Yapay Sinir Ağları ile Ağ Üzerinde Saldırı Tespiti ve Paralel Optimizasyonu

Mehmet Zahid Yıldırım[1], Abdullah Çavuşoğlu[2], Baha Şen[3], İdris Budak[4]

[1] Karabük Üniversitesi, Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bölümü, Karabük

[2] Yıldırım Beyazıt Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü, Ankara

3 Yıldırım Beyazıt Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü, Ankara

[4] Karabük Üniversitesi, Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bölümü, Karabük

abdullah.cavusoglu@ybu.edu.tr, bsen@ybu.edu.tr, m.zahidyildirim@karabuk.edu.tr, idrisbudak@karabuk.edu.tr

Özet: Günümüzde her gün binlerce sistem saldırıya uğramaktadır. Bu saldırılar temelde 2 tür olup ilki otomatik araçlarla yapılan saldırılar diğeri ise uzman saldırganların çeşitli yöntemler kullanarak gerçekleştirdiği saldırılardır. Firewall ve Anti-Virüs gibi sistemler ilk saldırı tipi için etkili olsa da uzman saldırganlara karşı aynı oranda saldırıyı engelleme imkanı sunamamaktadır. Bu sebeple Saldırı Tespit Sistemleri geliştirilmiş ve günümüze kadar veri madenciliği yöntemleri, yapay sinir ağları, istatistiki yöntemler gibi birçok farklı yöntemlerle kullanılmıştır. Bu çalışmada saldırı tespit sistemleriyle ilgili uygulamalarda en çok kullanılan veri setlerinden biri olan "KDD Cup'99" veri seti kümesi kullanılmıştır. Bu veri seti üzerinde çok katmanlı (multi-layer perceptron: MLP) yapay sinir ağlarının uygulanabilirliği test edilmiş ve paralel programlama ile performans analizleri yapılmıştır.

Anahtar Sözcükler: Saldırı Tespit Sistemleri, Bilgisayar Ağlarında Anomali Tespiti, Yapay Sinir Ağları, KDD Cup'99, Paralel Optimizasyon

Intrusion Detection on Network with Artificial Neural Networks and Parallel Optimization

Abstract: Nowadays, thousands of system is attacked in each day. These attacks are in 2 types basically; the first one is the attacks that made with automatic tools and the other one is made by professional cyber militants with using various methods. As though firewall and antivirus systems are effective for the type of first attacks, they cannot be effective to prevent attacks for professional cyber militants as much as to prevent attacks made with automatic tools. Therefore, Intrusion Detection Systems are developed up to now and used in many methods like data mining, artificial neural networks, statistical methods. In this study "KDD Cup'99" one of the most widely used data set in this area, is used for Intrusion Detection Systems. With this data set, applicability of multi-layer perceptron (MLP) neural networks has been tested and performance analysis was made with parallel programming.

Keywords: Intrusion Detection Systems, Anomaly Detection in Computer Networks, Artifical Neural Networks, KDD Cup'99, Parallel Optimization

1. Giriş

Günümüzde bilgi çok yoğun ve hızlı bir şekilde paylaşılmaktadır. Bu paylaşımın gerçekleşmesinde ise hiç kuşkusuz en büyük payı bilgisayar ağları sağlamaktadır. Bu sebeple bilgisayar ağlarının güvenliği büyük bir öneme sahiptir. Literatürde Anomali Tespit Sistemleri olarakta geçen Saldırı Tespit Sistemleri ağ üzerindeki düzensizlikleri algılayan ve bu durumu ağdan sorumlu yazılıma veya kişilere ileten sistemlerdir. Bu sistemlerde ağda oluşan anormalliğin tespit edilip anında müdahale edilmesi çok önemlidir. Saldırı tespit sistemimiz ne kadar iyide olsa eğer saldırılara gerekli zaman aralığında karşılık veremezse bu sistem uygulanabilir bir sistem olamaz. Günümüzde bu sistemlere verilebilecek gerek açık kaynak kodlu gerek ticari birçok örnek bulunmaktadır. Çalışmamızda ise çok katmanlı bir yapay sinir ağının ağ trafiği üzerindeki muhtemel saldırıları tespit etmesinde paralel programlamanın zaman etkisi analiz edilmektedir.

