Gazi Üniv. Müh. Mim. Fak. Der.
Cilt 26, No 2, 325-340, 2011

J. Fac. Eng. Arch. Gazi Univ.
Vol 26, No 2, 325-340, 2011

ZEKİ SALDIRI TESPİT SİSTEMİ TASARIMI ve GERÇEKLEŞTİRİLMESİ

Şeref SAĞIROĞLU[1], Esra Nergis YOLAÇAN[2], Uraz YAVANOĞLU[1]
[1]Gazi Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Maltepe, Ankara.
[2]Northeastern University, Electrical & Computer Engineering, Boston, MA, USA
ss@gazi.edu.tr, **yolacan.e@husky.neu.edu**, **uraz@gazi.edu.tr**,

(Geliş/Received: 10.03.2010 ; Kabul/Accepted: 15.02.2011)

ÖZET

Bu çalışmada, bilgi ve bilgisayar güvenliğini sağlamak için geliştirilen araçlardan birisi olan saldırı tespit sistemleri (STS) incelenmiş, STS geliştirmek için kullanılan yöntemler araştırılmış, mevcut STS çalışmaları tartışılmış ve zeki bir STS geliştirilmiştir. Bu sebeple STS geliştirilirken, yapay sinir ağı (YSA) ve zeki STS'ler bu çalışma kapsamında araştırılmış, STS'lerin uygulanması sırasında kullanılan veritabanları incelenmiş ve işlem yeteneği mevcut STS'lerin üzerinde YSA tabanlı zeki bir STS yöntemi önerilmiş ve KDD'99 verileriyle test edilmiştir.

Geliştirilen STS'den ve yapılan çalışmalardan elde edilen sonuçlar değerlendirildiğinde; 65536 test verisinin 22sn de hesaplandığı ve her bir giriş verisi için geliştirilen sistemin sonuç üretme süresinin ise 0.00048sn olduğu ve farklı eğitim ve test kümeleriyle yapılan testlerden elde edilen en yüksek başarım oranının %97,92 ve en düşük başarım oranının ise %81,93 olduğu görülmüştür. Son olarak, bu çalışmanın ülkemizde bilgi ve bilgisayar güvenliği konusunda yapılacak çalışmalara büyük katkılar sağlaması ve yeni ufuklar açması beklenmektedir.

Anahtar Kelimeler: Saldırı Tespit Sistemi, Yapay Sinir Ağları, KDD'99, Zeki STS.

DESIGNING and DEVELOPING an INTELLIGENT INTRUSION DETECTION SYSTEM

ABSTRACT

In this study, intrusion detection systems (IDSs) which are important tools for providing information and computer security were analyzed, the methods used in developing IDSs were reviewed, the studies on IDS's were revised, and an intelligent IDS was developed and introduced. In the development phase, artificial neural network (ANN), intelligent IDS and database used in developing and testing IDS applications were also reviewed. An intelligent IDS was designed and developed with the help of KDD'99 database. Evaluating the results obtained from this study and the developed IDS.

Enlightening the reviews, it can be concluded that selecting appropriate model and database are very crucial for better intelligent IDS design. The developed model takes 22sec to evaluate 65536 data set of network packets. Every dataset is processed in 0.00048sec. The developed IDS models were performed the tasks with high accuracies between %81.93 and %97.92. Finally, it is expected that this study might help to improve and bring new insight for information and computer security studies to be made in Turkey.

Keywords: Intrusion Detection System, Artificial Neural Networks, KDD'99, Intelligent IDS.

1. GİRİŞ (INTRODUCTION)

Bilgi teknolojilerinin hızla gelişmesi ile bilginin işlenmesi, saklanması, taşınması ve korunması daha da önem kazanmıştır. Casus yazılımlar [2], mobil tehditler [4], saldırganların ve saldırıların sayısındaki artışlar, bilgi güvenliğinin önemini arttırmış, kurumların bilgi güvenliğini sağlayabilmeleri konusunda geliştirilen pek çok yaklaşım kullanılmaya başlanmıştır [1,3]. Elektronik ortamdaki tehdit ve tehlikelerin sistemlere çoğunlukla dışarıdan gelmesinden dolayı, bu ortamlara giren tehditleri ve saldırıları önceden tespit etmek çok önemli olduğundan, saldırı tespit sistemleri bu amaçla kullanılmaktadır.

Saldırı Tespit Sistemleri (STS), ağ üzerinden yapılan saldırılara karşı bilgi sistemlerinin korunmasında "alarm" niteliği taşıyan yazılım ve/veya donanım bileşenleridir. STS'lerin kullanılması ile sistemlere yapılan yetkisiz erişimler ve kötüye kullanımlar tespit edilerek, saldırganların sistemlere sızma girişimleri engellenebilmektedir. Bilgisayar sistemlerinde STS'lerin kullanılması ile birlikte, sisteme ne tür saldırıların yapıldığı, mevcut açıklar ve saldırgan profili gibi önemli bilgiler elde edilebilmektedir.

STS'ler üzerine yapılan çalışmalar incelendiğinde;

- Veri Toplama,
- Etiketleme,
- Depolama,
- Veri Azaltma (filtreleme, özellik seçme ve sınıflandırma),
- Davranış modellerinin belirlenmesi ve sınıflandırılması,
- Kural tabanlı sistemler için kuralların belirlenmesi,
- Raporlama ve sonuç üretme aşamalarında

araştırma konuları olduğu bilinmektedir [5]. Bu sorunların ortaya çıkması, işlenmesi gereken trafik büyüklüğü ile doğru orantılıdır. Bu sebeple karar verme sürecinde çalışacak hızlı ve basit bir yönteme gereksinim duyulmaktadır. STS tasarımları yapılırken tanımlama yöntemi çıkartmak için saldırı veritabanları kullanılmaktadır. Uygulamalarda kullanılabilir mevcut bir veritabanının olması veya oluşturulması, gerçekleştirilmesi planlanan sistemin hızlı sonuç üretilebilmesi açısından önemlidir [5-28].

STS'lerde karşılaşılan problemlerden bir diğeri ise yanlış alarm (false positive) oranlarının düşürülememesidir [29]. Anormallik tespiti yapan sistemlerde, gerek davranış gerekse kullanıcı profillerinin modellenmesi sırasında bu durumla karşılaşılmaktadır. Bu tip STS tasarımında yanlış alarm oranlarının indirgenmesinde zeki STS'lere ihtiyaç vardır.

Zeki STS tasarımlarında klasik STS'lerde karşılaşılan problemlerin çözülmesi için çalışmalar yapılmaktadır [8]. Mevcut çalışmalar incelendiğinde önceden hazırlanmış veri kümelerinin kullanılmış olduğu tespit edilmesine rağmen, günümüz şartlarında araştırmacıların uygulamalarında kullanılabilecekleri güncel bir veritabanına rastlanmamıştır. DARPA'nın 1998 ve 1999'da STS'lerin başarılarının değerlendirilmesi için yaptığı çalışma ve DARPA çalışmasının özelleşmiş bir versiyonu olan KDD'99 verileri, halen araştırmacılar tarafından tercih edilen saldırı veritabanlarıdır [33-35].

Mevcut sorunların ortadan kaldırılabilmesi için zeki STS tasarımlarının artması, sistemlere yapılan saldırıların güncel veritabanlarında tutulması ve tasarlanan sistemlerin güncel verilerle test edilmesi önem arz etmektedir. Bu çalışmada, sorunların kısmen ortadan kaldırılabilmesine yönelik zeki bir STS tasarımı gerçekleştirilmiş ve farklı saldırı kümeleriyle test edilmiştir.

Makalenin 2. Bölümünde klasik STS'ler, 3. Bölümde ise zeki STS'ler incelenmiştir. 4. Bölümde makale kapsamında tasarlanan ve geliştirilen zeki STS tasarımı sunulmakla birlikte 5. Bölümde test sonuçları verilmiştir. Makale bulguların tartışılması ve performans analiziyle tamamlanmıştır.

2. SALDIRI TESPİT SİSTEMLERİ (INTRUSION DETECTION SYSTEMS)

Bilgi ve bilgisayar teknolojilerinin gelişmesine paralel olarak, elektronik ortamların kullanım oranlarının gün geçtikçe artması bu ortamlarda saklanan bilgilerin güvenliğinin sağlanması ihtiyaç haline getirmiştir. Korunacak bilginin değerine göre farklılık gösteren sistemlerinin tek amacı, saldırganlara ve saldırılara karşı önlem alarak, bilginin mahremiyetinin korunmasıdır. Bilgisayar sistemlerine yönelik tehditler ve bu sistemlerde oluşabilecek zafiyetler olduğu sürece, saldırıların tespit edilmesi, yüksek seviyede bilgi güvenliğinin sağlanmasında önemli rol oynamaktadır [16].

Bilgiyi korurken, erişebilirliğin ve sürekliliğin sağlanması ise önemli bir konudur. Saldırılara karşı alınan önlemlerin güncel olması, değişen ve gelişen tekniklerin bilinmesi ve STS tasarımların güncellenmesi gerekmektedir [32]. Saldırı deyimi izin almadan bilgiye ulaşma, değiştirme, kullanılmaz veya güvenilmez hale getirme olarak tanımlanmaktadır [6]. Günümüzde ise saldırı, "bilgilerin gizliliğini, bütünlüğünü ve erişilebilirliğini tehlikeye atabilecek girişimler" olarak adlandırılmaktadır [7].

