



**KARADENİZ TEKNİK ÜNİVERSİTESİ OF TEKNOLOJİ FAKÜLTESİ**  
**YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**

SEZGİSEL YAPAY SİNİR AĞI İLE SEZGİSEL K-EN YAKIN KOMŞULUK  
ALGORİTMASININ SINIFLANDIRMA PERFORMANSLARININ DEĞERLENDİRİLMESİ

## 1. Giriş

Sınıflandırma, veri madenciliğinin önemli bir parçasıdır. Sınıflandırma kavramı, basitçe bir veri kümesi üzerinde tanımlı olan çeşitli sınıflar arasında veriyi dağıtmaktır. Sınıflandırma algoritmaları, verilen eğitim kümesinden bu dağılım şeklini öğrenirler ve daha sonra sınıfının belirli olmadığı test verileri geldiğinde doğru şekilde sınıflandırmaya çalışırlar. Veri kümesi üzerinde verilen bu sınıfları belirten değerlere etiket ismi verilir ve gerek eğitim gerekse test sırasında verinin sınıfının belirlenmesi için kullanılırlar.

Sınıflandırma alanında birçok yöntem ve algoritma geliştirilmiştir. Bu yöntemler arasında en yaygın kullanılan algoritmalarından bir tanesi k- en yakın komşu (k-nn) algoritmasıdır. Genel olarak k-nn, sınıflandırma sırasında çıkarılan özelliklerden, sınıflandırılmak istenen yeni bireyin daha önceki bireylerden k tanesine olan yakınlığına bakılmasıdır.

Bir diğer yöntem olan yapay sinir ağları da sıklıkla sınıflandırma problemlerinde kullanılmaktadır. Yapay sinir ağları insan beyninden esinlenerek insanda bulunan yeni bilgiler öğrenme, yeni çıkarımlar yapma gibi yetenekleri otomatik olarak gerçekleştiren bir yöntemdir ve sınıflandırma, modelleme ve tahmin gibi birçok günlük hayat probleminin çözümünde kabul edilebilir başarılar gösterebilmektedir.

K-NN algoritmasında komşulara olan uzaklık hesaplamasında bağımsız değişkenlerin etki oranının daima 1 alınması ve YSA ağırlıkların bulunması için kullanılan geleneksel algoritmalar, yerel optimum tuzaklarına takılabilmektedirler. Bu nedenle yüksek doğruluk oranı gösteremeyebilirler. YSA ve K-NN de en iyi değerlerinin bulunması bir optimizasyon problemidir. Literatürde optimizasyon alanında sıklıkla kullanılan ve özellikle karmaşık problemlerde kabul edilebilir çözümler bulabilen meta sezgisel algoritmalar ile YSA ve K-NN modellerinde ağırlıkların en uygun değerlerinin bulunmasında kullanılabilir. Bu çalışmada Sezgisel Algoritma olarak SOS (Symbiosis Organisms Search) algoritması kullanılmıştır.