Andrew Sung ve arkadaşları 2002 yılın yaptıkları "Intrusion Detection Using Neural Networks and Support Vector Machines" başlıklı çalışmalarında yapay sinir ağı modellerinden olan SVM modelini kullanarak bilinen saldırılar ve kullanıcı davranışlarını modellemek için gerçek zamanlı bir çalışma yapmışlardır. Bu çalışmalarında destek vektör makinaları ve yapay sinir ağının performanslarını karşılaştırmışlardır. Sonuç olarak doğruluk oranları birbirine benzer olmasına karşılık SVM' nin eğitim sürelerinde YSA' ya göre çok daha performanslı olduğunu göstermişlerdir(17,77 sn - 18 dk).[1]

Mehmet Özgür DEPREN ve arkadaşları 2004 yılında "SOM Yapısı Kullanarak Ağ Tabanlı Olağandışılık Tespiti" başlıklı çalışmalarında yapay sinir ağı modellerinden olan SOM (Self Organizing Map) yapısı kullanarak ağ tabanlı olağandışılık tespiti yapmışlardır. Bu çalışmada sadece normal bağlantılar için bir matematiksel model oluşturularak, gelen diğer bağlantıların normal davranış modelinden sapmalarına bakılmıştır. Bu modeli oluştururken KDDcup99 veri setinin %10' luk eğitim kümesinde yer alan verileri kullanmışlardır.[2]

Literatürde bu alanda yapılan bazı çalışmalar:

Murat H. SAZLI ve Haluk TANRIKULU 2007 yılındaki "Saldırı Tespit Sistemlerinde Yapay Sinir Ağlarının Kullanılması" başlıklı çalışmalarında yapay sinir ağı kullanarak bir ağ üzerinde akan paketlerin hangi saldırı yöntemini kullandığını tespit etmeye çalışmışlardır. Bu saldırılardan "Neptune" ve "the ping of death" 'in bulunması için Çok Katman Algılayıcılı(Multi Layer Perseptron) yapay sinir ağ modelini kullanmışlar ve DARPA veri setlerini örnek alarak ağlarını eğitmişlerdir. Bu çalışmanın neticesinde internetten gelebilecek DoS ataklarının algılanması başarı ile sağlanmıştır. [3]

Fan ZHANG ve arkadaşları 2009 yılında yaptıkları "Network Instrusion Detection Method Based on Radial Basic Function Neural Network" başlıklı çalışmalarında Radyal tabanlı yapay sinir ağı kullanarak ağlar üzerindeki saldırıları tespit etmeye çalışmışlardır.[4]

Şeref SAĞIROĞLU ve arkadaşları 2010' da "Zeki Saldırı Tespit Sistemi ve Gerçekleştirilmesi" başlıklı çalışmalarında yine KDDcup99 veri setlerini kullanarak MLP modelini kullanarak zeki bir sistem tasarlamışlardır.[5]

2.Ağ Üzerinde Oluşabilecek Saldırı Türleri

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

problemlerdir.[6] Bu tip saldırılar 4 temel kategoride incelenebilirler

1) Bilgi Tarama (Probe ya da scan): Bu saldırılar bir sunucunun ya da herhangi makinanın, geçerli ip adreslerini, aktif portlarını veya işletim sistemini öğrenmek için yapılan saldırılardır. Bilinen saldırılardan bunlara örnek olarak:

ipsweep: belirli bir protu sürekli tarama saldırısı.

Portsweep: bir sunucu üzerindeki hizmetleri bulmak için tüm portları tarama saldırısı.

verilebilir.

2) Hizmet Engelleme (Denial of Service - DoS): Bu saldırılar genelde TCP/IP protokol yapısındaki açıklardan faydalanarak veya bir sunucuya çok sayıda istek yönelterek onu tıkamaya sebep olan saldırılardır. DoS saldırıları kendi içinde gruplara ayrılır. [6]

Özel olarak kullanılan saldırılardan bazıları:

1. **Smurf**: ICMP mesajlarının broadcast ile tüm ağa dağıtılmasıyla oluşur.
2. **Selfping**: Kullanıcının makinaya sürekli ping atmasıyla gerçekleşir.
3. **tcpreset**: Saldırgan kurbanın kurmaya çalıştığı bağlantılar için kurban adına reset göndererek bağlantısını engeller.
4. **mailbomb**: Saldırgan sunucuya sürekli mail gönderir.

3) Yönetici Hesabı ile Yerel Oturum Açma (Remote to Local - R2L): Kullanıcı haklarına sahip olunmadığı durumda misafir ya da başka bir kullanıcı olarak izinsiz erişim yapılmasıdır. Bunlara örnek:

1. **Sshtrojan**: Unix üzerinde çalışan bir trojan saldırısıdır.
2. **guest**: Tahmini kolay şifreleri bularak sisteme girilmesidir.