Saldırı tespiti, bir bilgisayar sisteminde veya ağda meydana gelen olayların izlenerek bilginin gizliliğini, bütünlüğünü ve erişilebilirliğini bozmak amacıyla sistemin güvenlik mekanizmalarını aşmak için yapılan hareketlerin işaretlerini analiz etme işlemidir [8].

Saldırı tespit sistemleri; bilgisayar sistemlerine yapılan saldırıları ve kötüye kullanımları belirlemek için tasarlanmış [9], tercihen gerçek zamanlı olarak, bilgisayar sistemlerinin yetkisiz ve kötüye kullanımı

ve amaç dışı ihlalleri tespit etmek için kullanılan [10], saldırıyı durdurma girişiminde bulunmayan ve olası güvenlik ihlali durumlarında, güvenlik uzmanına uyarı mesajı (alarm) veren [11], bilgisayar sistemlerinin kaynaklarına veya verilerine yetkisiz erişimleri belirleyen [12], bilgisayar güvenliği alanındaki "hırsız alarm" sistemleri olan [13] ve bilgisayar veya ağ sistemine yapılan yetkisiz erişimleri tespit etmek için kullanılan yazılım araçları [14] olarak farklı şekillerde tanımlanmaktadır.
Literatür incelendiğinde; STS'leri karşılaştıran çalışmalar yapıldığı [22] tek bir yaklaşım ile bilinen ve bilinmeyen saldırıları tespit eden yüksek performanslı bir STS bulunmadığı, dolayısıyla saldırıları kısa sürede tespit eden yüksek performanslı STS'lere ihtiyaç olduğu vurgulanmıştır.

STS'ler değerlendirildiğinde, olası güvenlik açıklarını belirleyebilmek için bilgisayar veya ağ içerisinde değişik alanlardan bilgileri toplayan, analiz ederek raporlayan sistemlerdir. Bu sebeple, güvenlik duvarlarını tamamlayan dinamik izleme elemanlarıdır [32].

3. ZEKİ SALDIRI TESPİT SİSTEMLERİ (INTELLIGENT INTRUSION DETECTION SYSTEMS)

Yapay sinir ağları (YSA), saldırı tespit sistemlerinde kullanılmaya başlanan zeki sınıflandırma yöntemidir. YSA'lar, anormallik tespiti yapan STS'ler için istatistiksel yöntemlere alternatif olarak kullanılan yöntemlerdendir [15].

YSA'lar, giriş ve çıkış vektörleri arasında ilişkileri öğrenen zeki sistemlerden oluşmaktadır. Bu sayede sisteme uygulanan yeni girişleri/çıkışları da ilişkilendirebilmektedir. YSA'ların STS'lerde kullanımı, YSA'nın sistem davranış izleriyle eğitilmesiyle başlayan bir süreçtir. Normal veya anormal olarak sınıflandırılan olay akışları YSA'ya verilir. Toplanan veriler ile sistemin davranışına bağlı olarak öğrenme gerçekleştirilir. Süreç kullanıcının n adet hareket veya komutundan, sonraki hareket veya komutuna yakınsayarak eğitilmesi işlemlerini kapsar [17].

YSA tabanlı STS'lerin oluşturulması için öncelikle eğitim veri seti oluşturulmalıdır. Eğitim verileri, belirli zaman periyodunda her kullanıcı için, sistem hesaplarından toplanan bilgilerden oluşmaktadır. Her gün ve her kullanıcı için veriler vektörel hale getirilir ve kullanıcının her komutu ne kadar sıklıkta yürüttüğü saptanır. Eğitim aşamasında ise daha önce elde edilen vektörler, kullanıcıları tanımlamak için eğitilir. YSA'lar, anormallik tespitinde kullanıldığı gibi kötüye kullanım tespitinde de kullanılan bir tekniktir. Ağ ataklarının sürekli değişen yapısı nedeniyle, ağ trafiğini geniş çapta analiz edebilen ve kural tabanlı sistemlerden daha esnek bir savunma sistemine duyulan ihtiyaca cevap verebilecek düzeydedir [15].

YSA'lar, kötüye kullanım tespitinde iki farklı şekilde kullanılmaktadır [42]. İlk yaklaşımda, var olan uzman sistemin parçası olarak kullanılmaktadır. Bu yaklaşımda, saldırı tespitinin daha etkin yapılması amacıyla gelen verilerin filtrelenmesi ve uzman sisteme gönderilmesinde YSA'dan faydalanılır. İkinci yaklaşımda ise, tek başına bir sistem olarak kötüye kullanımı tespit etmekte kullanılır. Bu yöntem YSA'ların, ağ akış bilgilerinden, kötüye kullanım tespitinin analizinde kullanılır. YSA'lar, STS'lerde karşılaşılan pek çok probleme esnek çözümler sunabildikleri için STS tasarımlarında önem arz etmektedir. YSA'nın çözüm sunduğu iki temel çözüm, giriş olarak uygulanması gereken verinin azaltılması ve sınıflandırma probleminin matematiksel ilişkiler kullanılarak çözülmesidir [17]. Veri azaltma, işleme zamanını, iletişim yükünü ve depolama gereksinimlerini azaltmak amacıyla veri koleksiyonunu analiz ederek en önemli girişleri tanımlama işidir. Sınıflandırma ise saldırgan profilini tanımlama işlemidir. YSA'lar, literatürde önerilen zeki STS'lerde bu iki problemi çözmek için kullanılmıştır [12,15,19,20-27,36-43].

STS tasarımlarında karşılaşılan diğer problemler ise istatistiksel yayılımı doğrulama ihtiyacı, tespit ölçütlerinin değerlendirilmesinin zorluğu, algoritma geliştirmenin yüksek maliyeti ve ölçeklemede zorluk olarak sıralanabilir [18]. İstatistiksel yayılımı doğrulama ihtiyacı probleminde; istatistiksel metotlar kullanıcı davranışlarının yayılımı hakkında varsayımlara dayanır. Bu varsayımlar doğru olmayabilir ve yanlış alarm oranının artmasına neden olabilir. YSA'lar veri yayılımında bu tür varsayımları ortadan kaldıran çözümler sunmaktadır. Tespit ölçütlerinin değerlendirilmesinin zorluğunda ise; istatistiksel metotlarda belirlenen ölçütler deneyim ve gözlemler sonucunda elde edilir. Ancak bu belirleme sırasında bir ölçütün ne kadar önemli olduğuna kesin olarak karar verilemez. Genelde etkili olmadığı için kullanılmayan bir ölçüt, özel bir çözüm için önemli olabilir. Algoritma geliştirmenin yüksek maliyeti probleminde ise; yeni bir istatistiksel algoritma önermenin ve yeni bir yazılım geliştirmenin süreci önemlidir. YSA kullanılması ise yeniden yapılandırma işlemi için daha kolay olup kullanılan yöntem değiştirilebilir. Ayrıca YSA'lar, çeşitli ölçüt kümelerinin ne kadar etkili olduğunu değerlendirmede de çözümler sunabilir. Ölçeklemede karşılaşılan zorluk; STS'lerin geniş ağ trafik yoğunluğunun yüksek olduğu yönlendiricilerde kullanılması ile binlerce kullanıcıya göre çalışmak gibi yeni problemler oluşturmaktadır. Bu sorunu çözmek için, güvenlik yöneticisi tarafından kullanıcı ve maruz kaldıkları saldırıları hedef alan bir profil çıkartılması gerekir. YSA kullanılarak, var olan davranışlara göre sınıflandırılma yapılması, saldırıları izlemeyi daha etkili hale getirecektir.

STS tasarımında farklı çözümler sunabilen farklı YSA çalışmaları aşağıda özetlenmiştir. Peddabachigari ve arkadaşları tarafından geliştirilen sistem, kullanıcı tarafından daha önce girilen komutların sırasına bağlı olarak, sonraki komutları tahmin edilmesinde kullanılmıştır [18]. YSA'ya giriş olarak gerçek

zamanlı veya geçmişteki belirli sayıda komut verilir. Aynı zamanda pencere boyutu olan bu sayının, sistemde önemli bir rolü vardır. Eğer bu sayı çok küçük olursa "yanlış alarm (false positive)" artar, büyük olması durumunda ise bazı ataklar tespit edilemez.

Çizelge 1. Literatürdeki YSA temelli STS örneklerinin karşılaştırılması (Comparison with ANN based IDS studies)

