ÖZDÜZENLEMELİ AĞLAR KULLANILARAK AĞ TRAFİĞİNİ ETKİLEYECEK SALDIRI TESPİTİ

Tevfik KIZILÖREN

Yüksek Lisans Tezi

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ocak – 2009

JÜRİ VE ENSTİTÜ ONAYI

Tevfik Kızılören'ın **Özdüzenlemeli Ağlar Kullanılarak Ağ Trafiğini Etkileyecek Saldırı Tespiti** başlıklı **Elektrik Elektronik Mühendisliği** Anabilim Dalındaki, Yüksek Lisans tezi 15/01/2009 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	**İmza**
Üye (Tez Danışmanı)	**: Yard. Doç Dr. EMİN GERMEN**	**……..**
Üye	**: Doç. Dr. YUSUF OYSAL**	**……..**
Üye	**: Yard. Doç. Dr. ATAKAN DOĞAN**	**……..**

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ……...…… tarih ve …… sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

ÖZDÜZENLEMELİ HARİTALAR KULLANARAK AĞ TRAFİĞİNİ EKTİLEYECEK SALDIRILARIN TESPİTİ

Tevfik KIZILÖREN

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr. Emin GERMEN
2009, 51 sayfa

Ağ trafiği üzerindeki anomalileri tespit etme bilgisayar biliminin üzerinde en çok durulan konularından bir tanesidir. Bu çalışma ağ trafik davranışını analiz etmek için yeni bir sınıflandırma yöntemi içermektedir. Ağ trafiğini, saldırılar, dosya indirme, port tarama gibi belli başlı bilinen anomalilerden ayırabilmek için çok bilinen bir Yapay Sinir Ağı mimarisi olan Özdüzenlemeli Ağlar kullanılmıştır. Trafiğin ölçümü çalışmanın ilk kısmında Basit Ağ Yönetim Protokolü (SNMP) aracılığıyla yapılmıştır. İkinci kısımda ise KDD Cup tarafından 1999 yılında hazırlanmış olan bir veri kümesi kullanılmıştır. Bu veri kümesi ilk kısımdan farklı olarak Temel Bileşen Analizine tabi tutulmuş ve Temel Bileşen Analizinin bu sistemin karar ve0rme başarısına olan etkisi gözlenmiştir. İkinci kısımda eğer optimum temel bileşen sayısı kullanılırsa sistemin başarı oranının düşmediği aksine bir miktar yükseldiği görülmüştür. Her iki kısımda da elde edilen sonuçlar başarı yüzdesi olarak literatürdeki diğer uygulamalara oranla oldukça tatmin edicidir.

Anahtar Kelimeler: Ağ trafiğinin sınıflandırılması, saldırı tespiti, SNMP, Özdüzenlemeli Ağlar, Temel Bileşen Analizi, Yapay Sinir Ağları

ABSTRACT

Master of Science Thesis

NETWORK INTRUSION DETECTION BY USING SELF ORGANIZING MAPS

Tevfik KIZILÖREN

Anadolu University
Graduate School of Sciences
Electrical Electronics Engineering Program

Supervisor: Assist. Prof. Dr. Emin GERMEN
2009, 51 pages

Anomaly detection in network traffic is one of the most challenging topics in the study of computer science and networking. This work introduces a classification method for analyzing network traffic behavior. In order to distinguish the normal traffic with well-known anomalies such as port scanning and Denial of Service (DOS) attacks, Self Organizing Maps (SOMs), one of the well-known artificial neural network architecture, is used. In the first part of this work, Simple Network Management Protocol (SNMP) performs the measurement of network traffic. In the second part, the dataset prepared by KDD is used. Unlike first part, the dataset is subjected to Principal Component Analysis. In this part, the result we obtained implies that if optimum number of Principle Components is used the decision rate of system is improved. It is worth to mention that impressively satisfactory results have been obtained.

Keywords: network traffic classification, intrusion detection, anomaly detection, SNMP, SOM, self organizing maps, neural networks, principle component analysis

İÇİNDEKİLER

ÖZET ..... **i**
ABSTRACT ..... **ii**
İÇİNDEKİLER ..... **iii**
ŞEKİLLER DİZİNİ ..... **v**
ÇİZELGELER DİZİNİ ..... **vi**
SİMGELER VE KISALTMALAR DİZİNİ ..... **vii**

1. GİRİŞ **1**
2. YAPAY SİNİR AĞLARI VE ÖZDÜZENLEMELİ HARİTALAR **4**
2.1 Yapay Sinir Ağları 4
2.2 Eğiticili ve Eğiticisiz Öğrenme 5
2.3 Özdüzenlemeli Haritalar 6

3. TEMEL BİLEŞEN ANALİZİ **12**
3.1 Tarihçe ve Tanım 12
3.2 Temel Bileşen Analizinin Özellikleri 15
3.2.1 Doğrusallık Varsayımı 15
3.2.2 Ortalama ve Varyansın İstatistiksel Önemi Varsayımı 15
3.2.3 Yüksek Varyansların Önemli Olduğu Varsayımı 16
3.3 Temel Bileşen Analizinin Kovaryans Yöntemiyle Hesaplanması 16
3.4 Temel Bileşen Analizinin Uygulamaları 19

4. ÖZDÜZENLEMELİ HARİTALAR KULLANILARAK AĞ TRAFİĞİNİN SINIFLANDIRILMASI **20**
4.1 Ağ Trafiğinin Ölçümü 20
4.1.1 Basit Ağ Yönetim Protokolü (SNMP) 20
4.1.2 SNMP Aracılığıyla Trafik Bilgisinin Toplanmazı 23
4.2 Verinin Toplanması 23

4.3 Toplanan Veriden Girdi Vektörlerinin Oluşturulması 25
4.4 Özdüzenlemeli Haritanın Eğitilmesi 29
4.5 Sonuçlar 29
5. TEMEL BİLEŞEN ANALİZİNİN SINIFLANDIRMAYA ETKİSİ VE KDD CUP 1999 VERİ KÜMESİNİN ÖZDÜZENLEMELİ HARİTALAR YARDIMIYLA SINIFLANDIRILMASI 33
5.1 Giriş 33
5.2 KDD Cup Veri Kümesi Üzerine 33
5.3 Girdi Vektörlerinin Sayısallaştırılması 38
5.4 Girdi Vektörlerinin Temel Bileşen Analizine Tabi Tutulması ve Boyut Düşürülmesi İşlemi 41
5.5 Özdüzenlemeli Haritanın Oluşturulması ve Eğitilmesi 43
5.6 Sonuçlar 45
6. SONUÇLAR VE İLERİDEKİ ÇALIŞMALAR 47

KAYNAKLAR 49

ŞEKİLLER DİZİNİ

2.1 Yapay Sinir Hücresi Şeması....4
2.2 Kohonen Özdüzenlemeli Haritası....6
2.3 Ağırlık Girdi Uzayında ve Fiziksel Uzayda Kohonen Haritası....9
2.4 Sınıflandırmak İçin Rastgele Seçilmiş 60 Adet Üçgen....10
2.5 Eğitimden Sonra Üçgenlere Karşılık Gelen Vektörler....10
3.1 Temel Bileşen Analizi Örnek Şekil -1....13
3.2 Temel Bileşen Analizi Örnek Şekil -2....13
3.3 Temel Bileşen Analizi Örnek Şekil -3....14
4.1 SNMP Protokolünün Çalışma Yapısı....21
4.2 MYSNMP Programının Bir Ekran Görüntüsü....25
4.3 Dosya İndirme Sırasında Oluşan Ağ Trafiği....27
4.4 Port Tarama Sırasında Oluşan Ağ Trafiği....27
4.5 Normal Ağ Trafiği....28
4.6 Eğitim Sonucunda Elde Edile U-Matrisi....29
4.7 Renklendirilmiş Harita....30
4.8 Dosya İndirme Sırasında Vektörlerin Haritada İzlediği Yol....31
5.1 Girdi Vektörlerinin Orjinal Görüntüsü....41
5.2 Girdi Vektörlerinin Sayısallaştırılmış Görüntüsü....42
5.3 İlk 3 Temel Bileşenin Kullanımı Sonucu Oluşan Harita....43
5.4 İlk 5 Temel Bileşenin Kullanımı Sonucu Oluşan Harita....44
5.5 İlk 9 Temel Bileşenin Kullanımı Sonucu Oluşan Harita....45

ÇİZELGELER DİZİNİ

4.1 N=5 iken oluşan girdi vektörlerini.....26
5.1 TCP bağlantısının temel özellikleri 35
5.2 Bağlantıya ait detaylı özellikler.....36
5.3 Bağlantıya ait trafik özellikleri.....37
5.4 Eğitim veri kümesinde bulunan saldırıların listesi ve türleri 38
5.5 Üçüncü kolonun alabileceği değerler.....39
5.6 Son kolonun alabileceği değerler.....40
5.7 Değişen N değerlerine göre sonuçlar.....46

SİMGELER ve KISALTMALAR DİZİNİ

X_i	: Girdi vektörü
W_i	: Her bir nodun ağırlık vektörü
$\beta(i,c,k)$	: Komşuluk fonksiyonu
$\alpha(k)$	: Öğrenme oranı parametresi
c	: En iyi eşleşen sinir hücresi
X	: Temel bileşen analizine tabi tutulacak girdi matrisi
u	: Deneysel ortalama vektörü
H	: Sapma matrisi
E	: Beklenen değer
C	: Kovaryans matrisi
V	: Köşegenleştirilmiş özvektör matrisi
Z	: Z-score matrisi
Y	: Temel bileşen matrisi
KLT	: Karhunen-Loeve Dönüşümü
BMU	: En iyi eşleşen sinir hücresi
SOM	: Özdüzenlemeli Harita

1. GİRİŞ

Ağ trafiğinin sınıflandırılması ve ağ trafiğinde oluşan anomalileri tespit etmek bilgisayar biliminin en ilgi çeken konularından bir tanesidir. Ağ trafiğini sınıflandırmak için kullanılan yöntemler sınıflandırma yaparken izledikleri yola göre ikiye ayrılabilirler [1]:

1. Anomali tespit etme(Anomali Detection).
2. Yanlış kullanım tespit etme(Misuse Detection).

Anomali tespit etme yeni karşılaşılan davranışların sistemin normal davranışlarıyla uyumunu inceleyip herhangi bir uyumsuzluk olduğunda alarm üretme prensibine göre çalışmaktadır. [1] Yanlış kullanım tespitinde ise sistemde daha önce meydana gelen anormal davranışların imzaları kaydedilir ve yeni davranışların bu anormal davranışlarla benzerliklerinin olup olmadığı kontrol edilir [2]. Anomali tespit sistemleri genelde belli bir sınırın aşılıp aşılmadığını kontrol ederken, yanlış kullanım tespit sistemleri genelde kural tabanlı bir yöntem izlerler. Bu kurallar genelde daha önceden elde edilmiş yanlış kullanım senaryolarıdır [2].

Her iki yöntemin de avantajları ve dezavantajları vardır. Anomali tespit sistemleri yeni durumlara karşı daha başarılıdır; fakat bulguları kesinlikten uzaktır. Yanlış kullanım tespiti prensibine göre çalışan sistemler ise yeni durumlara karşı savunmasızdır [2].

Sistemden beklenen davranışa göre her iki yöntemden biri seçilebilir. Bu çalışmada ağ trafik davranışlarının sınıflandırılması amaçlanmıştır. Ağdaki anomalileri tespit etmek için literatürde çok sayıda çalışma bulunmaktadır. Anomali tespiti için yapılan çalışmalarda uygulanan yöntemlerden bazıları şunlardır:

- Yapay Sinir Ağları [3].
- Destekçi Vektör Makinesi ve türevleri [4][5][6].
- Çekirdek Bazlı Yöntemler.
- Bulanık Mantık [7].
- Fonksiyonel Ağlar ve Yapay Sinir Ağlarının bazı genellemeleri [8].
- Çeşitli hibrid yaklaşımlar [9].

Tezin ilk kısmında ağ trafiğinin sınıflandırılması için bir test ortamı oluşturulmuş, belirli zaman dilimlerinde trafik üretilmiş ve SNMP(Simple Network Management Protocol) başta olmak üzere çeşitli network araçları ve protokolleri kullanılarak üretilen trafiğin ölçümü yapılmıştır.

Trafik davranışlarını sınıflandırabilmek için bir Yapay Sinir Ağı (YSA) türü olan Özdüzenlemeli Haritalar (SOM-Self Organizing Maps) kullanılmıştır. Bu çalışmada özdüzenlemeli haritaların kullanılmasının nedeni ağ trafiğinin önceden tahmin edilemeyen bir yapıya sahip olmasıdır. Özdüzenlemeli haritalar hem kendi kendilerine öğrenerek bilinmeyen girdiler üzerinde sınıflandırma yaparlar hem de yüksek boyutlu girdi uzaylarını düşük boyutlarda görselleştirebilirler. Bu da bize takip eden girdi vektörlerinin harita üzerinde hangi bölgelerde yoğunlaştığının görülebilmesini sağlamaktadır.

Tezin ilk aşamasında belli başlı bazı trafik türlerini birbirinden ayırmak amaçlanmıştır. Bu trafik türleri normal ağ trafiği, port tarama sırasında üretilen trafik ve dosya indirme sırasında üretilen trafik olarak belirlenmiştir.

Trafik ölçümleri yapılırken belirli zaman aralıklarında ölçüm yapılan anahtar cihazının portlarından geçen trafik miktarı bulunmuş ve kaydedilmiştir. Kaydelilen vektörler üzerinde bir zaman penceresi oluşturulmuş ve bu pencere vektörler üzerinde kaydırılarak 20 elemanlı girdi vektörleri oluşturulmuştur.

Girdi vektörleri sisteme sokulmadan önce hangi ağ trafik çeşidine karşılık geldiğinin belirlenebilmesi ve sınıflandırma işlemi sonucunda sistemin başarısının ölçülebilmesi için etiketlenmiştir. Bu girdi vektörleri kullanılarak çeşitli boyutlarda özdüzenlemeli haritalar oluşturulmuştur.

Daha önceden elde edilen vektörler sisteme girdi olarak verilmiş ve bu haritalar girdi vektörleri ile eğitilmiştir. Sonuç olarak birbirine matematiksel olarak benzerlik gösteren girdi vektörlerinin, çok büyük oranlarda, haritada birbirlerine yakın konumlarda toplandığını, birbirlerinden farklı olan vektör guruplarının ise haritanın farklı bölgelerinde bulunduğu görülmüş ve başarılı sonuçlar elde edilmiştir.

Ağ trafiği süregelen bir süreç olduğu için bulunan sonuçlara bir iyileştirme olarak ard arda gelen vektörlerin harita üzerinde izlediği yolların takibi yapılmış ve vektörlerin belirli bir zaman aralığı içerisinde haritanın hangi kısımlarında

toplandığını bulan bir iz sürme algoritması oluşturulmuştur. Bu algoritma haritanın sınıflandırma yaparak oluşturduğu karar verme mekanizmasını büyük oranda iyileştirmiştir.

Tezin ikinci kısmında ise akademik araştırmalarda geçerliliği olan bir test veri kümesi olan Bilgi Keşfi ve Veri Madenciliği (KDD - Knowledge Discovery ve Data Mining Tools Conference) yarışmasının [10] 1999 yılında hazırladığı veri kümesi kullanılarak bu test verisinin sınıflandırılması yapılmıştır.

Bu yarışmada yarışmacılardan Bilgisayarlı Hesaplama yöntemlerini kullanarak DARPA (Defense Advanced Research Projects Agency) tarafından oluşturulmuş bir test ortamında normal ve anormal trafik verileri elde edilmiş ve bunların sınıflandırılması istenmiştir [11].

Bu yarışmadan sonra bu veri kümesi anomali tespit etme ve saldırı tespit etme ile ilgili akademik çalışmalar için bir referans veri kümesi haline gelmiştir. Bu test kümesinde tezin birinci kısmından farklı olarak elde edilen girdi vektörlerin belirli bir kısmı Temel Bileşen Analizine (PCA – Principle Component Analysis) tabi tutulmuştur.