4) Kullanıcı Hesabının Yönetici Hesabına Yükseltilmesi (User to root - U2R): Bu tip saldırılarda sisteme girme izni olan fakat yönetici olmayan bir kullanıcının yönetici izni gerektirecek işler yapmaya çalışmasıdır. Örnekleri:

1. **Eject**: Solaris üzerinde eject programı ile tampon taşmasına yol açıp, yönetici haklarına sahip olunmasıdır.
2. **Sqlattack**: SQL veritabanı kurulu Linux makinalarda sunucuya bağlanan kullanıcının belirli komutlarla yönetici hakları ile komut satırı elde etmesidir.

Kullandığımız KDDcup99 veri seti de temelde bu belirttiğimiz 4 temel kategorideki saldırı türlerine ait verileri içermektedir.

3. Saldırı Tespitinde Kullanılan Yöntemler

Saldırı Tespit Sistemleri(STS), saldırı tespit yöntemi olarak anormallik tespiti ve kötüye kullanım tespiti olmak üzere iki farklı yaklaşım kullanır. Anormallik tespitine dayanan yaklaşım, sistemdeki kullanıcı davranışlarını modellerken, kötüye kullanım (imza) tespitine dayanan yaklaşım, saldırganların davranışlarını modeller.[7]

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. Saldırıları tespit etme yöntemlerine göre ikiye ayrıldığını bildiğimiz STS' lerin, hangi yöntemin seçileceğine dair dikkat edilmesi gereken noktası, anormallik tespiti yönteminin, bütün kötü davranışları tespit etmeye çalışırken, kötüye kullanım tespiti yönteminin kötü olarak bilinen davranışları tanımaya çalışmak olduğudur. Her iki yöntemin de avantaj ve dezavantajları olduğu göz önünde bulundurularak, tasarımlarda avantajları bir araya toplayan hibrit yaklaşımlardan faydalanmanın daha gerçekçi

olacağı değerlendirilmektedir. [7]

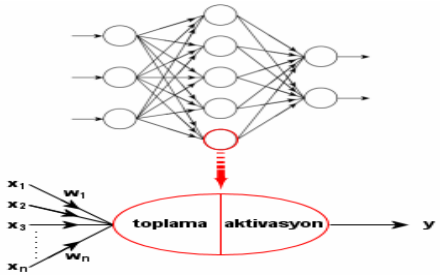
Saldırı tespit sistemleri, daha çok firewall'larda bulunan kural veya imza tabanlı sistemlerden farklı olarak daha dinamiktir ve henüz hakkında bir imza bilinmeyen saldırıları da algılama avantajına sahiptir.

Saldırı tespitinde günümüze kadar en fazla istatistiksel yöntemler kullanılmasına rağmen bunun dışında: durum geçiş diyagramları (state transition diagrams), yapay sinir ağları (artificial neural networks), veri madenciliği (data mining), yapay bağışıklık sistemi (artificial immune system), örüntü eşleme, bulanık mantık (fuzzy logic) gibi farklı birçok yaklaşım uygulanmıştır. [7]

4. Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beyninden esinlenerek geliştirilmiş, ağırlıklı bağlantılar aracılığıyla birbirine bağlanan ve her biri kendi belleğine sahip işlem elemanlarından oluşan paralel ve dağıtılmış bilgi işleme elemanlarıdır.[8]

Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaptik (synaptic) bağlantıların ayarlanması ile olur. İnsanlar doğumlarından itibaren bir yaşayarak öğrenme süreci içerisine girerler. Bu süreç içinde beyin sürekli bir gelişme göstermektedir. Yaşayıp tecrübe ettikçe sinaptik bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşir. Bu durum YSA için de geçerlidir. Öğrenme, eğitme yoluyla örnekler kullanarak olur; başka bir deyişle, gerçekleşme girdi/çıktı verilerinin işlenmesiyle, yani eğitme algoritmasının bu verileri kullanarak bağlantı ağırlıklarını (weights of the synapses) bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla olur.[9]



Şekil 1. Örnek Yapay Sinir Ağı Modeli [10]

Girişler (x1, x2, ..., xn) : Giriş katmanındaki hücreler için, kullanıcı tarafından örnekler ile oluşturulmuş veri kümesidir. Diğer katmandaki hücreler için, herhangi bir katmandaki hücrenin çıkışı olabilir.