Çalışmalar	Kullanılan YZ Tekniği	YSA Yapısı	Nöron Sayıları	Veri Kümesi	Özellik Vektörü	Atak Tipleri	Örnek Sayısı	Başarı oranları (%)
[12]	YSA	MLP	12-25-x	6 haftalık trafik verisi	-	-	22444-25457	-
[15]	YSA	MLP	9-x-x-2	3000 örnek (benzetimi yapılmış atak)	TCP Paket Başlıkları	SYNflood, SATAN, ISS Scan	8462-1000	89-99
[19]	YSA	MLP	100-30-10	2 günlük trafik verisi	100 önemli komut	-	65-24	93
[22]	YSA	MLP	-	Rasgele üretilmiştir.	-	SYNflood, SYNport Scan, Stealthy	4000-6000	98
[23]	YSA	SOM	8-8-1	DARPA'99	Dinlenen portlardan geçen paket sayısı	DoS, Probe, U2R, R2L, Normal	-	-
[24]	YSA	MLP	41x40x40x1	KDD'99	KDD özellik vektörü	DoS, Probe,U2R, R2L, Normal	7312-6980	99
[25]	YBS	-	-	-	TCP ve UDP Paket Başlıkları	-	-	-
[26]	YSA	MLP	35x35x35x3	KDD'99	KDD özellik vektörü	Normal, Satan ve Neptune	900	80-93
[27]	YSA	MLP 10 Farklı öğrenme algoritması	-	KDD'99	KDD özellik vektörü	DoS, Probe, U2R, R2L, Normal	5092 - 6890	48-95
[36]	YSA	MLP	-	DARPA 99	KDD özellik vektörü	-	48871	90
[37]	YSA	MLP	9x1	DataPro	TCP ve UDP Paket Başlıkları	-	10000	90
[38]	YSA	SOM	16x27	Snort	DNS, SMTP, HTTP vektörleri	DoS, Probe, U2R, R2L, Normal	5092 - 6890	-
[39]	YSA	MLP	-	DARPA	BAM Sensör Datası	-	Sensör dataları	Judge Karar Sistemi
[40]	YSA	MLP	5x8	UNM sendmail veri kümesi	-	-	1000000	-
[41]	YSA	MLP	48-72	KDD'99	IP Protokolü tabanlı	-	169000	99
[42]	YSA	MLP/SVM	34-41	DARPA	TCP ve UDP Paket Başlıkları	DOS R2L U2S Probing	-	99
[43]	YSA	SOM	8x8	Güvenlik Duvarı verileri	TCP ve UDP Paket Başlıkları	-	30000	-
Sunulan Çalışma	YSA	MLP	6-8	KDD'99	KDD özellik vektörü	DoS, Probe, U2R, R2L, Normal	65536	87-96

Ryan ve arkadaşlarının geliştirdiği NNID (Neural Network Intrusion Detector), kullanıcılar tarafından kullanılan komutların dağılımına göre saldırıların tespit edilmesini sağlamaktadır [19]. Sistem üç bölümden oluşmaktadır. İlk olarak, kullanıcıların belirli periyotlarda komut yürütme dağılımlarının gösterilmesi için her kullanıcı için günlük kayıtlarından eğitim verileri toplanarak sonrasında komut dağılım vektörlerine bağlı olarak kullanıcıları tanımlamak için YSA eğitilir. Son olarak YSA, test için verilen her yeni komut dağılım vektörü için kullanıcıları tanımlar. YSA'nın bulduğu sonuç tanımlanmışlardan farklı olursa, sistemde anormallik olduğu tespit edilmiş olur [19].

YSA'nın STS'lerde kullanılmasının üstünlükleri [15,17,19,22-24,26-28];

1. Kötüye kullanım ataklarının karakteristiğini öğrenmesi ve daha önce kaydedilen örneklere benzemeyenleri ayırt edebilmesi,
2. Hızlı sonuç üretmesi sayesinde gerçek zamanlı uygulamalar için kullanışlı olması,
3. Gürültülü veriler ile çalışabilmesi ve diğer yöntemlere göre daha net karar verebilmesi
4. Farklı sistemlere bütünleşik olarak çalışabilmesi
5. Sistemin veya atağın genel davranışını sınırlı sayıda örnekle öğrenebilme yeteneği ve
6. Bilinmeyen saldırıları tespit edebilmesi

olarak sıralanabilir.

Tüm bu üstünlüklerin yanı sıra, karşılaşılan en büyük problem ise YSA modellerinin oluşturulması ve eğitilmesidir. YSA modellerinin sağlıklı ve etkin şekilde olarak oluşturulabilmesi için farklı parametreler seçilerek en uygun ağ parametresinin başlangıçta belirlenmesi gereklidir. Bu çalışmada da birçok deneme-yanılma yaklaşımı ile en uygun parametrelerin seçimi konusunda başlangıçta çalışılmış ve en uygun ağ yapısı ve parametresi belirlenmeye çalışılmıştır. Bu süreçler ve yapılan ağ eğitimlerinde tüm saldırıları eşit olarak kapsayacak bir veri kümesi kullanılmaya çalışılmıştır.

Farklı özellikler dikkate alındığında, zeki STS'lerin ve kullanılan tekniklerin karşılaştırılması, farklı yapay zeka yaklaşımlarının STS'lerin performansını arttırmak için kullanıldığını göstermektedir [19,21,27]. Bu tekniklerden literatürde en sık kullanılanlar ise, YSA, yapay bağışıklı sistemi, genetik algoritma, bulanık mantık ve son günlerde de veri madenciliği yaklaşımlarıdır.

Yapılan son dönem çalışmalarında ise bu metotların hibrid kullanılmasıyla birlikte yüksek performanslar elde edilmeye çalışılmaktadır.

Literatür çalışmasında incelenen yapay zeka temelli STS yaklaşımları farklı özellikler dikkate alınarak Çizelge 1'de verilmiştir. Çizelge 1'den görülebileceği gibi yapılan çalışmalarda, DARPA ve KDD verileri temel alınmış ve saldırı tipleri gruplanmıştır. Makale kapsamında geliştirilen sistem tablonun son bölümünde karşılaştırmaya dâhil edilmiştir. Çalışmanın benzer performans değerlerine küçük ölçekli YSA modelleri ile ulaşılabildiği gösterilmiştir.

Araştırmacılar, farklı ortamlarda, farklı sistemler üzerinde ve farklı tekniklerle birçok STS yazılımı geliştirmişlerdir [12,15,19,24-30]. Bu çalışmalar, Çizelge 2'de listelenerek, karakteristik özelliklerine göre karşılaştırılmıştır.

4. GELİŞTİRİLEN ZEKİ SALDIRI TESPİT SİSTEMİ (DEVELOPMENT of INTELLIGENT INTRUSION DETECTION SYSTEMS)

Bu çalışma kapsamında geliştirilen zeki STS'nin tasarımı ve bu tasarım esnasında kullanılan yaklaşımlar alt başlıklarda detaylı olarak aşağıda sunulmuştur.

4.1. Veri Kümeleri

Bir STS tasarlanırken ele alınması gereken hususların başında kullanılacak olan veri kümesi gelmektedir. STS veri kümesi, geliştirilecek olan STS'nin eğitim ve test aşamalarında saldırıyı tanımlamak için gereken ve içerisinde saldırı verileri barındıran ağ paketleri veya günlük kayıtlardan elde edilen veriler bütünüdür.

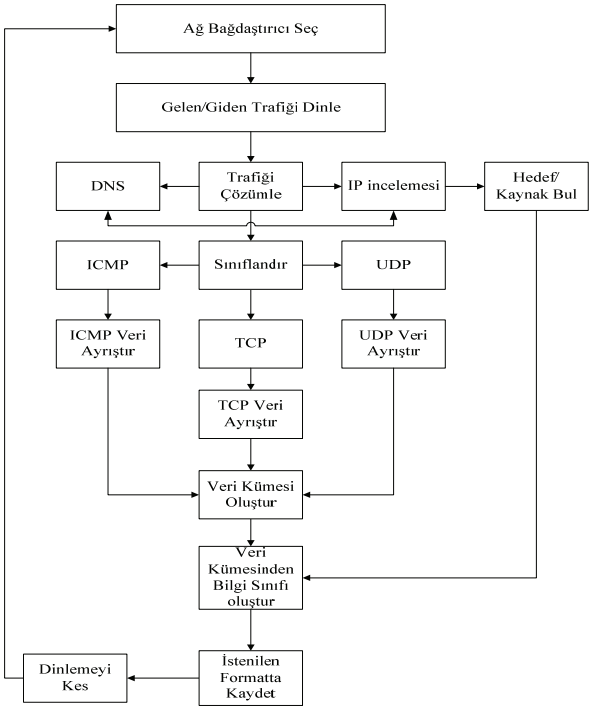
STS uygulamalarında, geliştiriciler veri kümelerini literatürde kabul görmüş hazır saldırı veritabanları yerine kendi veri kümelerini oluşturarak yapmayı tercih ederlerse oldukça zahmetli, bir o kadar da maliyetli bir çalışma gerçekleştirmenin zorlukları ortaya çıkacaktır.

STS'lerin tasarım ve uygulama çalışmalarının hızlandırılması iç geliştirilen sistemin daha objektif olarak değerlendirilebilmesi ve başarımının test edilmesi için bir saldırı veritabanı olan KDD'99 kullanılmıştır. Bununla birlikte yöntemin gerçek zamanlı oluşturulan veri kümeleri ile başarısının test edilebilmesi için ağ trafiğini dinleyen ve YSA modeline giriş sağlamayı amaçlayan bir yazılım geliştirilmiştir. Geliştirilen bu yazılımın akış diyagramı Şekil 1'de, örnek çalışma ekran görüntüsü ise Şekil 2'de sunulmuştur.

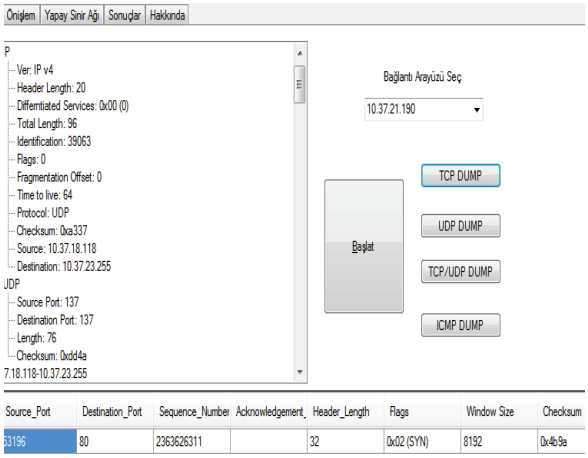
Geliştirilen yazılım sayesinde KDD'99 veritabanında olduğu gibi ulusal saldırı veritabanı oluşturulacaktır. Yapılan testlerin önceden saldırı olduğu bilinen trafik örüntüsüyle karşılaştırılabilmesi için bu çalışmada KDD'99 veri kümesi kullanılmıştır.

