Derin Öğrenme Yöntemleri ile Bilgisayar Ağlarında Güvenliğe Yönelik Anormallik Tespiti

R. Can AYGÜN
Danışman: Doç. Dr. A. Gökhan YAVUZ

Yıldız Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
Akıllı Sistemler Laboratuvarı

İçerik

- Giriş
- Saldırı Tespit Sistemleri
- Anomali Tespiti
- NSL-KDD Veri Kümesi
- Derin Öğrenmenin Kısa Tarihçesi
- Derin Öğrenmeye Dayalı Anomali Tespit Modelleri
- Performans Değerlendirmesi
- Test Sonuçları

Giriş

- Bilgisayar ağları; e-ticaret, bankacılık ve finansal işlemleri, sağlık, ulaşım, eğitim ve savunma sanayisi gibi hayatın her alanında etkin bir şekilde kullanılmaktadır.
- Bilgisayar ağlarına yapılan saldırılar her geçen gün artmakta ve güçlenmektedir.
- Mevcut saldırı tespit yöntemlerinden daha etkili ve akıllı bir sistemine ihtiyaç vardır.
- Saldırı tespit sistemlerinin geliştirilmesi aktif bir araştırma konusudur.

Ağ Saldırıları

- *Bilgisayar ağlarına ve sistemlerine yapılan saldırılar, yetkilendirilmemiş bir erişim ile ağın ve/veya ağa bağlı sistemlerdeki servis devamlılığını, veri bütünlüğünü ve gizliliğini tehdit ederler.*
- DOS:
  - *Belirli bir ağın veya sunucunun hizmet vermesini engelleyecek şekilde çok sayıda istek gönderilmesi soncunda sunucunun legal hiçbir isteğe cevap veremeyecek duruma getirilmesidir.*
- Probe:
  - *Belirli bir ağdan veya sistemden, saldırı için ön hazırlık yapmak amacı ile kullanılması planlanan bilgilerin elde edilmesi olarak tanımlanır.*
- User to Root (U2R):
  - *Normal kullanıcı olarak login olunan bir sistemdeki açıkların kullanılarak root yetkilerinin elde edilmesi.*
- Remote to User /Remote to Local (R2U /R2L)
  - *Legal kullanıcı hesabı olmadan uzaktaki bir sisteme zararlı paketler gönderilerek root veya normal kullanıcı olarak erişim sağlanması.*

Saldırı Tespit Sistemleri

- Kullanıcıların sistem ve ağ aktivitelerini izlenmesi
- Sistemin olası saldırılara karşı hazır olacak şekilde düzenlenmesi
- Sistem ve veri bütünlüğünün korunması
- Saldırların ve kullanıcı yetki ihlallerinin tespiti

Fonksiyonalitelerine sahip güvenlik mekanizmalarıdır.

Saldırı Tespit Sistemlerinin Sınıflandırılması

Saldırı Tespit Yaklaşımı Bazında	Uygulama Alanına göre
İmza Tabanlı Saldırı Tespit Sistemleri	Sistem Tabanlı Saldırı Tespit Sistemleri
Anomali Tespiti Tabanlı Saldırı Tespit Sistemleri	Ağ Tabanlı Saldırı Tespit Sistemleri

İmza Tabanlı Saldırı Tespiti

- Daha önceden meydana gelmiş saldırıların imzaları kullanılarak saldırı tespiti yapılır.
- Saldırı imzası nedir ?
  - Programlara ilişkin bilgiler (Sistem çağrıları, izinler, ağ erişimi, program çalışma periyodu)
  - Ağ paketine ilişkin bilgiler (Protokol tipi, akış süresi, veri boyutu)
- Oluşturulan imzalar, İmza veri tabanlarında saklanmaktadır.
- Uzman kişiler tarafından imza veri tabanlarının sürekli güncel tutulması gerekmektedir.
- Sıfırıncı gün saldırılarının tespiti konusunda başarısızdırlar.
- Antivirüs sistemlerine benzer şekilde çalışırlar.

Anomali Tespiti Tabanlı Saldırı Tespiti

- Anomali, verinin büyük çoğunluğundan farklı bir örüntüye sahip olan ve genellikle başka bir sistem tarafından oluşturulmuş örnekler olarak tanımlanmaktadır.
- Bilgisayar ağ güvenliği, tıp ve toplum sağlığı, finansal sahtekarlık tespiti, askeri uygulamalar, endüstriyel uygulamalar, sensor ağlar, robotik uygulamalar ve astronomik verilerin analizi
- Makine öğrenmesi algoritmaları ve İstatistiksel yöntemler anomali tespiti amacı ile kullanılmaktadır.
- Kötüye kullanım tespitindeki amaç bilgisayar ağlarına veya sistemlerine yapılan spesifik saldırıları tespit etmek iken anomali tespitindeki amaç, normal sistem profiline uymayan her türlü faaliyetin tespitini yapmaktadır.

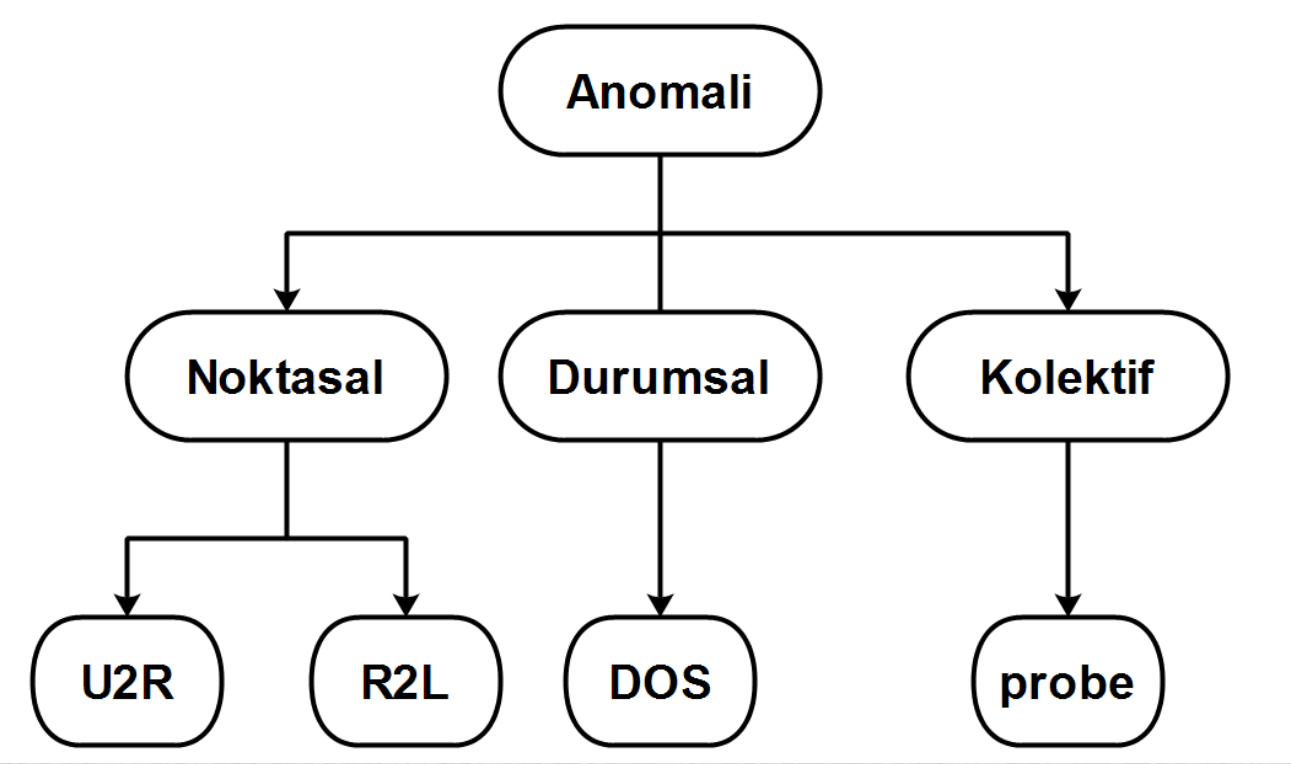
Anomali Türleri

Noktasal Anomali: Belirli bir örnek, normal verinin tamamından farklı bir karakteristiğe sahipse noktasal anomali olarak kabul edilir.

Durumsal Anomali: Belirli bir örnek, sadece belirli durumlarda normal veriden farklı oluyor ise durumsal anomali olarak kabul edilir.

Kolektif Anomali: Belirli bir grup veriyi oluşturan örnekler bireysel olarak değerlendirildiklerinde normal veriden farklı olarak nitelendirilmezken, toplu şekilde değerlendirildiklerinde normal veriden farklı bir yapı ortaya koyuyorlar ise bu şekildeki veri grupları kolektif anomali olarak adlandırılırlar.

Ağ Saldırının Anomali Türlerine Göre Sınıflandırılması



Anomali Tespiti Başarımını Etkileyen Faktörler

- Yöntem:
  - *Makine öğrenmesi yöntemleri, istatistiksel yöntemler*
- Veri yapısı
  - *Nümerik özellikler, alfanümerik özellikler*
- Veri kümesine ait önemli özelliklerin seçilmesi
- Mesafe ölçüm yöntemleri
- Eğitim türü

Makine Öğrenmesi Tabanlı Anomali Tespiti

- Anomali tespit modelleri; gözetimli , yarı gözetimli ve gözetimsiz şekilde eğitilebilmektedir.
- Gözetimli Eğitim Yöntemleri;
  - İmza tabanlı saldırı tespit sistemlerine göre daha başarılıdırlar.
  - Sıfırıncı gün saldırılarının tespiti konusunda başarısızdırlar.
- Yarı Gözetimli Eğitim Yöntemleri;
  - Model sadece normal veri kullanılarak eğitilir.
  - Eğitilmiş modele benzemeyen tüm veriler anormal olarak etiketlenir.
  - Sıfırıncı gün saldırılarının tespiti konusunda daha başarılıdırlar.

Bayes ağları, yapay sinir ağları, destek vektör makineleri , gözetimli veya yarı gözetimli şeklide eğitilerek anomali tespiti amacı ile kullanılabilmektedir.