Ağırlıklar (w1, w2, ..., wn) : Girişlerin, çıkışa ne oranda aktarılacağını gösterir. Örneğin w1 ağırlığı, x1 girişinin, çıkışa olan setkisini göstermektedir.

Toplama Fonksyonu: Bir hücrenin net girdisini hesaplamak için kullanılır. Bu amaç ile değişik fonksiyonlar kullanılmaktadır. En fazla tercih edilen, ağırlıklı toplam fonksiyonudur.

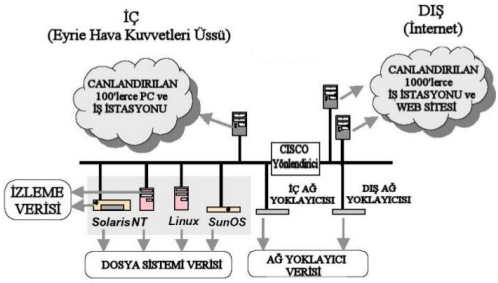
$$NET = \sum_{i=0}^{n} Xi.Wi$$

Nöronların değerleri **aktivasyon fonksiyonu** ile belirlenir. Çıkış nöronunda elde edilen değer ile hedef değer farklı ise geri yayılımlı olarak ağırlıklar tekrar hesaplanır ve öğrenme gerçekleştirilir. Yine bu amaçla çeşitli fonksiyonlar kullanılmaktadır. En çok tercih edilen fonksiyon ise Sigmoid fonksiyonudur.

5. Kullanılan Veri Seti

Saldırı Tespit Sistemleriyle ilgili çalışmalarda en sık kullanılan veri seti DARPA 1998 ve 1999 veri setleridir. Biz de model oluşturma çalışmalarımızda yine bu verilerden türetilen KDD Cup'99 veri setlerini kullanacağız. Veri setini oluşturan kaynak aşağıdaki şekilde de görüldüğü gibi saldırının hedefi olan bir iç ağ ve saldırıyı gerçekleştiren bir dış ağ olmak

üzere iki farklı ağdan oluşmaktadır:



Şekil 2. Darpa Ağının Yapısı [8]

Kullanacağımız veri seti Şekil 2' deki ağda görülen dış ağdan gelen verilerin her gün 22 saatlik periyodlar halinde dinlenilmesiyle oluşturulmuştur.

5.1 Veri Setinin Hazırlanması

Çalışmamızda gerçek kddCupp-99 veri setinin %10' luk kısmına karşılık gelen "kddcup.data_10_ percent_corrected" dosya ismi ile internetten indirilebilen yaklaşık 75Mb büyüklüğünde ve içinde yaklaşık 500bin kayıt bulunan veri seti kullanmıştır. Veri setimizin %60' lık kısmı eğitim için kalanı ise ağımızın test edilmesi için kullanılmıştır. Veri setinin içerisinde toplam 41 adet değişken yani giriş ve 1 adet çıkışımız mevcuttur. Bu 41 adet değişken içerisinde protokol türü, kullanılan servisler, bağlantının normal bir şekilde sonlanıp sonlanmadığı gibi bağlantı hakkındaki bilgiler, bir bağlantı içerisinde çalıştırılan komut sayısı ve yanlış login işlemi sayısı gibi gerçekleştirilen bağlantılarla ilgili çeşitli özellikler tutulmaktadır.

Çalışmamızda kullanılan kddCupp-99 veri setini yapay sinir ağları ile işlemek ve daha verimli sonuçlar alabilmek için bu veri seti üzerinde çeşitli normalizasyon işlemleri uygulanmıştır. Bunlar;

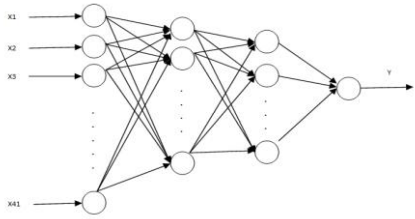
- Tekrar eden kayıtların silinmesi
- Karaktersel ifadelerin sayısallaştırılması
- Sayısallaştırılan ifadelerin 0-1 aralığına normalize edilmesi
- Tutarsız verilerin veri setinden çıkarılmasıdır.

Kullanılacak olan veri setinin normalizasyon işlemleri sonundaki örnek görüntüsü Tablo 1' deki gibidir.