### 1.1 Sezgisel K- En Yakın Komşuluk Algoritması

Sezgisel k-nn, k- en yakın komşu algoritması ve sezgisel arama algoritmalarının teknikleri ile hibrit bir yapı oluşturularak geliştirilmiş bir yöntemdir. K-nn algoritmasında sınıflandırma yapılırken eldeki verilerin birbirlerine olan uzaklığı veya benzerliği kullanılır. Veriler arasındaki mesafe ölçülürken kullanılan yöntemler Öklid, Manhattan ve Minkovski uzaklık metriklerinden biri olabilir. Klasik K-nn algoritmasında uzaklık hesaplaması yapılırken kullanılan niteliklerin her biri çıktılar üzerinde aynı etkiye sahiptir. Ancak bu çıktılar üzerinde etkisi aynı olan niteliklerin ayrı bir ağırlığı olabilir. Niteliklerin farklı etkilerinin olması durumunda çözüm olarak bu niteliklere etkilerine göre bir ağırlık değeri atanır. Doğru ağırlıkların bulunması ise bir optimizasyon problemidir dolayısıyla bu noktada devreye meta-sezgisel arama algoritmaları girmektedir. Sezgisel algoritma bu niteliklerin ağırlık değerleri için optimum değerleri bulacak olan yöntemdir. Optimizasyon bakış açısıyla k-en yakın algoritmasındaki ilgili uzaklık metriği amaç fonksiyonuna tekabül etmektedir. Uzaklık hesabı yapılarak k-nn algoritmasından geriye dönecek hata değeri minimize edilecek değerdir. Çözüm adaylarının boyutu tasarım değişkenlerinin sayısı kadar olurken, çözüm adaylarındaki her bir parametre etki oranı yani ağırlık değeridir. Arama uzayındaki ağırlık değerleri 0-1 arasındadır. 1 etki oranı %100 iken 0'da etki oranı %0'dır.

Aşağıda klasik k-en yakın komşu algoritmasının ve sezgisel k-en yakın komşu algoritmasının uzaklık hesabına ait denklemler verilmektedir. Klasik K-nn algoritmasında uzaklık hesaplaması yapılırken kullanılan niteliklerin her biri çıktılar üzerinde aynı etkiye sahiptir. Klasik k- en yakın komşu algoritmasına ait denklemler:

$$\text{Öklid Uzaklığı : } d(q_x, q_y) = \sqrt{\sum_{i=1}^n (Fq_{x[i]} - Fq_{y[i]})^2} \quad (1)$$

$$\text{Manhattan Uzaklığı : } d(q_x, q_y) = \sum_{i=1}^n (|Fq_{x[i]} - Fq_{y[i]}|) \quad (2)$$

Çıktılar üzerinde farklı etkisi olan niteliklere ayrı bir etki oranı/ağırlık verildiğinde Sezgisel k-en yakın komşu algoritmasına ait uzaklık metrikleri:

$$\text{Öklid Uzaklığı : } d(q_x, q_y) = \sqrt{\sum_{i=1}^n W_{FT[i]} * (Fq_{x[i]} - Fq_{y[i]})^2} \quad (3)$$

$$\text{Manhattan Uzaklığı : } d(q_x, q_y) = \sum_{i=1}^n W_{FT[i]} * (|Fq_{x[i]} - Fq_{y[i]}|) \quad (4)$$

Bulunan optimum değerlerin ağırlık matrisi:

$$W_{FT[m]} = \begin{bmatrix} W_{FT[0,0]} & \cdots & W_{FT[0,m-1]} \\ \vdots & \ddots & \vdots \\ W_{FT[n-1,0]} & \cdots & W_{FT[n-1,m-1]} \end{bmatrix} \quad (5)$$

Yapılan çalışmada KNN için Öklid ve Manhattan uzaklık formülleri kullanılmıştır. Oylama yöntemleri olarak klasik ve ağırlıklı oylama kullanılmıştır. En uygun komşu sayısının belirlenmesinde 1'den 30'a kadar ilgili veri setinde sınıflandırma yapılarak hata oranı en düşük komşu belirlenmiştir.