İlk kısımdan farklı olarak özdüzenlemeli haritaya girdi olarak temel bileşen analizi sonucunda elde edilen vektörler sunulmuştur. Bu kısımda da normal trafikten saldırı trafiğinin ayrılması konusunda çok başarılı sonuçlar elde edilmiş ve temel bileşen analizinin getirdiği boyut düşürme işlevi sonucunda vektörlerin kendisinin sisteme girdi olarak sunulduğu duruma göre çok daha hızlı eğitim sonuçları alınmıştır.

2.YAPAY SİNİR AĞLARI VE ÖZ DÜZENLEMELİ HARİTALAR

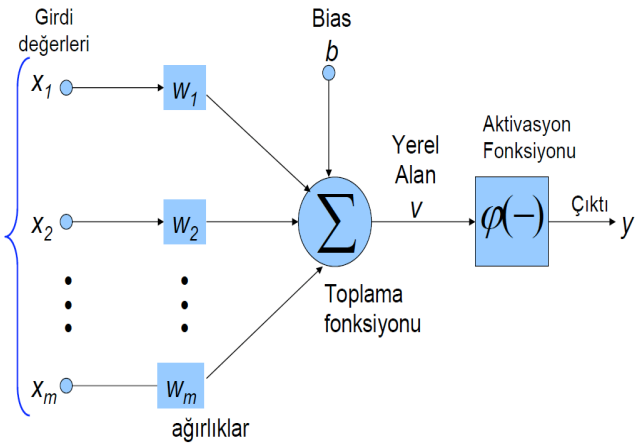
2.1 Yapay Sinir Ağları

Yapay Sinir Ağları (ANN - Artificial Neural Networks), birbirine bağlı çok sayıda işlem elemanlarından oluşan genellikle paralel işleyen yapılar olarak adlandırılabilir. Yapay Sinir Ağları insan beynindeki sinir hücrelerinin çalışma biçimleri temel alınarak geliştirilmiş ve benzerlik kurularak çeşitli uygulamalarla bilgisayarın yüksek hesap gücünü kullanarak karmaşık problemleri çözebilmek amacıyla geliştirilmiş bir yöntemdir.

Yapay sinir ağlarının temel birim işlem elemanı ya da düğüm (node) olarak adlandırılan Şekil 2.1 'de görüldüğü gibi yapay bir sinir hücresidir. Yapay sinir ağları, ağırlıklandırılmış şekilde birbirlerine bağlanmış birçok işlem biriminden (sinir hücreleri) oluşan matematiksel sistemlerdir.

Bir işlem birimi, genelde transfer fonksiyonu olarak kullanılan bir denklemdir [12]. Bu işlem birimi diğer sinir hücrelerinden sinyalleri alır; bunları birleştirir, dönüştürür ve sayısal bir sonuç ortaya çıkartır.

Yapay sinir ağı temelli hesaplamanın merkezinde dağıtılmış, adaptif ve doğrusal olmayan işlem kavramları vardır. Yapay sinir ağları, geleneksel işlemcilerden farklı şekilde işlem yapmaktadırlar.



Şekil 2.1: Yapay sinir hücresi şeması.

Geleneksel işlemcilerde tek bir merkezi işlem birimi her hareketi sırasıyla gerçekleştirir. Yapay sinir ağları ise herbiri büyük bir problemin bir parçası ile ilgilenen, çok sayıda basit işlem birimlerinden oluşmaktadır. Böylece paralelleştirmeye olan yatkınlıkları ortaya çıkmaktadır [12].

Herhangi bir yapay sinir ağının temel birimine sinir hücresi adı verilir. Bir sinir hücresinin temel özellikleri ve parçaları aşağıda listelenmiştir [13]:

- Dışarıdan veya başka bir işlem biriminden gelen belirli sayıda girdi vektörü (input vectors).
- Aktivasyon Fonksiyonu.
- Dışarıdan gelen girdi vektörünün temel işlem birimi olan sinir hücres üzerindeki etkisinin derecesini belirleyen ağırlık (weights) olarak adlandırılan elemanlar.
- Girdi vektörlerinin genellikle ağırlıklandırılmış toplamlarından oluşan bir toplam fonksiyonu (summation function).
- Girdi vektörlerinin toplam fonksiyonu kullanarak bulunan toplamlarından elde edilen sinyalin diğer işlem birimlerine, sinir hücrelerine, iletilip iletilmeyeceğine karar verilmesini sağlayan bir sınır değeri (treshold value).
- Bir çıktı sinyali.

2.2 Eğitimcili ve Eğitimcisiz Öğrenme

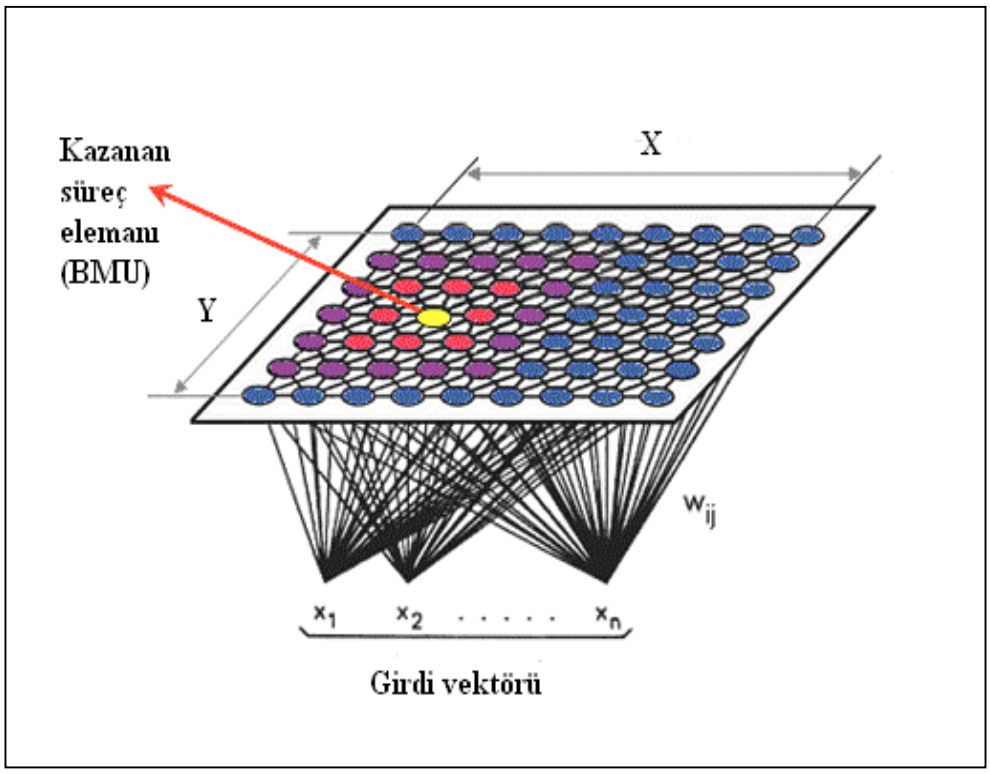
Bir yapay sinir ağı elemanları arasındaki ağırlıkların ayarlanmasıyla herhagi bir özel fonksiyonu gerçekleştirmek için eğitilebilir. Sıklıkla karşılaşılan bu tip durumlarda yapay sinir ağının elemanları belirli girdilere belirli çıktılar verecek şekilde eğitilirler. Bu tip öğrenme işlemine Eğiticili Öğrenme(Supervised Learning) adı verilir. Bazı tek ve çok katmanlı algılama algoritmaları (single/multi layer perception) ve Geri Yayılımlı Ağlar (Backpropogation Networks) eğitimcili öğrenme kullanan ağlara örnektir.

Eğiticili öğrenmenin aksine, yapay sinir ağlarının tamamen kendi başlarına bırakıldığı ve sonuç ağırlıklarına ve değerlerine kendiliğinden ulaşıldığı bir başka

eğitim türü vardır. Buna Eğiticisiz Öğrenme (Unsupervised Learning) adı verilmektedir. Öz Düzenlemeli Haritalar (Self Organizing Maps) ise eğiticisiz öğrenme kullanan ağlara örnek olarak gösterilebilir.

2.3 Özdüzenlemeli Haritalar

Özdüzenlemeli haritalar (SOM – Self Organizing Maps), yapay sinir ağlarının özel bir biçimidir. Teuvo Kohonen tarafından geliştirilen öz düzenlemeli haritalar Kohonen haritaları adıyla da bilinirler [15].



Şekil 2.2: Kohonen Özdüzenlemeli Haritası.

Özdüzenlemeli haritalar topoloji-korumalı bir harita türüdür. Bu harita; yüksek boyutlu vektörleri (üç veya daha fazla), tipik bir iki boyutlu ızgara formundaki haritada temsil edilebilmesine olanak tanır. Özdüzenlemeli haritanın ana amacı, girdi uzayındaki komşuluk ilişkilerini mümkün olduğunca koruyan bir harita yaratmaktır [12].

Öz Düzenlemeli Haritalar çok güçlü rekabetci Hebbian tipi bir yapay sinir ağı türüdür [14]. Bu yapay sinir ağı türünün rekabetçi olarak nitelendirilmesinin nedeni eğitim safhasında her zaman bir kazanan sinir hücresi bulunmasından kaynaklanır. Bu sinir hücresine En iyi uyumlu birim (BMU – Best Matching Unit) adı verilir [15].

Özdüzenlemeli haritalarda öğrenme sırasında eğiticisiz öğrenme kullanılır. Özdüzenlemeli haritalar iki boyutludur ve her bir sinir hücresi genelde kare veya altıgen şeklinde tasarlanmıştır [14]. İlk çalışma anında sistem kendini eğitmektedir.

Bu çalışma şeklindeyken yarışmacı öğrenme (Competitive Learning) kullanılmaktadır. İkinci safhası olan haritalama safhasında ise sistem gelen girdiyi doğru haritalamak için çalışır. Sistem çalışırken gerçekleştirilen işlem temel olarak çok boyutlu girdi vektörlerinin daha az boyuttaki vektör çıktılarına indirgenmesine dayanan çalışma şekline sahiptir.

Bu durum aslında problemin basitleştirilmesini amaçlayan bir boyut azaltma (Dimensionality Reduction) işlemine karşılık gelir. Özdüzenlemeli haritalar çalışırken kullanılan algoritmanın çalışması aşağıda listelenen 5 basamakta verilmiştir:

1. Kohonen ağındaki sinir hücrelerinin ağırlık değerleri rastgele olarak seçilir.
2. Girdi vektörleri alınır. Girdi vektörü, $X = \left[x_1, x_2, \ldots, x_n\right]^T \in R^n$ olsun. Bir *i* indisi ile düzenlenmiş birimlerin ayrık bir ızgarasını göz önüne alalım. Her bir düğümün kendine ait bir ağırlık verktörü $W_i = \left[w_1, w_2, \ldots, w_n\right]^T \in R^n$ bulunmaktadır.
3. Girdi vektörlerinin herbiri için haritadaki bütün sinir hücreleri dolaşılır ve
    a. Üzerinden işlem yapılan girdi vektörü ile dolaşılmakta olan sinir hücresi arasındaki mesafe genellikle öklit mesafesi(ED–Euclidian Distance) olarak hesaplanır. Öklit

mesafesi dışında başka matematiksel mesafe hesaplamaları da kullanılabilir

b. Hesaplanan mesafelere göre en küçük uzaklığa sahip olan düğüm bulunur. Bu düğüme En İyi Eşleşen Sinir Hücresi adı verilir. En iyi eşleşen sinir hücresinin bulunması için aşağıdaki formül kullanılır:

$$\left\|X - W_j\right\| = \min_i \left\|X - W_i\right\| \qquad (2.1)$$

4. En iyi eşleşen sinir hücresine komşu olan bütün sinir hücreleri güncellenerek girdi vektörüne yaklaştırılır. Yakınlaştırma işlemi için aşağıdaki formül kullanılır [5]:

$$M_i(k) = M_i(k-1) + \beta(i,c,k)\alpha(k)(\lambda(k) - M_i(k)) \quad 1 \le i \le m \qquad (2.2)$$

5. $k < \lambda$ olduğu sürece 2. adıma dönülerek işlemler tekrar edilir. Buradaki λ eğitim başında belirlenmiş adım sayısıdır.

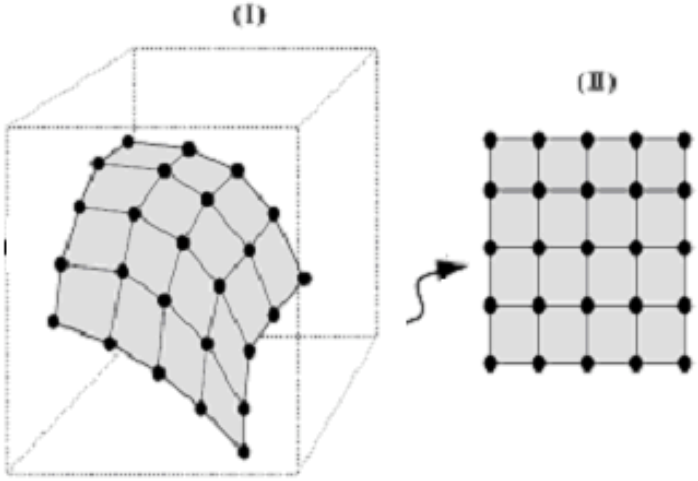
Burada $\alpha(k)$ öğrenme oranı parametresidir ve adaptasyon safhasında azalarak değişen bir fonksiyondur. $\beta(i,c,k)$ ise En İyi Eşleşen Sinir Hücresi olan c sinir hücresinin çevresindeki sinir hücrelerinin yaklaşıp uzaklaşmalarını belirleyen komşuluk fonksiyonudur.

c sinir hücresi eğitim aşamasında bulunurken aşağıdaki formül kullanılır [14]:

$$c = \arg\min \|\lambda(k) - M_i(k)\| \qquad (2.3)$$

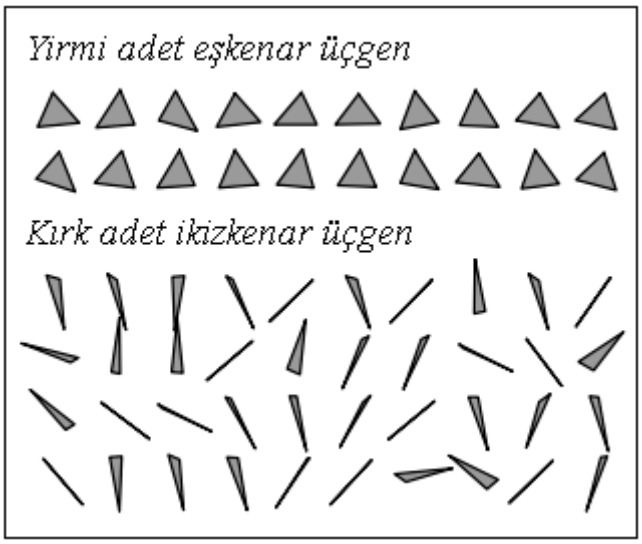
Özdüzenlemeli harita genelde iki katmana sahiptir [16]. Bu tip haritada Giriş katmanı tamamıyla çift boyutlu Kohonen katmanına bağlanmıştır. Kohonen katmanı işlem elemanlarının her biri, gelen giriş değerlerinden onların ağırlıklarının mesafesini ölçmektedir. Birimin ağırlık vektörü ile girdi vektörü arasındaki öklid mesafesi bu durumda aktivasyon fonksiyonu görevi görmektedir.

Şekil 2.3'te görüleceği gibi fiziksel uzayda iki boyutlu bir ızgara yapısı sergileyen özdüzenlemeli ağ, ağırlık/girdi uzayında eğimli bir yapı sergilemektedir. Özdüzenlemeli haritada Yarışmacı Öğrenme (competitive learning) kullanılmaktadır [17].



