otonom robot sistemine (DARS-Distributed Autonomous Robot System) uygulamışlardır. Her robot; bir “B” hücresi, her bir çevresel durum; bir antijen, bir davranış stratejisi; bir antikor ve bir kontrol parametresi de bir “T” hücresi olarak ele alınmıştır. Sistemin çalışmasında; çevresel durum değiştiğinde her robot uygun bir davranış stratejisi seçer ve onun bu davranış stratejisi iletişim ile diğer robotlar tarafından tetiklenir ve yayılır. Sonuçta en çok kabul gören strateji, kümenin davranış stratejisi olarak belirlenir. Bu kontrol şeması

klonal seçim ve idotopik ağ hipotezine dayanır. T hücre modellemesi kullanılarak, robotun dinamik ortamlara uyum yeteneği geliştirilmiştir[57].

Dasgupta ve Forrest, zaman serileri verilerindeki farklılaşmaları tespit etmek üzere bir negatif seçim algoritması önermişlerdir. Zamanla kesme kuvveti değerleri değişmektedir. İzin verilen sapmanın ötesindeki değerler sistemdeki 'non-self' leri belirtmektedir. Sistemin elemanlarını tasvir etmek için bir ikilik Hamming şekiluzayı uygulamışlar ve algılayıcılar ile kodu çözülmüş veri arasındaki algılamanın derecesini belirlemek için bir r ardışık bit kuralı uygulamışlardır. Yazarlar iki veri seti için sonuçları vermişlerdir: bir tornalama operasyonunun kesici dinamikleri ve bir sentetik sinyal. Eşleştirme fonksiyonu ile seçilen *r*-ardışık bitlerinin sayısının hataların tespitindeki riskin güvenilirliğini etkilediğini göstermişlerdir[58].

BÖLÜM 4. STS YAZILIMI GELİŞTİRME

4.1. Giriş

Bu bölümde tez kapsamında yapay bağışıklık sistemi ile gerçekleştirmiş uygulamanın tasarımı için yapılan işlemler anlatılmıştır. Öncelikle eğitim ve test için kullanılan veri kümeleri hakkında detaylı bilgi verilmiştir. Uygulamanın en önemli kısmını oluşturan YBS modelinin yapısı açıklanmış ve YBS modelinin test sonuçları ise bu bölümün sonunda sunulmuştur.

Uygulamanın gerçekleştirildiği programlama dili C# olarak belirlenmiştir. Gelişmiş fonksiyon tanımlamalarına sahip olması, kullanım kolaylığı ve geniş kullanım alanlarına sahip olması bu dilin seçilmesinin başlıca nedenlerindendir.

Uygulama alt yapısında sınıflandırma ve saldırı tespiti için YBS modeli kullanılmıştır.YBS kullanılmasının nedenleri şunlardır;

Doğrusal olmama : Doğrusal değişim göstermeyen problemlerin çözümü için kullanılabilir.

Öğrenme : Eğitim işlemi sonucunda YBS sistemleri öğrenme yeteneğine sahiptir.

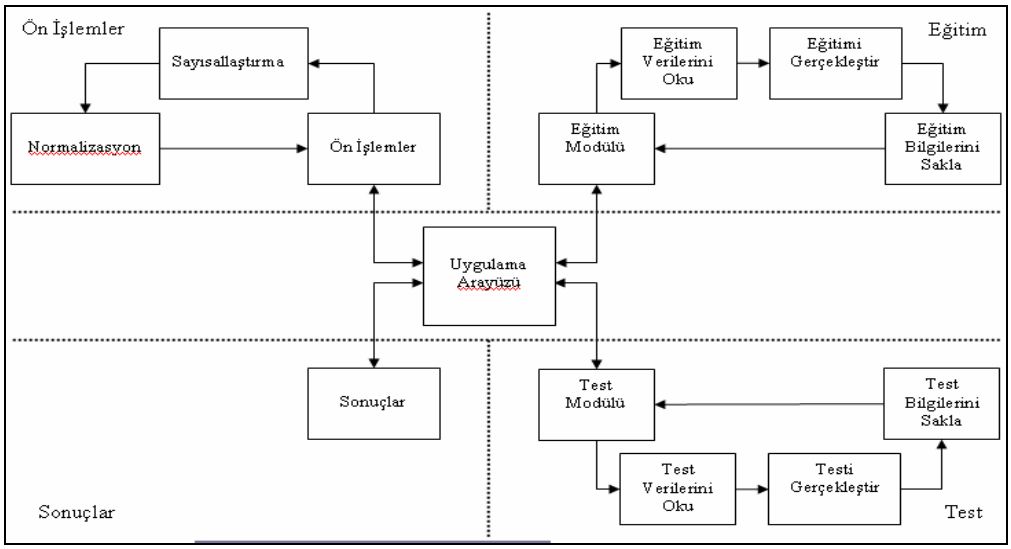
Genelleme : Eğitilmiş YBS sistemleri sisteme daha önce gösterilmeyen test örnekleri için yorumlama yeteneğine sahiptir.

Uyarlanabilirlik : YBS, ilgilendiği problemdeki değişikliklere göre parametre değerlerini (eşik ve affinite) ayarlayabilir.Yani, belirli bir problemi çözmek amacıyla

eğitilen YBS, problemdeki değişimlere göre tekrar eğitilebilir ve değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir.

Uygulama alanları : YBS sistemleri sınıflandırma, ses tanıma, karakter tanıma, robot kontrolleri, resim işleme ve yüz tanıma sistemlerinde başarılı sonuçlar veren sistemlerdir.

Geliştirilen uygulama 4 modülden oluşmaktadır. Uygulamanın modül yapısı Şekil 4.1 de gösterilmiştir.



Şekil 4.1 Uygulama modül yapısı

Uygulamanın genel işleyiş prensipleri incelendiğinde sisteme giriş olarak verilen eğitim ve test veri setlerinin belirlenmesi önceliklidir.

4.2. Veri Setlerinin Belirlenmesi

Sistemin eğitim ve test aşamalarında kullanılmak üzere veri kümeleri elde edebilmek için literatürde yapılan benzer çalışmalar incelenmiştir. Bunlardan, DARPA ve KDD'99 veri kümeleri Bölüm 2'de açıklanmıştır. DARPA veri kümelerinin incelenmesinden edinilen bilgiler ışığında oluşturulan KDD'99 veri kümeleri

sistemimizde kullanılmak üzere seçilmiştir. Uygulama örneğini geliştirmek için KDD'99 veri kümelerinden alınan verilerin formatı Tablo 4.1'de gösterilmiştir.

Tablo 4.1. Veri kümesi örneği

0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00, 1.00,0.00, 0.00,255,252,0.99,0.01,0.00,0.00,0.00,0.00,0.00,0.00,snmpgetattack.
1,tcp,smtp,SF,3170,329,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,2,0.00,0.00,0.00,0.00, 1.00,0.00, 1.00,54,39,0.72,0.11,0.02,0.00,0.02,0.00,0.09,0.13,normal.
0,tcp,http,SF,297,13787,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00, 1.00,0.00, 0.00,177,255,1.00,0.00,0.01,0.01,0.00,0.00,0.00,0.00,normal.

Verilerin kullanılabilmesi için sütunlara ayrıştırılması ve bazı ön işlemlerden geçmesi gereklidir. Bu veri hazırlama işlemlerin yapılabilmesi için uygulama içerisine ön işlemler modülü eklenmiştir.

4.2.1 Ön işlemler

KDD'99 veri kümesinde, tek bir örneğe ait olan her veri satırındaki 42 farklı alan, birbirlerinden ayrılmıştır. Tablo 4.2'de özelliklerine göre sütunlara ayrılmış veriler gösterilmiştir.

Tablo 4.2. Özelliklerin ayrıştırılmış formu

0 udp private SF 105 146 … … … 0.00 0.00 0.00 0.00 snmpgetattack.
1 tcp smtp SF 3170 329 … … … 0.02 0.00 0.09 0.13 normal.
0 tcp http SF 297 13787 … … … 0.00 0.00 0.00 0.00 normal.

Sütunlara ayrılan bu verilerden bazıları sayısal formatta olmadığından, YBS'de veri olarak kullanılabilmesi için, sayısal formata çevrilmesi gerekmektedir. Bu ihtiyaçtan dolayı veri kümesinde yer alan protokol, servis, bayrak (flag) ve saldırı tipleri alanlarının sayısal forma dönüştürülmesi gerçekleştirilmiştir.Sayısallaştırma işlemleri KDD'99 standartlarına göre yapılmıştır. Bu alanlardaki veriler ve sayısal karşılıkları Tablo 4.3, Tablo 4.4, Tablo 4.5, Tablo 4.6 ve Tablo 4.7'de verilmiştir.

Tablo 4.3. Saldırı isimlerinin sayısal forma dönüştürülmesi [39]

Sayısal Değer	Saldırı Türü
0	normal.
2	apache2.
2	back.
3	buffer_overflow.
4	ftp_write.
4	guess_passwd.
4	httptunnel.
3	httptunnel.
4	imap.
1	ipsweep.
2	land.
3	loadmodule.
2	mailbomb.
1	mscan.
4	multihop.
4	named.
2	neptune.
1	nmap.
3	perl.
4	phf.
2	pod.
1	portsweep.
2	processtable.
3	ps.
3	rootkit.
1	saint.
2	satan.
4	sendmail.
2	smurf.
4	snmpgetattack.
4	snmpguess.
4	spy
3	sqlattack.
2	teardrop.
2	udpstorm.
2	warezmaster.
4	worm.
4	xlock.
4	xsnoop.
3	xterm.