## 1.2 Sezgisel Yapay Sinir Ağı

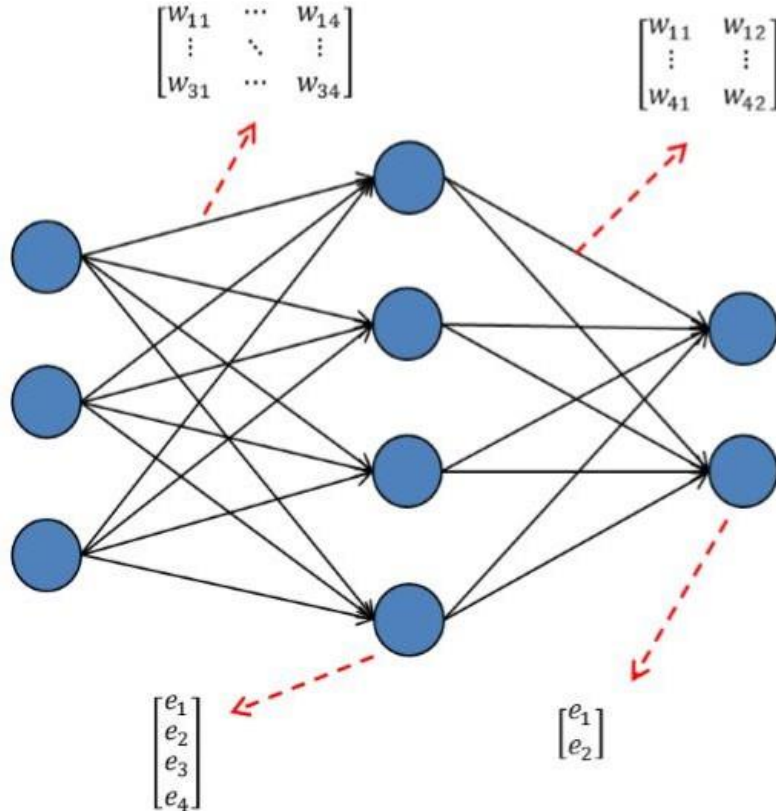
Sezgisel YSA, yapay sinir ağları ve sezgisel arama algoritmalarının teknikleri ile hibrit bir yapı oluşturularak geliştirilmiş bir yöntemdir. Yapay sinir ağlarında eğitim, ilgili YSA'nın nöronları arasındaki ağırlıkların değiştirilmesiyle gerçekleştirilmektedir. Bu ağırlık değerleri değiştikçe bilgilerin ağ içerisindeki dağılımları, dolayısıyla çıktı değerleri değişmektedir. Ağırlıkların sürekli yenilenip istenilen sonuca ulaşana kadar geçen zamana öğrenme denir. Eğitim sürecinin ne zaman sonlandırılacağına karar vermek ezberleme probleminin önüne geçmek ve en etkili modeli yaratmak için önemlidir. Ağın eğitimini sonlandırmaya karar vermek için doğrulama veri

setindeki hata değişimi dikkate alınır. Ağırlıkların hatayı en küçükleyecek şekilde değiştirilmesiyle sınıflandırma işlemini gerçekleştirmektedir. Meta-sezgisel algoritmalarda popülasyondaki her bir dizi, YSA ağırlıklarını ifade etmektedir. Ağırlık değerleri matrisler aracılığıyla tutulmaktadır ve matris boyutları Tablo 1’de verilmiştir.

Tablo 1. YSA eğitimi için tutulan ağırlık matrisleri

Katmanlar	Boyut
Girdi katmanı ile gizli katman arası	Girdi sayısı $\times$ gizli katmandaki nöron sayısı
Gizli katman ile çıktı katmanı arası	Gizli katmandaki nöron sayısı $\times$ çıktı sayısı
Gizli katman eşik değerleri	Gizli katman nöron sayısı
Çıktı katmanı eşik değerleri	Çıktı sayısı

Şekil 1’de 3 girdi 2 çıktılı bir problem için gizli katmanında 4 nöron içeren YSA yapısında ağırlık matrisleri gösterilmektedir. Şekilden de görüldüğü gibi girdi-gizli katman ağırlıklarının tutulduğu matris  $3 \times 4$  boyutunda, gizli-çıktı ağırlıklarının tutulduğu matris  $4 \times 2$  boyutunda, gizli katman eşik ağırlıklarının tutulduğu matris  $6 \times 1$  boyutunda ve çıktı eşik ağırlıklarının tutulduğu matris  $2 \times 1$  boyutundadır.