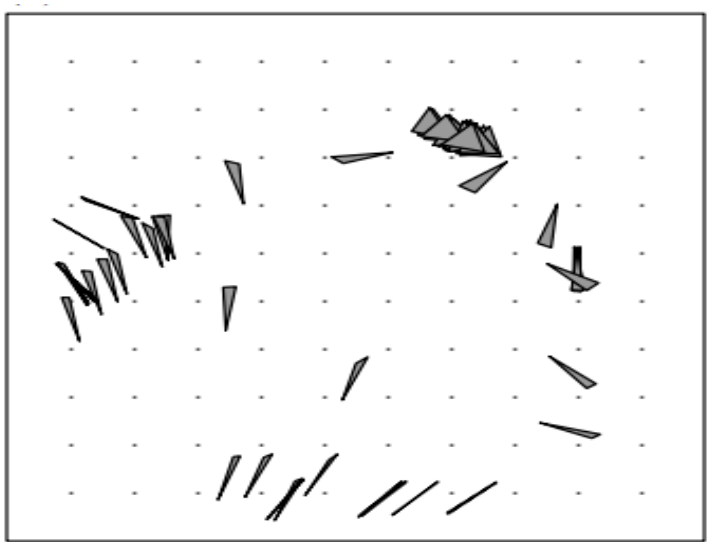
Şekil 2.3: (I) Ağırlık/Girdi uzayında ve **(II)** Fiziksel uzayda Kohonen haritası.

Özdüzenlemeli haritaların etkin bir şekilde kullanılması için sınıflandırma yapılacak veri kümesinde gereken özelliklerin uygun vektörler şeklinde dönüştürülmesi işlemi (Feature Extraction) sınıflandırmanın başarısı için temel bir rol oynar. Girdi vektörleri aslında genelde sınıflandırma yapmak istenen nesnelerin özelliklerini betimlemeyen bir kodlamadır [18]. Örnek olarak, eğer özdüzenlemeli haritaları kullanarak sınıflandıracağımız nesneler üçgenler olsaydı üçgenleri girdi vektörlerini üçgenleri betimleyecek şekilde seçmemiz gerekecekti (Şekil 2.4).



Şekil 2.4. Sınıflandırmak istenen rastgele seçilmiş 60 adet üçgen.

Girdi vektörlerinin seçimine üçgen örneğini kullanarak örnek verecek olursak; kenar uzunlukları, takip eden kenarlar arasındaki açılar ve üçgenlerin alanlarında oluştan 7 haneli bir girdi vektörü oluşturulabilir. Vektörler eğitildikten sonra üçgenlere karşılık gelen vektörlerin harita üzerindeki görüntüsü aşağıdaki şekilde verilmiştir (Şekil 2.5).



Şekil 2.5 Eğitimden sonra üçgenlere karşılık gelen vektörlerin harita üzerindeki görüntüsünün temsil ettikleri üçgenler cinsinden görüntüsü

İlgili şekilde her bir üçgen, eğitimden sonra kendisini betimleyen vektörün harita üzerindeki yerini gösterecek şekilde çizilmiştir [18]. Şekilden de

görülebileceği üzere eşkenar üçgenler harita üzerinde yakın koordinatlara toplanmışlardır.

İkizkenar üçgenlerden alanları büyük olanlar eş kenar üçgenlere yakın bölgelerde toplanmışlardır. Haritada birbirine yakın koordinatlarda sınıflandırılan üçgenlerin harita üzerindeki konumları arasındaki benzerlik dikkat çekicidir.

3. TEMEL BİLEŞEN ANALİZİ

3.1 Tarihçe ve Tanım

Temel Bileşen Analizi (PCA - Principal Components Analysis) 1901 yılında Karl Pearson [19] tarafından bulunmuştur ve günümüzde veri analizi ve tahminsel modelleme yapmak için kullanılan en önemli araçlardan biri haline gelmiştir. Temel Bileşen Analizinin literatürde bilinen bir başka adı Karhunen-Loeve dönüşümüdür [20].

Temel Bileşen Analizi veri faktörleri arasında kovaryans analizi gerçekleştirerek veri boyutunu küçülten matematiksel bir yöntemdir. Bu işlem aracılığıyla incelenen veri kümesindeki modelleri inceleyerek incelenen veri kümesinin benzerlikleri ve farklılıkları bulunabilir.

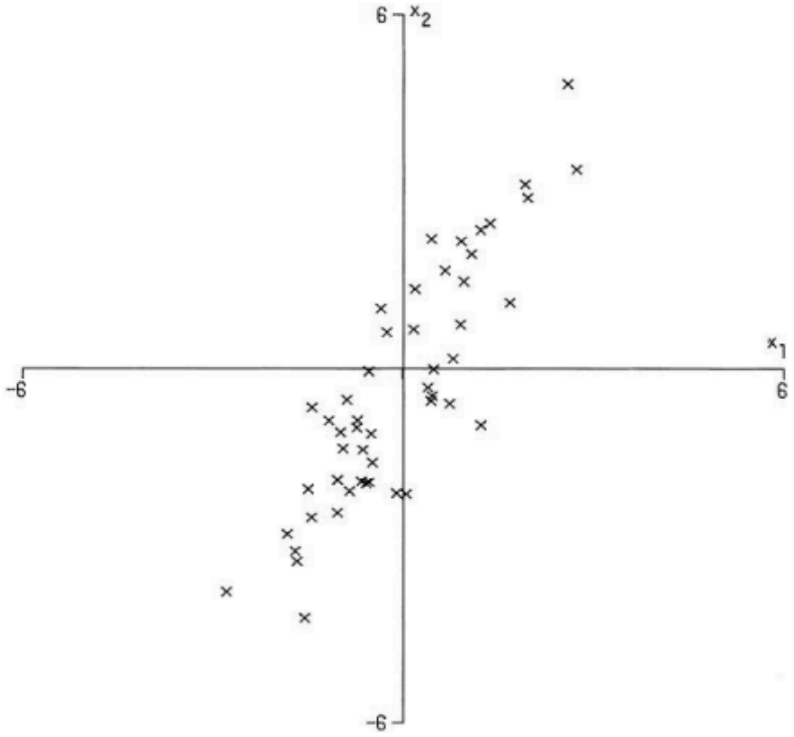
Temel Bileşan Analizinin ortaya çıkmasına neden olan başlıca neden işleme tabi tutulan veri kümesinin istatistiksel yapısına etki eden en önemli etkenin varyans olduğu düşüncesidir. Eğer çok değişkenli bir veri kümesi çok değişkenli bir veri uzayında görselleştirilirse, bu veri kümesininden elde edilen Temel Bileşen Analizi çok boyutlu veri kümesinin daha düşük boyutlarda elde edilmiş bir gölgesi olarak düşünülebilir. [20] En genel anlamda Temel Bileşen analizi çok boyutlu uzayların daha düşük boyutlu uzaylarda temsil etmek için kullanılan bir yöntemdir.

Örnek olarak Şekil 3.1 de verilen grafiksel örnek de 2 boyutlu 50 tane verinin koordinat sistemi üzerindeki görüntüsüne bakılırsa, x_1 ve x_2 değişkenlerinin yüksek oranda ilişkili olduğunu söyleyebiliriz. Fakat şekil dikkatli incelenirse x_2 yönündeki değişimin x_1 yönündekinden çok daha yüksek miktarda olduğu görülebilir. [20]

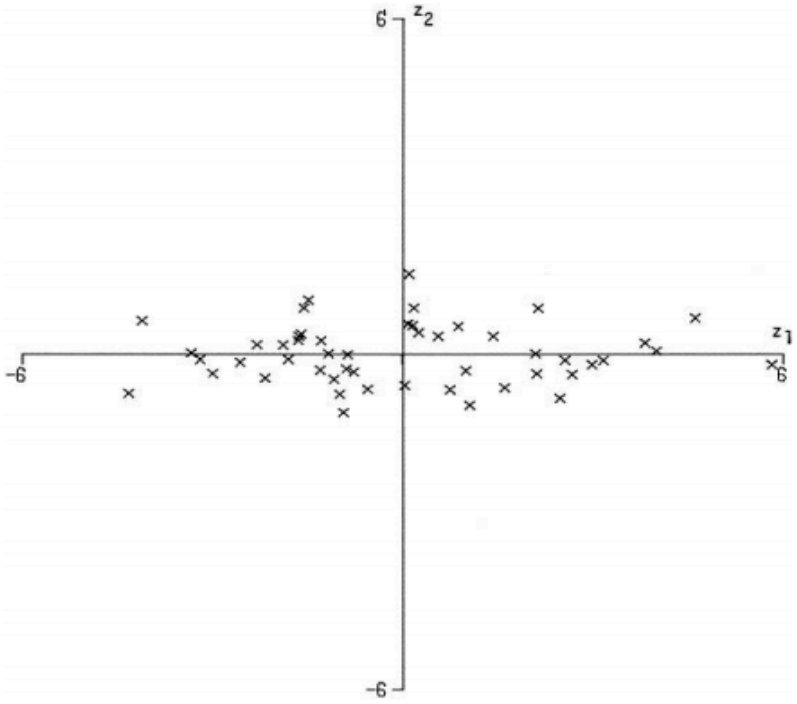
Eğer Şekil 3.1 de verilen veri kümesi üzerinde Temel Bileşen Analizi gerçekleştirilirse, Şekil 3.2 deki bir sonuçla karşılaşırız.

Şekil 3.2 nin de gösterdiği gibi birinci temel bileşen olan z_1 tarafındaki değişim oranı çok daha büyüktür fakat z_2 yönünde de küçük de olsa bir değişim görülmektedir. Daha genel olarak, eğer çok boyutlu veri kümeleri incelenirse ilk

temel bileşenden son temel bileşene doğru gidildiğinde bileşen üzerindeki değişim oranının azaldığı görülür [20]. Başka bir ifadeyle ilk birkaç temel bileşen veri kümesinin nasıl değiştiği hakkında yeterli bilgiyi barındırırlar.

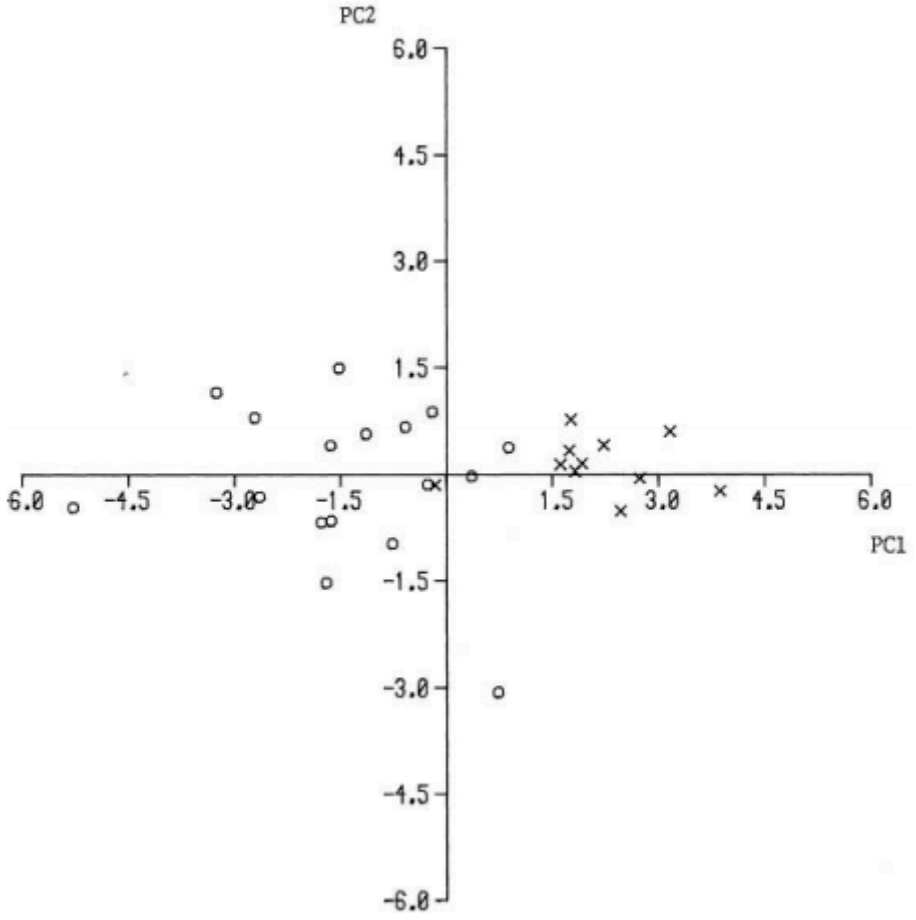


Şekil 3.1 2 Boyutlu 50 adet değişkenin koordinat sistemi üzerinde gösterimi



Şekil 3.2 Şekil 3.1 de gösterilen veri kümesinin temel bileşenler eksenindeki (z_1 ve z_2) gösterimi.

Yukarıda anlatılan örneğin ışığında Şekil 3.1 her bir elemanı 7 değişkenden oluşan bir veri kümesinin ilk iki temel bileşen koordinatına göre bir reprezentasyonunu göstermektedir. Şekil 3.3 e konu olan veri kümesi 11'i bayan 17 si erken olmak üzere 28 öğrenciden alınan 7 anatomik ölçümünü içermektedir. Şekil 3.3 bu veri kümesinin temel bileşenlerinin bir temsilidir. Şekilde de görüldüğü gibi ilk temel bileşen PC1 veri kümesi üzerindeki değişim yaklaşık %80 ini içermektedir.



Şekil 3.3 28 öğrencinin anatomik ölçümlerini içeren veri kümesinin ilk iki temel bileşeni PC1 ve PC2 koordinatlarındaki görüntüsü. Şekildeki **o** erkeklere **x** ise bayanlara karşılık gelmektedir.

Şekilden de görüleceği üzere veri genellikle ilk temel bileşen üzerinde değişmektedir ve bilginin neredeyse tamamını içerisinde barındırmaktadır. Öyleki birinci temel bileşenin 0 dan büyük olup olmadığı incelenerek ilgili veri

kümesinin bir erkeğe mi yoksa bir bayana mı ait olduğu büyük oranda anlaşılabilmektedir [20].

3.2 Temel Bileşen Analizinin Özellikleri

Temel bileşen analizi matematiksel olarak tanımlandığında ortagonal bir doğrusal dönüşüme karşılık gelir. Bu dönüşüm, veri kümesini öyle bir koordinat sistemine yerleştirir ki bu koordinat sisteminde ilk koordinat, veri kümesinde en yüksek değişime sahip olan bileşene karşılık gelir. İkinci en büyük değişim gösteren bileşen ikinci koordinat sistemine yerleştirilir ve bu durum bu şekilde devam eder.

Temel bileşen analizinin boyut düşürme özelliği de buradan gelir. Temel bileşen analizi veri kümesinin özelliklerini bozmadan varyansını en çok değiştiren kısımlarını ve en az değiştiren kısımlarının bulunmasına olanak tanır. İlk sıradaki temel bileşenler varyansı en çok değiştirenler olmakla birlikte diğerleri varyansa çok bir katkıda bulunmazlar. Bu nedenle genellikle ilk sıradaki temel bileşenler verinin en önemli özelliklerini taşırlar.

Temel bileşen analizinin uygulanması için analizin uygulanacağı verinin aşağıda verilen 3 varsayıma uygun olduğu düşünülmelidir.

3.2.1 Doğrusallık varsayımı:

Temel bileşen analizinin uygulanacağı veri kümesinin doğrusal bir şekilde değiştiği varsayılır. Doğrusal olmayan şekilde değişen veri kümeleri için Çekirdek Temel Bileşen Analizi isimli yöntemin uygulanması daha uygundur [20].

3.2.2 Ortalama ve Varyansın istatistiksel önemi varsayımı:

Temel bileşen analizi verinin Gaussian bir şekilde dağıldığını kabul eder. Bu şekilde dağılmayan veri kümeleri için en çok değişim gösteren, en çok varyansa sahip özvektörlerin veri kümesini her zaman en doğru bir şekilde betimleyeceğinin bir garantisi yoktur [20].