Çizelge 2. STS Yazılımlarının karşılaştırılması (Comparison for IDS software)

Yıl	STS	Saldırı Tespiti Yöntemi	Gerçek Zamanlı	Veri Kaynağı	Mimari yapı (veri işleme)
1988	Haystack	Hibrit	Hayır	Sunucu	Merkezi
1988	MIDAS	Hibrit	Evet	Sunucu	Merkezi
1988	IDES	Anormallik	Evet	Sunucu	Merkezi
1989	W&S	Anormallik	Evet	Sunucu	Merkezi
1990	Comp-Watch	Anormallik	Hayır	Sunucu	Merkezi
1990	NSM	Hibrit	Evet	Ağ	Merkezi
1991	NADIR	İmza	Hayır	Sunucu	Merkezi
1991	Computer misuse detection system	Hibrit	Evet	Sunucu	Dağıtık
1992	Hyperview	Hibrit	Evet	Sunucu	Merkezi
1992	DIDS	Hibrit	Evet	Hibrit	Dağıtık
1992	ASAX	İmza	Evet	Sunucu	Merkezi
1992	Intruder alert	İmza	Evet	Hibrit	Dağıtık
1993	USTAT	İmza	Evet	Sunucu	Merkezi
1994	DPEM	İmza	Evet	Sunucu	Dağıtık
1994	IDIOT	İmza	Evet	Sunucu	Merkezi
1995	NIDES	Hibrit	Evet	Sunucu	Merkezi
1996	GrIDS	Hibrit	Hayır	Hibrit	Dağıtık
1996	CSM	İmza	Evet	Sunucu	Dağıtık
1996	JANUS	İmza	Evet	Sunucu	Merkezi
1996	Kane security monitor	İmza	Evet	Sunucu	Dağıtık
1996	Netranger	İmza	Evet	Ağ	Dağıtık
1996	Realsecure	İmza	Evet	Ağ	Dağıtık
1997	JiNao	Hibrit	Evet	Sunucu	Dağıtık
1997	EMERALD	Hibrit	Evet	Hibrit	Dağıtık
1997	Anzen flight jacket	Hibrit	Evet	Ağ	Hibrit
1997	Netprowler (née id-trak)	İmza	Evet	Sunucu	Dağıtık
1997	Securenet Pro	İmza	Evet	Ağ	Dağıtık
1997	Sessionwall-3	İmza	Evet	Ağ	Dağıtık
1998	Bro	İmza	Evet	Ağ	Merkezi
1998	Centrax security suite	Hibrit	Evet	Hibrit	Dağıtık
1998	Cross-site for security	İmza	Evet	Ağ	Dağıtık
1998	Smartwatch	İmza	Evet	Sunucu	Dağıtık
1998	Stake out	İmza	Evet	Ağ	Hibrit
1998	Tripwire	İmza	Hayır	Sunucu	Merkezi
1999	Cybercop monitor	İmza	Evet	Sunucu	Dağıtık
2000	FIRE	Anormallik	Hayır	Ağ	Merkezi



Şekil 1. Paket koklayıcı yazılımın akış diyagramı (Block diagram of packet sniffer software)



Şekil 2. Paket koklayıcı yazılımın örnek ekran görüntüsü (Screenshot of packet sniffer software)

4.2. Hazır STS Veri Tabanı : KDD'99

Bu veri tabanı 1999 yılında DARPA [33-35] veri kümesinin bazı önişlemlerden geçirilmesi ile elde edilmiş 41 özellikten oluşan bir veri kümesidir. Bu veri kümesinin amacı, son yıllarda farklı tekniklerle gerçekleştirilen STS'ler için eğitim ve test işlemlerinde kolaylık sağlamaktır. STS'ler için veri kümesi problemi, DARPA ile kullanılabilmesi için yüksek sayıda önişleme ihtiyaç duymaktadır. KDD veri kümesi ile eğitim ve test sonuçlarının daha hızlı alınabilmesi yapılan çalışmaların sonuca ulaşmasını kolaylaştıran bir faktör olmuştur.

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

Bunlar;

- İçerik özellikleri (content features),
- Sunucu tabanlı trafik özellikleri (host-based trafic features),
- Zamana bağlı trafik özellikleri (time-based traffic features),

olarak sıralanabilir.

Çizelge 3'de İçerik özellikleri kategorisi verilmiştir.

Çizelge 3. İçerik özellikleri (Contents of TCP packets)

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", TCP bağlantılarından sağlanmaktadır. Bu özelikleri elde etmek için ağ trafiği verileri üzerinde önişlem yapılması gerekmektedir

"Sunucu tabanlı trafik özellikleri", etki alanı (domain) bilgisi ile sağlanmaktadır.

"Zamana bağlı trafik özellikleri", "aynı sunucu" ve "aynı servis" özellikleri kullanılarak sağlanmaktadır. "Aynı sunucu" özellikleri, son iki saniye içerisinde aynı sunucuya yapılan bağlantıların "aynı servis" özellikleri ise son iki saniye içerisinde aynı servise yapılan bağlantıların izlenmesiyle elde edilir.

KDD'99 eğitim veri kümesinde bulunan 24 saldırı ve bu saldırıların ait oldukları saldırı tipleri ile saldırı veri kümesinde bulunan örnek sayıları Çizelge 4'de gösterilmiştir.

Test veri kümesinde yer alan ve "KDD doğrulanmış" (KDD corrected) dosyasından alınan 14 farklı saldırıya ait saldırı tipi ve örnek sayıları Çizelge 5'de verilmiştir [20].

Çizelge 4. KDD'99 veri kümesinin %10'luk kısmından alınan saldırı örneklerinin sayıları (%10 of sample intrusions in KDD'99 data set)

Saldırılar	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	r2l
guess_passwd	53	r2l
warezmaster	20	r2l
imap	12	r2l
ftp_write	8	r2l
multihop	7	r2l
phf	4	r2l
spy	2	r2l
buffer_overflow	30	u2r
rootkit	10	u2r
loadmodule	9	u2r
perl	3	u2r

Çizelge 4'de YSA eğitimi için kullanılan ve tanınması amaçlanan saldırı tipleri ve her saldırı tipinden eğitim için önerilen örnek sayıları verilmiştir. Çizelge 5'de ise KDD veri kümesinde bulunmasına rağmen YSA eğitiminde kullanılmayan ve YSA'nın eğitim kümesinde yer almayan saldırıları tanıma başarısını ölçmek için kullanılacak olan saldırı tipleri ve örnek sayıları verilmiştir.

4.3. Zeki STS Geliştirme Adımları

Bu çalışma kapsamında, incelediğimiz KDD verileri ile zeki STS tasarlanmıştır. Şekil 3'de blok diyagramı verilen zeki STS, YSA tabanlı geliştirilmiştir.

Geliştirilen Zeki STS yazılımı; Ön İşlem, Eğitim, Test ve Sonuç modüllerinden oluşmaktadır. Ön işlem modülü koklayıcı yazılım ile dinlenen ve ayrıştırılmamış verilerin KDD örüntüsüne uygun hale getirilmesini sağlamaktadır.

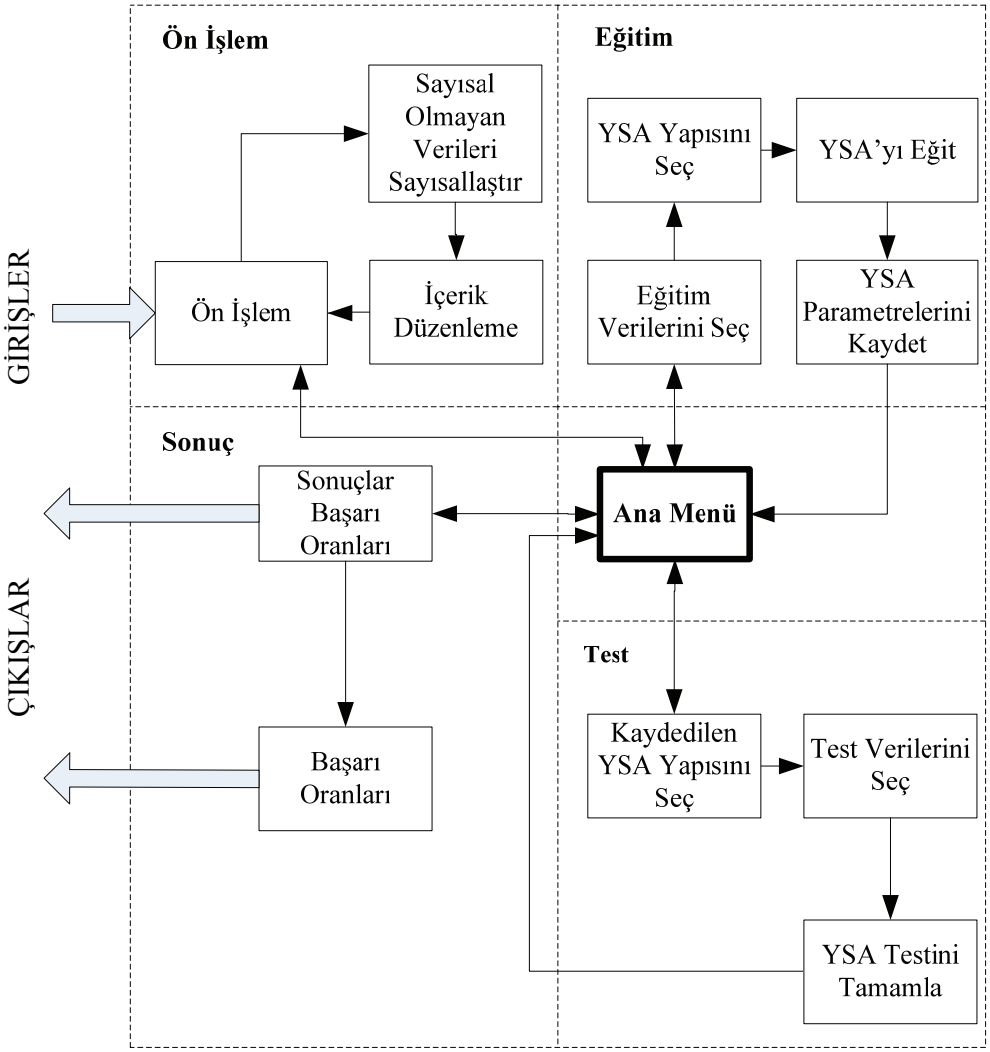
Çizelge 5. Eğitim kümesinde yer almayan saldırılar (Intrusions not included in ANN training set)