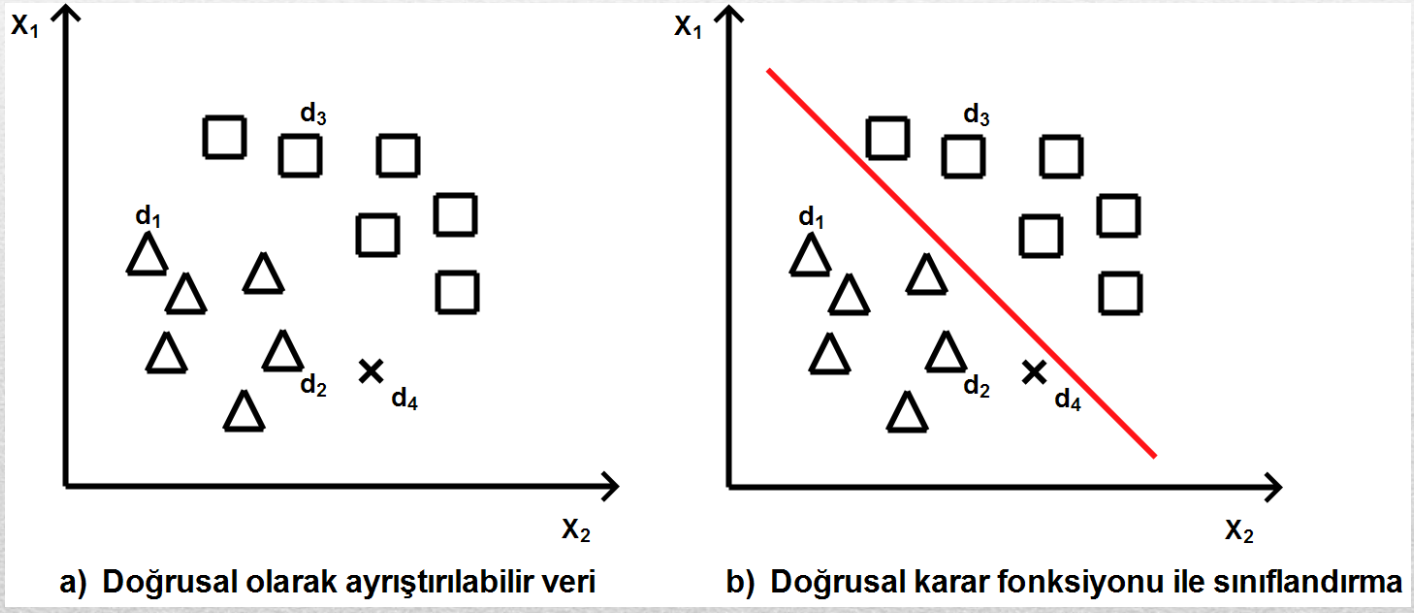
Makine Öğrenmesi Tabanlı Anomali Tespiti

- Gözetimsiz Eğitim Yöntemleri ;
  - Model etiketsiz veri kullanılarak eğitilir ancak eğitim kümesindeki anormal veri yoğunluğunun normal veri yoğunluğuna kıyasla daha az olması gerekmektedir.
  - Normal ve anormal verilerin ayrı ayrı kümelenmesi amaçlanmaktadır.
  - Sıfırıncı gün saldırılarının tespiti konusunda başarılıdırlar.

K-means, k-meodids gibi yöntemler kümeleme tabanlı anomali tespiti amacı ile kullanılabilmektedir.

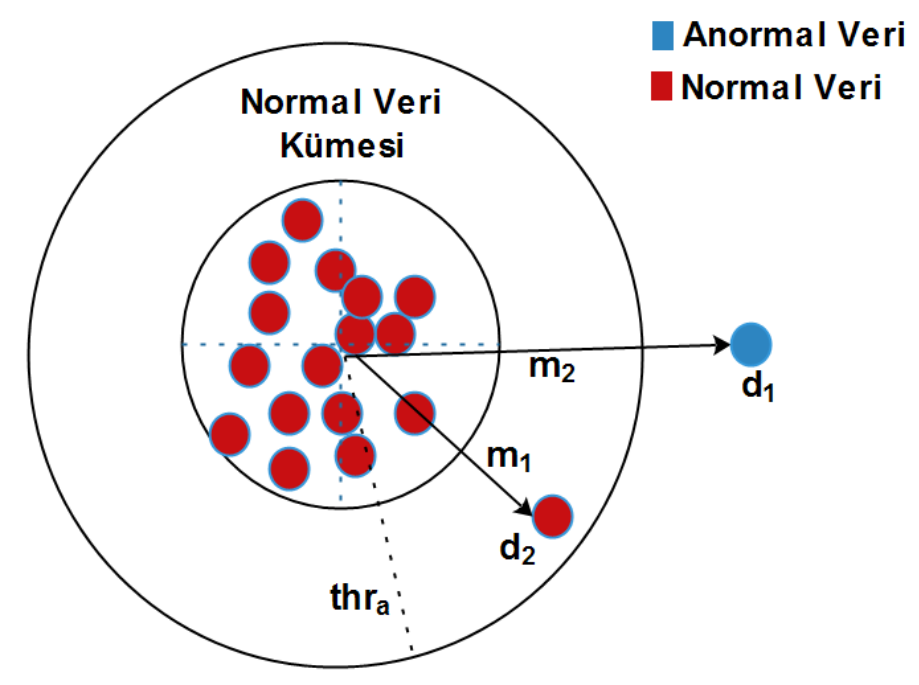
Makine Öğrenmesi Yöntemleri Bakımından Anomali Tespit Yöntemlerinin Sınıflandırılması

- Sınıflandırma tabanlı yöntemler
  - Gözetimli eğitim, yarı gözetimli eğitim



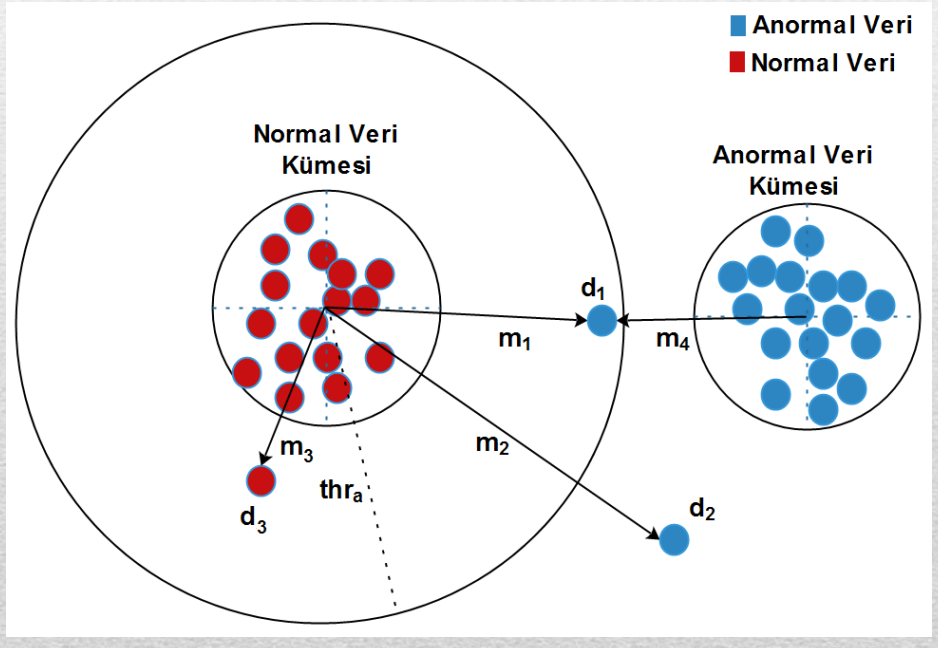
Makine Öğrenmesi Yöntemleri Bakımından Anomali Tespit Yöntemlerinin Sınıflandırılması

- Uç değer tespiti
  - Yarı gözetimli eğitim (sınıflandırıcı veya kümelemeye tabanlı)



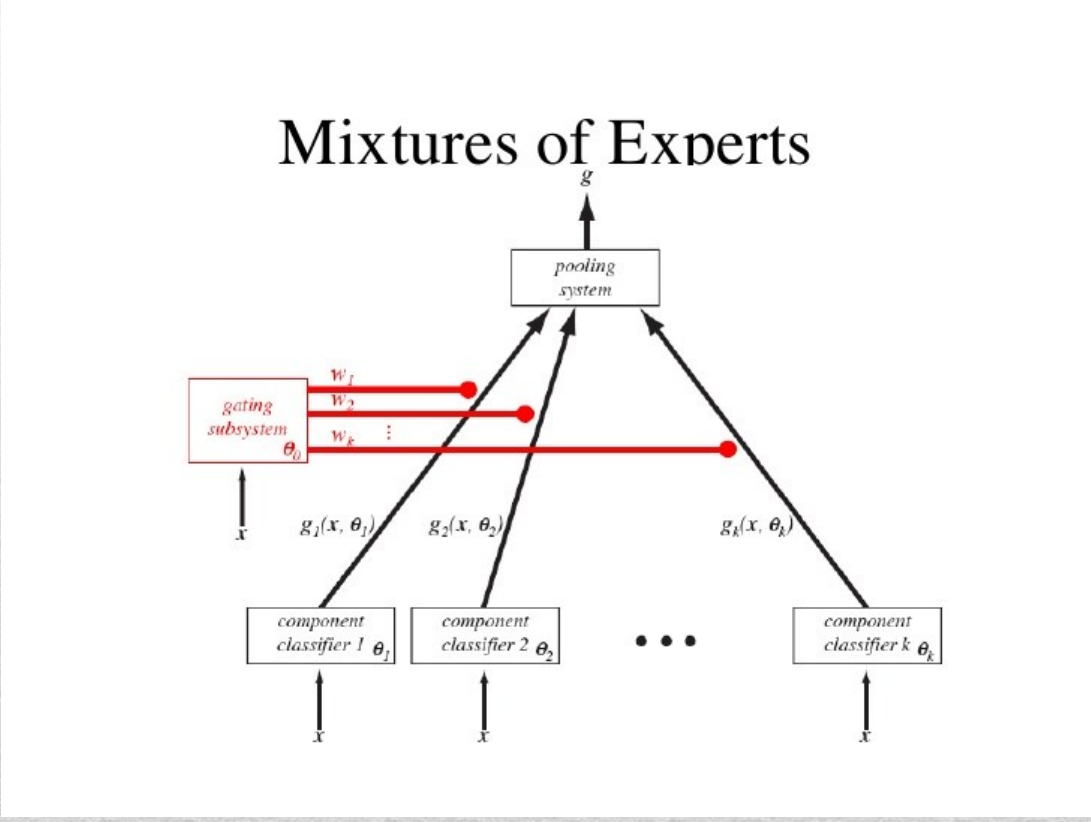
Makine Öğrenmesi Yöntemleri Bakımından Anomali Tespit Yöntemlerinin Sınıflandırılması