3.2.3 Yüksek varyansların önemli olduğu varsayımı:

Temel bileşen analizi basitçe bir koordinat döndürme işlemidir. Bu dönüşümde veri en çok varyansa sahip koordinatların olduğu düzleme oturtulur. Verinin sinyal/gürültü oranı yüksek olduğu durumlarda temel bileşen analizi sinyalin gürültüden ayrılmasını sağlar.

3.3 Temel Bileşen Analizinin Kovaryans Yöntemi ile Hesaplanması

Aşağıdaki adımlar Temel Bileşen Analizinin kovaryans yöntemi ile hesaplanmasını gösterir. Burada amaç **M** boyutlu bir **X** veri kümesinin (**L<M**) olmak üzere **L** boyutlu bir **Y** veri kümesine temel bileşen analizi ile dönüştürülmesini sağlamaktır. Hesaplama işlemi aşağıda verilen 10 adımda gerçekleştirilir.

1. **Adım (Veri Kümesini Düzenleme):**
   $\mathbf{X_1}$ $\mathbf{X_2}$ ..... $\mathbf{X_N}$ olmak üzere **M** boyutlu elemanlardan oluşan **N** tane sütun vektöründen oluşan bir veri kümesi olsun.
   Bu sütun vektörlerini **MxN** boyutlu bir **X** matrisi içerisine yerleştirilir.
2. **Adım (Deneysel Ortalamanın Hesaplanması):**
   Her bir boyutun deneysel ortalamasını Formül 3.1 de gösterildiği şekilde bulunur ve hesaplanan ortalamalar **m x 1** boyutlu **u** matrisinin içerisine yerleştirilir.

$$u[m] = \frac{1}{N}\sum_{n=1}^{N} X[m,n] \tag{3.1}$$

3. **Adım (Ortalamadan Sapmanın Hesaplanması):**
   Bulunan deneysel ortalama vektörü **(u)**, **X** matrisinin her bir sütunundan çıkarılır. Girdi vektöründen ortalamanın çıkartılmış hali **B** matrisinde tutulur.

$$B = X - uh \tag{3.2}$$

$$h[n] = 1; \quad n = 1...N \tag{3.3}$$

4. **Adım(Kovaryans Matrisinin Hesaplanması):**

   B matrisinin kendisiyle dış çarpımı hesaplanarak **MxM** boyutundaki kovaryans matrisi **C** bulunur ki bu aynı zamanda **B** matrisinin beklenen değeridir:

   $$C = E[B \cdot B^*] = (1/N)B.B^* \quad (3.4)$$

   Burada **E** operatörü beklenen değere ***** operatörü ise kompleks eşlenik transpoz operatörüne karşılık gelir. Eğer **B** matrisi reel sayılardan oluşuyorsa bu ***** işlemi normal transpoz işlemine karşılık gelir.

5. **Adım(Kovaryans matrisinin öz değer ve öz vektörlerinin bulunması):**

   C matrisini köşegenleştiren öz vektör matrisi **V** aşağdaki denklemi sağlayacak şekilde bulunur:

   $$V^{-1}CV = D \quad (3.5)$$

   3.5 denklemindeki **C** matrisi **C'**nin özdeğerlerinden oluşan bir köşegen matristir. Bu işlemden sonra **D** matrisinin köşegenlerindeki elemanlar **C** matrisinin öz değerlerine karşılık gelirler.

   $$D[p,q]= \Lambda_m ; p=q=m \quad (3.6)$$

   D matrisinin köşegenleri dışındaki diğer elemanlarını değeri sıfırdır.

   $$D[p,q]=0 \quad (3.7)$$

   Bu işlem sonrasında öz değerler sıralanmış ve eşleşmiş olur. Yani **m.** özdeğer **m.** özvektöre karşılık gelir.

6. **Adım (Öz değer ve Öz vektörlerin yeniden düzenlenmesi):**

   Özvektör matrisi olan **V'**nin öz değer matrisi olan D deki özdeğerlere göre en büyükten en küçüğe göre sıralanır.

7. **Adım(Her bir özvektör için kümülatif enerjinin hesaplanması):**

   m. özvektör için kümülatif enerjinin hesaplanması için aşağıdaki formül kullanılır:

$$g[m] = \sum_{q=1}^{m} D[p,q] \;;\; p = q \;\text{ve}\; m = 1...M \qquad (3.8)$$

8. **Adım (Öz değerlerin L elemanlı alt kümesinin baz vektörleri olarak seçilmesi):**

   V matrisinin ilk **L** sütununun seçilerek **MxL** boyutunda bir **W** matrisinin aşağıdaki gibi oluşturulur:

   $$W[p,q] = V[p,q] \;;\; p = 1....M,\; q = 1.....L \qquad (3.9)$$

   Burada **g** vektörü **L** değeri için optimum değeri seçmek için kullanılabilir. Buradaki amaç seçtiğimiz **L** değeri için bulduğumuz toplamsal enerjinin, bütün toplamsal enerjinin %90 ından fazlasını vermesini sağlamak olmalıdır. Bunu sağlayan minimum **L** değeri boyut düşürme işlemi için optimum değeri sağlar.

   $$G[m = L] > 90\% \qquad (3.10)$$

9. **Adım (Kaynak verilerinin z-skorlarının hesaplanması):**

   Bu adımda **C** matrisinin köşegenlerindeki herbir elemanın karekökleri bulunur ve **Mx1** boyutunda standart sapma vektörü oluşturulur. Bu vektöre **s** adı verilirse hesaplanışı aşağıdaki gibi olur:

   $$\mathbf{s} = \{s[m]\} = \sqrt{C[p,q]} \;;\; p = q = m = 1 \ldots M \qquad (3.11)$$

   s kullanılarak MxN boyutundaki z-skorlarının hesaplanması aşağıdaki formülde verilmiştir:

   $$Z = B/(s.h) \qquad (3.12)$$

10. **Adım (Z-skorlarının yeni tabana oturtularak Temel Bileşenlerin Bulunması):**

    İz-düşümü bulunan vektörler aşağıdaki formüldeki gibi bulunurlar formülde verilen KLT işlemi **X** vektörünün **Karhunen-Loeve** dönüşümüne karşılık gelir.

    $$Y = W^* \cdot Z = KLT\{X\} \qquad (3.13)$$

    Karhunen-Loeve dönüşümü sonucunda oluşan matris, **MxN** boyutlu **X** matrisinin boyutlu temel bileşenlerinden oluşan **LxN**

boyutlu temel bileşen matrisini **(Y)** verir. Y matrisinin satırları X girdi matrisinin temel bileşenleridir.

3.4 Temel Bileşen Analizinin Uygulamaları

Veri boyutu büyüdükçe incelenen veri kümelerindeki benzerliklerin grafiksel olarak ifade edilmesi oldukça zordur. Bu nedenle Temel Bileşen Analizi çok yüksek boyuttaki verileri analiz etemek için oldukça uygun bir araçtır [21].

Temel Bileşen Analizi çok boyutlu verilerin incelenmesini gerektiren problemleri çözmek için çok elverişli bir yol olduğu için ekonomiden biyolojiye bilgisayar mühendisliğinden meteorolojiye çok sayıda uygulaması vardır. Temel Bileşen Analizinin en çok kullanıldığı alanlardan birkaçı aşağıdaki gibi listelenebilir:

- Görüntü sıkıştırma [26].
- Gen haritalarının incelenmesi [27].
- Biyoloji ve Çevre Mühendisliği [27].
- Deniz Bilimi [30].
- Meteoroloji [28].
- Örüntü Tanıma [29].
- Finansal Tahmin Üretme [30].

4. ÖZDÜZENLEMELİ AĞLAR KULLANILARAK AĞ TRAFİĞİNİN SINIFLANDIRILMASI

Çalışmanın bu bölümünde ağ trafiğinin davranışının belirlenebilmesi için Özdüzenlemeli Ağlar kullanılarak bir sınıflandırma yöntemi ortaya konulmuştur. Bilinen belli başlı ağ trafiği anomalileri dosya indirme, port tarama ve ağa yöneltilen Servis Dışı Bırakma (DOS – Denial of Service) saldırılarıdır [14]. Ağ trafiğinin ölçümü için çalışmanın bu aşamasında Basit Network Yönetim Protokolü (SNMP - Simple Network Management Protocol) kullanılmıştır.

Bu çalışmada dosya indirme, port tarama ve normal network trafiğinin birbirinden ayırt edilebilmesi için Özdüzenlemeli Ağ bazlı bir sınıflandırıcı tasarlanmıştır.Elde edilen sonuçlara bir iyileştirme olarak eğitilen özdüzenlemeli ağ üzerinde birbirini takip eden zamana bağlı trafik vektörlerinin izledikleri yolların bulunması ve nerede yoğunlaştıklarının belirlenmesi ile sistemin karar verme başarısı artırılmıştır.

4.1 Ağ Trafiğinin Ölçümü

Bu çalışmada ağ trafiğinin ölçümü için bir test ortamı oluşturulmuştur. Kampüs ağı içerisindeki bir anahtara bir bilgisayar bağlanmış ve bu bilgisayarın üzerinden dosya indirme, port tarama ve saldırı gibi çeşitli işlemler gerçekleştirilmiştir. Bu işlemler gerçekleştirilirken bilgisayarın bağlı olduğu ağ anahtarının ilgili portu yazılan bir bilgisayar programı ile başka bir bilgisayardan dinlenmiş ve bu port üzerinden geçen ağ trafiği ölçülmüştür. Ağ trafiğinin ölçümü için SNMP protokolü kullanılmış ve ilgili ağ anahtarının üzerinde gerekli SNMP izinleri ağ trafiğini dinleyen programın yüklü olduğu programın kullanabileceği şekilde ayarlanmıştır.

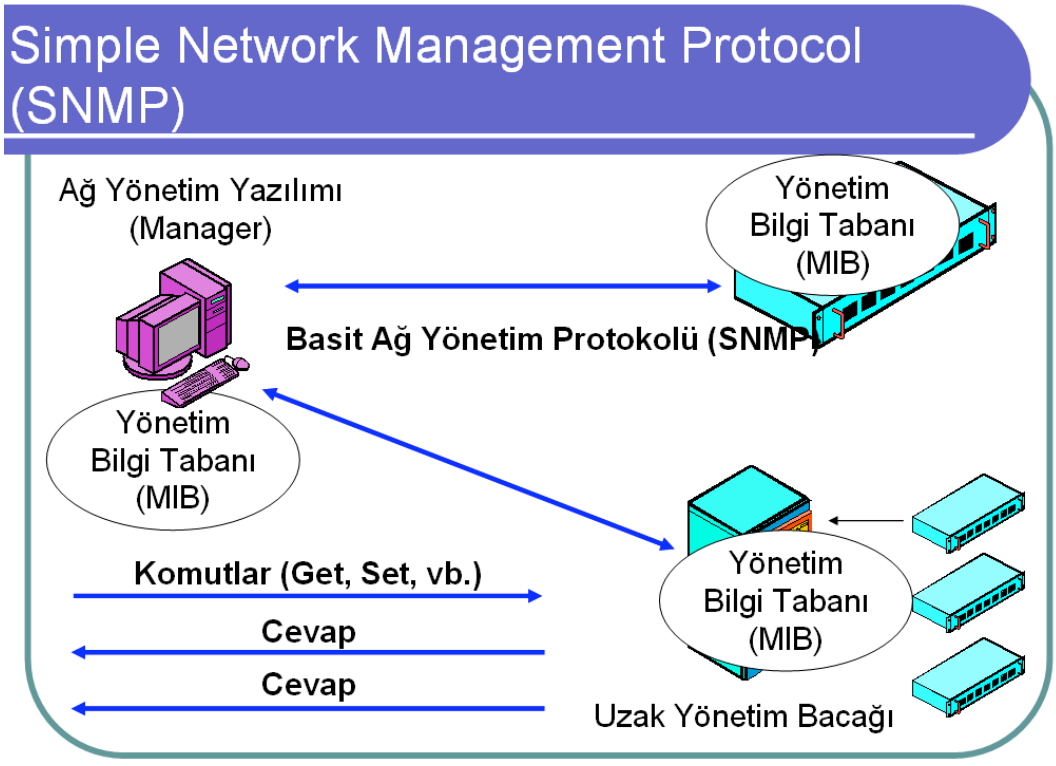
4.1.1 Basit Ağ Yönetim Protokolü (SNMP)

Basit Ağ Yönetim Protokolü ağ üzerindeki cihazları denetlemek ve bunlardan bilgi toplamak için kullanılan, hemen hemen bütün ağ cihazlarının

desteklediği çok güçlü bir protokoldür. Günümüzde ağ araçlarını yönetmek ve için en çok kullanılan protokoldür [22].

Basit Ağ Yönetim Protokolü 1998 de İnternet Aktiviteleri Forumu (IAB – Internet Activities Board) tarafından tasarlanmış ve çok kısa bir sürede TCP/IP ağların üzerinde çalışan ağ anahtarlarının, ağ köprülerinin ve yönlendiricileri yönetmek için geliştirilmiş ve günümüzdeki kullanım şeklini almıştır.

Bu protokolün çalışma şekli sunucu/istemci yapısı ile çalışır. SNMP'nin üzerinde çalıştığı protokol UDP (User Datagram Protocol) protokolüdür. Bu protokolde sunucu tarafı bütün bilgileri toplayan taraftır. Ağ cihazı üreticilerinin hemen hemen hepsi ürettikleri cihazları SNMP protokolüne uygun halde üretmektedirler ve bu SNMP sunucuları bu cihazlar üzerinde bulunan ve üreticiler tarafından tasarlanmış küçük programlardır.



Şekil 4.1 SNMP Protokolünün Genel Çalışma Yapısı

Eğer ağa yeni eklenmiş bir cihaz üzerinde SNMP sunucusu varsa Ağ Yönetim Sistemleri (NMS- Network Managemenet System) otomatik olarak yeni eklenen bu bileşeni izlemeye ve bilgi toplamaya başlarlar. Bu nedenle üzerinde

SNMP ajanı bulunduran cihazlar aynı zamanda yönetilebilir cihazlar olarak da adlandırılırlar [23].

1988 de tasarlandığı haldeki SNMP protokolüne SNMP versiyon 1 (SNMP v1) adı verilmiştir. SNMP v1 beş temel mesaj tipine sahiptir ve mesajların maksimum boyutları kısıtlıdır. Bu eksiklikeri gidermek için 1993 SNMP versiyon iki ve 1998 yılında SNMP versiyon 3 çıkartılmıştır. İkinci versiyonda cihazlardan veri toplamak için çeşitli yetkilendirme mekanizmaları eklenmiş, versiyon 3 ile birlikte yetkisiz kişilerin ağ cihazlarından bilgi toplamalarını önlemek amacıyla sertifika bazlı kimlik tanımlama yöntemine geçiş yapılmıştır [23]. SNMP protokolünün ağ cihazlarını tanımayı kolaylaştırması ve ağ cihazlarını uzaktan kolayca ayarlamayı sağlaması yaygınlığını artıran ve yaygınlaşmasını sağlayan en önemli etmen olarak söylenebilir [24].

SNMP protokolünün çalışma şekli şöyledir: Cihaz üreticileri ürettikleri cihazlara özel Yönetim Bilgi Tabanları (MIB - Management Information Base) yayınlarlar. Her cihazın kendine özel bir yönetim bilgi tabanı vardır ve bu bilgi tabanlarında cihazın dışarıya sağlayacağı bigiler tutulur.

Yönetim bilgi tabanlarında her bir bilgi tipinin kendine özel bir kodlaması vardır ve bu kodlamaya Nesne Kimlik Bilgisi (OID – Object Identifier) adı verilir. Cihaz hakkında bilgi toplamak isteyen kişi veya yazılım bu yönetim bilgi tabanlarını inceleyerek cihaz hakkında öğrenmek istediği bilginin tipinin nesne kimlik bilgisini öğrenir.

Daha sonra bu nesne kimlik bilgisi kullanılarak cihaza bir GET mesajı gönderilir. Cihaz ilgili bilgiyi içeren bir paket oluşturur ve bu paketi kullanıcıya UDP protokolünü kullanarak gönderir. Bu pakete Protokol Datagram Birimi (PDU – Protocol Datagram Unit) adı verilir. Protokol Datagram Birimi SNMP protokolün anladığı yegane paket türüdür.