Saldırılar	Örnek sayısı	Kategori
apache	794	Dos
mailbomb	5000	Dos
processtable	759	dos
udpstorm	2	dos
mscan	1053	probe
saint	736	probe
httptunnel.	138	r2l
named	17	r2l
sendmail	17	r2l
snmpgetattack	1040	r2l
xlock	9	r2l
xsnoop	4	r2l
ps	16	u2r
xterm	13	u2r

Bu örüntüye ait paketlerin başlık analizlerinin yapılması ve her paket başlığına bağlı içeriklerin analiz edilmesi saldırı tespiti için önemlidir. Bu çalışmanın konusu olan sistem KDD'99 veri setini kullandığı için ön işlem KDD'99 verilerinin sayılaştırılması ve YSA modeline giriş için matematiksel olarak ifade edilebilir hale dönüştürülmesi işlemini kapsamaktadır.

Eğitim modülü, ön işlem modülü içerisinde sayısallaştırılan verilerin bilgi üretimi için sınıflandırılarak modellendiği bölümdür. Sistemin eğitim kümesinde bulunmayan saldırı tekniklerini de tanımlayabilecek bir yapı kazanması için farklı saldırı türlerini kapsayan eğitim veri kümeleriyle eğitilmesi gerekmektedir. Bu amaçla KDD'99 veri kümesi her biri 65536 örnekleme sayısına sahip rasgele seçilen parçalara bölünmüştür. Bu modül genel sistem tasarımından bağımsız olarak bir Dinamik Bağlantı Kütüphanesi (DLL) ile diğer sistemler bütünleştirilmiştir. Sistemin daha önce karşılaşmadığı trafik verisi için eğitim sonucunda elde edilen matematiksel model kullanılmaktadır.

Test modülü, önişlemlerden geçen ağ trafiğinin Eğitim modülünde bulunan matematiksel ilişki modeline, giriş olarak uygulandığı bölümü olup,



Şekil 3. Geliştirilen zeki STS blok şeması (Block diagram of intelligent IDS platform)

KDD'99 veri kümesinin veya ön işlem modülünde işlenmiş herhangi bir trafik örüntüsünün, anlaşılır bilgiye dönüştürüldüğü ve trafik örüntüsünde yer alan belirsizliğin giderildiği kısımdır. Bu çalışmada gerçek trafik örüntüsü yerine KDD'99 veri kümesinin kullanılmasının nedeni KDD'99 veri kümesinde yer alan saldırı tekniklerinin gerçek zamanlı olarak elde edilmesinin mümkün olmamasıdır. Bu tip bir veritabanı ancak ulusal çapta hizmet veren internet servis sağlayıcı geçitlerinden elde edilebileceği için ülke ulusal saldırı veritabanı oluşturulması ihtiyacı oluşmuştur.

Sonuç modülü, esas olarak sistemin raporlama bölümüdür. Test modülünde matematiksel ilişki modeline giriş olarak uygulanan trafik içerisinden elde edilen normal trafik veya saldırı istatistikleri, bu kısımda raporlanabilir grafiklere dönüştürülmektedir.

Bu çalışma kapsamında tasarlanan zeki STS için yapılan işlemler, eğitim ve test için kullanılan veri kümeleri ile YSA yapısı detaylı olarak sunulmuştur.

4.4. YSA Yapısı

KDD'99 verilerinden elde edilen tüm girişler temel alınarak, 41 giriş nöronu kullanılmıştır.

Bölüm 3'de Zeki STS'lerin tasarımında kullanılan YSA yapıları ve öğrenme algoritmaları Çizelge 1'de özetlenmiştir. Zeki STS tasarımda en çok kullanılan YSA yapısı olan MLP bu çalışmada kapsamında tercih edilmiştir. MLP yapısının tercih edilmesinin nedeni sınıflandırma problemlerinde başarılı sonuçlar üretmesi ve farklı öğrenme algoritmaları ile kullanıma uygun olmasıdır. Şekil 4'de genel olarak gösterilen MLP modeli, bir giriş, bir veya daha fazla ara katman ve bir de çıkış katmanından oluşur.

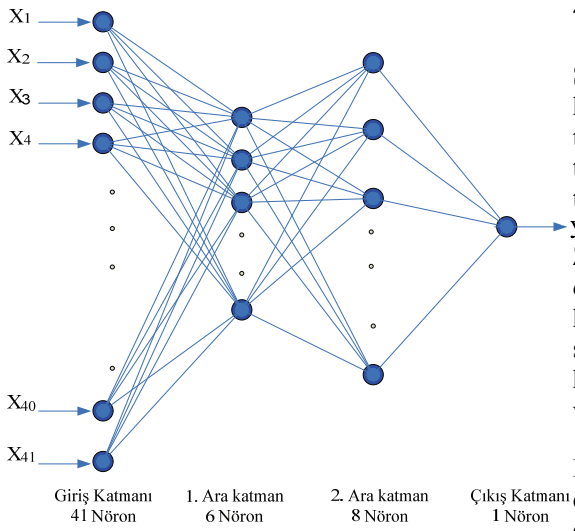
Katmanlarda kullanılan nöron sayıları, Çizelge 6'da verilen karar süreci sonucu öğrenme etkisi temel alınarak deneme-yanılma yoluyla tespit edilmiştir. Çizelge 6'da karar verme süreci içerisinde denenen YSA modellerinden bazıları ve ulaştıkları hata oranları sunulmuştur.

Çizelge 6. YSA modeli tercihi için yapılan testler (Experimental results for ANN structures tested)

AKS	HKNS	TF	ÖA	IT	EMH(%)
2	6,8	LOG,LOG,LIN	LM	24	0,00121
2	10,15	LOG,TAN,LIN	LM	16	0,00154
3	10,15,5	TAN,TAN,LOG,LIN	LM	8	0,00988
2	4,8	LOG,LOG,LIN	LM	50	0,0223
2	10,15	LOG,LOG,LIN	GD	160	0,123
3	10,15,5	TAN,LOG,LOG,LIN	GD	18	0,0336
2	9,16	LOG,LOG,LIN	RP	64	0,00422

AKS: Ara Katman Sayısı
HKNS: Her Katmanda Nöron Sayısı
TF:Transfer Fonksiyonu
ÖA: Öğrenme Algoritması
IT: İterasyon
EMH: Erişilen Minimum Hata

LOG: Logaritmik
TAN: Tanjant Hiperbolik
LIN: Doğrusal
LM: Leverberg Marquardt
GD: Gradient Descent
RP: Resilent Backpropagation



Şekil 4. Zeki STS için kullanılan YSA yapısı (ANN Structure of intelligent IDS platform)

Birinci ara katman 6 ve ikinci ara katman ise 8 nörondan oluşmaktadır. YSA'dan elde edilen çıkış "saldırı" (0) ya da "normal" (1) olarak belirlendiğinden, tek bir çıkış nöronu kullanılmıştır. Makale kapsamında YSA tabanlı zeki STS'lerin tasarımı ve uygulaması için temel bilgileri içeren yol gösterici bir çalışma yapılmasına özen gösterilmiştir.Levenberg-Marquardt (LM) öğrenme algoritması, YSA öğrenme sürecinde başarılı sonuçlar vermektedir [31]. LM algoritması, maksimum komşuluk fikri üzerine kurulmuş en az kareler hesaplama metodudur [31]. Bu algoritma, Gauss-Newton ve En Dik İniş (Steepest Descent) algoritmalarının en iyi özelliklerinden oluşmakla birlikte her iki metodun kısıtlamalarını ortadan kaldırmaktadır. Genel olarak kullanılan metot yavaş yakınsama probleminden etkilenmediği için tercih edilmiş ve Zeki STS'nin ana bileşeni olan YSA yapısının eğitiminde kullanılmıştır.

4.5. Eğitim veri kümeleri

STS tasarımında en önemli bileşenlerden birisi veri kümeleridir. KDD'99 veri kümesi araştırmacılar tarafından, büyük boyutlu olması, farklı saldırı tiplerini ve dengeli saldırı tiplerini içermesi sebebiyle tercih edilmektedir.

Zeki STS tasarımı için kullanılan eğitim veri kümeleri oluşturulurken, KDD'99 veri kümesinin %10'luk kısmından rasgele seçilen 65536 adet örnek sayısına sahip 3 farklı eğitim kümesi kullanılmıştır. Eğitim kümelerinde yer alan bir örneklem Çizelge 7'de verilmiştir.