- Kümeleme yöntemleri
  - Gözetimsiz eğitim yöntemleri



Makine Öğrenmesi Yöntemleri Bakımından Anomali Tespit Yöntemlerinin Sınıflandırılması

- Hybrid ve Ensemble yöntemler



Ağ Saldırı Tespit Sistemi Değerlendirme Veri Kümeleri

- **CAIDA**
  - *Veriler etiketli değildir. Çoklu saldırı senaryoları bulunmaktadır.*
- **DEFCON**
  - *Hacker yarışmalarındaki trafikten elde edilmiş bir veri kümesi*
  - *Geneli saldırı verisi, gerçek ağ verilerinden farklı veriler içermektedir.*
- **ADFA-LD12**
  - Güncel bir ağ ortamından toplanan verileri içermektedir. (Ubuntu 11.0,2012)
- **ISCX UNB**
  - *Gerçek; http, smtp, ssh, imap, pop3 ve ftp trafiği oluşturacak profiller yaratılmıştır. Ek olarak, veri kümesi çeşitli çok katmanlı atak senaryolarına ilişkin verileri de içermektedir.*
- **KDDCUP'99**
- **NSL-KDD**

NSL–KDD Veri Kümesi

- Hala en popüler saldırı tespit sistemi değerlendirme veri kümesi olan KDDCUP’99’un güncellenmiş halidir. KDDCUP’99 veri kümesi;
    - DARPA IDS performans ölçüm programı kapsamında ABD Hava Kuvvetleri üstlerinin ağlarına benzer şekilde oluşturulmuş simülasyon ortamlarından yakalanmış paketlerden oluşturulmuştur.
    - Çok fazla tekrarlı veri içermektedir.
    - Verilerin büyük bölümü en basit yöntemlerle bile kolayca sınıflandırılabilmektedir.
- KDDCUP’99 içindeki tekrarlı veriler temizlenmiştir.
- Veri kümesi çeşitli zorluk seviyelerini oluşturacak şekilde tekrar oluşturulmuştur.
- NSL-KDD, KDDTrain+, KDDTrain+_%20, KDDTest+ ve KDDTest-21 isminde dört ayrı alt veri kümesinden oluşmaktadır.
- Bu çalışmada KDDTrain+_%20 ve KDDTest+ veri kümeleri kullanılmıştır.

NSL–KDD Veri Kümesi

- KDDTrain+_%20, Normal trafiği ve 21 ayrı saldırı tipini içermektedir.
- KDDTest+, KDDTrain+_%20 veri kümesindeki 19 saldırı tipine ek olarak 18 farklı saldırı tipini de içermektedir.
- Bu açıdan KDDTest+, modelin sıfırıncı gün saldırılarının tespiti konusundaki başarımının ölçülmesi açısından etkin bir veri kümesidir.

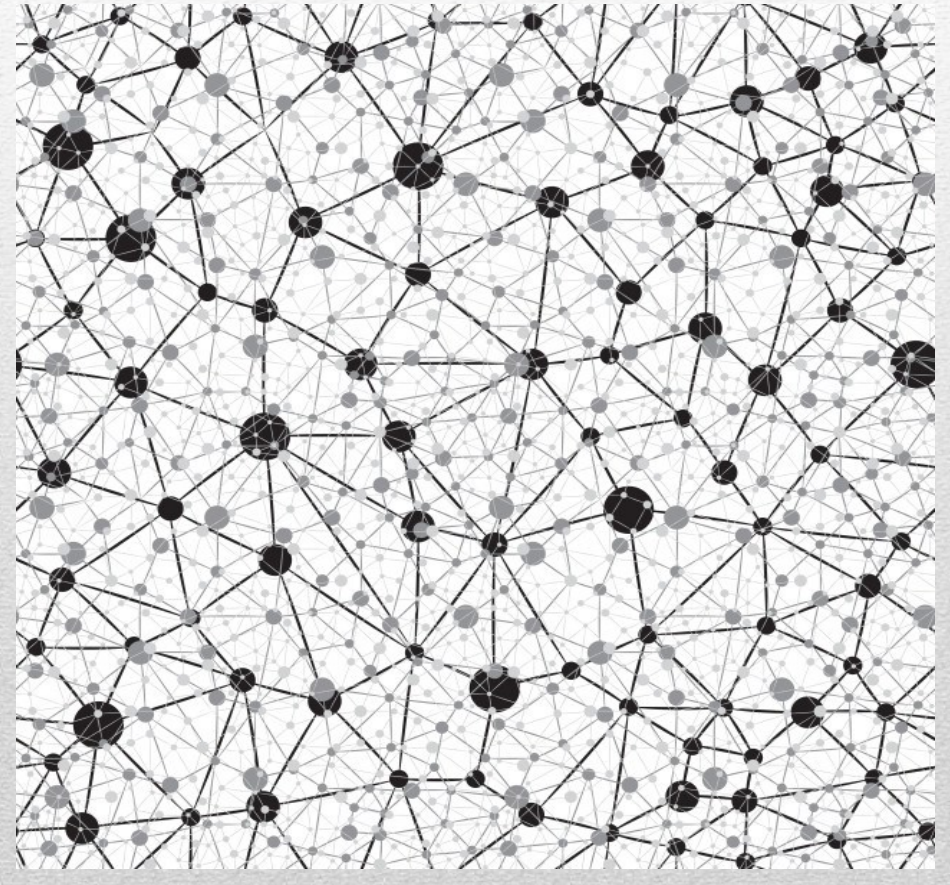
NSL–KDD Veri Kümesi

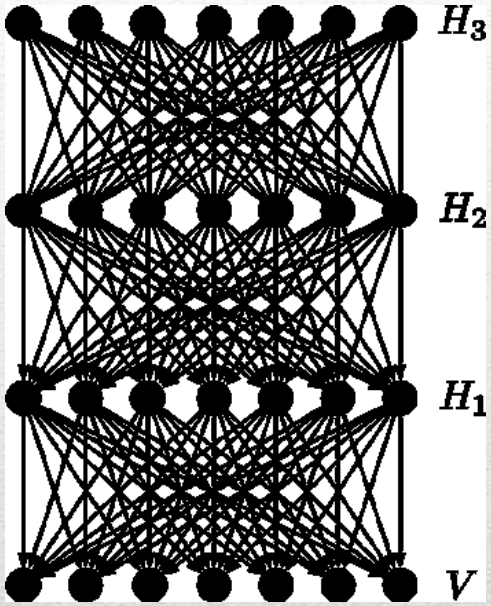
		KDDTrain+%20	KDDTest+
Normal		13449	9711
ANORMAL	DOS	9233	7458
	U2R	11	67
	R2L	347	2887
	PROBE	2289	2421

Makine Öğrenmesi Modellerinin Performansını Etkileyen Faktörler

- Modelin iyi eğitilmesi için veri kümesine ait hangi özellikler seçilmeli?
- Manuel özellik seçimi
- Konusunda uzman olan kişilere ihtiyaç vardır.
    - 200x200 piksellik bir resmi ifade eden en iyi özelliklerin seçilmesi kaç veri uzmanı gerektirir ?
- Klasik yöntemler; PCA, Genetik programlama
- YSA eğitiminde, Stochastic Gradient Descent (SGD) yönteminin lokal minimum değerlerine takılması.

Derin Öğrenme





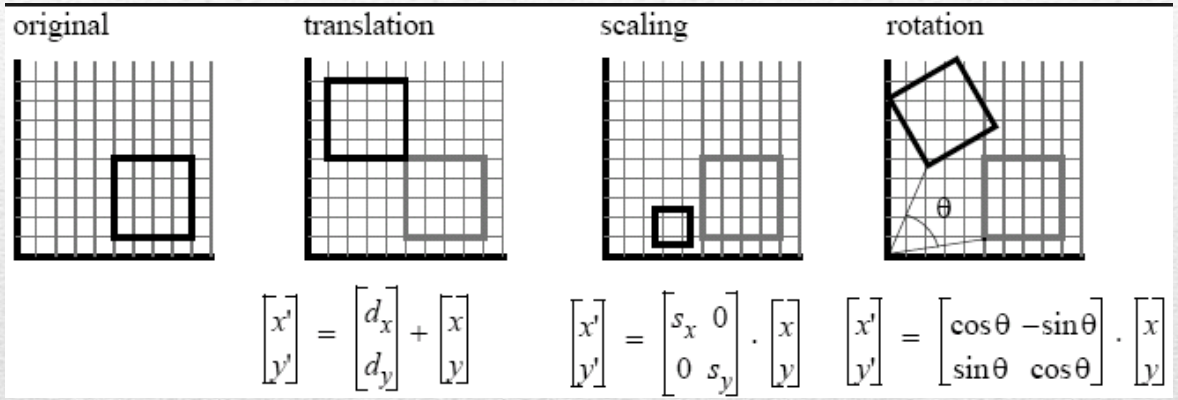
Derin Öğrenme Yöntemlerinin Kısa Tarihçesi

- 1950'lerde perceptron algoritması
- 1980'lerde çok katmanlı perceptron
- 1980-2000 arasında hesaplama gücündeki yetersizlikler nedeniyle ilgi görmemiştir.
- 2000'li yılların sonlarına doğru paralel hesaplama teknolojilerindeki gelişmeler(Clusters , Fast GPUs)
- Bu süreçte yapay sinir ağı mimarisinde çok fazla değişiklik olmadı.
- Derin öğrenme yöntemleri = Daha yüksek başarım (Eğer çok sayıda veri varsa)
- Derin öğrenme gözetimsiz eğitim yapılmasına da imkan tanır.