İstemci kendisine cihaz üzerindeki sunucu tarafından gönderilen bilgiyi alır ve kullanır. İstemci SNMP protokolünü yalnızca cihazdan bilgi toplamak için değil aynı zamanda cihazı konfigure etmek için de kullanabilir. Cihazı konfigüre edebilmek için ilgili Nesne Kimlik Bilgisi bulunur ve konfigurasyon bilgisini içeren bir SET mesajı gönderilir. Cihaz üzerindeki sunucu bu SET mesajını ve içeriğini algılayarak cihazın ilgili konfigürasyonunu değiştirebilir.

4.1.2 SNMP Aracılığıyla Ağ Trafik Bilgisinin Toplanması

SNMP Protokolünde her hangi bir cihazın ilgili port'undan geçen trafiğin belirlenebilmesi için Yönetim Bilgi Tabanında iki adet nesne tanımlıdır. Bunlardan birincisine **IfInOctets** ikincisine ise **IfOutOctets** adı verilir [25].

IfInOctets nesnesine karşılık gelen nesne adresi (Object ID: 1.3.6.1.2.1.2.2.1.10)dir ve bu nesne ağ cihazının bir portuna giren paketlerin boyutlarının toplamlarını byte cinsinden tutar.

IfOutOctets nesnesine karşılık gelen nesne adresi (Object ID: 1.3.6.1.2.1.2.2.1.16)dır ve bu nesne ağ cihazının bir portundan çıkan paketlerin boyutlarının toplamlarını byte cinsinden tutar.

Cihaz üzerinden belirli bir zaman aralığında geçen trafik verisi aşağıdaki formül kullanılarak hesaplanabilmektedir [25]:

$$\frac{\Delta(T) = (\Delta(ifInOctets) + \Delta(ifOutOctets))}{\Delta(t)} \tag{4.1}$$

İlgili ağ verisinin toplanması ve trafiğin ölçülmesi için MYSNMP isimli bir yazılım geliştirilmiştir. Bu yazılımın genel özellikleri 4.2 nolu başlıkta verilmiştir.

4.2 Veri'nin Toplanması

Özdüzenlemeli ağa girdi olarak sağlanacak verinin toplaması işlemi SNMP protokolü konuşabilen bir ağ anahtarından trafik değerlerinin sorgulanmasına dayanır. Cihaz üzerinden geçen trafik oranının zaman içerisindeki değişimi bu çalışmada toplanan verinin temel kaynağıdır.

Bir ağ anahtarından veri toplayabilmek için ya cihazı hazır yazılımlar aracılığıyla elle sorgulamak ya da SNMP protokolünü kullanarak ihtiyaç duyulan veriyi otomatik olarak toplayan bir yazılım geliştirmek gerekmektedir. Bu nedenle

bu çalışmada network trafiğini toplamak ve incelemek için MYSNMP isimli bir yazılım geliştirildi.

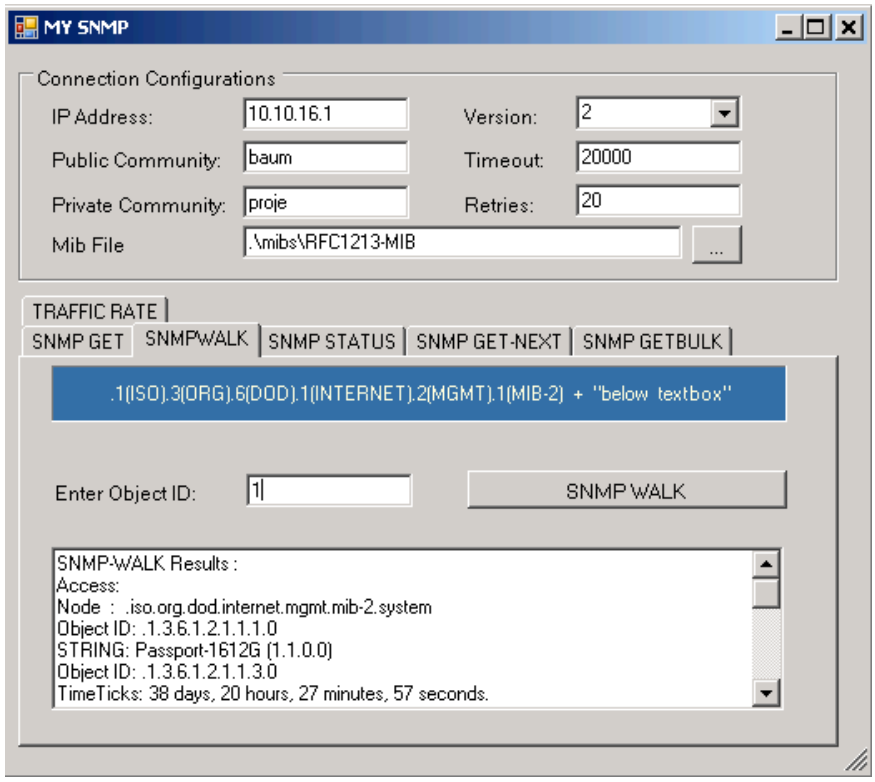
MYSNMP temelde bir Yönetim Bilgi Tabanı (MIB) tarayıcısı olarak tasarlanmıştır. Bu programa üzerinden geçen trafik konusunda bilgi toplanmak istenen cihazın IP adresi kullanıcı adı ve şifre bilgileri girilir ve cihazla bağlantı kurulur. Bu aşamadan sonra yazılım çekilmek istenen bilginin Nesne Kimlik Bilgisini (OID) girerek cihaz üzerinde temel SNMP işlemleri yapmaya olanak tanımaktadır.

Temel SNMP işlemleri şu şekilde tanımlanabilir:

- SNMP-Get: Belirli bir nesne kimlik bilgisi gönderilerek o kimlik bilgisine karşılık gelen değerin cihazdan çekilmesi işlemi.
- SNMP-GetNext: Belirli bir nesne kimlik bilgisi gönderilerek o kimlik bilgisinden sonra gelen nesne kimlik bilgisine karşılık gelen değerin cihazdan çekilmesi işlemi
- SNMP-Status: Belirli bir cihazın durumu hakkında özet bilgilerin çekilmesi işlemi.
- SNMP-Walk: Belirli bir nesne kimlik bilgisi üst grubunun altındaki bütün OID'lerin bilgisinin tek bir istek yapılarak cihazdan çekilmesi işlemi.

Şekil 4.2 de MYSNMP programının SNMP-Walk işlemine verdiği cevap görülebilir.

MYSNMP programı Temel SNMP işlemlerine ek olarak bir ağ cihazı (yönlendirici, ağ anahtarı, ağ köprüsü) üzerinde bulunan herhangi bir porttan geçen trafik oranını hesaplayıp günlük dosyalarına yazan bir modülü daha bulunmaktadır. Program trafik ölçümünü formül 4.1 de verilene göre yapmaktadır ve bu trafik verisi özdüzenlemeli ağlar için oluşturulan girdinin kaynağını olşuturmaktadır.



Şekil 4.2 MYSNMP Programı'nın bir ekran görüntüsü.

4.3 Toplanan veriden girdi vektörlerinin oluşturulması

Ölçüm işlemi tamamlandıktan sonar toplanan vektörlerden ilk N tanesi birinci vector olarak kabul edilir. Bu vektöre **V1** adını verelim. Toplanan trafik değerlerinden 2 den N+1 tanesine kadar olanı 2. vektör olan **V2**'yi oluştururlar. Diğer vektörler de bu algoritmayı kullanarak oluşturulurlar. Trafik oranlarının ölçümü her 20 saniyede bir geçekleştirilmektedir. Dolayısıyla **V1** vektörü ağın **1** ile **20N** saniye arasındaki trafik davranışını vektörel olarak gösterimidir. Benzer olarak V2 vektörü ağın trafik davranışının **20** ile **20(N+1)** saniye arasındaki davranışına karşılık gelir. Bu işlem bütün veri kümesi için tekrarlanır.

Deneysel sonuçlar eğer pencere boyutuna karşılık gelen **N** değeri 10 ile 30 arasında seçilmesinin uygun olacağını göstermiştir. Fakat pencere boyutu değeri bu değerlerin dışarısında seçilirse girdi kalitesinin düştüğünü göstermiştir.

Bu performans düşüşünün olası nedeni port tarama ve dosya indirme işlemeleri sırasında üretilen ağ trafiğinin gerçekleşme periyodunun uzunluğundan kaynaklanmaktadır. Genelde 400 saniye herhangi bir ağ trafiği davranışının

başlayıp bitmesi için yeterli bir süredir. Bu nedenle bu çalışmada zaman penceresi boyutu değeri **N**=20 olarak seçilmiştir. Bu durumda oluşturulan her bir vektör **400** saniyelik zaman dilimleri olarak düşünülebilirler. Çizelge 4.1 de N = 5 değeri için vektörlerin seçimi görülebilir.

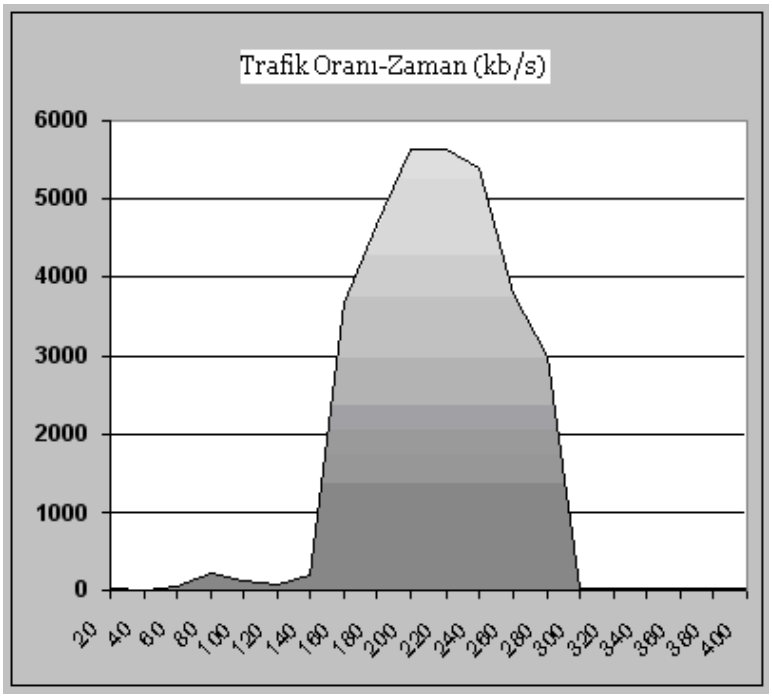
Çizelge 4. 1 Zaman penceresi değeri N = 5 iken oluşturulan ilk 5 vektörün gösterimi

V1	***V2***	***V3***	***V4***	***V5***
15.2	14.6	25	120	135
14.6	25	120	135	10.4
25	120	135	10.4	17
120	135	10.4	17	19
135	10.4	17	19	15,5

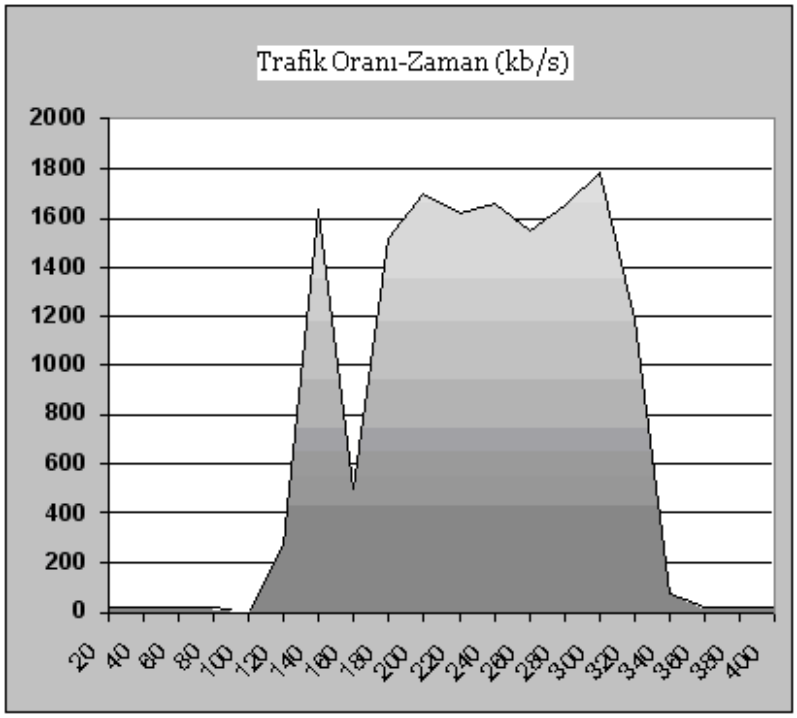
Sınıflandırma sonuçlarını gözlemleyebilmek için özdüzenlemeli haritaya girdi olarak sunulan her bir vektörün etiketlendirilmesi gerekmektedir. Etiketleme işlemi için MYSNMP programı ölçüm yapılan bilgisayarın trafiğini ölçerken, ölçüm yapılan bilgisayarda üretilen trafiğin türü(dosya indirme, port tarama) ile başlangıç ve bitiş zamanları kaydedilmiştir. Dosya indirme ve port tarama dışındaki bütün trafik normal network trafiği olarak kabul edilmiş ve sınıflandırmada karşılık gelen vektörler bu şekilde etiketlenmiştir.

Böylece özdüzenlemeli haritalara girdi olarak sunacağımız vektörleri zaman ekseninde kaydırarak, özdüzenlemeli haritanın zamana göre trafik değişimini sınıflandırmasını sağlayacak bir girdi vektör kümesi oluşturuldu. Bu çalışmanın ana fikri özdüzenlemeli haritaları bu vektörlerle eğittikten sonra harita üzerinde çeşitli stabil bölgeler oluşmasını beklenmektedir ve herbir stabil bölgenin belli bir ağ trafik tipine karşılık geleceği öngörülmüştür.

Ağ trafik çeşitlerini şekiller üzerinde gösterecek olursak Şekil 4.3 bir dosya indirme işlemine karşılık gelmektedir.



Şekil 4.3 Dosya indirme sırasında oluşan ağ trafiğinin zamana göre değişimi



Şekil 4.4 Port tarama sırasında oluşan ağ trafiğinin zamana göre değişimi

.

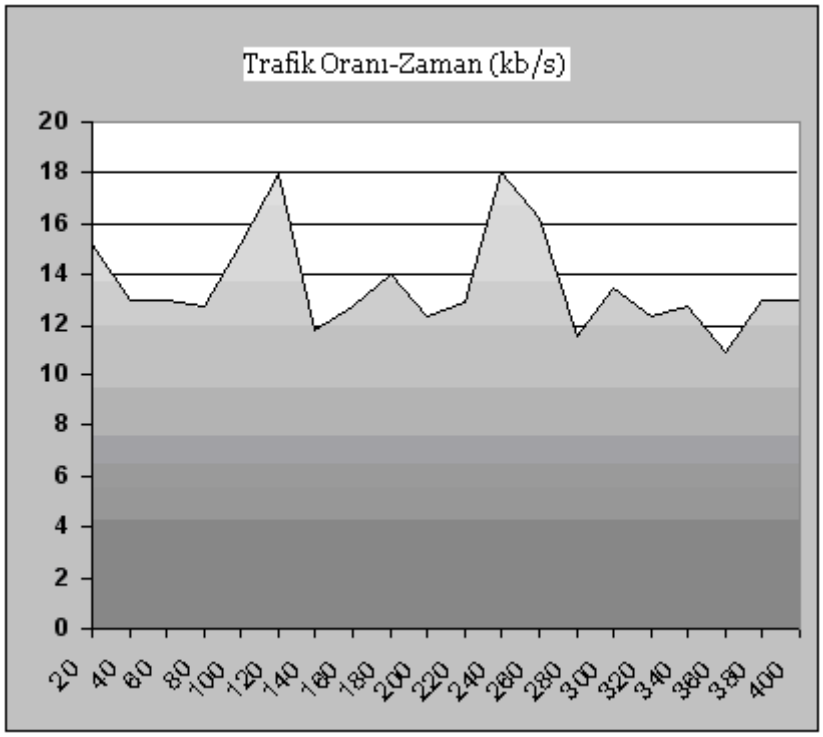
Şekilden de görüldüğü gibi dosya indirmenin başladığı sırada ağ trafiği ani bir biçimde artış göstermiş ve dosya indirme işlemi bittiğinde yine ani bir şekilde

azalış göstermiştir. Şekil 4.4 ise ölçüm yapılan ağ anahtarının bağlı olduğu cihaza yönelik gerçekleştirilen tipik bir port tarama işleminin bir gösterimidir.