Tablo 1. Veri Setinin Örnek Görüntüsü

0	1	0,169492	1	0,011677	0	1		0	0	0
0	1	0,169492	1	0,011677	0	1		0	0	0
0	1	0,169492	1	0,011677	0	1		0	0	0
0	1	0,169492	1	0,011677	0	1		0	0	0
0	1	0,169492	1	0,011677	0	1		0	0	0
0	1	0,169492	1	0,011677	0	1		0	0	0
0	1	0,169492	1	0,011677	0	1	...	0	0	0
0	1	0,169492	1	0,011677	0	1		0	0	0
0	1	0,169492	1	0,011677	0	1		0	0	0
0	0	0,305085	0,142857	0	0	1		0	1	1
0	0	0,305085	0,142857	0	0	0,09		0	1	1
0	0,5	0,644068	1	0,001188	0,000132	0		0	0	0

6. Ağın Eğitimi ve Test Edilmesi



Şekil 3. Kullanılan YSA Modeli

Gerekli önişlemler neticesinde elde etmiş olduğumuz veri seti üzerinde ağımızı Matlab R2012b programının neural network kütüphanesinden faydalanarak eğitimi gerçekleştirildi. Bu eğitim için kullandığımız veri setinin rastgele seçilmiş %60' lık kısmı eğitim, kalan %40' lık kısmı ise test için kullanıldı. Şekil 3' te kullanılan ağ modelinin örnek bir görüntüsü bulunmaktadır. Şekilde de görüldüğü üzere giriş katmanında 41, çıkış katmanında ise 1 adet nöron bulunmaktadır.

Veri seti üzerinde ağımızın eğitimi ve testi sırasında 1, 2 ve 3 ara katmanlı ağlar üzerinde değişken nöron sayıları ile bir çok ağ modeli test edilmiştir.

Tablo 2. Eğitim sonuçları

				Eğitim		Paralel İşlemler (Eğitim için)		
Ara Katman Sayısı	Nöron sayıları	İterasyon Sayısı	TF	Başarı (%)	Süre (sn)	2 Çekirdek	3 Çekirdek	4 Çekirdek
1	6	35	logsig	99,943	184,450776	94,052562	70,868346	63,916461
1	8	7	logsig	99,294	72,308183	35,717875	26,900167	23,814835
1	16	34	tansig	99,957	1096,041348	556,599006	464,577901	632,977564
2	4,8	37	logsig, tansig	99,929	201,200155	101,347288	73,776957	61,004517
2	6,8	36	tansig, logsig	99,964	382,646972	192,973639	143,687963	119,576525
2	6,8	41	logsig, logsig	99,98	435,223957	219,346103	161,004495	138,698294
2	6,10	24	logsig, logsig	99,956	299,967844	151,601056	112,001011	91,765822
2	10,16	12	logsig, tansig	99,827	495,64142	252,110441	185,469767	180,048401
2	12,16	53	tansig, tansig	99,98	2663,964064	1349,96074	1226,313452	1457,838476
2	12,20	112	logsig, logsig	99,98	6801,565301	3487,563989	3630,138808	3988,68265
2	20,30	54	logsig, logsig	99,979	14491,66607	9205,892797	7777,757488	7482,953821
3	3,4,5	17	tansig, logsig, logsig	99,86	71,962346	37,468074	27,790284	25,069898
3	4,6,12	29	tansig, tansig, logsig	99,928	367,277419	185,986939	139,933876	122,166367
3	6,8,10	22	logsig, logsig, tansig	99,958	462,255387	234,292991	170,962308	142,116613

Eğitim sonuçları üzerinde genel olarak %99' un üzerinde bir başarı sağlanmıştır. Eğitim süreleri ise ağdaki katman ve nöron sayılarına bağlı olarak artış göstermiştir. Paralel işlemler neticesinde ise Tablo 2' gösterilen oranlarda bir zaman performansı sağlanmıştır.

Tablo 3. Eğitim setindeki başarı yüzdeleri

Paralel İşlemler (Eğitim için)		
2 Çekirdek	3 Çekirdek	4 Çekirdek
%46,81	%57,29	%61,28

Paralel işlemler uygulanırken fiziksel olarak 4 çekirdeği bulunan bir işlemci üzerinde bu işlemler uygulanmıştır. Matlab programı 12 paralel işleme kadar izin vermesine rağmen fiziksel çekirdek sayısı aşıldıktan sonra zaman değerleri tutarsız değerler vermiştir. Bu sebeple paralelleştirme uygulamaları 4 çekirdekte sınırlandırılmıştır.