Saldırı isimleri KDD ’99 organizasyonunda belirlenen sınıflara ayrılmıştır. Yukarıda belirlenen sayısal değerler ve karşılıkları olan sınıf bilgisi Tablo 4.4’te verilmiştir.

Tablo 4.4. Saldırı sınıfları ve sınıfların sayısal değerleri [39]

Sayısal Değer	Sınıf
0	normal
1	probe
2	denial of service (DOS)
3	user-to-root (U2R)
4	remote-to-local (R2L)

Tablo 4.5. Servis isimlerinin sayısal forma dönüştürülme tablosu [39]

Servis	Sayısal Değeri	Servis	Sayısal Değeri
http	0	exec	33
smtp	1	printer	34
finger	2	efs	35
domain_u	3	courier	36
auth	4	uucp	37
telnet	5	klogin	38
ftp	6	kshell	39
eco_i	7	echo	40
ntp_u	8	discard	41
ecr_i	9	systat	42
other	10	supdup	43
private	11	iso_tsap	44
pop_3	12	hostnames	45
ftp_data	13	csnet_ns	46
rje	14	pop_2	47
time	15	sunrpc	48
mtp	16	uucp_path	49
link	17	netbios_ns	50
remote_job	18	netbios_ssn	51
gopher	19	netbios_dgm	52
ssh	20	sql_net	53
name	21	vmnet	54
whois	22	bgp	55
domain	23	Z39_50	56
login	24	ldap	57
imap4	25	netstat	58
daytime	26	urh_i	59
ctf	27	X11	60
nntp	28	urp_i	61
shell	29	pm_dump	62
IRC	30	tftp_u	63
nnsp	31	tim_i	64
http_443	32	red_i	65

Tablo 4.6. Protokol isimlerinin sayısal dönüşümleri [39]

Protokol	Sayısal Değeri
Tcp	0
Udp	1
Icmp	2

Tablo 4.7. Bayrak (Flag) isimlerinin sayısal forma dönüştürülmesi [39]

Bayrak (Flag)	Sayısal Değeri
S0	0
S1	1
S2	2
S3	3
SF	4
SH	5
OTH	6
REJ	7
RSTO	8
RSTOSO	9
RSTR	10

Sayısal formata dönüştürme işlemi sonucunda veri görünümü Tablo 4.8. verilmiştir.

Tablo 4.8. Sayısal forma dönüştürme sonrası veri görünümü

0 1 11 4 105 146 … … … 0.00 0.00 0.00 0.00 4
1 0 1 4 3170 329 … … … 0.02 0.00 0.09 0.13 0
0 0 0 4 297 13787 … … … 0.00 0.00 0.00 0.00 0

Sayısal formata dönüştürme işleminin ardından her bir özellik için sayısal değerlerin dağılımı incelenmiştir. Dağılımının çok farklılık gösterdiği bazı özelliklerin normalize edilmesinin faydalı olacağı görülmüştür. Bu sebepten dolayı hazırlık işlemlerine normalizasyon işlemi de eklenmiştir.

4.2.2. Eğitim veri kümesi

Geliştirilecek olan eğitim veri kümesi içerisinde, hangi sınıftaki verilerden ne kadar yer alacağına karar verilmelidir. Tercih edilen durum, en çok karşılaşılan sınıfları içinde barındıran bir veri kümesidir. Ancak sistemin yapısına göre, daha önceden elde edilen istatistiksel sonuç verileri sayesinde, en çok karşılaşılan saldırıların tespit edilmesi yararlı olacaktır. Veri kümesi içerisinde yer alacak olan örneklerinin miktarı, yine aynı istatistiksel verilerle orantılı olarak tasarlanabilir. Örneğin bir sistemde, en çok DoS saldırıları ile karşılaşıldığı tespit edilmişse, veri kümesinin içeriğinde DoS saldırılarının yoğun olması beklenir.

Bu ihtiyacın karşılanabilmesi amacıyla uygulamada kullanılmak üzere eğitim ve test verilerinin oluşturulması için bir modül geliştirilmiştir. Modülün gerçekleştirilmesinin bir diğer nedeni de ayrıştırılması gereken başlangıç veri setinin çok büyük olması durumunda manual ayrıştırmanın mümkün olamamasıdır. Modülün işleyişinde toplam veri seti belirlenen sayı kadar veri setlerine ayrıştırılmakta ve bu işlem yapılırken sınıfların homojen dağılımı esas alınmaktadır. Bu tür bir dağılımın performansı pozitif yönde etkileyeceği bilinmektedir. Bu uygulama içerisinde KDD ’99 eğitim veri seti kullanılacağı için veri seti bölme işlemi yapılmamıştır. Geliştirilen uygulama farklı alanlarda da kullanılmak üzere tasarlanmış olduğundan ihtiyaç dâhilinde bu kolaylıklar da kullanıcıya sağlanmıştır. Eğitim setinde sınıfların sayısal dağılımı Tablo 4.9. de gösterilmiştir.

Tablo 4.9. Eğitim setindeki sınıf sayıları

Sınıf	Eğitim Veri Sayısı
0	97278
1	4107
2	391458
3	52
4	1126

4.2.3. Test veri kümesi

Farklı eğitim veri kümesiyle eğitilen YBS uygulaması, eğitim veri kümesinden farklı olan test veri kümesi ile test edilmiştir. Test veri kümesi KDD'99 doğrulanmış test veri kümesinden alınan verilerden oluşmaktadır.

Eğitim ve test veri kümelerinde bulunan örnek sayıları incelendiğinde, birbirine yakın olmayan değerler gözlenmiştir. Örneğin eğitim veri kümesinde bir saldırı türünden 1126 örnek varken test veri kümesinde 16189 adet örnek vardır. Bu durum, test veri kümesi için önemsizken, YBS'nin öğrenmesini etkileyeceğinden eğitim veri kümesi için önemlidir. KDD'99 veri kümesinde genel tarama yapıldığında benzer bir durum olduğu görülmüştür. Test setinde sınıfların sayısal dağılımı Tablo 4.10 da gösterilmiştir.

Tablo 4.10. Test setindeki sınıf sayıları

Sınıf	Test Veri Sayısı
0	60593
1	4166
2	229853
3	228
4	16189

4.3. YBS Yapısı

Yapay bağışıklık sistemi kullanılarak gerçekleştirilen uygulamanın YBS algoritma yapısı şu şekildedir.

Adım 1: Eğitim verilerini oku ve antibody populasyonunu oluştur.

Adım 2: Eşik değer belirle.

Adım 3: Populasyondaki antibody hücrelerinden klonlama ile yeni antibodyler oluştur.

Adım 4: Oluşan antibody hücrelerinin affinitelerini mevcut antibody populasyonu ile test et, affinite degeri eşik değeri altında kalan antibody leri yok et,digerlerini populasyona ekle.

Adım 5: Eğer populasyondaki antibody miktarı değiştiyse, Adım 3 ‘e dön.

Adım 6: Test verilerini oku.

Adım 7: Test hücrelerini eğitim sonucu oluşan antibody populasyonu ile karşılaştır, sınıflandır.

Algoritma adımlarını açıklamak gerekirse;

Adım 1 de Antibody populasyonu başlangıç eğitim verilerinden oluşturulmaktadır.

Adım 2 de eğitim işleminin ilk kısmı olan eşik değer belirleme işlemi gerçekleşir.Eşik değer aynı sınıfa dahil olan veriler arasında her bir özellik için belirlenen, kabul edilebilir fark değerini ifade eder.Eşik değerin belirlenebilmesi için başlangıç eğitim seti kullanılmıştır.

Adım 3 ile mevcut populasyon türetim ile genişletilir. Eğitim verilerinden bilgi olarak sisteme verilen sınıflar bir kümeyi ifade eder. Klonlama işlemindeki amaç sınıflar arasındaki hiçbir sınıfa dahil olmayan boşlukları bir sınıfa dahil etmek ve bu sayede sınıfların tanım alanlarını genişletmektir.

Adım 4 ile klonlama ile elde edilen yeni verilerin (türetilmiş antibody hücreleri) sağlıklı olup olmadığı kontrol edilir. Yani türetilmiş yeni veriler eğer başka sınıfların kümesi içine girmiş ise bu türetim yanlış yapılmıştır. Yanlış türetilen veri populasyona dahil edilmez.

Adım 5 ile eğer klonlama başarılı ve populasyon genişledi ise yeni türetimler halen yapılabilir demektir, klonlama işlemine devam edilmelidir. Aksi taktirde türetim bitmiştir ve eğitim tamamlanmıştır.

Adım 6 ile test verileri okunur. Test işlemi için hazır hale getirilir.