Hemen hemen bütün port tarama işlemlerinde Şekil 4.4'e benzer bir değişim ve dalgalanma görülmüştür. Dosya indirme ve port tarama işlemleri grafikler üzerinde karşılaştırılacak olursa bu iki işlemi birbirinden ayırt eden temel etmenler grafiklerin tepe noktalarındaki trafik değerleri ve grafiklerin şekillerinin farklılıklarıdır.

Dosya indirme sırasında trafik oranı port taramayla karşılaştırıldığında oldukça yüksektir. Fakat port tarama işleminde dosya indirme işlemine oranla trafik değerlerinde daha fazla dalgalanma olmaktadır.

Şekil 4.5 bu dosya indirme ve port tarama işlemleri dışında dinlenilen cihaz üzerindeki normal ağ trafiğinin bir gösterimini içermektedir. Bu durumda cihaz üzerinden geçen trafiğin durumu port tarama ve dosya indirme işlemleri ile karşılaştırıldığında göreceli olarak daha düşüktür.

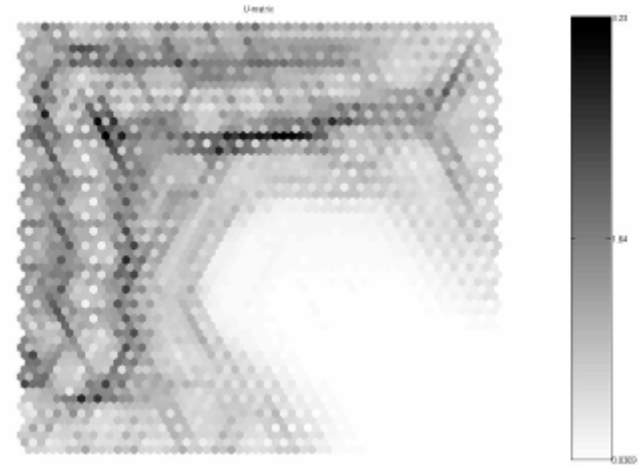


Şekil 4.5 Port tarama ve dosya indirme dışındaki ağ trafiğinin zamana göre değişimi

4.4 Özdüzenlemeli ağın eğitilmesi.

Bu çalışmada sınıflandırma işleminin yapılabilmesi için 30x30 sinir hücresinden oluşan bir özdüzenlemeli harita oluşturuldu. Harita eğitilirken yığın eğitim algoritması kullanıldı. Eğitim uzunluğu 100 basamak olarak seçildi ve başlangıç yarıçapı 15 olarak belirlendi.

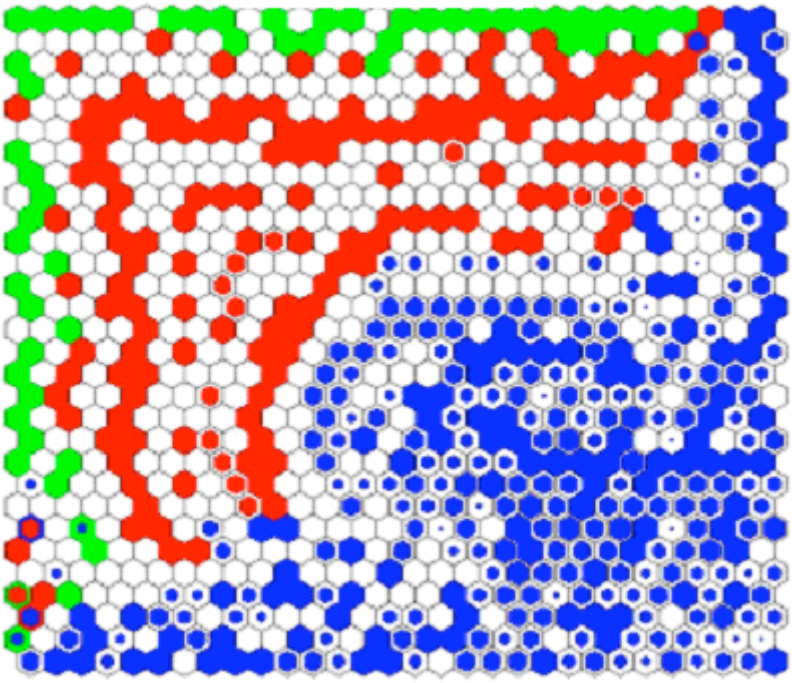
Eğitim işlemi tamamlandıktan sonra özdüzenlemeli harita ile port tarama ve dosya indirme işlemlerinin normal network trafiğinden ayrılması konusunda oldukça başarılı sonuçlar elde edildi. Eğitim işlemi için MATLAB programı üzerinde çalışan SOM-TOOLBOX isimli kütüphane kullanıldı. Bu kütüphane kullanılarak elde edilen U-matrisi aşağıdaki Şekil 4.6 da görülebilir.



Şekil 4.6 Eğitim sonucunda elde edile U – matrisi

4.5 Sonuçlar

Şekil 4.7 de 30x30 luk bir haritanın verinin tipine göre renklendirilmiş bir gösterimi bulunmaktadır:

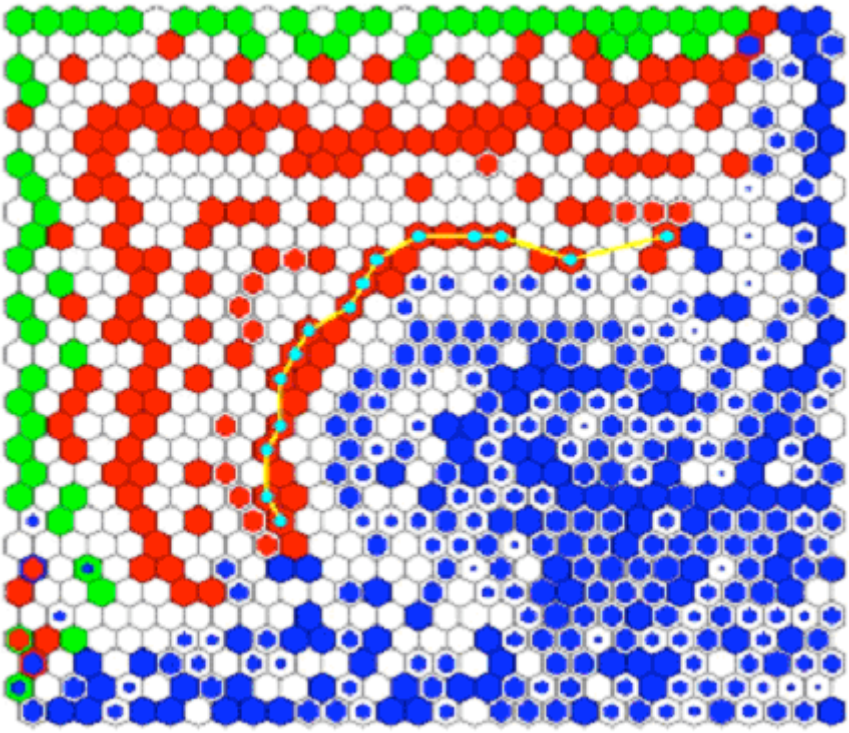


Şekil 4.7 Renklendirilmiş harita. Şekildeki yeşil kısımlar port taramaya, kırmızı kısımlar dosya indirmeye ve mavi kısımlar ise normal ağ trafiğine karşılık gelmektedir.

Şekil 4.7 de port tarama işlemine karşılık gelen vektörler yeşil olarak renklendirilmiş, aynı şekilde dosya indirme işlemine karşılık gelen işlemler de kırmızı ile renklendirilmiştir. Diğer vektörler normal ağ trafiği olarak kabul edilmiş ve mavi ile renklendirilmişlerdir.

Dikkat edilirse haritanın bazı düğümlerinde birden fazla renk bulunmaktadır. Yani bir düğüm üzerinde hem kırmızı hem mavi ya da hem yeşil hem de kırmızı renklendirmeler göze çarpmaktadır. Bunun nedeni bazı dosya indirme, port tarama ve normal ağ trafiği vektörlerinin diğer sınıftan vektörlerle istatistiksel olarak benzerlik içermeleridir.

Bu durum, harita üzerinde zaman içerisinde birbirini takip eden vektörlerin izlediği yolun takip edilmesi ile çözülebilir. Şekil 4.8'de bu durum ele alınmıştır.



Şekil 4.8 Dosya indirme sırasında takip eden vektörlerin harita üzerinde izledği yol. Takip eden vektörlerin hangi sınıf üzerinde yoğunlaştığı gözlemlenerek karar verme mekanizmasında iyileştirme sağlanabilir.

Sonuç olarak anomali içeren ağ trafiği sadece trafikteki değişim incelenerek özdüzenlemeli bir harita yardımıyla tespit edilebilir. Şekil 4.7 ve 4.8 den görüleceği gibi sonuçlar oldukça tatmin edicidir. Bu sistemin gerçek zamanlı uygulamalar için kenar anahtarlarını dinleyen ağ yönetim yazılımlarında kullanılması oldukça uygundur. Sınıflandırılacak uygulamalar olarak port tarama ve dosya indirme işlemi dışında ağ üzerinde gerçekleştirilen çok sayıda başka uygulama vardır. Sınıflandırılacak uygulama sayısı arttıkça bunların trafiksel davranışları incelemek için daha büyük boyutlu haritalar kullanılabilir.

Sisteme girdi olarak sağlanan vektörler trafik değerlerin zaman içerisinde değişimini temsil etmektedir. Yani vektörlerden herbiri kendisinden önce gelen vektörden sonraki zaman diliminin bilgisini içermektedir. Eğer harita eğitildikten sonra sisteme girdi olarak verilecek test vektörlerinin harita üzerinde hangi konuma düştükleri belirlenebilirse, belirli bir zaman diliminde oluşan trafiğin

hangi tipde olduğu daha kolay bir şekilde bulunabilir. Sonuç olarak birbirini takip eden vektörlerin harita üzerinde izlediği yolların incelenmesi sistemin kararlılığını artırmaktadır.

5. TEMEL BİLEŞEN ANALİZİNİN SINIFLANDIRMAYA ETKİSİ VE KDD CUP 1999 VERİ KÜMESİNİN ÖZDÜZENLEMELİ HARİTALAR YARDIMIYLA SINIFLANDIRILMASI

5.1 Giriş

Çalışmanın ikinci kısmında ise akademik araştırmalarda geçerliliği olan bir test veri kümesi olan Bilgi Keşfi ve Veri Madenciliği (KDD - Knowledge Discovery ve Data Mining Tools Conference) yarışmasının [10] 1999 yılında yapılan versiyonundaki veri kümesi kullanılarak bu veri kümesinin sınıflandırılması yapılmıştır.

Bu yarışmada yarışmacılardan Bilgisayarlı Hesaplama yöntemlerini kullanarak DARPA (Defense Advanced Research Projects Agency) tarafından oluşturulmuş bir test ortamında elde edilmiş ağ trafiğinin sınıflandırılması istenmiştir [11]. Yarışmadan sonra bu veri kümesi anomali tespit etme ve saldırı tespit etme ile ilgili akademik çalışmalar için bir referans veri kümesi haline gelmiştir. Bu test kümesinde tezin birinci kısmından farklı olarak elde edilen girdi vektörlerin belirli bir kısmı Temel Bileşen Analizine(PCA – Principle Component Analysis) tabi tutulmuştur.

Çalışmanın ilk kısmından farklı olarak, bu kısımda özdüzenlemeli haritaya girdi olarak temel bileşen analizi sonucunda elde edilen vektörler sunulmuştur. Bu kısımda incelenen temel öge sınıflandırma yapmanın yanı sıra Temel Bileşen Analizinin sınıflandırma üzerindeki etkisidir. Bu yöntemle yapılan sınıflandırma sonucunda normal trafikten saldırı trafiğinin ayrılması konusunda çok başarılı sonuçlar elde edilmiş ve temel bileşen analizinin getirdiği boyut düşürme işlevi sonucunda vektörlerin kendisinin sisteme girdi olarak sunulduğu duruma göre çok daha hızlı eğitim sonuçları alınmıştır.

5.2 KDD Cup 1999 Veri Kümesi

KDD Cup 1999 Veri Kümesi, 1998 yılında DARPA tarafından geliştirilen bir ağ simülasyonunun verilerini içerir. Bu ağ simülasyonunda Amerikan Hava

Kuvvetlerinin Yerel Ağının trafiği dokuz hafta boyunca dinlenmiş ve bu trafik çeşitli saldırılarla beslenmiştir. Dokuz hafta boyunca saldırıların başladığı ve bittiği süreler kaydedilmiş olup trafiğin kendisi de tcpdump [33] günlükleri halinde tutulmuştur.

KDD Cup için bu tcpdump günlükleri incelenerek bir veri kümesi oluşturmuş ve bu veriler üzerinde bir saldırı tespit sistemi geliştirilmesi için bir yarışma açmıştır. KDD Cup veri kümesi iki alt kümeden meydana gelmektedir. Birincisi ilk 7 haftalık eğitim veri kümesidir. Bu 7 haftalık veri kümesi yaklaşık 5 milyon bağlantı kaydı ve 4 gigabyte'lık sıkıştırılmış bir dosya içerisinde yarışmacılara sunulmuştur.

İkinci alt küme son 2 haftalık trafik kayıtlarından meydana gelmiştir. Bu test kümesi de benzer bir şekilde yaklaşık 2 milyon kadar bağlantı kaydı içermektedir. Veri kümesi içerisindeki her bir bağlantı başlangıç ve bitiş zamanları bilinen bir TCP paket grubuna karşılık gelir. Her bir bağlantı saldırı veya normal etiketleriyle etiketlenmiştir ve her bir bağlantı kaydı yaklaşık 4 byte tutmaktadır.

Saldırılar temelde 4 ana kategori içerisinde toplanmıştır:

- DOS saldırıları: syn-flood veya smurf saldırıları gibi,
- R2L (Uzaktan Yerel Ağa olan saldırılar – Remote to local): Uzak bilgisayarlardan yerel ağdaki bilgisayarlara yönelik olarak yapılan saldırılar. Yöneteici şifresini bulmaya çalışmak bu tip saldırılara örnek olarak gösterilebilir.
- U2R (Yerel Ağdaki Kullanıcıların ürettiği saldırılar – User to Root): Yerel ağdaki kullanıcıların ağdaki başka bilgisayarların yönetici şifresini elde etmeye çalışması ya da bufferoverflow saldırıları buna örnek olarak verilebilir.
- Probing (Tanıma): Ağdaki bilgisayarlar hakkında herhangi bir saldırı gerçekleştirmeksizin bilgi toplama işlemi. Port tarama işlemi bunlara örnek olarak verilebilir.

Veri kümesi içerisindeki her bir vektör Çizelge 5.1, Çizelge 5.2 ve Çizelge 5.3 de gösterilen bileşenlerden oluşmaktadır. Çizelge 5.1 temel TCP bağlantısı

bilgisinin bir açıklamasıdır ve burada bağlantı vektörlerinin her birinde bulunan değerlerin tanımlarını gösterilmektedir.