Bir eğitim setinde 65536 veri kullanılmasının sebebi eğitim ve test verilerinin ofis yazılımlarıyla organize edilmesinden kaynaklanmaktadır. Bir ofis yazılımı tek bir tablo sayfasında 65536 veri satırı saklayabilmektedir. Bu sebeple KDD'99 veri seti kendi içinde dengeli saldırı tiplerinin saklandığı 3 adet tablo olarak oluşturulmuştur. Bu örneklemeler alınırken rastgele seçilen 65536 satır yerine tasarlanan matematiksel modelin daha geniş örneklemeler ile test edilebilmesi için birden fazla eğitim seti kullanılmıştır.

Zeki STS eğitilen atakları değil, kendisine öğretilmeyen atak tiplerini de belirleyebilmelidir. Bu kural ve istatistik tabanlı klasik STS tasarımı ile zeki matematiksel ilişki modeline dayalı STS tasarımını ayıran en temel özelliktir.

Test kümeleri oluşturulurken 65536 sınır değerli 5 adet test veri kümesi oluşturulmuştur. Bu veri kümeleri eğitim kümesinde kullanılmayan ve kalan örneklerden rastgele seçilen KDD'99 bilgisidir. Bu sayede test verileri eğitim kümesinde yer alan atak tiplerini de içermekle birlikte daha önce hiçbir eğitim kümesinde yer almayan atak tiplerinden harmanlanarak oluşturulmuştur.

Çizelge 7. Eğitim kümelerinin oluşturulması (Establising training sets)

Saldırı Tipi	Saldırı Adı	Veri Sayısı	Saldırı tipine göre Toplam	Tüm saldırılar toplamı	Toplam eğitim verisi sayısı
DoS	Land	16	42681	44098	65536
	Neptune	29086			
	Pod	20			
	Smurf	13459			
	Teardrop	100			
U2R	Buffer-overflow	2	4		
	Load module	1			
	Perl	1			
R2L	Imap	11	34		
	Multihop	1			
	Warez master	20			
	Phf	2			
Probe	Ipsweep	102	1379		
	Nmap	101			
	Portsweep	238			
	Satan	938			
Normal	Normal	21438	21438	21438	

Eğitim ve test veri kümelerinde bulunan örnek sayıları incelendiğinde, yakın olmayan değerler gözlenmiştir. Örneğin; eğitim veri kümesinde tek bir saldırı türünden 1000 örnek olmasına karşın diğer veri kümesinde aynı saldırı türünden 5 örnek bulunmaktadır. Bu durum, test veri kümesi için önemsizken, YSA'nın öğrenmesini etkileyen eğitim veri kümesi için önem taşımaktadır. Bu çalışmada rastgele alınan veriler kullanıldığı için mümkün olduğunca bağıl bir dağılım oranı aranarak eğitim veri kümeleri oluşturulmuştur.

4.6. Test veri kümeleri

Test kümeleri, KDD'99 un doğrulanmış veri kümelerinden oluşturulan 65536 veriden oluşmaktadır. Bu çalışmada kullanılan ve başarı oranları gösterilen 5 adet test veri kümesi için örnek saldırı örüntüleri Çizelge 8'de verilmiştir.

4.7. Ön işlemler

Zeki STS tasarımında KDD'99 dan alınan verilerin formatı Çizelge 9'da gösterilmiştir. Bu verilerin YSA tabanlı matematiksel ilişki modeline giriş olarak uygulanabilmesi için önişlemden geçirerek 41 özellik ve 1 çıkış haline getirilmesi gerekmektedir.

YSA yapısı tasarlanırken matematiksel olarak işlem yeteneği olan değerler kullanılmaktadır. Bu sebeple STS tasarımında kullanılan kavramların matematiksel ilişki modeline dönüştürülmesi sürecinde KDD'99 veri kümesi sayısal formata dönüştürülmüştür. Geliştirilen platform eğitim ve test kümelerine ait sayısal verileri işledikten sonra, sonuçları KDD'99 veri kümesi içerisinde tanımlandığı gibi saldırı ve normal trafik olarak raporlayabilme yeteneğine sahiptir.

Çizelge 8. Test kümelerinin oluşturulması (Establishing test sets)

Saldırı Tipi	Saldırı Adı	Veri Sayısı	Saldırı tipine göre	Tüm saldırılar toplamı	Toplam
DoS	Pod	22	57256	59168	65536
	Smurf	56835			
	Apache2	398			
	Udpstorm	1			
U2R	Buffer_ overflow	3	3		
R2L	Guess_ passwd	2	1100		
	Multihop	4			
	Phf	1			
	Named	10			
	Sendmail	6			
	Snmp getattack	1069			
	Xlock	6			
	Xsnoop	2			
Probe	Ipsweep	80	809		
	Portsweep	106			
	Saint	623			
Normal	Normal	6367	6368	6368	

Çizelge 9. Veri kümesi örneği (Sample of data set)

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,3170,329,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,2,0.00,0.00,0.00,0.00,1.00,0.00,1.00,54,39,0.72,0.11,0.02,0.00,0.02,0.00,0.09,0.13,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.

0,tcp,http,SF,291,3542,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,12,12,0.00,0.00,0.00,0.00,1.00,0.00,0.00,187,255,1.00,0.00,0.01,0.01,0.00,0.00,0.00,0.00,normal.

KDD'99 veri kümesinde, tek bir örneğe ait olan her veri satırında 42 alan bulunmaktadır. Buradaki özellik vektörleri Bölüm 3'de detaylı bir şekilde açıklanmıştır. Çizelge 10'da özelliklerine göre sütunlara ayrıştırılmış veriler gösterilmiştir.

Çizelge 10. Özelliklerin ayrıştırılmış veri formatı (data format after selection process)

0	udp	private	SF	105	146	…	snmpgetattack
1	tcp	smtp	SF	3170	329	…	normal.
0	tcp	http	SF	297	13787	…	normal.
0	tcp	http	SF	291	3542	…	normal.

Çizelge 11. Saldırı isimlerinin sayısal formata dönüştürülmesi (Conversion of intrusion features to corresponding numeric forms)

Saldırı	Sayısal Değeri	Saldırı	Sayısal Değeri	Saldırı	Sayısal Değeri	Saldırı	Sayısal Değeri
normal	0	teardrop	20	neptune	10	xsnoop	30
back	1	warezclient	21	nmap	11	mailbomb	31
buffer_overflow	2	warezmaster	22	perl	12	processtable	32
ftp_write	3	apache2	23	phf	13	mscan	33
guess_passwd	4	named	24	pod	14	httptunnel	34
imap	5	saint	25	portsweep	15	ps	35
ipsweep	6	sendmail	26	rootkit	16	xterm	36
land	7	snmpgetattack	27	satan	17	snmpguess	37
loadmodule	8	udpstorm	28	smurf	18	worm	38
multihop	9	xlock	29	spy	19	sqlattack	39

Sütunlara ayrılan verilerden bazıları sayısal formatta olmadığından, YSA'nın giriş olarak kullanılabilmesi için, sayısal formata dönüştürülmüştür. Ancak bu çevirimlerin elle yapılması, veri setinin büyüklüğünden dolayı mümkün değildir. Bu nedenle, önişlemlerin yapılması için bir modül geliştirilmiştir. Bu modül, veri kümesinde yer alan protokol, servis, bayrak (flag) ve saldırı tipleri alanlarının sayısal forma dönüştürülmesi gerçekleştirmektedir. Çizelge 11 ve Çizelge 12'de örnek çevrimler gösterilmiştir.

Çizelge 12. Protokol isimleri ve sayısal dönüşümleri (Protocol names and numeric offset)

Protokol	Sayısal Değeri
Tcp	0
Udp	1
Icmp	2

YSA yapısı tasarlanırken matematiksel olarak işlem yeteneği olan değerler kullanılmaktadır. Bu sebeple STS tasarımında kullanılan kavramların matematiksel ilişki modeline dönüştürülmesi sürecinde KDD'99 veri kümesi sayısal formata dönüştürülmüştür. Geliştirilen platform eğitim ve test kümelerine ait sayısal verileri işledikten sonra, sonuçları KDD'99 veri kümesi içerisinde tanımlandığı gibi saldırı ve normal trafik olarak raporlayabilme yeteneğine sahiptir.

4.8. Genel Sistem Tasarımı

Yazılım C# programlama dili ile MS Visual Studio ortamında geliştirilmiştir. Zeki STS DLL bileşeni hazır bir program kullanılarak modellenmiştir. Yazılımın kullanılabilmesi için gerekli çalıştırma kütüphanelerinin (framework) bulunduğu bilgisayara kurulması yeterlidir. Sistemin raporlama kabiliyetleri MS Ofis programının alt sınıfları ile modellenmiştir. Bu model sayesinde sistem ön tanımlı raporlama isterlerine cevap verebilecek düzeydedir. Geliştirilen yazılımın sonraki aşamalarında miniportlar aracılığıyla güvenlik duvarlarıyla uyumlu olarak çalışması hedeflenmektedir.

Zeki STS yazılımının YSA bileşeni kullanılırken takip edilen adımlar aşağıda verilmiştir.