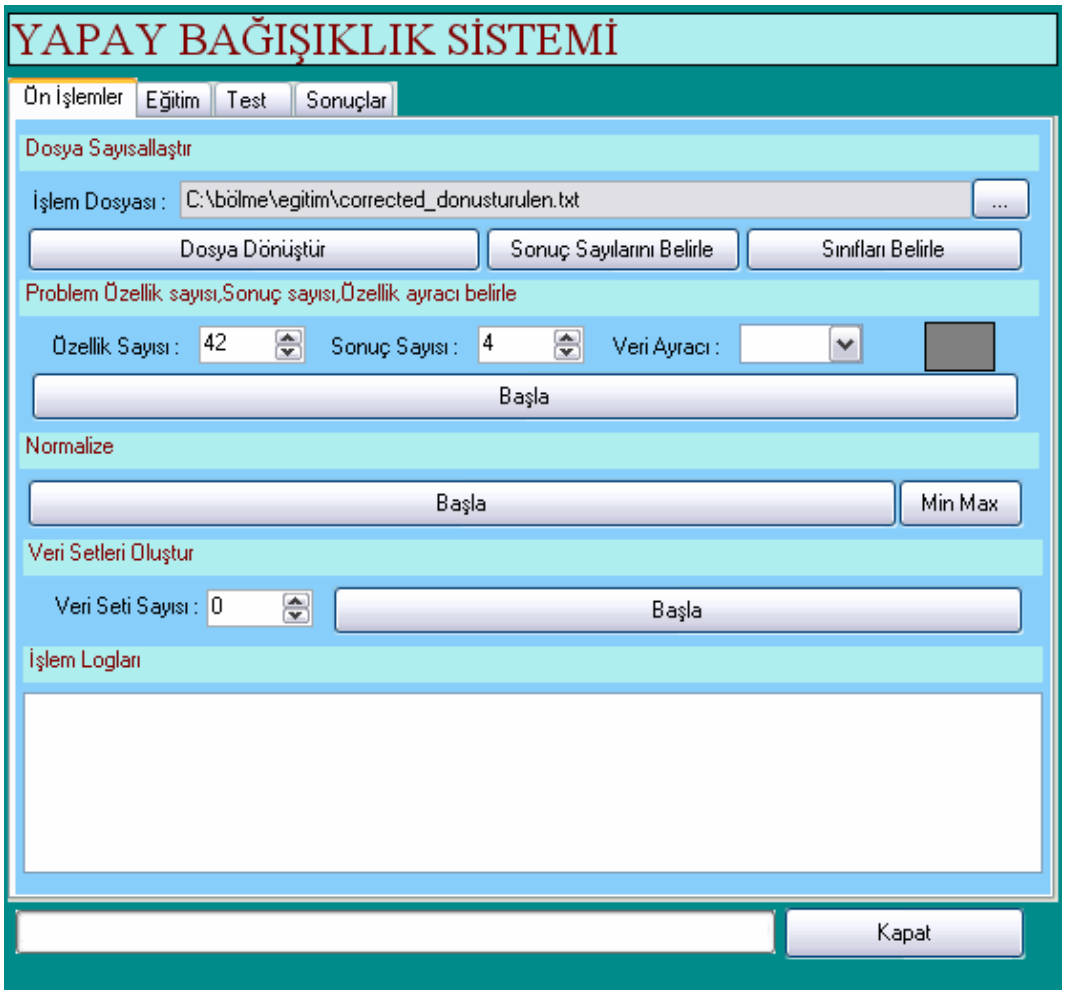
Adım 7 de eğitim ile oluşturulmuş son antibody populasyonu ile test verileri karşılaştırılır.Bu işlem için her bir populasyon verisi ile her bir test verisi sırasıyla karşılaştırılır eşik değer sınırları içerisinde olmak koşulu ile özellikler arasındaki farklardan benzerlik oranları bulunur.Bu oranlar her bir özellik için ayrı ayrı hesaplanır , toplanır ve ortalaması alınır,sonuç oran ilgili test verisinin karşılaştırıldığı eğitim verisinin sınıfına benzerlik oranıdır.Bu orani elde eden eğitim veri sınıfı,benzerlik oranı ile birlikte test verisinin yanındaki sutunlara yazılır.Elde edilen yeni başarı oranı eğer mevcut orandan yüksek ise bu test verisi bu sınıfa dahildir denir ve yeni oran yine yanındaki sütuna yazılır. Bu işlem test verisi her bir eğitim verisi ile karşılaştırılıncaya kadar devam eder. Test verisinin karşılaştırması sonucunda sınıflar artık belirlenmiştir.

4.4. YBS Similatörü

Uygulamanın takip ettiği adımlar şu şekildedir;

1. Eğitim veri setinin okunması ve normalize edilmesi
2. Eğitime başlanması
3. Eşik değerlerinin belirlenmesi
4. Mutasyon ve veri türetimi
5. Eğitimin tamamlanması
6. Test veri setinin okunması ve normalize edilmesi
7. Sınıflandırma işleminin tamamlanması
8. Sonuçların değerlendirilmesi

Uygulama ön işlemler arayüzü Şekil 4.2. de verilmiştir.



Şekil 4.2. Uygulama ön işlemler arayüzü

Ön işlemler arayüzünde eğitim ve test veri seti için ön işlem seçenekleri yer almaktadır. Bu şeçenekler şu şekildedir;

Dosya dönüştürme işlemi : Veri setleri içerisinde sayısal olmayan verileri sayısallaştırmak için kullanılır.Örneğin veri setlerinde yer alan saldırı isimleri bu şekilde sayısallaştırılmıştır.İşlem sonucunda oluşan veriler diske kaydedilir.

Sonuç sayılarının belirlenmesi işlemi : Veri setindeki verilerin özellik sayısı ,toplamda tekil kaç sınıfın yer aldığı ve özellikleri ayırmada kullanılan ayracın ne olduğu uygulama tarafından tespit edilebilmektedir.İşlem sonucu bilgileri diske kaydedilir.

Sınıfların belirlenmesi işlemi : Veri setinde hangi sınıftan ne kadar veri olduğunun belirlenme işlemidir.Sınıf sayıları elde edildikten sonra diske txt formatında kaydedilir.Benzer şekilde istatistiki calişmalarının yapılabilmesi için ikinci bir dosya olarak excell formatında kaydedilir.

Normalizasyon işlemi : Verilerin her bir özelliginin minimum ve maximum değerleri bulunmakta ve bu değerler esas alınarak her bir özellik 0 – 1 arasında normalize işlemine tabi tutulmaktadır. Normalizasyon sonucu da oluşan veri dosyası diske kaydedilir.

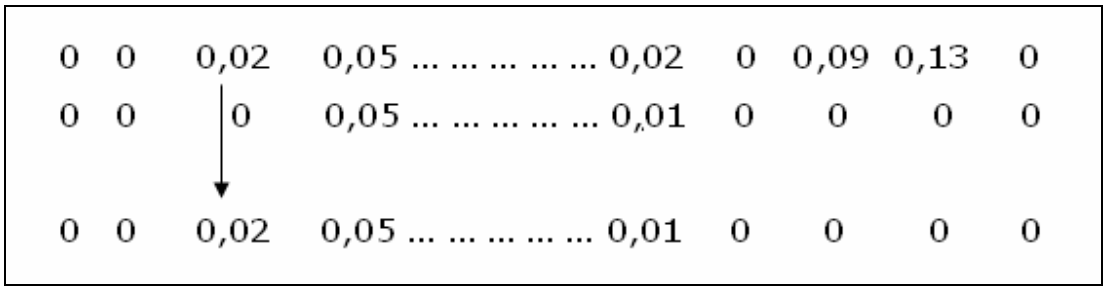
Veri setlerinin oluşturulması işlemi : Bu özellik öncelikle büyük veri setlerinin kullanılacağı durumlar için düşünülmüştür.Giriş olarak verilen veri dosyası, arayüzde belirlenen sayı kadar veri setlerine bölünmektedir ve sonuçlar ayrı dosyalarda arşivlenmektedir.

İşlem logları ile yapılan bu işlemlerin durumu, geldikleri aşama kullanıcıya bilgi amaçlı gösterilmektedir.

Eğitim dosyasının okunması : Yolu gösterilen eğitim veri dosyası sistem tarafından okunur ve özelliklerine ayrıştırılır.Bu sayede eğitim populasyonu ,ybs diliyle antibody başlangıç populasyonu oluşturulmuştur.

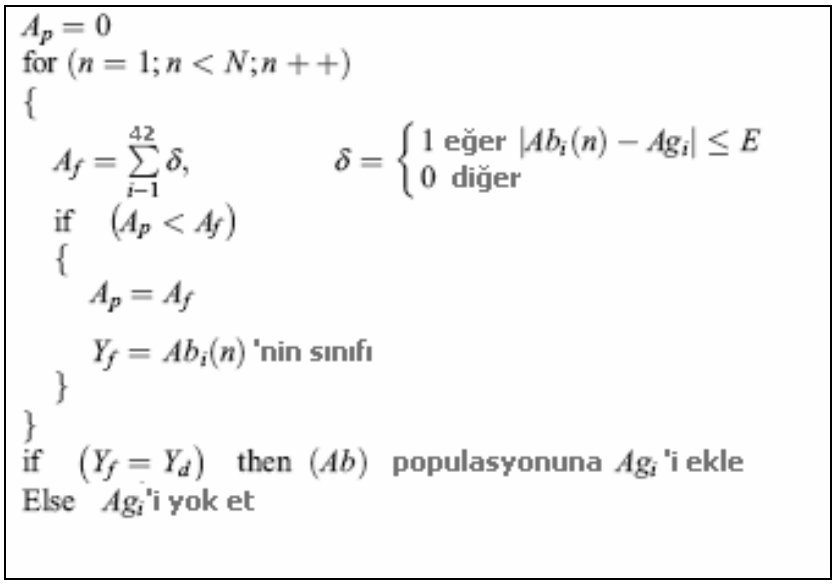
Eşik değer belirleme işlemi : Okunan eğitim ve test veri setleri baz alınarak öğrenme algoritmalarından ilki olan eşik değer belirleme işlemi gerçekleşir.Buradan elde edilen değerler belirli işlem aralıkları ile saklanır, bu sayede eğitim ile öğrenen sistemin öğrenme diagramı oluşturulur ve bu aralıklarda elde edilen test başarı durumu da diagram olarak oluşturulmaktadır.Buradan elde edeceğimiz bilgiler doğrultusunda eğitimi bitirmemiz gereken iterasyon sayısı ve maximum başarının sağlandığı eşik değerler belirlenmiş olur.Eşik değer belirleme işleminde Öklit bağıntısı kullanılmıştır.