Çizelge 5.1 TCP Bağlantılarının temel özellikleri

Özellik ismi	***Tanım***	***Tip***
Duration	Bağlantının gerçekleştiği süre	sürekli
Protocol_type	Kullanılan protokolün ismi, tcp, udp, vb	kesikli
Service	Alıcı üzerinde çalışan servis ismi, http, telnet vb	kesikli
src_bytes	Hedeften kaynağa gönderilen byte sayısı	sürekli
dst_bytes	Kaynaktan hedefe gönderilen byte sayısı	sürekli
flag	Bağlantıın hatalı veya normal olup olmadığı	kesikli
land	Eğer bağlantı anı sunucuya ve porta yapılıyorsa 1 değilse 0	kesikli
wrong_fragment	Yanlış paketleri sayısı	sürekli
urgent	Bağlantı içindeki acil paketlerin sayısı	sürekli

Birçok DOS saldırısı ve port tarama işleminin aksine R2L ve U2R saldırılarının önemli bir kısmında takip eden bağlantılar olmamaktadır. Bu nedenle DOS saldırıları ve tanımlama işlemleri aynı sunucuya ard arda çok kısa süre aralıklarla bağlantılar içerirken, R2L ve U2R işlemleri sadece tek bir bağlantı kaydı içerisinde gerçekleşmiş olabilir. Çizelge 5.2 bağlantınıya ait daha ayrıntılı özelliklerin bir listesini içermektedir. Bu çizelgede gösterilen terimler her bir bağlantı vektörünün ikinci kısmını oluşturmaktadır. Çizelge 5.3 Bağlantı

vektörünün içerisinde bulunan ve bağlantıya ait trafik özelliklerine ait açıklamaları içermektedir Çizelge 5.4 ise Eğitim veri kümesinde gerçekleştirilen atakların isimlerini ve tiplerini vermektedir.

Çizelge 5.2 Bağlantıya ait detaylı özellikler

Özellik ismi	***Tanım***	***Tip***
Hot	"hot" belirteçlerinin sayısı	sürekli
num_failed_logins	Yanlış login işlemlerin sayısı	sürekli
Logged_in	Eğer başrıyla giriş yapılmış ise 1 diğer durumda 0	kesikli
num_compromised	Gizliliği ihlal edilmiş durumların sayısı	sürekli
root_shell	Eğer yönetici kabuğu ele geçirilmişse 1 diğer durumda 0	kesikli
su_attempted	Eğer ``su root'' komutu denenmişse 1 diğer durumda 0	kesikli
num_root	Yönetici işlemlerine erişim sayısı	sürekli
num_file_creations	Dosya yaratma işlemlerinin sayısı	sürekli
num_shells	Bağlantı süresince kabuk içerisinde gerçekleştirilen komutların sayısı	sürekli
num_access_files	Erişim izinlerini belirleyen dosyalar üzerinde yapılan işlemlerin sayısı	sürekli
num_outbound_cmds	Bir ftp bağlantısındaki outbound komutlarının sayısı	sürekli
is_hot_login	Eğer giriş işlemi "hot" listesinin içerisindeyse 1 diğer durumda 0 değerini alır	kesikli
is_guest_login	Eğer işlemi "misafir" işlemiyse 1.	kesikli

Çizelge 5.3 Bağlantıya ait trafik özellikleri

Özellik ismi	***Tanım***	***Tip***
count	Aynı bilgisayardan iki saniye içerisinde yapılan bağlantıların sayısı	sürekli
Serror_rate	TCP SYN Hatası veren bağlantıların yüzdesi	sürekli
Rerror_rate	TCP REJECT Hatası veren bağlantıların yüzdesi	sürekli
same_srv_rate	Aynı servise yapılan bağlantıların yüzdesi	sürekli
diff_srv_rate	Farklı servise yapılan bağlantıların yüzdesi	sürekli
Srv_count	Aynı servise 2 saniye içerisinde yapılan bağlantıların sayısı	sürekli
Srv_serror_rate	Aynı servise yapılan bağlantılardaki TCP SYN hatasının yüzdesi	sürekli
Srv_rerror_rate	Aynı servise yapılan bağlantılardaki TCP REJECT hatasının yüzdesi	sürekli
Srv_diff_host_rate	Farklı servislere yapılan bağlantıların yüzdesi.	sürekli

Çizelge 5.4 Eğitim veri kümesinde bulunan saldırıların listesi ve türleri

Saldırı Adı	**Saldırı Tipi**
Back	Dos Saldırısı
Buffer_overflow	U2R Saldırısı
ftp_write	R2L Saldırısı
Guess_passwd	R2L Saldırısı
İmap	R2L Saldırısı
İpsweep	Probe işlemi
Land	Dos Saldırısı
loadmodule	U2R Saldırısı
Multihope	R2L Saldırısı
Neptune	Dos Saldırısı
Nmap	Probe işlemi
Perl	U2R Saldırısı
Phf	R2L Saldırısı
Pod	Dos Saldırısı
Portsweep	Probe işlemi
Rootkit	U2R Saldırısı
Satan	Probe işlemi
Smurf	Dos Saldırısı
Spy	R2L Saldırısı
Teardrop	Dos Saldırısı
Warezclient	R2L Saldırısı
Waresmaster	R2L Saldırısı

5.3 Girdi Vektörlerinin Sayısallaştırılması

KDD Cup veri kümesinde her bir sütunun belli başlı değerleri vardır. Bu değerlerin önemli bir kısmı sayısal değerlerdir fakat ikinci, üçüncü, dördüncü ve son sütunlarda değerler metin bazlıdır. İkinci sütuna karşılık gelen bilgiler bağlantının hangi protokol üzerinden yapıldığını göstermektedirler. Bu sütunun aldığı değerler aşağıdaki gibi listelenmiştir:

1. udp
2. tcp
3. icmp

Üçüncü sütun ise tam olarak 64 farklı değer almaktadır. Bu sütun bağlantının hangi servis üzerinden gerçekleştiğinin bilgisini tutar. Bu veriler çizelge 5.5 de gösterilmiştir.

Çizelge 5.5 Üçüncü sütunun alabileceği değerler.

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

Dördüncü sütun 11 değişkenden değeri alabilir ve TCP bağlantısının içerisindeki bayrak değerinin bir listesini içerir. Bu değerler aşağıda listelenmiştir.

1. SF
2. RSTR
3. S1
4. REJ
5. S3
6. RSTO
7. S0
8. S2
9. RSTOS0
10. SH
11. OTH

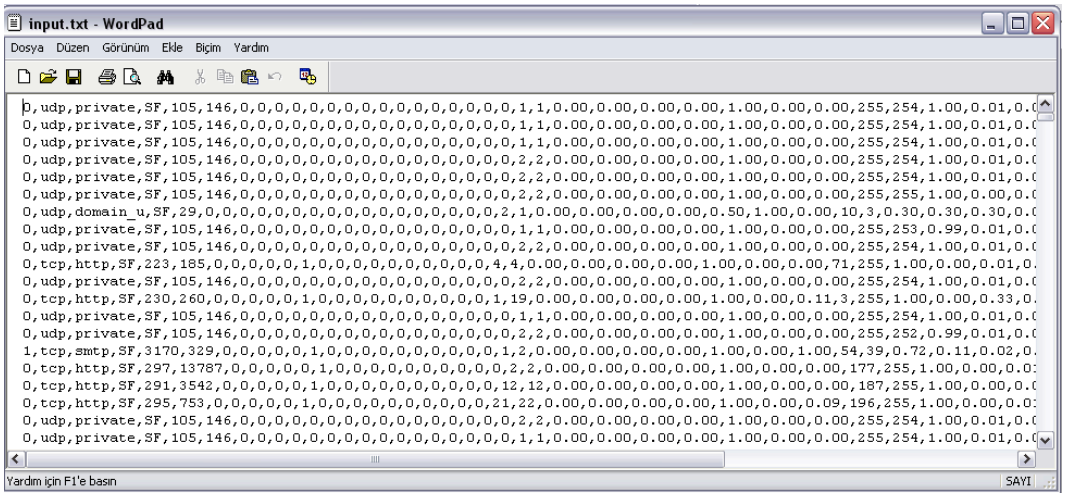
Son sütun ise bağlantının normal mi yoksa bir saldırı mı olduğunu, eğer saldırıysa hangi saldırı tipini içeridiğini belirleyen değerlerin bir listesini içerir. Bu sütun 31 farklı değer almaktadır. Bu değerler Çizelge 5.6 da listelenmektedir.

Çizelge 5.6 Son sütunun alabileceği değerler.

1. normal.	9. sendmail.	17. udpstorm.	25. nmap.
2. snmpgetattack.	10. guess_passwd.	18. warezmaster.	26. rootkit.
3. named.	11. saint.	19. perl.	27. neptune.
4. xlock.	12. bfr_overflow.	20. satan.	28. loadmodule.
5. smurf.	13. portsweep.	21. xterm.	29. imap.
6. ipsweep.	14. pod.	22. mscan.	30. back.
7. multihop.	15. apache2.	23. processtable.	31. httptunnel.
8. xsnoop.	16. phf.	24. ps.	

İşlenmemiş vektörler Şekil 5.1 de gösterilmiştir. Şekilde görülen her bir satır yeni bir bağlantı vektörüne karşılık gelmekte her bir sütun ise bağlantının karakteristik özellikleri hakkında bilgi barındırmaktadır. Bu vektörlerin

özdüzenlemeli ağa girdi olarak verilebilmesi için sayısallaştırılması gerekmektedir.



Şekil 5.1 Girdi vektörlerinin orjinal görüntüsü.

Sayısallaştırma işlemi şu şekilde yapılmaktadır. Son sütun hariç metin değeri alan her bir sütun için bu değer yerine, çizelgelerde verilen karşılık gelen sayısal değerler kullanılmıştır.

Bir örnek verilecek olursa ikinci kolon için udp metnine karşılık gelen her yere 1 sayısı atanmış, tcp metininin bulunduğu her yere 2 sayısı, icmp metninin yazıldığı her yere ise 3 sayısı atanmıştır.

Son sütunda yapılan işlem ise bundan biraz farklıdır. Son sütunda bağlantının normal veya saldırı verisi olup olmadığı gösterilmektedir. Normal kelimesi bulunan bütün alanlara 0 ve diğerlerine de 1 değeri atanmıştır.

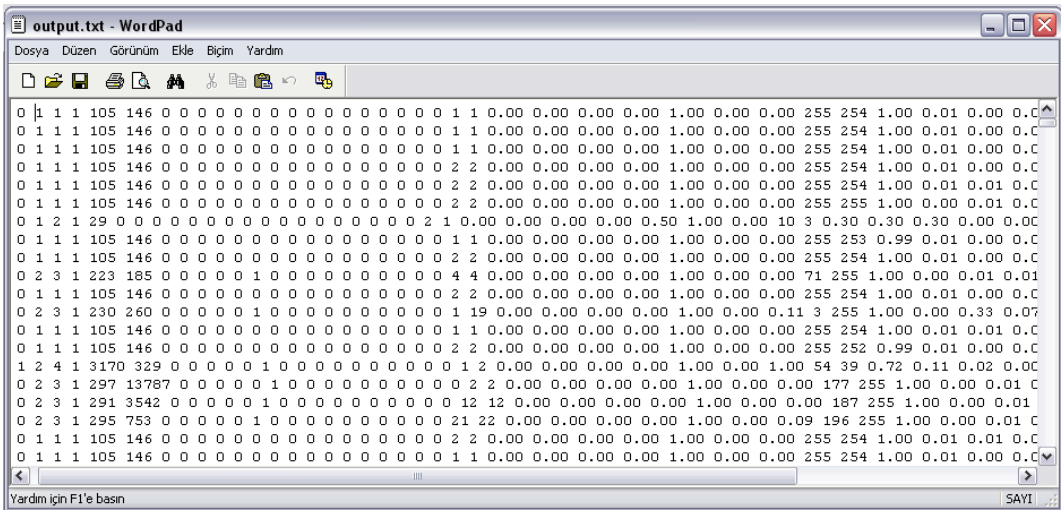
Bu işlem sonrasında elde edilen girdi vektörlerinin bir görüntüsü Şekil 5.2 de verilmiştir.

5.4 Girdi vektörlerinin Temel Bileşen Analizine tabi tutulması ve boyut düşürülmesi işlemi.

Çalışmanın bu kısmındaki amaç, Temel Bileşen Analizinin sistemin karar verme hızına ve verilen kararın doğruluk oranına etkisinin araştırılmasıdır.. Bu nedenle tasarlanan özdüzenlemeli haritaya doğrudan sayısallaştırılmış vektörleri

kullanmak yerine bu vektörlerin Temel Bileşenlerinin bulunup sistemin temel bileşenlerle eğitilmesinin sistemin karar verme başarısına ne derecede etki ettiğinin gözlenmesi amaçlanmıştır.

KDD Cup verisi yaklaşık 120.000 vektörden oluşmaktadır. Her bir vektörün 41 adet sütunu vardır. Bu 120.000 vektörden rastgele 1000 tanesi seçilmiş 1000x41 boyutlu bir örnek vektör kümesi oluşturulmuştur. Elde edilen 1000x41 örnek vektör matrisi Temel Bileşen Analizine tabi tutulmuş ve 41x41 lik bir öz vektör matrisi elde edilmiştir.



Şekil 5.2 Girdi vektörlerinin sayısallaştırılmış görüntüsü.

Bu aşamadan sonra elde edilen **V** öz vektör matrisinin sütun vektörleri öz değerleri büyükten küçüğe doğru olacak şekilde sıralanmıştır. Bu işlemden sonra ilk N özvektör alınmış ve 41xN 'lik bir temel bileşen matrisi oluşturulmuştur. Bu matrise **P** adını verelim. Burada N değerinin seçimi deneyseldir ve bu değer değiştirilerek sonuçlarda en iyi ayrıştırmayı yapabilecek vektör kümesinin bulunması hedeflenmiştir.

41xN boyutlu **P** matrisi ile 120.000x41 boyutlu **G** girdi matrisinin çarpımı hesaplanarak **S = GxP** işlemi yapılmıştır. Elde edilen **S** sonuç matirisinin boyutunun 120.000xN olacağı aşikardır.

Buna göre ilk 5 temel bileşen değerlendirmeye alınırsa N değeri 5 olacağı için elde edilen sonuç matrisi **S'**nin boyutu da 120.000x5 olacaktır. Bu şekilde boyut düşürme işlemi tamamlanmış ve girdi matrisinin boyutu 120.000x41'den 120.000x5'e düşmüştür (N=5 olmak üzere). Temel Bileşen Analizi yapıldıktan

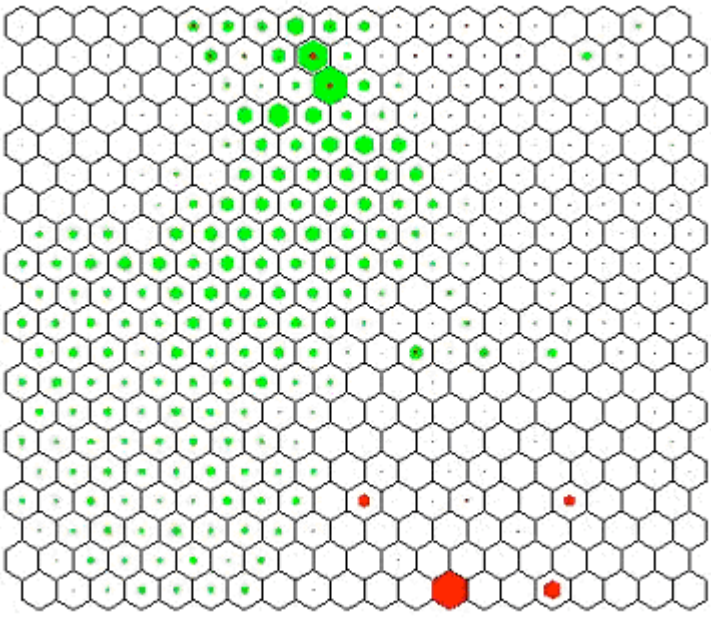
sonra elde edilen yeni girdi matrisi **S** matrisidir ve bu matris ayrıca çalışmada özdüzenlemeli ağın işleminde kullanacağı matristir.

5.5 Özdüzenlemeli haritanın oluşturulması ve eğitilmesi.