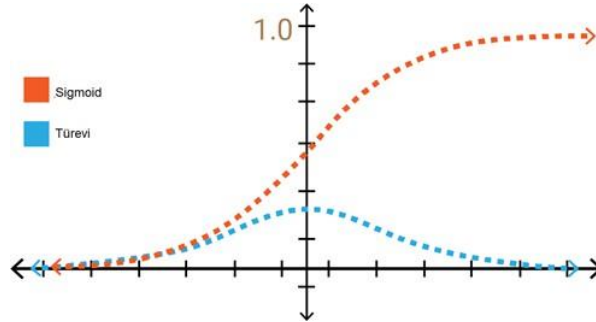
Şekil 1. YSA eğitimi için tutulan ağırlık değerleri matrisleri

## KULLANILAN AKTİVASYON FONKSİYONLARI

Yapılan çalışmada YSA için Sigmoid, Hiperbolik, Sinüs, Esik Değer, Relu ve Leaky Relu aktivasyon fonksiyonları kullanılmıştır. Sinir hücrelerinin birleştirme fonksiyonlarında ise Toplam, Çarpım, Min ve Max fonksiyonları kullanılmıştır. Sinir hücrelerinin ağırlıklarının yanında en uygun birleştirme ve aktivasyonlarının belirlenmesinde sezgisel algoritma kullanılmıştır.

Yapay sinir ağlarında sinir düğümlerinin çıktısını almakta kullanılan fonksiyona aktivasyon fonksiyonu denir. Transfer fonksiyonu olarak da bilinir. Yapay sinir ağlarına doğrusal olmayan gerçek dünya özelliklerini tanıtmak için aktivasyon fonksiyonuna ihtiyaç duyulur. Yapay sinir ağında x girdiler, w ağırlıklar ve ağın çıkışına aktarılan değere  $f(x)$  yani aktivasyon işlemi uygulanır. Aktivasyon fonksiyonu kullanılmayan bir sinir ağı sınırlı öğrenme gücüne sahip bir doğrusal bağlanım (linear regression) gibi davranacaktır. Sinir ağına öğrenmesi için görüntü, video, yazı ve ses gibi karmaşık gerçek dünya bilgileri verileceğinden doğrusal olmayan aktivasyon fonksiyonlarına ihtiyaç vardır ve bu sayede ağların daha güçlü öğrenmesi sağlanabilir. Ağırlıklar ile ilgili hata değerlerini hesaplamak için yapay sinir ağında hatanın geriye yayılımı algoritması uygulanmaktadır.

### Sigmoid Fonksiyonu

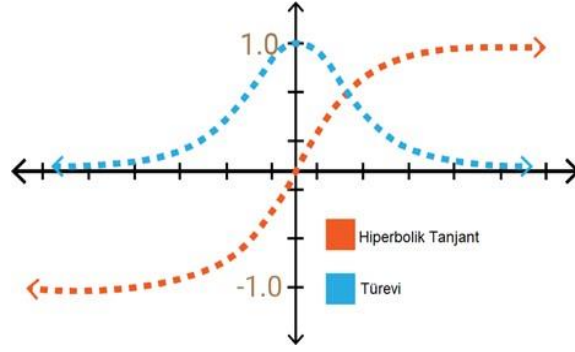


Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur Doğrusal olmayışı dolayısıyla yapay sinir ağları uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir. Katmanları sıralanabilir, basamak fonksiyonundan farklı olduğu için türevlenebilir. Yavaş bir öğrenme olayı gerçekleştiğinde hatayı minimize eden optimizasyon algoritması yerel (lokal) minimum değerlere takılabilir ve yapay sinir ağı modelinden alınabilecek maksimum performans alınamaz.