Türetim işlemi : Oluşturulan antibody populasyonundan mutasyon ve klonlama teknikleri ile yeni türetimler yapılmakta ve eğitim populasyonu genişletilmeye çalışılmaktadır.Bu aşama ile eğitim sona ermiştir.Örnek türetim işlemi Şekil 4.3.'de gösterilmiştir.



Şekil 4.3. Türetim işlemi

Türetim işlemi sırasında türetilen yeni antibody hücresinin affinity kontrolü için kullanılan algoritma Şekil 4.4. 'de gösterilmiştir.



Şekil 4.4. Türetim işlemi sonrasında affinity kontrolü

Burada N eğitim veri setindeki veri sayısını, Ab_i (n) eğitim veri setindeki n. antibody verisini, Ag_i (n) klonlama sonrasında oluşturulan yeni antibody verisini, *Ap* başlangıç

affinity değerini , *Af* son affinity değerini, *Yf* son sınıf değerini, *Yd* türetim yapılan antibodynin sınıfını ifade etmektedir.

Test dosyasının okunması :Test veri dosyası sistem tarafından okunur ve özelliklerine ayrıştırılır.Sınıf sayıları belirlenir ve normalizasyon işlemine tabi tutulur. Bu sayede test populasyonu ,ybs diliyle sınıflandırılmayı bekleyen antijen populasyonu oluşturulmuş olur.

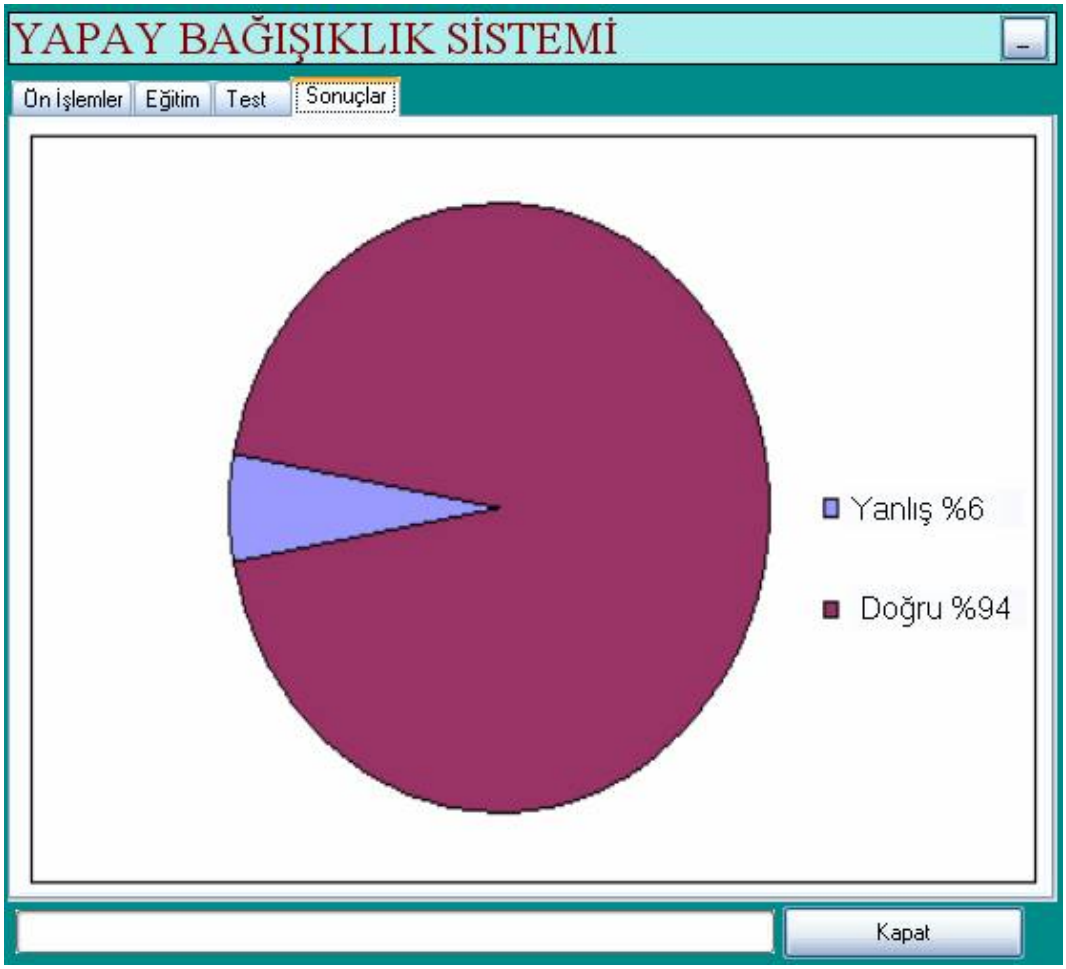
Test işlemi : Eğitim sonucunda oluşturulan antibody populasyonu baz alınarak test verileri sırayla karşılaştırılır ve sınıflama işlemi gerçekleştirilir.İşlem sonucunda elde edilen başarı oranları listelenir.Sınıflandırma işlemini gerçekleştiren YBS algoritması Şekil 4.5. 'te gösterilmiştir

for$(t = 1; t < T; t++)$
{$A_p = 0$
for $(n = 1; n < N; n++)$
{
$A_f = \sum_{i=1}^{42} \delta,$ $\delta = \begin{cases} 1 \text{ eğer } |Ab_i(n) - Ag_i(t)| \leq E \\ 0 \text{ diğer} \end{cases}$
if $(A_p < A_f)$
{
$A_p = A_f$
$Y_f = Ab_i(n)$'nin sınıfı
}
}
$Ag_i(t)$'nin sınıfı $= Y_f$
}

Şekil 4.5. Sınıflandırma algoritması

Burada T son eğitim veri setindeki veri sayısını,N test veri setindeki veri sayısını, Ab_i (t) son eğitim veri setindeki t. antibody verisini, Ag_i (n) test veri setindeki n. antibody verisini, *Ap* başlangıç benzerlik değerini , *Af* son benzerlik değerini, *Yf* test verisinin belirlenen sınıf bilgisini ifade etmektedir.

Sonuç menüsünde ise elde edilen başarı oranları grafiksel olarak gösterilmektedir. Geliştirilen STS'nin YBS modülü eğitim ve test sonucu değerlendirme arayüzü Şekil 4.6.'da gösterilmiştir.



Şekil 4.6. Sonuç ve başarı değerlendirme arayüzü

Şekil 4.6.'da gösterilen sonuç ve başarı değerlendirme arayüzü, test edilen veri kümesi sonuçlarının, elde edilmek istenen sonuçlar ile karşılaştırılarak, sınıflandırma işleminde doğruluk ve hata yüzde oranlarını vermektedir.

BÖLÜM 5. SONUÇLAR

Eğitim ve test aşamasında kullanılan veri setleri KDD ‘99 organizasyonu sonucu belirlenen veri setleridir. Organizasyon sonucunda oluşturulmuş veri setleri veri örnekleri açısından zengindir. STS çalışması yapan araştırmacıların kullanabilmesi amacıyla geniş veri setleri oluşturulmuştur. YBS sistemlerinin diğer zeki sistemlerde olduğu gibi öğrenebilmesi için veri setlerinin dikkatli oluşturulması gerekmektedir. KDD ’99 veri setleri uzun süren çalışmalar sonucunda bu önem dikkate alınarak oluşturulmuştur.Veri setlerinde sınıfların homojen dağılımı esas alınmıştır.Bu sayede veri setlerinde her sınıftan örneklerin yer alması da sağlanmıştır.Bu durum eğitimi olumlu yönde etkilemiştir. Eğitim ve test işlemi için birer veri seti kullanılmıştır.

Veri örnekleri normalizasyon işleminden geçirilmiştir.Bu işlem sayesinde eğitim aşamalarından olan eşik değer belirleme aşamasında geçen süreyi farkedilir derecede kısaltmıştır. Normalizasyon uygulanma durumuna göre sistemin elde ettiği sonuçlar incelenmiş ve başarı sonuçları Tablo 5.1 ve Tablo 5.2 ‘de verilmiştir.

Tablo 5.1 Normalizasyon işlemi yapılmadan sistemden elde edilen sonuçlar

Sınıf	0	1	2	3	4	başarı %
0	60342	186	58	3	4	99,59
1	319	3755	92	0	0	90,13
2	5456	984	223413	0	0	97,20
3	131	14	0	76	7	33,33
4	10848	263	0	7	5174	31,76

Tablo 5.2 Normalizasyon işlemi yapılarak sistemden elde edilen sonuçlar

Sınıf	0	1	2	3	4	başarı %
0	60351	178	57	3	4	99,60
1	316	3758	92	0	0	90,21
2	5451	984	223418	0	0	97,20
3	126	14	0	81	7	35,53
4	10841	263	0	7	5181	31,80

Normalizasyon sonucunda beklendiği gibi normalizasyon yapılan sistemin sonuçları ile normalizasyon işleminin yapılmadığı sistemin başarı sonuçları arasında önemli bir fark bulunmamaktadır. Geniş veri setlerinin kullanılması durumunda sistemin eğitim süresini kısaltmak amacıyla normalizasyon işlemi kullanılabilir. Diğer karşılaştırma işlemlerinde sistemin normalizasyon uygulanmış sonuçları karşılaştırılacaktır.