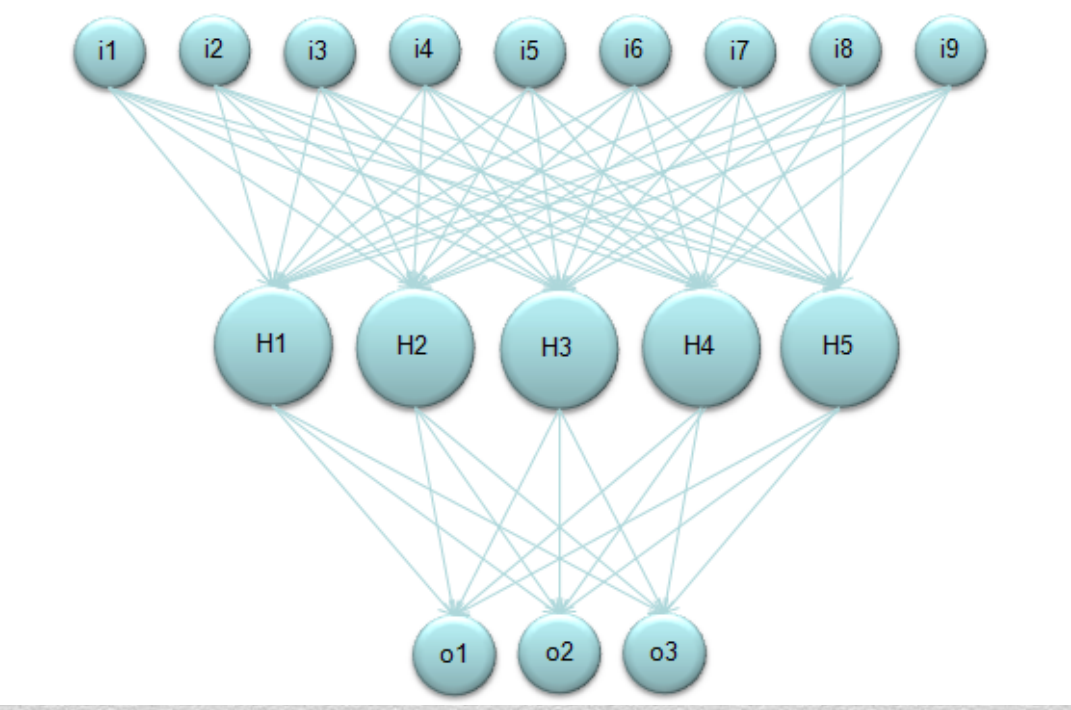
Grafik İşlemci – Derin Öğrenme İlişkisi

- CPU, genel amaçlı işlemci
- GPU, Grafik işlemeye konusunda özelleştirilmiş işlemciler.
- Grafik işleme ile Derin öğrenme arasındaki ilişki nedir ?
  - Neden Grafik işlemciler derin öğrenme eğitim süresini kısaltır ?

Grafik İşlemci – Derin Öğrenme İlişkisi



Grafik İşlemci – Derin Öğrenme İlişkisi



$$\begin{bmatrix} w_{11} & w_{21} & w_{31} & w_{41} & w_{51} & w_{61} & w_{71} & w_{81} & w_{91} \\ w_{12} & w_{22} & w_{32} & w_{42} & w_{52} & w_{62} & w_{72} & w_{82} & w_{92} \\ w_{13} & w_{23} & w_{33} & w_{43} & w_{53} & w_{63} & w_{73} & w_{83} & w_{93} \\ w_{14} & w_{24} & w_{34} & w_{44} & w_{54} & w_{64} & w_{74} & w_{84} & w_{94} \\ w_{15} & w_{25} & w_{35} & w_{45} & w_{55} & w_{65} & w_{75} & w_{85} & w_{95} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \\ i_8 \\ i_9 \end{bmatrix} = \begin{bmatrix} total\ input\ to\ H_1 \\ total\ input\ to\ H_2 \\ total\ input\ to\ H_3 \\ total\ input\ to\ H_4 \\ total\ input\ to\ H_5 \end{bmatrix}$$

$$\begin{bmatrix} w_{11} & w_{21} & w_{31} & w_{41} & w_{51} \\ w_{12} & w_{22} & w_{32} & w_{42} & w_{52} \\ w_{13} & w_{23} & w_{33} & w_{43} & w_{53} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{bmatrix} = \begin{bmatrix} total\ input\ to\ o_1 \\ total\ input\ to\ o_2 \\ total\ input\ to\ o_3 \\ total\ input\ to\ o_4 \\ total\ input\ to\ o_5 \end{bmatrix}$$

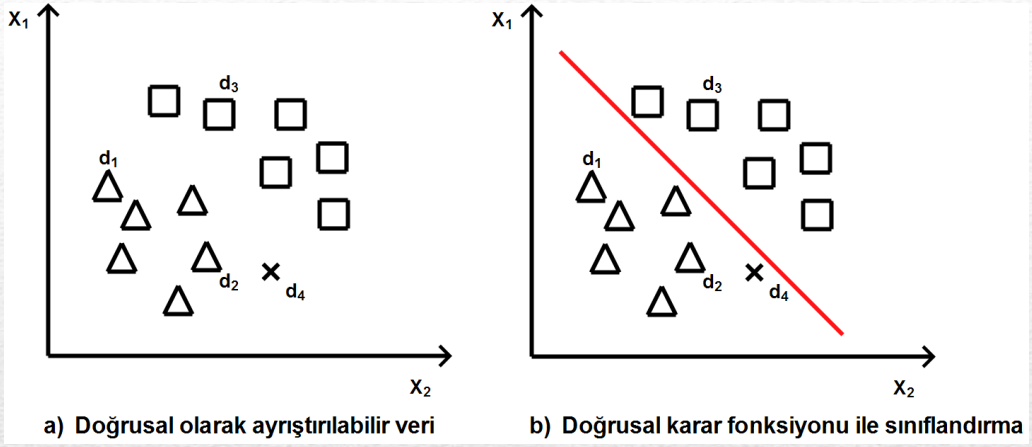
Deep Learning Frameworks

- Theano, Tensorflow, Caffe, Keras

- Derin öğrenme kütüphanelerinin temel Özellikleri :
  - Matematiksel fonksiyonların **sembolik graflar** olarak gösterilebilmesi
  - **Sembolik graflar** üzerinden türev alma
  - Tasarlanan modellerin Nvidia grafik işlemciler üzerinde çalıştırılmasına olanak tanıyan **CUDA** kütüphanesini kullanılır.
  - Matris işlemleri üzerinde etkin bir şekilde işlem yapılabilmesine olanak tanıyan **numPy** kütüphanesi kullanılır.
  - Sembolik graflar üzerinde otomatik türev alma özelliği, kompleks matematiksel fonksiyonların türevleri konusunda programcıların yapabilecekleri hataların önüne geçmektedir.

- Theano kütüphanesi ile gerçeklenen programlar python-numpy, scipy ve C++ dillerine göre, CPU üzerinde 6,5 kat, GPU üzerinde ise 44 kat daha hızlı çalışabilmektedir.

Softmax Aktivasyon Fonksiyonu Tabanlı Sınıflandırıcı



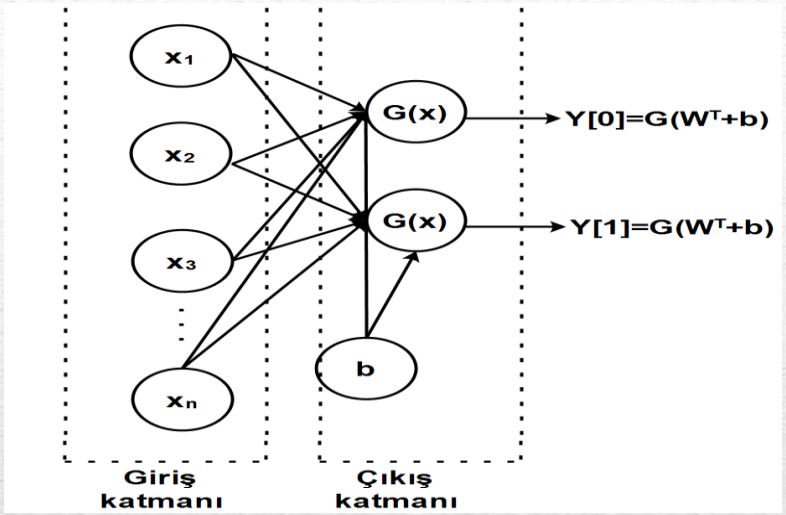
$$P(Y=i \mid x,W,b) = softmax(W_i^T x + b)$$

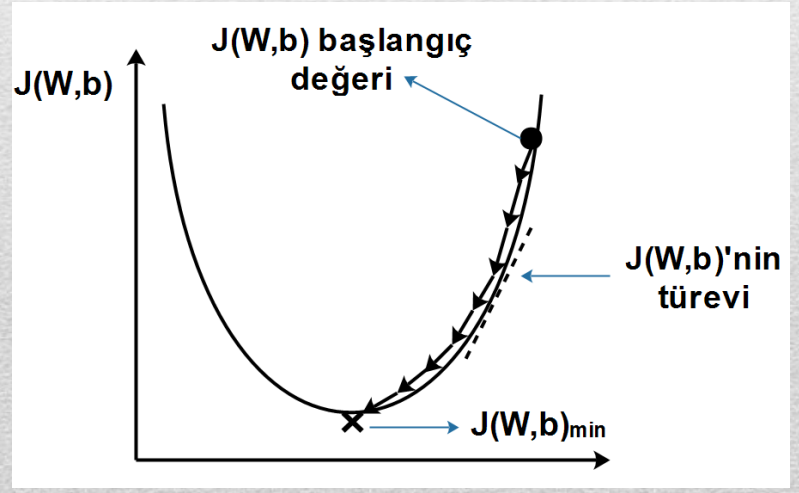
$$P(Y=i \mid x,W,b) = \frac{e^{W_i^T x + b_i}}{\sum_{j=1}^{n} e^{W_j^T x + b_j}}$$

$$y_{pred} = argmax_i P(Y=i \mid x,W,b)$$

$$LL(W,b) = \sum_{i=0}^{|D|} log\left(P(Y=y^{(i)} \mid x^{(i)},W,b)\right)$$

$$J(W,b) = -LL(W,b)$$





Çok Katmanlı Perceptron (Multilayer Perceptron)

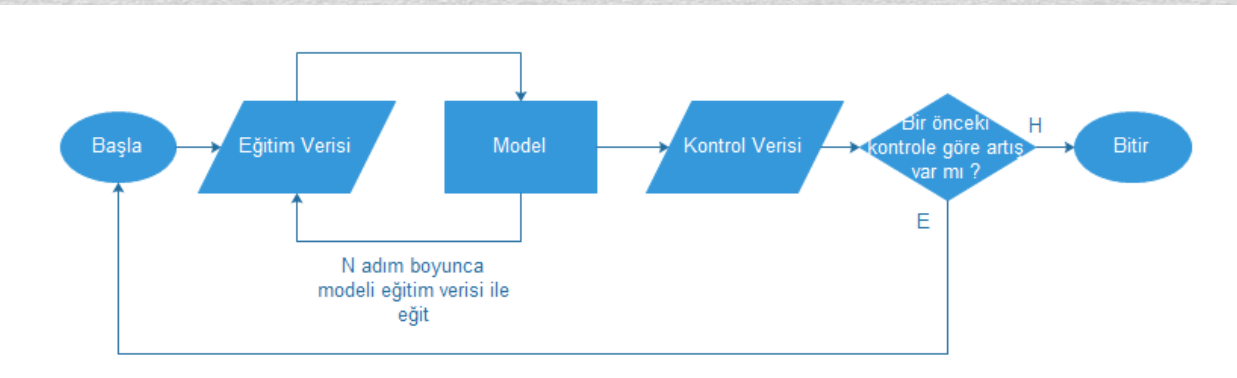
- Non-lineer olarak ayrıştırılabilen verilerin sınıflandırılması için kullanılır.
- Giriş katmanı, Gizli katman/katmanlar ve çıkış katmanından oluşur.
- Backprobagation algoritması – Stochastic gradient descent algoritması ile model eğitilir.
- L1/L2 Regularizasyonu ile model genelleştirilir.