Tablo 4. Test sonuçları

				Test		Paralel İşlemler (Test için)		
Ara Katman Sayısı	Nöron sayıları	İterasyon Sayısı	TF	Başarı (%)	Süre (sn)	2 Çekirdek	3 Çekirdek	4 Çekirdek
1	6	35	logsig	90,763	0,073482	0,073483	0,073124	0,07344
1	8	7	logsig	90,537	0,077363	0,078252	0,078092	0,07802
1	16	34	tansig	90,917	0,10592	0,105879	0,106032	0,106051
2	4,8	37	logsig, tansig	90,654	0,082982	0,081163	0,081263	0,08088
2	6,8	36	tansig, logsig	89,935	0,089923	0,089818	0,090372	0,089675
2	6,8	41	logsig, logsig	90,572	0,090363	0,089921	0,090973	0,090088
2	6,10	24	logsig, logsig	90,3	0,093511	0,094241	0,094413	0,095143
2	10,16	12	logsig, tansig	90,868	0,120942	0,119426	0,120438	0,119982
2	12,16	53	tansig, tansig	90,537	0,128718	0,126972	0,136185	0,126846
2	12,20	112	logsig, logsig	89,896	0,220357	0,140743	0,133481	0,131947
2	20,30	54	logsig, logsig	90,561	0,188073	0,189454	0,189647	0,203638
3	3,4,5	17	tansig, logsig, logsig	90,87	0,083032	0,08566	0,082705	0,081991
3	4,6,12	29	tansig, tansig, logsig	90,436	0,106178	0,104978	0,106491	0,105001
3	6,8,10	22	logsig, logsig, tansig	90,67	0,113692	0,113575	0,113226	0,112445

Test sonuçlarının başarı oranı ortalama %90,61 olarak bulunmuştur. Ancak test sürelerinin saniyenin yaklaşık %10' u kadarlık sürelerde gerçekleşmesi sebebiyle paralel işlemler neticesinde tutarlı sonuçlar alınamamıştır. Test için 2, 3 ve 4 çekirdek için elde edilen zaman değerleri yine tek çekirdekte elde edilen zaman değerlerine yakın sonuçlar üretmiştir.

7. Sonuç ve Öneriler

Sonuç olarak uyguladığımız ağ modelleri bilinen veriler üzerinde %99' un üzerinde, eğitim setimizde yer almayan test verileri üzerinde ise %90 üzerinde bir başarı oranı sağlamıştır. Paralel işlemlerde fiziksel olarak 4 çekirdekli bir işlemci kullanılmış ve 4 çekirdekte eğitim sürelerinde %61,28 oranında daha hızlı sonuçlar elde edilmiştir.

8. Kaynaklar

[1] Zhang, F., Gao, M. and Tian, J., "Network Intrusion Detection Method Based on Radial Basic Function Neural Network", (2009).

[2] Depren, M.Ö., Topallar, M., Anarım, E. ve Cılız, K., "SOM Yapısı Kullanarak Ağ Tabanlı Olağandışılık Tespiti Network-Based Anomaly Intrusion Detection System Using SOMs", (2004).

[3] Sağıroğlu, Ş., Yolaçan, E.N. ve Yavanoğlu, U., "Zeki Saldırı Tespit Sistemi Tasarımı ve Gerçekleştirilmesi", Ankara, 2011.

[4] SAZLI, M.H. , "Saldırı Tespit Sistemlerinde Yapay Sinir Ağlarının Kullanılması", Ankara, 2007.

[5] Sung, A., Janoski, G. and Mukkamala, S., "Intrusion Detection Using Neural Networks and Support Vector Machines", New Mexico, USA, 2002.

[6] M. Erol, "Saldırı Tespit Sistemlerinde İstatistiksel Anormallik Belirleme Kullanımı", (2005).

[7] Budak, İ, Şen, B ve YILDIRIM, M.Z., "Lojistik Regresyon ile Bilgisayar Ağlarında Anomali Tespiti", Akademik Bilişim (2013).

[8] Elmas, Ç., “Yapay Zeka Uygulamaları”, Ankara: Seçkin Yayınları, 2007.

[9]http://tr.wikipedia.org/wiki/Yapay_sinir_a%C4%9Flar%C4%B1 Sayfa Görüntüleme Tarihi: 08.12.2013

[10] Öğücü, M.O., “Destek Vektör Makinelerini Kullanarak Yüz Bulma”, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü (2006).