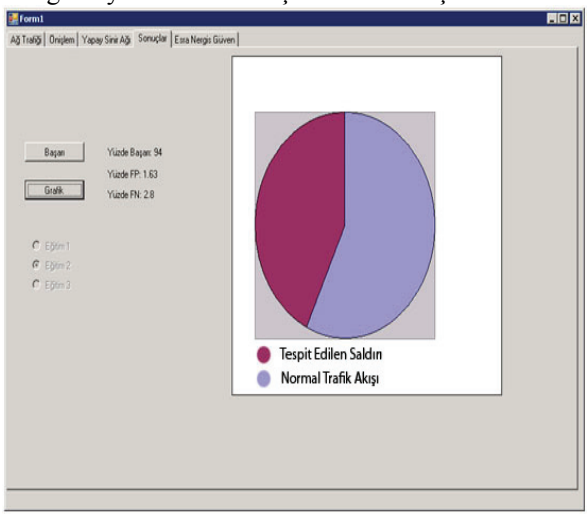
1. Verilerin önişlemden geçirilmesi
2. Eğitim ve test verilerinin oluşturulması
3. YSA yapısının oluşturulması
4. Eğitim veri setinin okunması ve normalize edilmesi
5. İlk ağırlık değerlerinin atanması ve eğitime başlanması,
6. Belirlenen epok sayısına kadar eğitimin devam ettirilmesi
7. Sonuçların belirlenen kriterlere göre değerlendirilmesi

Şekil 5'de verilen başarı değerlendirme grafiği nesnesi, test edilen veri kümesi sonuçlarının, beklenen sonuçlar ile karşılaştırılarak, tespit etme oranı, yanlış alarm ve yanlış negatif oranlarının hesaplanması için geliştirilmiştir. Geliştirilen arayüz içinde gösterilen grafik, test verilerinin ne kadarının saldırı, ne kadarının normal veri olduğunu göstermektedir.
Şekil 6'da YSA modelinin eğitimi sonucu elde edilen hata değerleri incelendiğinde yaklaşık 10^{-3} hata değerine ulaşılmıştır. 15 epoktan sonra da YSA hata değerlerinin ihmal edilebilir oranda değiştiği ve bunun da öğrenmeyi fazlaca iyileştirmediği gözlemlenmiştir.

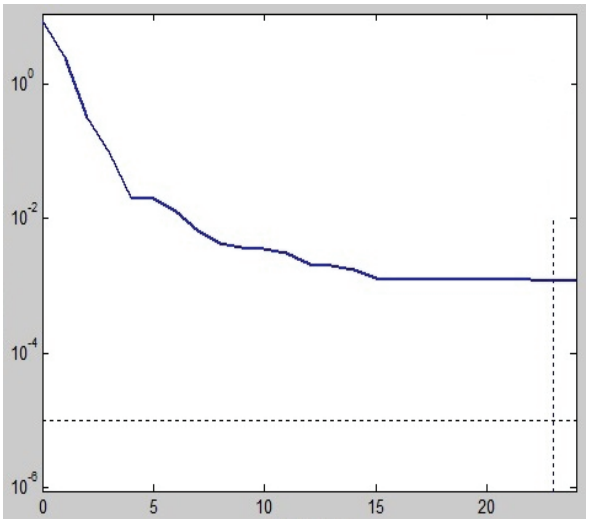
Oluşturulan modelin test edilmesi ile elde edilen sonuçlar ise Çizelge 13'de sunulmuştur. YSA eğitim kümelerinin doğruluk oranı normal yada saldırı olduğu tespit edilen trafik oranını, yanlış alarm oranını ve saldırı olduğu halde YSA tarafından hatalı tespit edilen trafik oranını içermektedir.

Çizelge 13'de matematiksel modelin doğruluğunu kanıtlamak için birbirlerinden bağımsız kullanılan 3 farklı eğitim kümesinin sonuçları verilmiştir. Farklı eğitim veri kümeleri ile eğitilen YSA'ların, 5 farklı test kümesi ile test edilmesinin sonuçları ise Çizelge 14'de sunulmuştur. Bu çizelgede test veri kümelerinin YSA eğitim kümesiyle oluşturulmuş modele

uygulanması sonucunda normal trafik ile saldırı trafiğini ayırt edebilme başarıları verilmiştir.



Şekil 5. Saldırı tespit başarısının normal trafik üzerine grafiksel raporlaması (Success rate of intrusion detection platform)



Şekil 6. Oluşturulan YSA modelinin eğitim hatası oranları (Graphical error rate of proposed ANN structure)

Zeki STS'nin Çizelge 14'de gösterilen test veri kümesi sonuçları ise sistemin başarılı olduğunu ve eğitim setinde olmayan yeni saldırı tiplerinin de algılandığını göstermektedir. Bu başarım oranları değerlendirildiğinde, Zeki STS'nin her biri 65536 trafik bilgisine sahip 1. Test kümesinde %97,77; 2. Test kümesinde %83,96; 3. Test kümesinde %86,53; 4. Test kümesinde %93,72; 5. Test veri kümesinde %93,49 oranında başarılı olduğu görülmüştür.

Elde edilen sonuçlar; bu çalışma kapsamında geliştirilen zeki STS sisteminin başarılı olduğunu göstermektedir. Öğrenmede en düşük performans oranı %96 iken test kümeleri için ortalama performans oranı %91'dir. Önerilen yöntem literatürde başarılı kabul edilen sistemlerden daha fazla sayıda saldırı tipini daha kısa sürede tanıyabilmektedir.

Çizelge 13. YSA eğitim kümelerinin performans sonuçları (Success rates of proposed training sets)

Eğitim veri kümesi	Doğruluk oranı (%)	Yanlış alarm (%)	Yanlış negatif (%)
Eğitim #1	99,12	1,04	0,99
Eğitim #2	96,45	2,63	0,73
Eğitim #3	98,57	0,35	0,92

5. SONUÇ VE ÖNERİLER (CONCLUSION & SUGGESTIONS)

Bu çalışmada, STS'ler genel olarak incelenmiş, klasik ve zeki STS'ler gözden geçirilmiş, STS tasarımında kullanılan veri kümeleri değerlendirilmiştir. Literatür incelemesinden elde edilen bilgiler doğrultusunda, YSA tabanlı zeki bir STS tasarımı yapılarak elde edilen sonuçlar sunulmuştur.

Elde edilen sonuçlardan;

- YSA'nın zeki STS tasarımında başarılı olduğu,
- YSA'nın başarısında saldırıların doğru şekilde tespit edilebilmesi için uygun veri kümesi tasarımının önemli olduğu,
- YSA'ların ara katman sayısı, nöron sayısı ve seçilecek olan fonksiyon tipine bağlı olarak STS performansını artırabildiği,
- YSA performanslarının farklı veri tiplerinde başarılarının ölçülmesinin şart olduğu,
- Eğitim kümesi içerisinde bulunmayan yeni saldırı tiplerinin test esnasında tespit edilebildiği,
- Tasarlanan YSA yapısının kolaylıkla gerçek-zamanlı uygulamalarda kullanılabileceği, Literatürde incelenen çalışmaların birçoğunda öğrenme algoritması olarak BP kullanılmışken, bu çalışmada farklı bir öğrenme algoritmasının da kullanılabileceği ve bununda performansı arttırdığı belirlenmiştir.

Literatürde bulunan %99 gibi yüksek kararlılıkla tanımlama başarısına sahip çalışmalar incelendiğinde zeki STS tasarımlarının nöron sayılarına göre eğitilen ağın her test verisi için sonuç üretme kapasitesinin 1-2 saniye olarak ölçüldüğü görülmektedir [42]. Bununla birlikte çalışmalar sınırlı sayıda saldırı tipi için gerçekleştirildiğinden tüm atak tiplerini içeren, hızlı işlem yapabilme yeteneğine sahip STS tasarımlarına ihtiyaç duyulmaktadır. Başarılı kabul edilen çalışmaların sınırlı sayıda atak (4 tip) için tespit başarısının verilmesi [42] veya bilimsel olmayan bir ifade ile gerçek zamanlı taramalarda ağ üzerinde düşük gecikmeler ile sistemin çalıştığının belirtilmesi [41], STS'lerin her atak tipi için gerçek zamanlı olarak çalışmasının önünde engel teşkil etmektedir.

Çizelge 14. Zeki STS test kümelerinin yüzde başarım oranları (Success rates of Intelligent IDS test sets)

Test Veri Kümesi	YSA#1 (%)	YSA#2 (%)	YSA#3 (%)	YSA Ortalaması (%)
Test Veri Kümesi #1	97,80	97,59	97,92	97,77
Test Veri Kümesi #2	83,48	83,66	84,76	83,96
Test Veri Kümesi #3	88,57	81,93	89,10	86,53
Test Veri Kümesi #4	96,11	88,97	96,09	93,72
Test Veri Kümesi #5	92,57	93,31	94,61	93,49
Test Veri Kümesi Ortalaması (%)	91,70	89,09	92,49	91,09

Bu çalışmada önerilen yöntem, literatürde önerilen ve yüksek başarı oranına sahip çalışmalara göre, daha düşük başarı oranına sahip olmasına rağmen literatürde önerilen yöntemlerin aksine ağın eğitimi için 22 farklı saldırı kullanılmış, literatürde önerilen ara katman nöron sayılarının dörtte birlik kısmı kullanılarak ağın işlem performansı artırılmış ve farklı alt kümelere bölünmüş eğitim ve test verileri kullanılarak ağa öğretilmeyen 14 farklı saldırı tanımlanmıştır.

Yapılan 5 farklı testte (5x65536), 65536 veri için ön işlem süresinin 10 s., YSA tabanlı STS eğitim süresinin 24 s., test süresinin ise 22 s olduğu ölçülmüştür. Bir diğer ifadeyle ağın her giriş verisini test etme süresi 0.00048 s sürmektedir.