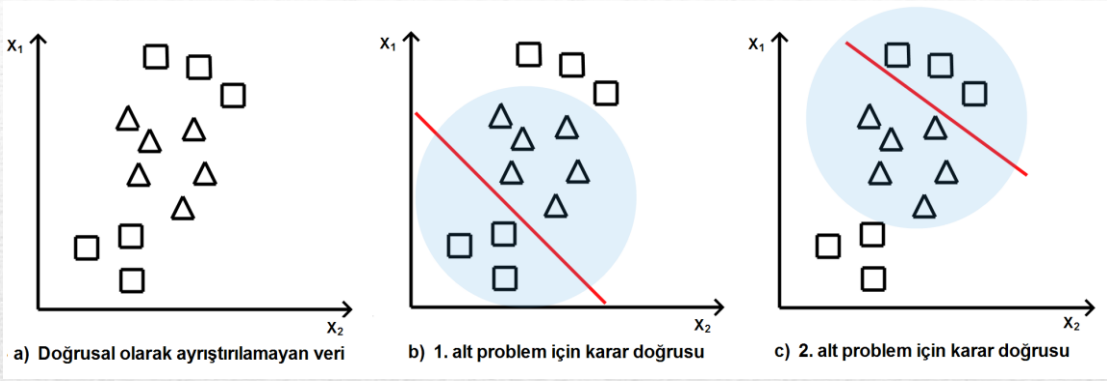
$$E_L(\theta) = J(\theta) + \lambda R(\theta)$$

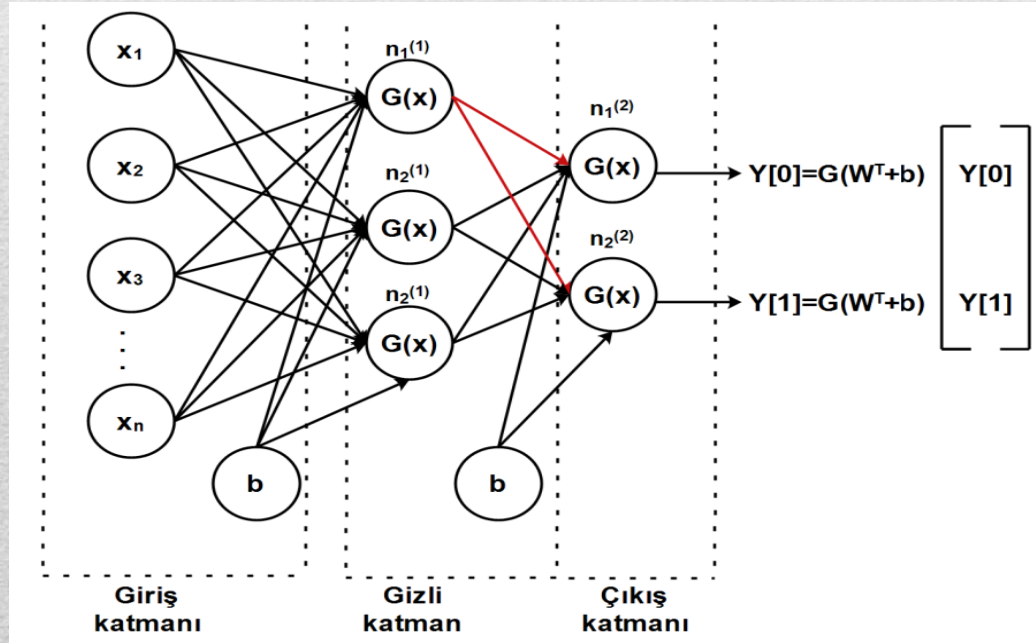
$$R(\theta) = \|\theta\|_p$$

$$\|\theta\|_p = \sum_{i=1}^{\theta} \left(|\theta_i|^p \right)^{\frac{1}{p}}$$

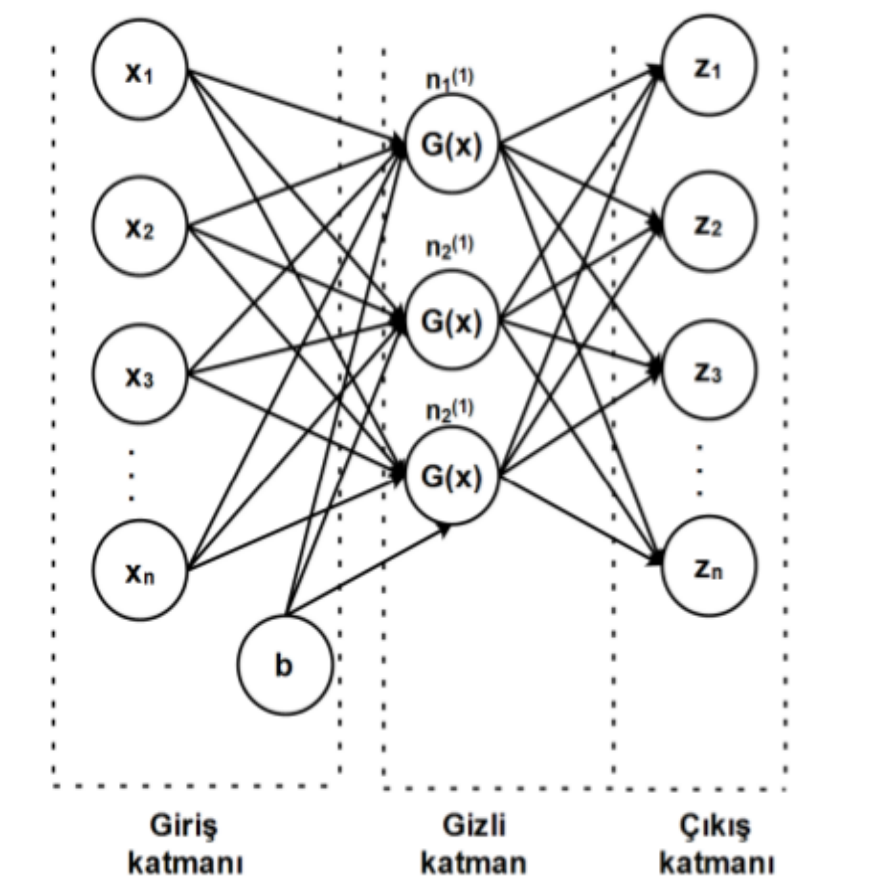
- Early- Stopping







Otomatik Kodlayıcı



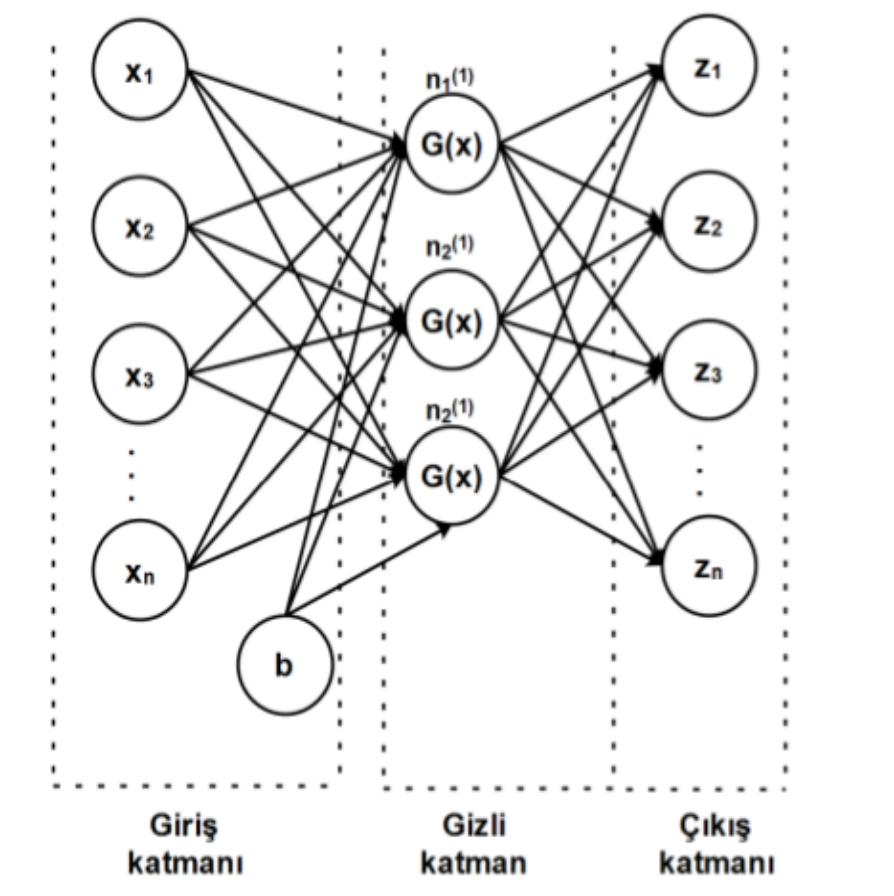
- Özellik boyut daraltımı
- Veri sıkıştırma
- Resim arama (Google)
- Anomali Tespiti
- Yapay sinir ağları için ön eğitim
- Deterministik yapıdadır.