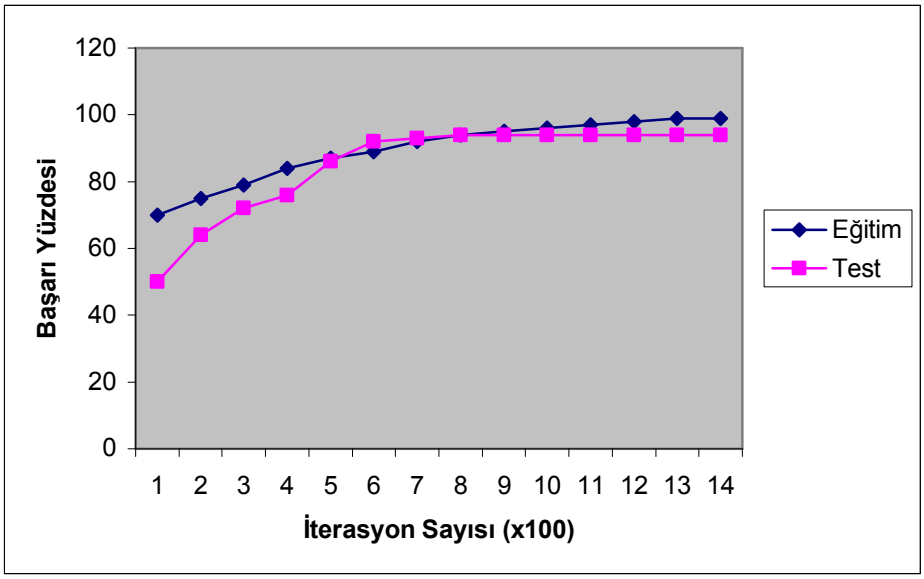
Öğrenme aşamasına geçilmesi ile birlikte sistemin eğitim işleminden elde ettiği sonuçlar Tablo 5.3'de verilmiştir.

Tablo 5.3 Eğitim sonuçları

	0	1	2	3	4	Başarı %
0	96986	256	36	0	0	99,7
1	306	3778	23	0	0	92
2	4376	321	386761	0	0	98,8
3	27	1	0	24	0	47
4	624	18	0	0	484	43

Eğitim aşaması sonuçları incelendiğinde sistemin eğitim sonrasında yine eğitim verilerine verdiği sonuçlar başarılı olarak nitelendirilebilir. Ancak bu tür sistemlerde, sistemin eğitim veri setini ezberleme olasılığı da bulunmaktadır. Bu durumun anlaşılabilmesi için eğitim sürecinde, belirlenen iterasyon aralıklarında test verileri ile sistem test edilmelidir. Eğitim başarı eğrisi çıkarılır ve o andaki test başarı durumu takip edilirse eğitimin nerede durdurulacağına karar verilebilir ve sistemin

ezber yapması engellenmiş olur. Bu ihtiyaç belirlenmiş ve sistemin eğitim süreci izlemeye alınmıştır. Sistemin eğitim eğrisi Şekil 5.1'de verilmiştir.



Şekil 5.1 Eğitim sürecinde eğitim ve test başarı değişimi

Eğitim süreci izlendiğinde, zeki tekniklerin kullanıldığı sistemlerde, eğitimin devam ettirilmesi durumunda gerçekleşen ezberleme durumu, YBS sisteminde açıkca görülmemiştir. Bu duruma sınıfların veri setlerine homojen dağılımı en büyük etkendir. Eğitim sürecinde dikkat edilen bir diğer durum eğitimin devam ettirilmesine karşın test başarı oranı %94 değerini aşmamıştır. Eğitim başarı oranı %99 olarak belirlenmiştir. Eğitim işlemleri herhangi bir başarı artışının artık yaşanmadığı 14 nolu iterasyon aşamasında durdurulmuştur.

Geliştirilen uygulama sonucunda elde edilen test sonuçları Tablo 5.4.'de sunulmuştur.

Tablo 5.4 Simülatör test sonuçları

Sınıf	0	1	2	3	4	başarı %
0	60351	178	57	3	4	99,60
1	316	3758	92	0	0	90,21
2	5451	984	223418	0	0	97,20
3	126	14	0	81	7	35,53
4	10841	263	0	7	5181	31,80

Yüksek başarı ile öğrenen sistemin, Tablo 5.1.'de gösterilen test veri kümesi sonuçları ile başarılı olduğunu ve eğitim setinde olmayan yeni saldırıları tiplerinin de algılandığını göstermektedir. KDD '99 organizasyonunda belirlenen en yüksek başarı değerine sahip olan çalışmanın sonuç değerleri Tablo 5.5. 'de verilmiştir.

Tablo 5.5. KDD '99 en yüksek başarı oranını gerçekleştiren çalışmanın sonuçları

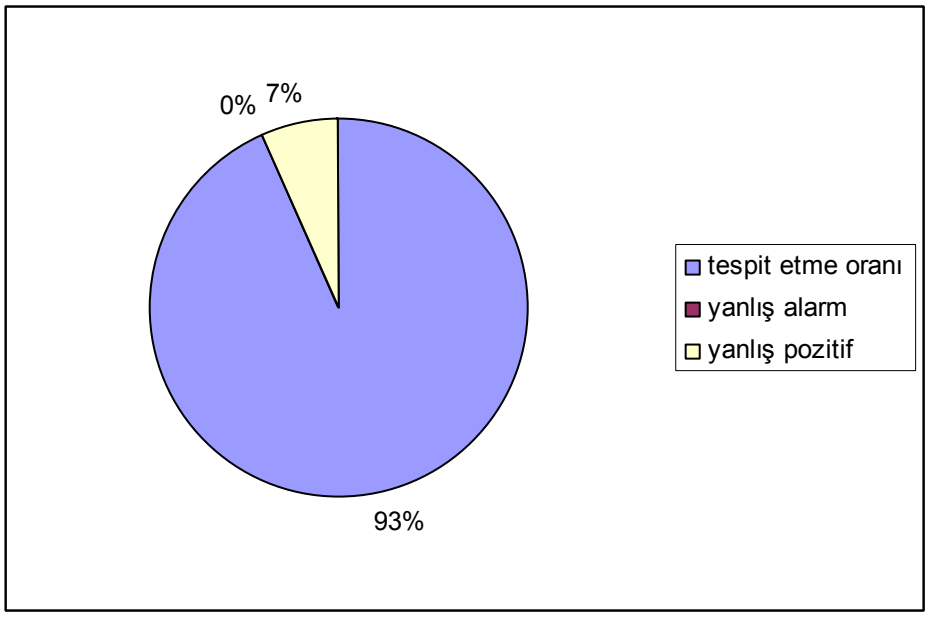
Sınıf	0	1	2	3	4	başarı %
0	60262	243	78	4	6	99,5
1	511	3471	184	0	0	83,3
2	5299	1328	223226	0	0	97,1
3	168	20	0	30	10	13,2
4	14527	294	0	8	1360	8,40

İki sistem birbiriyle karşılaştırıldığında YBS sisteminin KDD '99 organizasyonunda belirlenen en başarılı yöntemden daha başarılı olduğu görülmektedir. Nitekim test sonuçlarına göre 3 ve 4 nolu sınıfların başarı değerlerinin halen düşük seviyelerde olduğu görülmektedir. Bu durumun sebebi bu sınıflara ait veri setlerinde az sayıda veri bulunması en büyük etkendir. Bunun dışında sınıflandırma işleminde özellik değerlerinin ağırlıkları eşit olarak değerlendirilmiştir ve bir sınıfa dahil ederken ortalama benzerlik dikkate alınmıştır. Eğer özellikler farklı ağırlık değerlerine sahip ise ortalama benzerlik 3 ve 4 nolu sınıfların belirlenmesini etkilemiş olabilir ve ağırlıklar özelliklere dağıtmış diğer sınıflar bu durumda ortalama benzerlik

karşılaştırmasında daha yüksek başarıya ulaşmış olabilmektedir. Bu durum eğitimi etkilemiştir. Başarı oranlarının yükselebilmesi için bu sınıflara ait daha çok örnek veri , veri setlerine dahil edilebilir ve sınıflandırmada özelliklerin ağırlık değerleri de dikkate alınabilir.

Sistemden elde edilen sonuçlardan birisi de sistemin STS sistemi olarak kullanılması durumunda elde edilebilen değerlerdir. Bu bilgiler şu şekildedir,

Tespit etme oranı: Normal olan bir durumun normal, saldırı olan bir durumun da saldırı olarak sistem tarafından belirlenebilmesi,
Yanlış alarm: Saldırı olmadığı halde sistemin saldırı olarak yorumladığı durumlar,
Yanlış pozitif: Saldırı olan bir durumun normal olarak değerlendirilmesidir.



Şekil 5.2. STS değerlendirme sonuçları

Şekil 5.2.'de gösterilen STS değerlendirme sonucuna göre test edilen veri kümesi sonuçlarının, elde edilmek istenen sonuçlar ile karşılaştırılarak, tespit etme oranı, yanlış alarm ve yanlış negatif oranları hesaplanmıştır. Sonuçlar sistemin bilgisayar ve bilgi güvenliğinde kullanılabilecek başarılı bir STS uygulaması olduğunu göstermektedir.

Elde edilen sonuçlar sistemin başarılı olduğunu ve YBS'nin zeki STS'ler için oldukça başarılı bir yöntem olduğunu göstermektedir.

Geliştirilen sistem aynı zamanda genel bir sınıflandırma ve tanıma sistemi olarak kullanılabilir. Değişik problemler ile ilgili veri setlerinin sisteme giriş olarak verilmesi durumunda sistem ilgili problem için eğitilebilir, ilgili problemi öğrenebilir. Bu özelliği ile herhangi bir sınıflandırma ve tanıma problemi için genel bir simülatördür. Uygulama benzer şekilde hastalık tanısı yapmak amacıyla başka çalışmalarda da kullanılmıştır[59].