Bu çalışma kapsamında gerçekleştirilen zeki STS uygulamasında, farklı eğitim ve test kümeleriyle yapılan testlerden elde edilen en iyi sonuç, %97,92 başarılı iken, en düşük sonuç %81,93 olarak tespit edilmiştir. YSA #3 ise diğer YSA yapılarına göre en yüksek performansı göstermiştir. Gerçekleştirilen sistemin test sonuçları, diğer çalışmalarla karşılaştırıldığında gerek tanıma süresi gerekse tanımlanabilen atak sayısı kriterleri dikkate alındığında daha yüksek başarı elde ettiği görülmüştür. Elde edilen bu başarının sonucunda LM algoritmasının, zeki STS tasarımında kullanılabileceği veya uygulanabileceği görülmüştür.

Çalışma genel olarak değerlendirildiğinde;

- Literatürde yapılan çalışmalarda çok farklı YSA yapısının ve öğrenme algoritmasının kullanılmadığı,
- YSA'ları eğitmede en çok geri yayılım algoritmasının kullanıldığı,
- Literatürde, veri kümesi oluşturmak için yapılan çalışmaların güncel olmadığı,
- Literatürde birçok çalışma mevcut olsa da çalışmalarda veri paketlerinin ayrıştırılması ve anlaşılabilir bir forma dönüştürülmesi için kolaylıkla uygulanabilecek bir yazılım bulunmadığı,
- Ülkemizde zeki saldırı tespit sistemlerine yönelik yeterli çalışma bulunmadığı ve
- Ülkemizde geliştirilecek olan STS'lerin testinde kullanılabilecek bir veri kümesi bulunmadığı

anlaşılmıştır.

Bundan dolayı bu çalışmanın, ülkemizde ilk kapsamlı çalışma olması, YSA uygulamasının tüm adımlarının açıkça sunulması, veri kümelerinin anlamlı bir hale getirilmesi için bir yazılım geliştirilmesi, tasarlanan YSA'nın farklı test kümeleri için bile yüksek başarı sunması, bu çalışmanın önemini arttırmaktadır.

Gerçekleştirilen çalışma, bilgi güvenliğinin sağlanması açısından farklı bakış açıları kazanılmasını sağlamaktadır. Literatürde belirtildiği gibi zeki yaklaşımların STS tasarımında kullanılmasının doğru bir tercih olduğu, farklı yapılar seçerken MLP yapısının LM gibi güçlü algoritmalarla eğitilmesiyle YSA'ların hesaplama süresini düşürebileceği ve farklı saldırı tiplerini belirlemede başarıyı artıracağı görülmüştür.

6. KAYNAKLAR (REFERENCES)

1. Vural Y.**,** Sağıroğlu Ş.**,** Kurumsal Bilgi Güvenliği ve Standartları üzerine bir İnceleme**, Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi,** Cilt 23, No 2, s.507-522, 2008.

2. Canbek G., Sağıroğlu Ş, Casus Yazılımlar: Bulaşma Yöntemleri Ve Önlemler, **Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi**, Cilt 23, No 1, 165-180, 2008.

3. Vural Y., Sağıroğlu Ş, Kurumsal Bilgi Güvenliğinde Güvenlik Testleri ve Öneriler, **Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi**, Cilt:25, No:3, s.484-494, 2010.

4. Sağıroğlu Ş**,** Bulut H., Mobil Ortamlarda Bilgi ve Haberleşme Güvenliği Üzerine Bir İnceleme, **Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi,** Cilt:24, No:3, s.499-508, 2009.

5. 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.

6. Anderson, J.P., "Computer security threat monitoring and surveillance", **Technical Report,** Fort Washington, Pennsylvania**,** 1-30, 1980.

7. 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.

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

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

10. Mukherjee, B., Heberlein, L.T., Levitt, K.N., "Network intrusion detection", **IEEE Network**, 8(3): 26-41, 1994.

11. 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.

12. Endler, D., "Intrusion detection applying machine learning to solaris audit data", **Annual Computer Security Applications Conference (ACSAC'98)**, 268-269, 1998.

13. 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.

14. Patcha, A., Park, J.M., "An overview of anomaly detection techniques: Existing solutions and latest technological trends", **Computer Networks**, 51(12): 3448-3470, 2007.

15. Cannady J., "Artificial neural networks for misuse detection", **Proceedings of the 1998 National Information Systems Security Conference (NISSC'98),** Arlington, VA, 443-456, 1998.

16. Sundaram, A., "An introduction to intrusion detection", Crossroads: The ACM Student Magazine, New York, USA, 2(4), 3-7, 1996.

17. Frank, J., "Artificial intelligence and intrusion detection: current and future directions", **Division of Computer Science,** University of California at Davis**,** 1-12, 1994.

18. Peddabachigari, S., Abraham, A., Grosan, C., Thomas, J., "Modeling intrusion detection system using hybrid intelligent systems", **Journal of Network and Computer Applications**, Elsevier, 30:114–132, 2007.

19. Ryan, J., Lin, M.J., Mikkulainen, R., "Intrusion Detection with Neural Networks", **Advances in Neural Information Processing systems 10**, Cambridge, MA, MIT Press, 1-7, 1998.

20. Kayacik, H. G., Zincir-Heywood, A. N., Heywood M. I., "Selecting features for intrusion detection: A feature relevance analysis on KDD'99 intrusion detection datasets", **Third Annual Conference on Privacy, Security and Trust (PST-2005),** St. Andrews, Canada, 85-89, 2005.

21. Murali, A., Rao, M., "A survey on intrusion detection approaches", First International Conference on Information and Communication Technologies, **IEEE Communications Society Press**, 233-240, 2005.

22. Lee, S.C., Heinbuch, D.V., "Training a Neural-Network Based Intrusion Detector to Recognize Novel Attacks", **IEEE Transactions on systems, man, and Cybernetics-Part A: Systems and Humans,** 31(4), 294-299, 2001.

23. Bivens, A., Palagiri, C., Smith, R., Symanski, B., Embrechts, M., "Network-Based Intrusion Detection Using Neural Networks", **Intelligent Engineering Systems through Artificial Neural Networks (ANNIE-2002),** New York: ASME Press, 579-584, 2002.

24. 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.

25. Yang, X.R., Shen, J.Y., Wang, R., "Artificial Immune Theory Based Network Intrusion Detection System And The Algorithms Design", **First Internetional Conference on Machine Learning and Cybernetics**, Beijing, 73-77, 2002.

26. Moradi, M., Zulkernine, M., "A neural network based system for intrusion detection and classification of attacks," **2004 IEEE International Conference on Advances in Intelligent Systems** - Theory and Applications, Luxembourg-Kirchberg, Luxembourg, 148:1-6, 2004.

27. Mukkamala, S., Sung, A.H., "Artificial Intelligent Techniques for Intrusion Detection", **IEEE International Conference on Systems, Man and Cybernetics**, Washington D.C., USA, 2: 1266 – 1271, 2003.

28. Mukkamala, S., Sung, A.H., "Feature Selection for Intrusion Detection using Neural Networks and Support Vector Machines", **Journal of the Transportation Research Board of the National Academies**, Transportation Research Record No. 1822, Washington D.C., USA, 33-39, 2003.

29. Jackson, K.A., "Intrusion detection system (IDS) product survey", **Technical Report,** LA-UR-99-3883, Los Alamos National Laboratory, Los Alamos, New Mexico, 1-96, 1999.

30. Roy, A., "Artificial neural Networks - A science in trouble", **ACM SIGKDD Explorations,** ACM SIGKDD, 1(2): 33-38, 2000.

31. Sağıroğlu, Ş., Beşdok, E., Erler, M., "**Mühendislikte yapay zeka uygulamaları-1: Yapay sinir ağları**", Ufuk Kitabevi, Kayseri, 10-100, 2003.

32. Güven E.N**.**, **Zeki Saldırı Tespit Sistemlerinin İncelenmesi, Tasarımı ve Gerçekleştirilmesi**, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2007.

33. İnternet: 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, 2007.

34. İnternet: 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, 2007.

35. İnternet: Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları "**2000 DARPA Intrusion Detection Evaluation Data Set Overview**" http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html, 2007.

36. Ghosh, A.K. C. Michael, M. Schatz, A real-time intrusion detection system based on learning program behavior, in: **Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection**, Toulouse, France, 93–109, 2000.

37. Ghosh, A.K. Schwartzbard, A., A study in using neural networks for anomaly and misuse detection, in: **Proceedings of the Eighth USENIX Security Symposium**, Washington, DC, 141–151, 1999.

38. Ramadas, M., Tjaden, S.O.B., Detecting anomalous network traffic with self-organizing maps, in: **Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection**, Pittsburgh, PA, USA, 36–54, 2003.

39. 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, in: **Proceedings of the Second DARPA Information Survivability Conference and Exposition**, Anaheim, CA, 2001, 85–100, 2001.

40. Tan, K.M.C., Maxion, R.A., Determining the operational limits of an anomaly-based intrusion detector, **IEEE Journal on Selected Areas in Communication** 2, 96–110, 2003.

41. Sarasamma, S.T., Zhu, Q.A., Huff, J., Hierarchical Kohonen net for anomaly detection in network security, **IEEE Transactions on Systems, Man and Cybernetics**—PART B: Cybernetics 35, 302–312, 2005.

42. Sung, A.H., Mukkamala, S., Identifying important features for intrusion detection using support vector machines and neural networks, in: **Proceedings of the 2003 Symposium on Applications and the Interne**t, 209–216, 2003.

43. Girardin, L., and Brodbeck, D., "A Visual Approach or Monitoring Logs," **In Proceedings of the 12th System Administration Conference** (LISA'98), Boston, MA, December, 299-308, 1998.