$$s(Wx + b) = y \quad (1)$$

$$s(W'y + b') = z \quad (2)$$

$$HKT = \sum_{i=1}^{n} (z_i - x_i)^2 \quad (3)$$

Gürültü Giderici Otomatik Kodlayıcı



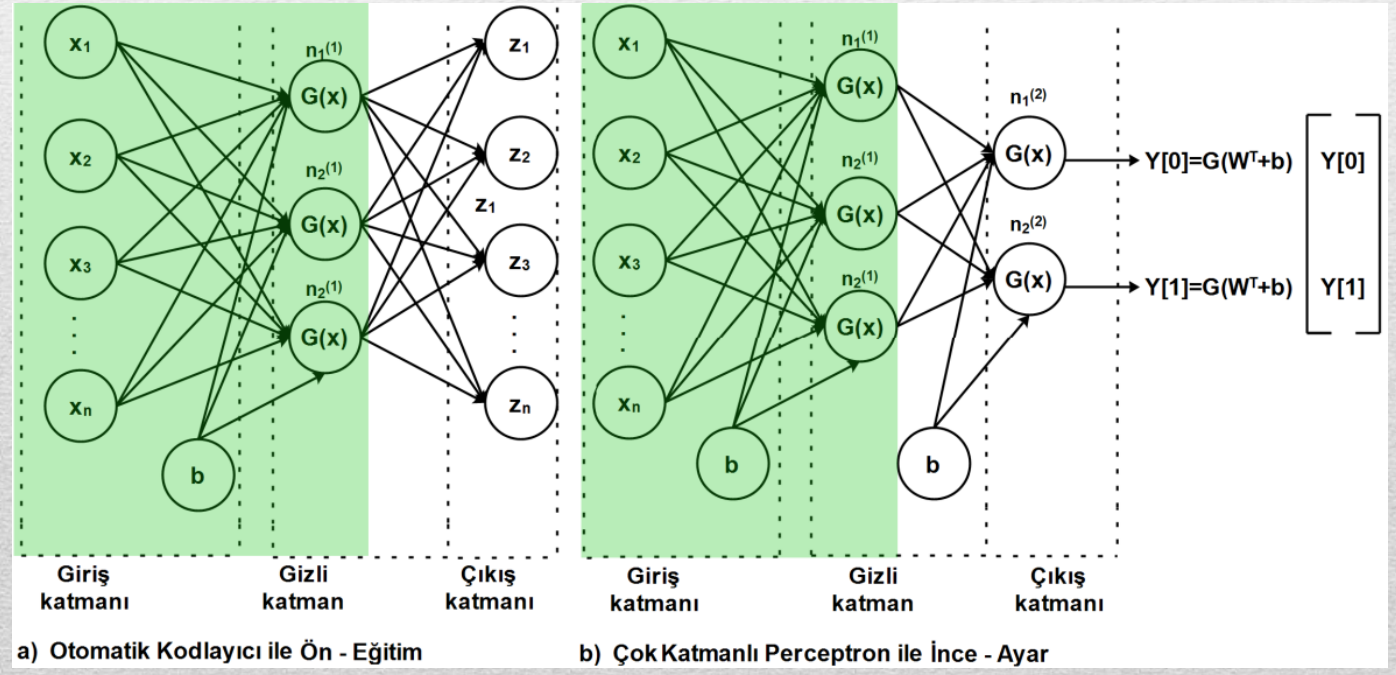
- Giriş verisinin belirli bir yüzdesi rastgele sıfırlanır.
- Model çıkış katmanında giriş verisini bozulmuş olan veriyi kullanarak tahmin etmeye çalışır
- Stokastik bir otomatik kodlayıcıdır.
- Modelin genelleştirilmesini sağlar.

Derin Yapay Sinir Ağı (DSA)

- Gözetimsiz eğitime dayalı otomatik kodlayıcılar
- Gözetimli eğitime dayalı çok katmanlı perceptron

Yapılarının birleşiminden oluşur.

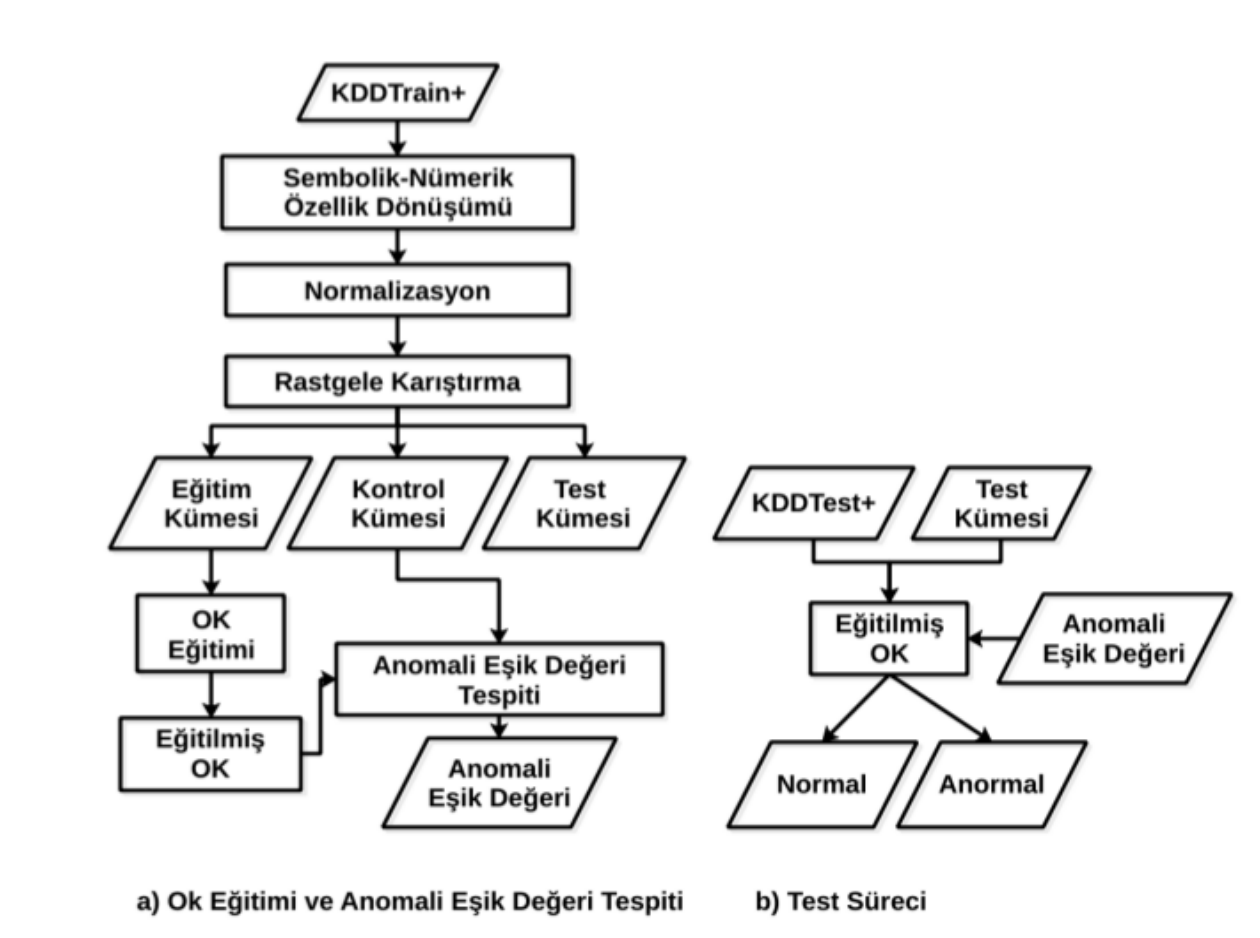
- Ağırlıkların uygun değerler ile ilklendirilmesi
- Önemli özellikleri seçilmesi ve verinin temsilinin çıkartılması

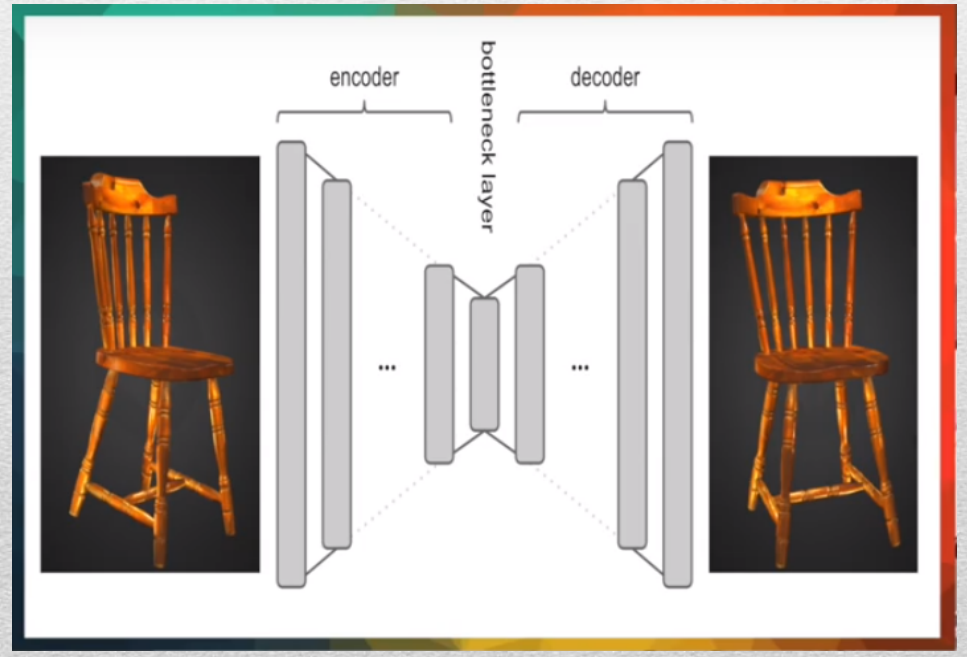


DSA'ya dayalı iki ayrı model oluşturulmuştur:

- OK-DSA
- GGOK-DSA

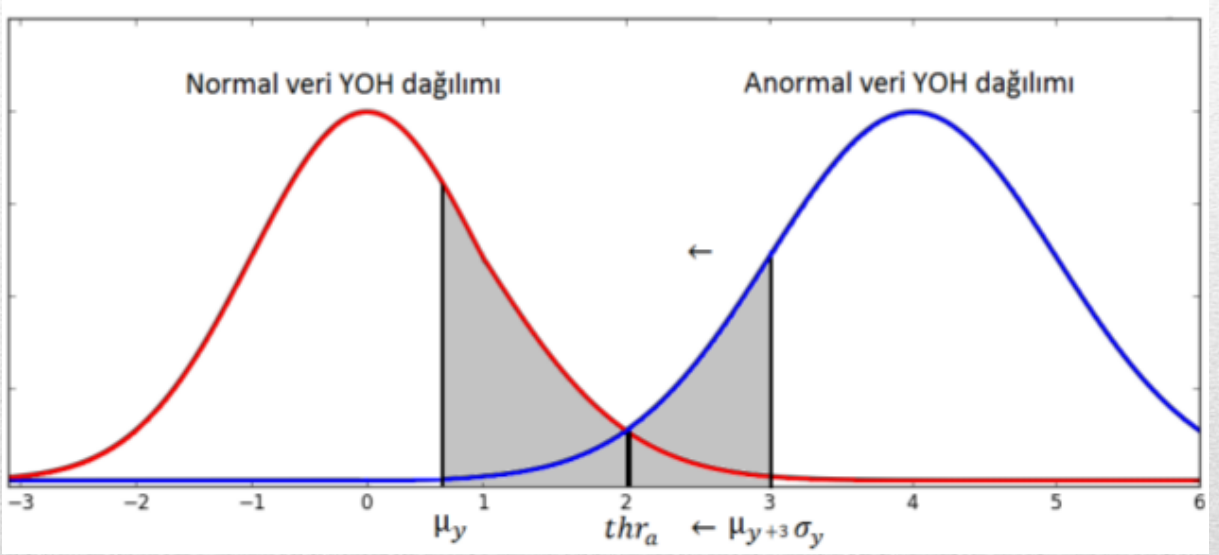
Otomatik Kodlayıcı Tabanlı Anomali Tespiti





Önerilen Stokastik Anomali Eşik Değeri Tespit Yöntemi

▷ ts(training set), vs(validation set), ae(autoencoder), re(reconstruction error), acc(accuracy)
procedure FINDTHRESHOLDVALUE(ts, vs, ae)
    train ae with ts
    calculate μ_y and σ_y values of re by using ts
    $thr_a \leftarrow thr_a + 3\mu_y$
    while $thr_a > \mu_y$ **do**
        calculate acc of the ae on vs by using thr_a
        add (thr_a - acc) key - value pair to $resultList$
        $thr_a \leftarrow thr_a - 0.01$
    end while
    select the thr with the best accuracy from $resultList$.
    $thr_a \leftarrow thr$
    return thr_a
end procedure



Önerilen eşik değeri belirleme yöntemini kullanan otomatik kodlayıcı tabanlı iki ayrı model oluşturulmuştur;

- OK-SAEDT
- GGOK-SAEDT

Performans Değerlendirmesi – Kullanılan Kütüphaneler

- Python dili
- Theano Kütüphanesi
  - Matematiksel fonksiyonların sembolik graflar halinde ifade edilmesini sağlar.
  - Sembolik graflar üzerinde optimizasyon yapar.
  - Sembolik graflar üzerinden türev alınmasını sağlar. Bu sayede manuel türev hesaplamalarında meydana gelen hataların önüne geçilmiş olunur.
  - Matris hesaplamaları için CUDA ve numPy kütüphanelerini kullanır.
  - Otomatik CUDA kodu üretir.
  - Theano kütüphanesi ile gerçeklenen programların numpy, scipy ve C++ dillerine göre CPU üzerinde 6,5 kat ve GPU üzerinde de 44 kat daha hızlı çalıştıkları tespit edilmiştir.
- CUDA Kütüphanesi
  - Çok boyutlu matrislerin GPU hafızasında saklanarak GPU üzerinde matris işlemleri yapılabilmesine olanak sağlayan bir kütüphanedir.

Performans Değerlendirmesi – Kullanılan Donanım

Grafik işlemci destekli hesaplama sunucusu

CPU	İ5-3470 @3.2GHZ x 4
RAM	16 GB
GPU	GTX TITAN X 12GB 384 bit, 3072 CUDA Cores
OS	Ubuntu 16.04 LTS

Performans Değerlendirmesi – Ön Hazırlık

- NSL-KDD'deki 41 özellikten alfanümerik olan üç adet özellik; *protocol-type, service* ve *flag*, 1-n kodlama yöntemiyle 84 ayrı ikili özelliğe dönüştürülmüştür.
- Örn: $Veri_1$, $Veri_2$, $Veri_3$ için F_x alfanümerik özelliğinin 1-n kodlama yöntemi ile nümerik değerlere dönüştürülmesi:

	F_x	v_1	v_2	v_3
$Veri_1$	v_1	1	0	0
$Veri_2$	v_2	0	1	0
$Veri_3$	v_3	0	0	1
.				
.				
$Veri_n$	v_1	1	0	0

- Tüm veriler için sıfır değerine sahip olan *num-outbound-cmds* isimli özellik silinmiştir.
- Toplam özellik sayısı 121'e çıkarılmıştır.

Performans Değerlendirmesi

- *Eğitim, Kontrol* ve *Test* veri kümeleri oluşturulmuştur.

	Training Set	Validation Set	Test Set
Normal	10086	1681	1681
Anomaly	-	1681	1681

- Maksimum-Minimum Normalizasyonu ile veri kümesindeki tüm veriler [0-1] aralığına ölçeklenmiştir.

$$x' = \frac{x - min\_deger}{max\_deger - min\_deger}$$

Performans Değerlendirme Ölçütleri

- Doğruluk, kesinlik, Duyarlılık, F-ölçütü (Accuracy, Precision, Recall, F-Measure)
  - **Doğru Pozitif:** Gerçekte anormal olan bir veri model tarafından anormal olarak sınıflandırılıyorsa **TP** olarak kabul edilir.
  - **Yanlış Pozitif:** Gerçekte normal olan bir veri model tarafından anormal olarak sınıflandırılıyorsa **FP** olarak kabul edilir.
  - **Doğru Negatif:** Gerçekte normal olan bir veri model tarafından normal olarak sınıflandırılıyorsa **TN** olarak kabul edilir.
  - **Yanlış Negatif:** Gerçekte anormal olan bir veri model tarafından normal olarak sınıflandırılıyorsa **FN** olarak kabul edilir.

$Doğruluk = \frac{TP + TN}{NormalÖrnekSayısı + AnormalÖrnekSayısı}$	$Kesinlik = \frac{TP}{TP + FP}$
$F - Ölçütü = \frac{2 * Doğruluk * Kesinlik}{(Doğruluk + Kesinlik)}$	$Duyarlılık = \frac{TP}{TP + FN}$

Test Konfigürasyonları

OK-DSA :

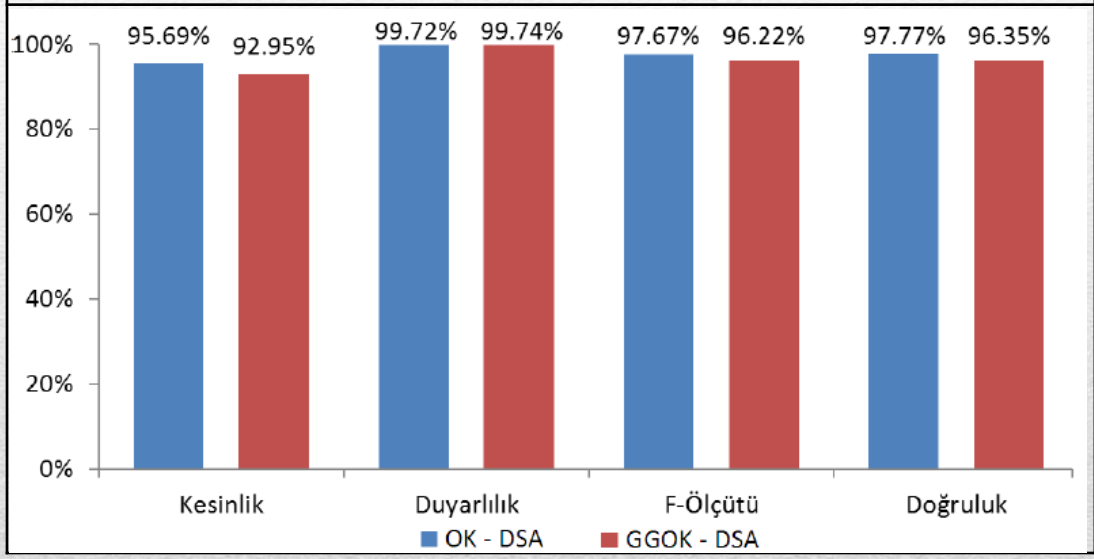
- 121 nöronluk giriş katmanı, sırası ile 60, 30 ve 15 nöron içerecek şekilde üç gizli katman ve iki nöronluk bir çıkış katmanından oluşmaktadır.
- Aktivasyon fonksiyonu olarak, gizli katmanlarda sigmoid, çıkış katmanında ise softmax fonksiyonu kullanılmıştır.
- Ön-eğitim aşamasında 200 tur boyunca gözetimsiz olarak eğitilmiştir.
- İnce-ayar aşamasında 3000 tur boyunca gözetimli olarak eğitilmiştir.
- Ön-eğitim ve ince-ayar aşaması için $\alpha=0.1$ olarak belirlenmiştir.