Ayırt edilecek sınıf sayısı saldırı ve normal olmak üzere 2 olduğu için haritanın boyutu 4. bölümdeki haritaya göre küçültülmüş ve 20x20 olarak seçilmiştir.

20x20'lik haritalar üzerinde girdi vektörlerinin Temel Bileşenlerinin Sayısını değiştirerek 15 ayrı veri kümesi oluşturulmuştur. Bu 15 veri kümesinin birbirinden farkı her bir veri kümesinde temel bileşen sayısının değiştirilmesidir. Başka bir deyişle birinci veri kümesi için N değeri 2 iken ikinci veri kümesi için 2 temel bileşen yerine ilk 3 temel bileşen kullanılmış ve N değeri 3 olarak seçilmiştir.

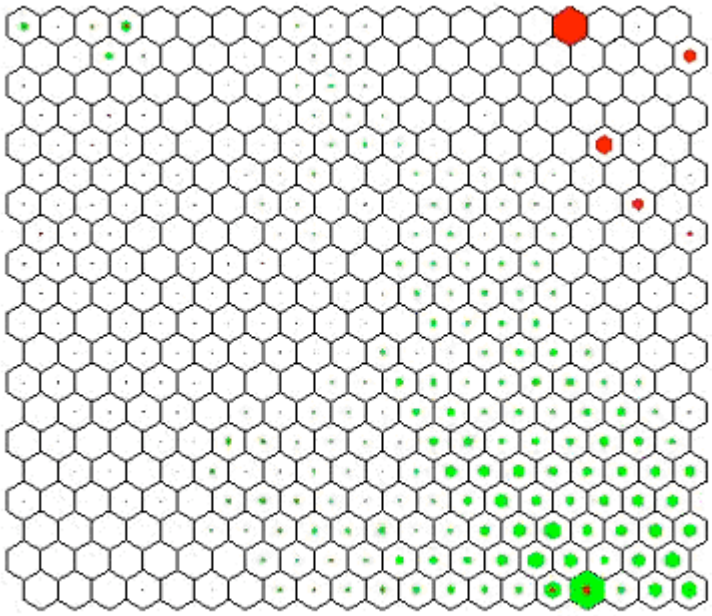
Temel bileşen analizi sonucunda elde edilen vektörlerle eğitilen özdüzenlemeli harita, normal ve saldırı vektörlerini ayırt etme konusunda %90 oranında başarılı olmuştur.



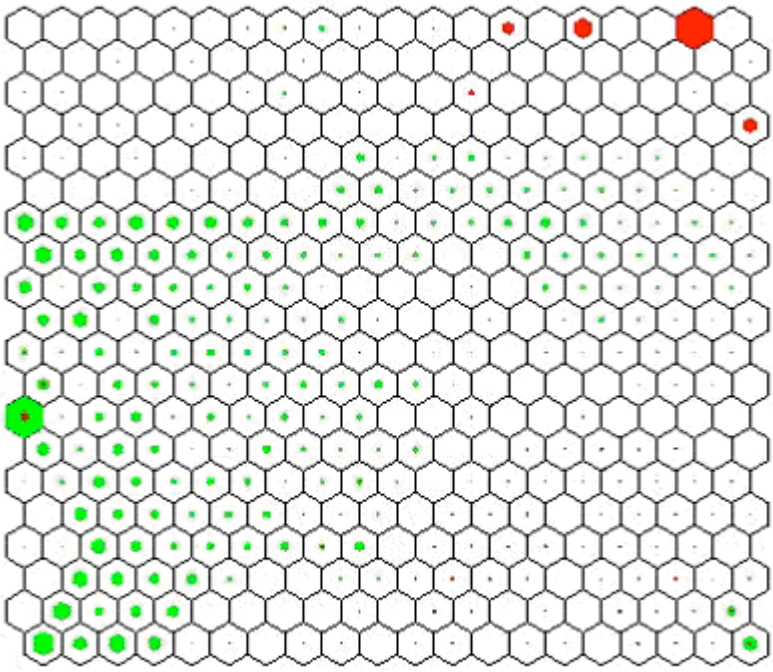
Şekil 5.3 İlk 3 temel bileşenin kullanımı sonucunda oluşan harita. Kırmızılar saldırı yeşiller normal vektörlere karşılık gelmektedir.

20x20 lik bir harita üzerinde ilk 3 temel bileşenin kullanıldığı sınıflandırma işlemi yapıldıktan sonra elde edilen renklendirilmiş haritanın görüntüsü Şekil 5.3 de verilmiştir. Şekil 5.4 ise ilk 5 temel bileşenin kullanıldığı sınıflandırma işlemi yapıldıktan sonra elde edilen renklendirilmiş haritayı içermektedir.

20x20 lik bir harita üzerinde ilk 10 temel bileşenin kullanıldığı sınıflandırma işlemi yapıldıktan sonra elde edilen renklendirilmiş haritanın görüntüsü Şekil 5.5 de verilmiştir:



Şekil 5.4 İlk 5 temel bileşenin kullanımı sonucunda oluşan harita. Kırmızılar saldırı yeşiller normal vektörlere karşılık gelmektedir.



Şekil 5.5 İlk 9 temel bileşenin kullanımı sonucunda oluşan harita. Kırmızılar saldırı yeşiller normal vektörlere karşılık gelmektedir.

5.6 Sonuçlar

Temel Bileşen Analizinde 1<N<16 için özdüzenlemeli ağ toplam 15 defa eğitilmiştir. Sistemin başarısını ölçen programın çıktısı aşağıdaki gibidir. Sonuçlardan da görülebileceği üzere temel bileşenlerin sayısının artışı sistemin başarısını belirli bir noktaya kadar artırmıştır. Aşağıdaki sonuçlara göre en yüksek saldırı ayırt etme yüzdesi N=10 değeri için sağlanmıştır. N=10 değeri için sistemin başarı oranı %98.83 olarak bulunmuştur. Sonuçlardan da görülebileceği gibi N=10 değerinden sonraki değerler için sistemin başarısı bir miktar düşmüştür. Bu durum N=10'dan büyük temel bileşenlerin sistemein karar verme başarı yüzdesine olumlu olarak etki etmediğini göstermektedir. Buradan N=10 değeri %98.83 başarı oranıyla optimum temel bileşen olarak gözükmektedir.

Çizelge 5.7 N değerine göre başarı yüzdeleri

	Normal Vektör Ayırt Etme Yüzdesi	Saldırı Vektörleri Ayırt Etme Yüzdesi	Genel Başarı Yüzdesi
N=2	87.76	85.73	87.38
N=3	90.46	80.31	88.56
N=4	87.94	87.93	87.94
N=5	96.42	88.25	94.40
N=6	96.45	90.32	95.31
N=7	95.76	88.35	94.38
N=8	93.99	86.64	92.25
N=9	99.23	94.77	98.40
N=10	98.97	98.22	98.83
N=11	97.58	97.26	97.52
N=12	97.73	85.92	95.53
N=13	98.94	88.52	97.00
N=14	97.46	96.58	97.30
N=15	97.25	96.43	96.76

6. SONUÇLAR VE İLERİDEKİ ÇALIŞMALAR

Çalışmanın ilk kısmında bir ağ cihazının herhangi bir portundan geçen trafik bilgisini kullanarak ağ trafiği sınıflandırılmaya çalışıldı. Ağ trafiği süregelen bir yapıya sahip olduğu için takip eden trafik vektörleri arasındaki bağlantıyı gözlemlek çok büyük bir öneme sahiptir. Bu yapının gözlemlenmesi harita üzerindeki izleri takip etmekle sağlanabilir.

Bu şekilde sistem tek bir vektör için hatalı sonuç üretse bile bu yanlış sonuç vektörün hangi noktada yoğunlaştığının takip edilmesi ile elenebilir. Çalışmanın bu kısmında üç farklı türden trafik verisinin sınıflandırılmasında %95 oranında başarı sağlanmıştır. Eğer harita üzerindeki izlerin takibi devreye sokulursa sistemin başarı oranı %99'lara çıkmaktadır. Böyle bir sistem ağ cihazlarının üzerinde rahatlıkla uygulanabilir fakat cihazları uzaktan algılayan ağ yönetim yazılımlarına entegre edilmesi çok daha uygundur. Çünkü sistemin ihtiyaç duyduğu tek şey son 400 saniyedeki trafik bilgisidir.

Çalışmanın ikinci kısmında ise girdilerimiz çok geniş bir ağ üzerinde kaydedilmiş ağ günlükleridir. Bu ağ günlükleri uygun bir şekilde sayısallaştırıldıktan sonra sisteme girdi olarak hazır hale getirilmiştir. Çalışmanın bu aşamasının getirdiği yenilik, girdilerin ilk önce temel bileşen analizi kullanılarak boyut küçültme işlemine tabi tutulmasıdır. Bu şekilde hem özdüzenlemeli haritayı eğitmek için geçmesi gereken süre azaltılacak hem de sistem dışarıdan gelen isteklere daha hızlı yanıt verecektir.

Bu yapı gerçek zamanlı sistemler için çok daha uygundur. Temel bileşen analizinde, hesaplanan temel bileşenlerden kaç tanesinin kullanılacağı sistemin başarsını doğrudan etkilemektedir. Denemelerden sonra en başarılı karar verme yüzdesinin N=10 iken elde edildiği görülmüştür. Bu durumda sistemin başarı oranı %98.83 olarak bulunmuştur. Bu yüzde KDD Cup veri kümesini özdüzenlemeli ağlarla sınıflandıran fakat temel bileşen analizi kullanmayan sistemlerin başarı oranlarından daha yüksektir. [32] ve [33] nolu çalışmalarda KDD Cup verisi üzerinde özdüzenlemeli ağlar kullanılarak anomali tespit'i yapılmış ve başarı oranları sırası ile %95 ve %91.5 olarak bulunmuştur.

İlerideki çalışmlarda KDD Cup veri kümesinde temel bileşen analizine tabi tutulacak veri kümesinin seçiminde çeşitli iyileştirilmeler yapılabilir. Bu vektörler rastgele seçilmek yerine çeşitli clustering algoritmalarına tabi tutulabilir ve daha sonra temel bileşen analizine tabi tutulabilir. İleriki çalışmalarda ikinci kısım için saldırı vektörlerinin de birbirinden ayrılabilmesini sağlayacak bir özdüzenlemeli ağ yapısı oluşturulması düşünülmektedir.

KAYNAKLAR

[1] Thottan, M. and Ji, C., "Anomaly Detection in IP Networks," IEEE Transactions on Signal Processing, **51(8),** pp. 2191-2203, August, 2003.

[2] Cannady, J., "Artificial Neural Networks for Misuse Detection," 21st National Information Systems Security Conference, p. 2, 1998.

[3] Mukkamala, S., Sung, A. and Abraham, A., "Intrusion Detection using an ensemble of intelligent paradigms," Journal of Network and Computer Applications, **28,** pp: 167-182, 2005.

[4] Fugate, M. and Gattiker, J.R., "Computer Intrusion Detection with Classification and Anomaly Detection, using SVMs," International Journal of Pattern Recognition and Artificial Intelligence, **17**, pp. 441-458, 2003.

[5] Fugate, M. and Gattiker, J.R., "Anomaly Detection Enhanced Classification in Computer Intrusion Detection, Applications of Support Vector Machines," Int. Conf. Pattern Recognition and Machine Learning, pp. 186-197, 2002.

[6] Mukkamala, S., Sung, A., "Feature Ranking and Selection for Intrusion Detection Systems Using Support Vector Machines," In Digital Forensic Research Workshop, 2002.

[7] Gomez, J., Gonzalez, F. and Dasgupta, D., "An Immuno-Fuzzy Approach to Anomaly Detection.", The IEEE International Conference on Fuzzy Systems, pp. 1219-1224.

[8] Castillo, E., Cobo, A., Gutierrez, J.M. and Pruneda, E., "Functional Networks with Applications", Kluwer Academic Publishers, 1998.

[9] Venkatachalam, V. and Selvan, S., "Intrusion Detection using an Improved Competitive Learning Lamstar Neural Network," IJCSNS International Journal of Computer Science and Network Security, **7(2)**, February 2007

[10] Knowledge Discovery and Data Mining Tools Conference Cup 1999 Data, <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>

[11] Lipmann, P., Fried, D., Graf, I., Haines, J., Kendall, K., McClung, D., Weber, D., Webster, S., Wyschogrod, D., Cunningham, R., and Zissman,

M., “Evaluating intrusion detection systems. In the 1998 DARPA on-line intrusion detection evaluation,” DARPA Information Survivability Conf. & Exposition, 2000.

[12] Karasulu B. and Uğur A., “Özörgütlenmeli Yapay Sinir Ağı Modelinin Kullanıldığı Kutup Dengeleme Problemi İçin Paralel Hesaplama Tekniği ile Bir Başarım Eniyileştirme Yöntemi”, Akademik Bilişim 2007, 2007.

[13] Gallant, Stephan I., “Neural Network Learning and Expert Systems”, MIT Press, 1993.

[14] Kızılören, T., Germen, E., “Network Traffic Classification With Self Organizing Maps”, ISCIS 2007, IEEE Explorer, 2007.

[15] Kohonen (Editor), Self-Organizing Maps, Springer-Verlag, Germany, 2001.

[16] Rauber A., Tomsich P., Merkl D., “parSOM: A Parallel Implementation of the Self-Organizing Map Exploiting Cache Effects: Making the SOM Fit for Interactive High-Performance Data Analysis”, HPCN Europe, 2000.

[17] Koikkalainen P. and Oja E., “Self-organizing hierarchical feature maps. In Proceedings of the International Joint Conference on Neural networks”, **2(1)**, pp. 279–285, Piscataway,NJ, IEEE Service Center, 1990.

[18] Rhodes B., Mahaffey J. A. and Cannady J.,Proceedings of the 23rd National Information Systems, 2001.

[19] Pearson, K. "On Lines and Planes of Closest Fit to Systems of Points in Space," Philosophical Magazine 2, 2006

[20] Jolliffe, I.T., “Principal Component Analysis, Series: Springer Series in Statistics,” 2nd Ed., Springer, NY, 2002

[21] Ding, S. and Shi, Z., “Analysis and Applications of PCA Information Features,” Journal of Communication and Computer, **2(9)**, 2005.

[22] Stallings, W., “SNMP and SNMP V2: The Infrastructure of Network Management”, Communications Magazine, IEEE, **36(3)**, pp:37-43, 1998.

[23] James F. Krouse, Keith W. Ross, “Computer Networking, A Top-Down Approach Featuring the Internet,” Addison-Wessley, 2005.

[24] Subramanian, “Network Management, Principles and Practice”, Addison-Wesley, 2000.

[25] M. Ilvesmäki and M. Luoma, "On the capabilities of application level traffic measurements to differentiate and classify Internet traffic", Internet Perfomance and Control of Network Systems II, Proceedings of SPIE, **4523**, 2001.

[26] Quian Du, "Low-Complexity Principal Component Analysis for Hyperspectral Image Compression", International Journal of High Performance Computing Applications, **22**, pp. 438-448,2008.

[27] R. Cangelosi, "Component retention in principal component analysis with application to cDNA microarray data," Biology Direct 2007, **2**, pp. 1186,2007.

[28] George D. Thompston, "A Multivariate Assessment of Meteorological Influences on Inhalable Particle Source Impacts," **24**, pp. 1245-1256, 2005

[29] A. Ferraz, "The use of principal component analysis (PCA) for pattern recognition in Eucalyptus grandis wood biodegradation experiments,". **14**, pp. 487-490.

[30] V. Zubko, Kaufman Y. J., "Principal component analysis of remote sensing of aerosols over oceans," EEE transactions on geoscience and remote sensing, **45**, pp. 730-745, 2007

[31] http://www.tcpcdump.org

[32] Ding Li, Ni Gui-qiang, Pan Zhi-Song, Hu Gu-Yu, "Ddos Intrusion Detection Using Generalized Gray Self-Organizing Maps," Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services, pp. 1458-1551, 2007

[33] Kayacik H. G., "Hierarchical Self Organizing Map Based IDS on KDD Benchmark," Master Thesis, Dalhousie Univeristy, Halifax, Nova Scotia, 2003