BÖLÜM 6. TARTIŞMALAR VE ÖNERİLER

Yapılan çalışmada, hızlı ve etkili bir problem çözme tekniği olan yapay bağışıklık sistemleri incelenmiş,literatür incelemesinden elde edilen bilgiler doğrultusunda, YBS tabanlı saldırı sınıflandırma yapabilen zeki bir sistem geliştirilmiştir.. Yapay bağışıklık sistemleri, yapay sinir ağları ve genetik algoritmalara benzer şekilde, bazı biyolojik sistemlerin özet modelleridir ve birçok alanda uygulaması bulunmaktadır. Çalışmada öncelikle Yapay Bağışıklık Sistemleri'nde kullanılan doğal bağışıklık mekanizmaları, bu mekanizmalara dayanan algoritmalar verilmiştir. Bu modeller ile farklı problemlerin çözümleri için yapılmış çalışmalardan bahsedilmiştir. Bu çalışmalardaki sonuçlara göre YBS farklı uygulama alanlarında oldukça tatmin edici sonuçlar vermiştir. Gerçekleştirilen STS tasarımından elde edilen sonuçlar çalışma içerisinde sunulmuştur. Elde edilen sonuçlardan;

YBS'nin saldırı tespit tasarımında başarılı olduğu görülmüştür.

YBS'nin başarısında saldırıların doğru şekilde tespit edilebilmesi için uygun veri kümesi tasarımının önemli olduğu görülmüştür.

YBS'nin sınıflandırma kriterlerinden olan eşik değer gibi değerlerin sistemden sisteme ve farklı uygulama alanlarına göre değişiklik göstereceği bilinmektedir.

Normalizasyon işlemleri YBS'nin başarı oranında önemli bir değişim yaratmaması karşın, eğitim sürecinde geçen süreyi önemli ölçüde azaltmaktadır.

Veri setlerindeki özelliklerin sınıfları belirleyicilik ağırlıkları belirlebilir ise elde edilen başarı oranı arttırılabilir.

Eğitim ve test verilerinin sayısının fazla olmasına karşın tamamının kullanılmasının yüksek hesaplama zamanı gerektirdiği ve yapılan testler sonucunda az sayıda örneğin olduğu veri kümeleri kullanıldığında saldırı tespitinde başarının düştüğü görülmüştür.

Tasarlanan YBS yapısı kolaylıkla gerçek-zamanlı uygulamalarda kullanılabilmektedir.

Kullanılabilecek birden çok öğrenme algoritması olduğundan dolayı, farklı algoritmalarda başarı ortalamasının değişebileceği görülmüştür.

Çalışma genel olarak değerlendirildiğinde;

Ülkemizde saldırı tespit sistemlerine yönelik yeterli çalışmanın bulunmadığı görülmüştür.

Ülkemizde saldırı tespit sistemlerine yönelik kullanılabilecek bir veri kümesi tasarımının bulunmadığı görülmüştür.

Literatürde, veri kümesi oluşturmak için yapılan çalışmaların güncel olmadığı görülmüştür.

Literatürde, kullanılabilecek veri kümelerini isteğe uygun formatlara dönüştürmek için bir yazılımın bulunmadığı, bu sebepten dolayı format dönüşümleri için uzun çalışma sürelerinin gerektiği anlaşılmıştır.

EKLER

EK-1 DARPA 1998 örnek veri kümesi

1 01/23/1998 16:56:12 00:01:26 telnet 1754 23 192.168.1.30 192.168.0.20 0 -
2 01/23/1998 16:56:15 00:00:13 ftp 1755 21 192.168.1.30 192.168.0.20 0 -
3 01/23/1998 16:56:17 00:00:01 smtp 43493 25 192.168.0.40 192.168.1.30 0 -
4 01/23/1998 16:56:17 00:00:00 auth 1756 113 192.168.1.30 192.168.0.40 0 -
5 01/23/1998 16:56:19 00:00:01 smtp 43494 25 192.168.0.40 192.168.1.30 0 -
6 01/23/1998 16:56:19 00:00:00 auth 1761 113 192.168.1.30 192.168.0.40 0 -
7 01/23/1998 16:56:19 00:00:01 ftp-data 20 1762 192.168.0.20 192.168.1.30 0 -
8 01/23/1998 16:56:22 00:00:00 ftp-data 20 1767 192.168.0.20 192.168.1.30 0 -
9 01/23/1998 16:56:24 00:00:02 ftp-data 20 1768 192.168.0.20 192.168.1.30 0 -
10 01/23/1998 16:56:25 00:01:01 telnet 1769 23 192.168.1.30 192.168.0.20 0 -
11 01/23/1998 16:56:27 00:00:00 ftp-data 20 1770 192.168.0.20 192.168.1.30 0 -
12 01/23/1998 16:56:36 00:00:03 finger 1772 79 192.168.1.30 192.168.0.20 0 -
13 01/23/1998 16:56:42 00:00:03 smtp 1778 25 192.168.1.30 192.168.0.20 0 -
14 01/23/1998 16:56:43 00:00:03 smtp 1783 25 192.168.1.30 192.168.0.20 0 -
15 01/23/1998 16:56:45 00:00:00 http 1784 80 192.168.1.30 192.168.0.40 1 phf
16 01/23/1998 16:56:49 00:00:14 ftp 43504 21 192.168.0.40 192.168.1.30 0 -
17 01/23/1998 16:56:56 00:00:00 ftp-data 20 43505 192.168.1.30 192.168.0.40 0 -
18 01/23/1998 16:56:57 00:00:00 ftp-data 20 43506 192.168.1.30 192.168.0.40 0 -
19 01/23/1998 16:56:59 00:00:00 ftp-data 20 43508 192.168.1.30 192.168.0.40 0 -
21 01/23/1998 16:57:00 00:00:00 ftp-data 20 43509 192.168.1.30 192.168.0.40 0 -
22 01/23/1998 16:57:02 00:00:00 ftp-data 20 43510 192.168.1.30 192.168.0.40 0 -
24 01/23/1998 16:57:13 00:00:48 telnet 43516 23 192.168.0.40 192.168.1.30 0 -
25 01/23/1998 16:57:15 00:00:12 ftp 1787 21 192.168.1.30 192.168.0.20 0 -
26 01/23/1998 16:57:16 00:00:01 http 1788 80 192.168.1.30 192.168.0.40 0 -
27 01/23/1998 16:57:19 00:00:02 http 1789 80 192.168.1.30 192.168.0.40 0 -
29 01/23/1998 16:57:20 00:00:05 smtp 43519 25 192.168.0.40 192.168.1.30 0 -
30 01/23/1998 16:57:22 00:00:00 auth 1790 113 192.168.1.30 192.168.0.40 0 -
31 01/23/1998 16:57:23 00:00:02 http 1796 80 192.168.1.30 192.168.0.40 0 -
32 01/23/1998 16:57:24 00:00:00 ftp-data 20 1801 192.168.0.20 192.168.1.30 0 -
33 01/23/1998 16:57:26 00:00:00 ftp-data 20 1802 192.168.0.20 192.168.1.30 0 -
34 01/23/1998 16:57:27 00:00:02 http 43521 80 192.168.0.40 192.168.1.30 0 -
35 01/23/1998 16:57:27 00:00:03 http 1804 80 192.168.1.30 192.168.0.40 0 -
36 01/23/1998 16:57:31 00:00:01 http 43522 80 192.168.0.40 192.168.1.30 0 -
37 01/23/1998 16:57:34 00:00:02 http 1806 80 192.168.1.30 192.168.0.40 0 -
38 01/23/1998 16:57:37 00:00:02 http 43524 80 192.168.0.40 192.168.1.30 0 -
39 01/23/1998 16:57:37 00:00:02 http 1807 80 192.168.1.30 192.168.0.40 0 -
41 01/23/1998 16:57:40 00:00:02 http 43525 80 192.168.0.40 192.168.1.30 0 -
42 01/23/1998 16:57:41 00:00:02 http 1808 80 192.168.1.30 192.168.0.40 0 -