GGOK-DSA :

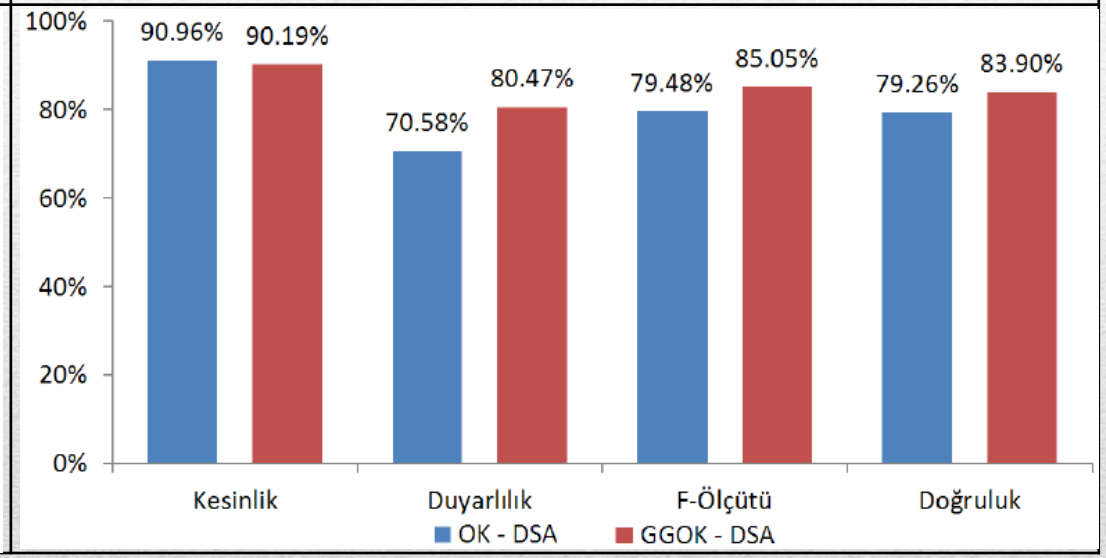
- 121 nöronluk giriş katmanı, her biri 150 nörondan oluşan üç gizli katman ve iki nöronluk bir çıkış katmanından oluşmaktadır.
- Aktivasyon fonksiyonu olarak, gizli katmanlarda sigmoid, çıkış katmanında ise softmax fonksiyonu kullanılmıştır.
- Ön-eğitim aşamasında 200 tur boyunca gözetimsiz olarak eğitilmiştir.
- İnce-ayar aşamasında 3000 tur boyunca gözetimli olarak eğitilmiştir.
- Ön-eğitim ve ince-ayar aşaması için $\alpha=0.1$ olarak belirlenmiştir.
- Bozulma oranı 1. ,2. ve 3. katmanları içi sırası ile %10,%20 ve %30 olarak belirlenmiştir.

Test Sonuçları OK-DSA, GGOK-DSA

KDDTrain+



KDDTest+



Yöntem	Türü	Doğruluk Oranı
MLP[16]	Tekil	% 77,41

Test Konfigürasyonları

OK - SAEDT:

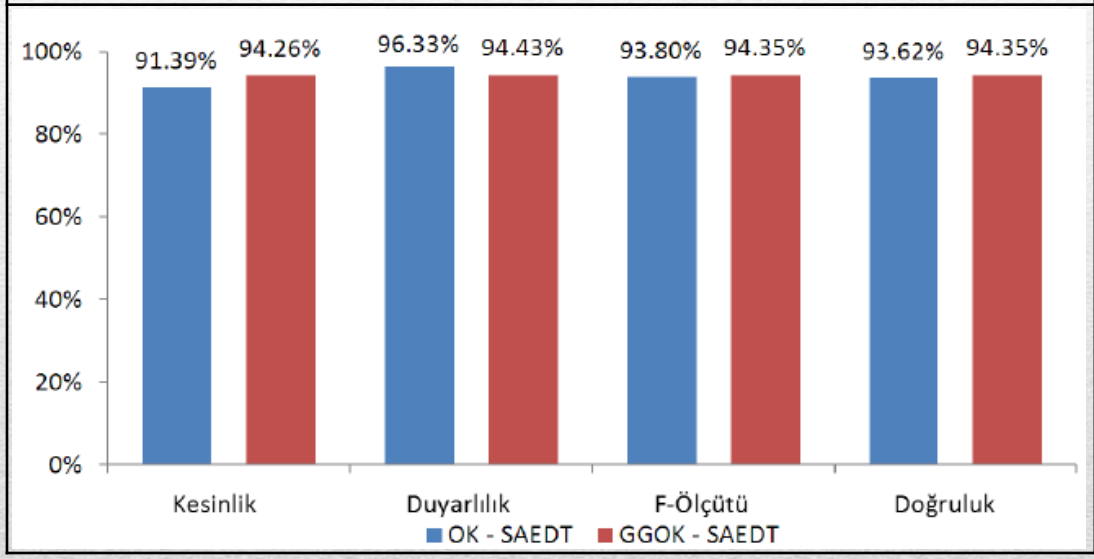
- 121 nöronluk giriş katmanı, 30 nöronluk bir gizli katman ve 121 nöronluk bir çıkış katmanından oluşmaktadır.
- Aktivasyon fonksiyonu olarak, gizli katmanlarda sigmoid fonksiyonu kullanılmıştır.
- YOH değerini minimuma indirgemek için 3000 tur boyunca yarı gözetimli olarak eğitilmiştir.
- α=0.01 olarak belirlenmiştir.

GGOK- SAEDT :

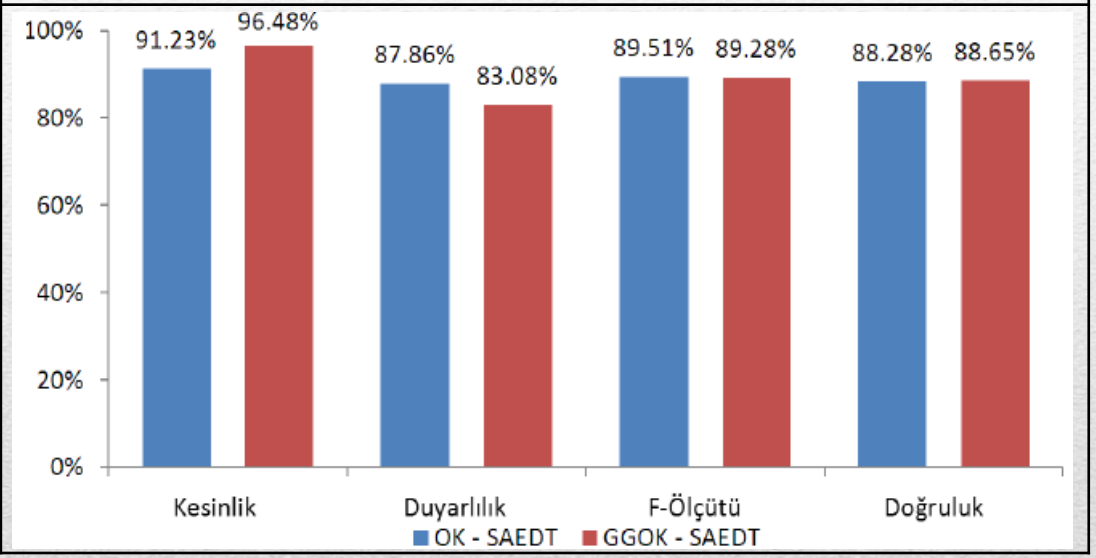
- 121 nöronluk giriş katmanı, 60 nöronluk bir gizli katman ve 121 nöronluk bir çıkış katmanından oluşmaktadır.
- Aktivasyon fonksiyonu olarak, gizli katmanlarda sigmoid fonksiyonu kullanılmıştır.
- YOH değerini minimuma indirgemek için 2500 tur boyunca yarı gözetimli olarak eğitilmiştir.
- Bozulma oranı %10 ve α=0.01 olarak belirlenmiştir.

Test Sonuçları OK-SAEDT, GGOK-SAEDT

KDDTrain+



KDDTest+



KDDTest+ Üzerinde Literatürdeki En Başarılı Diğer Çalışmalar ile Karşılaştırmalı Sonuçlar

Yöntem	Türü	Doğruluk Oranı
MLP[16]	Tekil	% 77,41
OK-DSA(Gerçekelenen Yöntem)	Hibrit	% 79,25
NB Tree[16]	Tekil	% 82,02
Fuzzy Classifier[19]	Tekil	% 82,74
GGOK-DSA(Gerçekelenen Yöntem)	Hibrit	% 83,39
SAE- MLP[24]	Hibrit	% 88,39
OK-SAEDT (Önerilen Yöntem)	Tekil	% 88,28
Random Tree[20]	Tekil	% 88,46
GGOK-SAEDT (Önerilen Yöntem)	Tekil	% 88,65
Random Tree - NB Tree[20]	Hibrit	% 89,24

Teşekkürler.

Referanslar

[1] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, “Intrusion detection by machine learning: A review,” Expert Systems with Applications, vol. 36, no.10, pp. 11994–12000, 2009.

[2]K. Leung and C. Leckie, “Unsupervised anomaly detection in network intrusion detection using clusters,” Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38, pp. 333–342, 2005.

[3]M. Ahmed, A. N. Mahmood, and J. Hu, “A survey of network anomalydetection techniques,” Journal of Network and Computer Applications,
vol. 60, pp. 19–31, 2016.

[4]P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, “A Survey of Outlier Detection Methods in Network Anomaly Identification," The Computer Journal, vol. 54, no. 4, pp. 570–588, 2011.

[5]Dau, Hoang Anh, Ciesielski, Vic, Song, Andy, “Anomaly DetectionUsing Replicator Neural Networks Trained on Examples of One Class,Simulated Evolution and Learning: 10th International Conference, SEAL 2014, Dunedin, New Zealand, Springer International Publishing, pp311—322, 2014.

[6]Li Deng, “A tutorial survey of architectures, algorithms, and applications for deep learning," APSIPA Transactions on Signal and Information Processing, 2014.

[7]Y. Bengio, “Learning Deep Architectures for AI,” Foundations and Trends in Machine Learning, vol. 2, no. 1, pp. 1-127, 2009.

[8]M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. 2nd IEEE International Conference on Computational Intelligence for Security and Defense Applications, USA: IEEE Press, pp. 53–58, 2009.

[9]unb.ca, “UNB ISCX NSL-KDD DataSet", [Online], Available: http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html, [Acces-ed: 25-11-2016]

[10] J. Kevric, S. Jukic, and A. Subasi, “An effective combining classifier approach using tree algorithms for network intrusion detection," Neural Computing and Applications, pp. 1-8, 2016.

[11]P. Kromer, J. Platos, V. Snasel, and A. Abraham, “Fuzzy classification by evolutionary algorithms,” in IEEE International Conference on Systems, Man, and Cybernetics, IEEE System, Man, and Cybernetics Society, pp. 313–18, 2011.

[12]S. Hawkins, H. X. He, G. J. Williams, and R. A. Baxter, “Outlier detection using replicator neural networks", In Proc. of the Fifth Int. Conf. and Data Warehousing and Knowledge Discovery (DaWaK02), 2002.

[13]Nicolau, Miguel, James McDermott. “A Hybrid Autoencoder and Density Estimation Model for Anomaly Detection," International Conferenceon Parallel Problem Solving from Nature. Springer International Publis-hing, 2016.

[14]M. Sakurada and T. Yairi, “Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction," Proc. of the 2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA), pp. 4–11, 2014.

[15]Javaid, Ahmad, et al. “A Deep Learning Approach for Network Intrusion Detection System," Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), New York, NY, USA. Vol. 35., 2015.