EK-1 (Devam) DARPA 1998 örnek veri kümesi

43 01/23/1998 16:57:44 00:00:03 http 43526 80 192.168.0.40 192.168.1.30 0 -
44 01/23/1998 16:57:45 00:00:01 http 1810 80 192.168.1.30 192.168.0.40 0 -
45 01/23/1998 16:57:47 00:00:00 finger 1811 79 192.168.1.30 192.168.0.20 0 -
46 01/23/1998 16:57:48 00:00:03 http 43527 80 192.168.0.40 192.168.1.30 0 -
47 01/23/1998 16:57:48 00:00:02 http 1814 80 192.168.1.30 192.168.0.40 0 -
48 01/23/1998 16:57:52 00:00:02 http 43528 80 192.168.0.40 192.168.1.30 0 -
49 01/23/1998 16:57:53 00:00:03 http 1816 80 192.168.1.30 192.168.0.40 0 –
50 01/23/1998 16:57:55 00:00:03 http 1818 80 192.168.1.30 192.168.0.40 0 -
51 01/23/1998 16:57:55 00:00:01 finger 1820 79 192.168.1.30 192.168.0.20 0 -
53 01/23/1998 16:57:57 00:00:02 smtp 1826 25 192.168.1.30 192.168.0.20 0 -
54 01/23/1998 16:57:59 00:00:02 http 1830 80 192.168.1.30 192.168.0.40 0 -
55 01/23/1998 16:57:59 00:00:04 smtp 1832 25 192.168.1.30 192.168.0.20 0 -
56 01/23/1998 16:57:59 00:00:03 http 1833 80 192.168.1.30 192.168.0.40 0 -
57 01/23/1998 16:58:02 00:00:03 finger 1834 79 192.168.1.30 192.168.0.20 0 -
58 01/23/1998 16:58:03 00:00:02 http 1835 80 192.168.1.30 192.168.0.40 0 -
59 01/23/1998 16:58:03 00:00:02 http 1836 80 192.168.1.30 192.168.0.40 0 -
60 01/23/1998 16:58:04 00:00:01 http 43529 80 192.168.0.40 192.168.1.30 0 -
61 01/23/1998 16:58:06 00:00:02 http 1837 80 192.168.1.30 192.168.0.40 0 -
62 01/23/1998 16:58:06 00:00:02 http 1838 80 192.168.1.30 192.168.0.40 0 -
63 01/23/1998 16:58:07 00:00:01 http 43530 80 192.168.0.40 192.168.1.30 0 -
64 01/23/1998 16:58:10 00:00:01 http 1839 80 192.168.1.30 192.168.0.40 0 -
65 01/23/1998 16:58:11 00:00:01 finger 1841 79 192.168.1.30 192.168.0.20 0 -
66 01/23/1998 16:58:13 00:00:02 http 1844 80 192.168.1.30 192.168.0.40 0 -
67 01/23/1998 16:58:13 00:00:19 ftp 43532 21 192.168.0.40 192.168.1.30 0 -
68 01/23/1998 16:58:16 00:00:03 http 1846 80 192.168.1.30 192.168.0.40 0 -
69 01/23/1998 16:58:18 00:00:04 finger 1847 79 192.168.1.30 192.168.0.20 0 -
70 01/23/1998 16:58:20 00:00:02 http 1848 80 192.168.1.30 192.168.0.40 0 -
71 01/23/1998 16:58:20 00:00:00 ftp-data 20 43534 192.168.1.30 192.168.0.40 0 -
72 01/23/1998 16:58:21 00:00:02 http 1849 80 192.168.1.30 192.168.0.40 0 -
73 01/23/1998 16:58:22 00:00:17 ftp 1850 21 192.168.1.30 192.168.0.20 0 -
74 01/23/1998 16:58:23 00:00:02 http 1851 80 192.168.1.30 192.168.0.40 0 -
75 01/23/1998 16:58:24 00:00:02 http 1852 80 192.168.1.30 192.168.0.40 0 -
76 01/23/1998 16:58:27 00:00:03 http 1853 80 192.168.1.30 192.168.0.40 0 -
77 01/23/1998 16:58:28 00:00:02 finger 1855 79 192.168.1.30 192.168.0.20 0 -
78 01/23/1998 16:58:28 00:00:00 ftp-data 20 43536 192.168.1.30 192.168.0.40 0 -
79 01/23/1998 16:58:28 00:00:01 ftp-data 20 1854 192.168.0.20 192.168.1.30 0 -
80 01/23/1998 16:58:31 00:00:00 ftp-data 20 43537 192.168.1.30 192.168.0.40 0 -
81 01/23/1998 16:58:31 00:00:02 http 1857 80 192.168.1.30 192.168.0.40 0 -
82 01/23/1998 16:58:31 00:00:01 ftp-data 20 1856 192.168.0.20 192.168.1.30 0 -
83 01/23/1998 16:58:34 00:00:00 ftp-data 20 1858 192.168.0.20 192.168.1.30 0 -
84 01/23/1998 16:58:34 00:00:02 http 1859 80 192.168.1.30 192.168.0.40 0 -
85 01/23/1998 16:58:36 00:00:00 ftp-data 20 1860 192.168.0.20 192.168.1.30 0 -
86 01/23/1998 16:58:38 00:00:02 http 1861 80 192.168.1.30 192.168.0.40 0 -
87 01/23/1998 16:58:38 00:00:00 ftp-data 20 1863 192.168.0.20 192.168.1.30 0 -
88 01/23/1998 16:58:41 00:00:02 http 1864 80 192.168.1.30 192.168.0.40 0 -
89 01/23/1998 16:58:44 00:00:02 http 1866 80 192.168.1.30 192.168.0.40 0 -
90 01/23/1998 16:58:45 00:00:22 telnet 1867 23 192.168.1.30 192.168.0.20 1 guess

EK-1 (Devam) DARPA 1998 örnek veri kümesi

91 01/23/1998 16:58:47 00:00:02 http 1868 80 192.168.1.30 192.168.0.40 0 -
92 01/23/1998 16:58:48 00:00:02 http 1869 80 192.168.1.30 192.168.0.40 0 -
93 01/23/1998 16:58:48 00:00:11 ftp 43540 21 192.168.0.40 192.168.1.30 0 -
94 01/23/1998 16:58:51 00:00:02 http 1870 80 192.168.1.30 192.168.0.40 0 -
95 01/23/1998 16:58:52 00:00:00 ftp-data 20 43542 192.168.1.30 192.168.0.40 0 -
96 01/23/1998 16:58:54 00:00:02 http 1873 80 192.168.1.30 192.168.0.40 0 -
97 01/23/1998 16:58:57 00:00:00 ftp-data 20 43544 192.168.1.30 192.168.0.40 0 -
98 01/23/1998 16:58:57 00:00:01 http 1874 80 192.168.1.30 192.168.0.40 0 –
100 01/23/1998 16:59:00 00:00:02 http 1875 80 192.168.1.30 192.168.0.40 0 –

EK-2 Saldırılar ve açıklamaları

Back	Denial of service attack against apache webserver where a client requests a URL containing many backslashes.
Dict	Guess passwords for a valid user using simple variants of the account name over a telnet connection.
Eject	Buffer overflow using eject program on Solaris. Leads to a user->root transition if successful.
Ffb	Buffer overflow using the ffbconfig UNIX system command leads to root shell
Format	Buffer overflow using the fdformat UNIX system command leads to root shell
ftp-write	Remote FTP user creates .rhost file in world writable anonymous FTP directory and obtains local login.
Guest	Try to guess password via telnet for guest account.
Imap	Remote buffer overflow using imap port leads to root shell
Ipsweep	Surveillance sweep performing either a port sweep or ping on multiple host addresses.
Land	Denial of service where a remote host is sent a UDP packet with the same source and destination
Loadmodule	Non-stealthy loadmodule attack which resets IFS for a normal user and creates a root shell
Multihop	Multi-day scenario in which a user first breaks into one machine
Neptune	Syn flood denial of service on one or more ports.
Nmap	Network mapping using the nmap tool. Mode of exploring network will vary--options include SYN
Perlmagic	Perl attack which sets the user id to root in a perl script and creates a root shell
Phf	Exploitable CGI script which allows a client to execute arbitrary commands on a machine with a misconfigured web server.
Pod	Denial of service ping of death
Portsweep	Surveillance sweep through many ports to determine which services are supported on a single host.
Rootkit	Multi-day scenario where a user installs one or more components of a rootkit
Satan	Network probing tool which looks for well-known weaknesses. Operates at three different levels. Level 0 is light
Smurf	Denial of service icmp echo reply flood.
Spy	Multi-day scenario in which a user breaks into a machine with the purpose of finding important information where the user tries to avoid detection. Uses several different exploit methods to gain access.
Syslog	Denial of service for the syslog service connects to port 514 with unresolvable source ip.
Teardrop	Denial of service where mis-fragmented UDP packets cause some systems to reboot.
Warez	User logs into anonymous FTP site and creates a hidden directory.
Warezclient	Users downloading illegal software which was previously posted via anonymous FTP by the warezmaster.
Warezmaster	Anonymous FTP upload of Warez (usually illegal copies of copywrited software) onto FTP server.

KAYNAKLAR

[1] SAĞIROĞLU, Ş., ALKAN, M., Her yönüyle elektronik imza (e-imza), Grafiker Yayınları, Ankara, 1-100 ,2005.

[2] ENDORF, C., SCHULTZ, E., MELLANDER, J., Intrusion Detection & Prevention, Jenn Tust, Jody McKenzie, Elizabeth Seymour, McGraw-Hill, California, 10-150, 2004.

[3] LEE, W., STOLFO, S.J., CHAN, P.K., ESKIN, E., FAN, W., MILLER, M., HERSHKOP, S., ZHANG, J., Real time data mining-based intrusion detection, Second {DARPA} Information Survivability Conference and Exposition (DISCEX II), Anaheim, CA, 89-100 ,2001.

[4] ILGUN, K., KEMMERER, R., PORRAS, P., State Transition Analysis: A RuleBased Intrusion Detection System, Software Engineering, 21(3): 181-199 ,1995.

[5] Knowledge Discovery and Delivery, KDD Cup 1999: General Information,http://www.sigkdd.org/kddcup/index.php?section=1999&method =info ,Ekim 2008.

[6] PORRAS, P.A., STAT: A State Transition Analysis Tool for intrusion detection, Yüksek Lisans Tezi, Computer Science Department, University of California, Santa Barbara, 1-150 ,1992.

[7] MELL, P., HU, V., LIPMANN, R.,HAINES, J.,ZISSMAN, M., An overview of issues in testing intrusion detection systems, Technical Report NIST IR 7007,National Institute of Standard and Technology, 1-18 ,2003.

[8] MAHONEY, M.V., CHAN, P.K., An analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for network anomaly detection", Recent Advances in Intrusion Detection (RAID2003), Lecture Notes in Computer Science., Springer-Verlag, 2820, 220-237 ,2003.

[9] ANDERSON, D., LUNT, T.F., JAVITZ, H., TAMARU, A., VALDES, A., Detecting unusual program behavior using the statistical component of the nextgeneration intrusion detection expert system (NIDES), SRI-CSL-95-06, Menlo Park, California, 1-22 ,1995.

[10] GUVEN, E.N., Zeki Saldırı Tespit Sistemlerinin İncelenmesi Tasarımı ve Gerçekleştirilmesi, Ankara, 2007.

[11] ANDERSON, J.P., "Computer security threat monitoring and surveillance", Technical Report, Fort Washington, Pennsylvania, 1-30 ,1980.

[12] 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.

[13] LUNT, T.F., Automated audit trail analysis and intrusion detection: A survey, 11th National Computer Security Conference, Baltimore, MD, 65-73 ,1988.

[14] DENNING, D.E., An intrusion detection model, IEEE Transactions on Software Engineering, 13(2): 118–131 ,1987.

[15] MUKHERJEE, B., HEBERLIN, L.T.,LEVITT, K.N., Network intrusion detection, IEEE Network, 8(3): 26-41 ,1994.

[16] CROSBIE, M., SPAFFORD, E.H., Defending a computer system using autonomous agents, Technical Report 95-022, Dept. of Comp. Sciences, Purdue University, West Lafayette, 1-11 ,1995.

[17] ENDLER, D., Intrusion detection applying machine learning to solaris audit data, 1998 Annual Computer Security Applications Conference (ACSAC'98), 268-269,1998.

[18] 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.

[19] PATCHA, A., PARK, J.M., An overview of anomaly detection techniques: Existing solutions and latest technological trends, Computer Networks, 51(12): 3448-3470 ,2007.

[20] HOFMEYR, S.A., An immunological model of distributed detection and its application to computer security, Doktora Tezi, Computer Science, University of New Mexico, 1-69 ,1999.

[21] SUNDARAM, A., An introduction to intrusion detection, Crossroads: The ACM Student Magazine, New York, USA, 2(4), 3-7 ,1996.

[22] BACE, R.,Intrusion Detection, Macmillan Technical Publishing, Indianapolis USA,2000.

[23] BACE, R. and MELL, P., Intrusion Detection Systems, NIST Special Publication on Intrusion Detection Systems,2001.

[24] MCHUGH J., CHRISTIE A., and ALLEN J., Defending Yourself: The

Role of Intrusion Detection Systems, IEEE Software, 17(5), 42- , SP 800-31, Gaithersburg,2000.

[25] ROESCH, M., Snort – Lightweight Intrusion Detection for Networks, In Proceedings of the 13th LISA Conference of USENIX Association, Berkeley, CA, USA, 07 – 12 November, pp. 229-238,1999.

[26] RUSSELL, R., Snort Intrusion Detection 2.0, Syngress Publishing, Inc., Rockland, MA,2003.

[27] POSTEL, J., Simple Mail Transfer Protocol, Internet Engineering Task Force Request for Comments, STD 10, RFC 821, Information Sciences Institute University of Southern California , California,1982.

[28] REHMAN, R. U., Intrusion Detection Systems with Snort, Publishing as Prentice Hall PTR, Upper Saddle River, New Jersey,2003.

[29] DAYIOĞLU, B., Use Of Passive Network Mapping To Enhance Network Intrusion Detection, Master Thesis, The Graduate School Of Natural And Applied Sciences of The Middle East Technical University, Ankara,2001.

[30] MAHONEY, M.V., CHAN, P.K., PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic, Florida Tech. Technical Report, CS-2001-04, Melbourne, Florida,2001.

[31] CABRERA, B.D., CABRERA, L. LEWIS and R.K. MEHRA, Detection and classification of intrusions and faults using sequence of system calls, ACM SIGMOD record, 30(4): 25-34 ,2001.

[32] BACE, R., MELL, P., Intrusion detection systems, Technical Report, National Institute of Standards and Technology, NIST SP300-31, Scotts Valley, CA, 5- 46 ,2001.

[33] Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları ,Off-line intrusion detection evaluation data, http://www.ll.mit.edu/IST/ideval/ ,Ekim 2008.

[34] MUKKAMALA, S., JANOSKI, G., SUNG, A., Intrusion detection using neural networks and support vector machines, IEEE International Joint Conference on Neural Networks, IEEE Computer Society Press, 1702-1707 ,2002.

[35] MITRE-CWE, Common Weakness Enumeration, ,Vulnerability Type Distributions in CVE, http://cwe.mitre.org/documents/vuln-trends/index.html, Ekim 2008.

[36] Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları ,1998 DARPA Intrusion Detection Evaluation Data Set Overview, http://www.ll.mit.edu/IST/ideval/data/1998/1998_data_index.html, Ekim 2008.

[37] Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları ,1999 DARPA Intrusion Detection Evaluation Data Set Overview, http://www.ll.mit.edu /IST/ideval/data/1999/1999_data_index.html, Ekim 2008.

[38] KAYACIK, H. G., HEYWOOD, A. N., HEYWOOD, M. I., A hierarchical SOMbased intrusion detection System", Engineering Applications of Artificial Intelligence, Elsevier, 20(4): 439-451, 2007.

[39] ELKAN, C., Results of the KDD'99 Classifier Learning Artificial Intelligence, University of California San Diego,63-64,2000.

[40] GARRETT, S.M., How Do We Evaluate Artificial Immune Systems?, Evolutionary Computation, MIT Press, 13(2): 145-178 ,2005.

[41] ESPONDA, F., FORREST , S., HELMAN, P., A formal framework for positive and negative detection schemes, IEEE Transactions on Systems, Man, and Cybernetics, Part B, Cybernetics, 34(1): 357-373 ,2004.

[42] DE CASTRO, L.N. ve TIMMIS, J., Artificial Immune Systems: A New Computational Intelligence Systems, Springer, 357 pages, 2002.

[43] GONZALEZ, F., A Randomized Real-Valued Negative Selection Algorithm', Proceedings of the 2nd International Conference on Artificial Immune Systems, UK, 2003.

[44] DE CASTRO, L. N. ve VON ZUBEN, F. J., The Clonal Selection Algorithm with Engineering Applications, GECCO'2000, pp. 36-37, 2000.

[45] DE CASTRO , L. N. ve VON ZUBEN, F. J., An Evolutionary Immune Network for Data Clustering, IEEE SBRN'2000, pp. 84-89, 2000.

[46] FORREST , S., PERELSON,A., ALLEN, L., Self-Nonself Discrimination in a Computer, Proceedings of the IEEE Symposium on Research in Security and Privacy: 202-212 ,1994.

[47] FOREST, S., and HOFMEYR S.A., John Holland's Invisible Hand: An Artificial Immune System, presented at the FESTSCHIRIFT ,1999.

[48] HOFMEYR, S.A., FORREST, S., Immunity by Design: An Artificial Immune System, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), San Francisco, CA, 1289-1296 ,1999.

[49] SOMAYAJI, A., HOFMEYR, S.A., FORREST, S., Principles of a Computer Immune System" Proceedings of the New Security Paradigms Workshop, 75-81 ,1997.

[50] DE CASTRO, L.N. , VON ZUBEN, F.J., Learning and optimization Using the Clonal Selection Principle, In the Special Issue on Artificial Immune Systems of the Journal IEEE Transactions on Evolutionary Computation,

June 6(3) ,2002.

[51] DE CASTRO, L.N. , VON ZUBEN, F.J., The Clonal Selection Algorithm with Engineering Applications, GECCO 2000, Las Vegas, Nevada, USA, July 8 ,2000.

[52] COSTA, A.M.,VARGAS,P.A.,VON ZUBEN ,F.J. and Franca, P.M., Makespan Minimization On Paralel Processors: An Immune-Based Approach, In the proceedings of the special sessions on artificial immune systems in the 2002 Congress on Evolutionary Computation, 2002 IEEE World Congress on Computational Intelligence, Honolulu, Hawaii ,2002.

[53] JENSEN, M., HANSEN, T., Robust solutions to job shop problems", http://wwwiiuf. unifr.ch/~wangl/immune_reference.html ,Ekim 2007.

[54] HART, E., ROSS, P., NELSON, J., Producing Robust Schedules via an Artificial Immune System, ICEC, 464- 469 ,1998.

[55] MORI,K., TSUKIYAMA,M , FUKUDA,T., Artificial Immunity Based Management System for a Semiconducter Production Line, In IEEE International Conference on Systems, Man, and Cybernetics, 1: 852-856 1997.

[56] DASGUPTA, D., FORREST, S., Artificial Immune systems in Industrial Applications, In the proceedings of the Second International Conference on Intelligent Processing and Manufacturing of materials (IPMM), Honolulu, July 10-15 1999.

[57] LEE, D., JUN, H., SIM, B., Artificial Immune System for Realization of Cooperative Strategies and Group Behaviour in collective Autonomous mobile Robots, Proceedings of 4th Int. Symp. On Artificial Life and Robotics:232-235 ,1999.

[58] DASGUPTA,D., FORREST, S. Novelty Detection in Time Series Data Using Ideas From Immunology, Proceedings of the ISCA'96,1996.

[59] TEMURTAS, F., ER, O., SERTKAYA, C., A Comparative Study on Chronic Obstructive Pulmonary and Pneumonia Diseases Diagnosis using Neural Networks and Artificial Immune System , 2008.

ÖZGEÇMİŞ

Cengiz SERTKAYA, 13.07.1983 te Ankara' da doğdu. İlkokul eğitimini Dumlupınar İlköğretim Okulu'nda , orta ve lise eğitimini Mamak Anadolu Lisesi'nde tamamladı. 2002 yılında Mamak Anadolu Lisesi'nden mezun oldu.2002 yılında Sakarya Üniversitesi, Bilgisayar Mühendisliği Bölümüne girdi ve 2006 yılında mezun oldu. 2007 yılında ADASU Genel Müdürlüğü'nde mühendis olarak çalışmaya başladı.Görev süresi içerisinde şirketin yazılım projelerinde yazılım mühendisi ve proje yöneticisi görevlerinde aktif rol aldı. Şu anda yine ADASU Genel Müdürlüğü'nde Proje Yöneticisi olarak görev yapmaktadır.