**Formül :**

$$F(Net) = \frac{1}{1 + e^{-Net}} \quad (6)$$

### Hiperbolik Tanjant Fonksiyonu

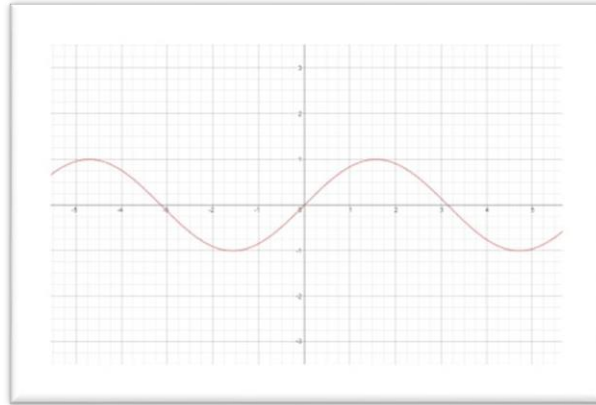


Sigmoid fonksiyonuna çok benzer bir fonksiyondur. Ancak hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişir. Türevinin daha dik olması yani daha çok değer alması sigmoid fonksiyonuna göre avantajdır. Bu da daha hızlı öğrenme ve sınıflama işlemi için daha geniş aralığa sahip olmasından dolayı daha verimli olacağı anlamına gelir. fonksiyonun uçlarında gradyanların ölmesi problemi sebebiyle maksimum performans alınamaz.

**Formül :**

$$F(Net) = \frac{e^{Net} + e^{-Net}}{e^{Net} - e^{-Net}} \quad (7)$$

### Sinüs Fonksiyonu



Öğrenilmesi düşünülen olayların sinüs fonksiyonuna uygun değer gösterdiği durumlarda kullanılır.

**Formül :**

$$F(Net) = \sin(Net) \quad (8)$$

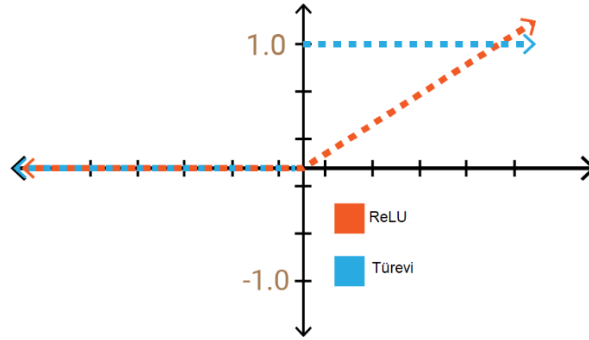
### Eşik Değer Fonksiyonu

Gelen bilgiler sıfırdan küçük eşit olduğunda 0 çıktısı, 1' den büyük eşit olduğunda 1 çıktısı, 0 ile 1 arasında olduğunda ise yine kendisini veren çıktılar üretebilir.

**Formül :**

$$F(Net) = \begin{cases} 0, & \text{if } Net \leq 0 \\ Net, & \text{if } 0 < Net < 1 \\ 1, & \text{if } Net \geq 1 \end{cases} \quad (9)$$

### ReLU (Rectified Linear Unit) Fonksiyonu

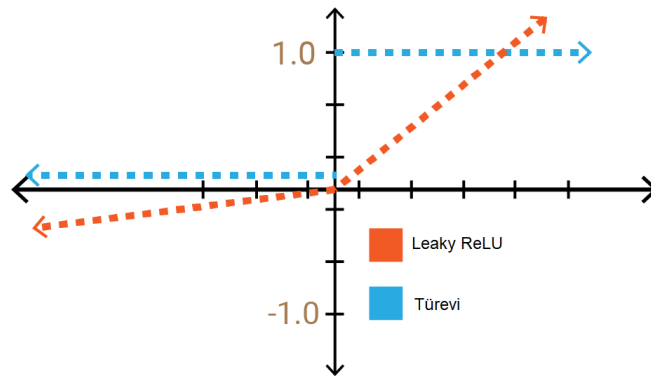


Doğrusal olmayan bir fonksiyondur. İyi bir tahmin edicidir. ReLU'nun kombinasyonları ile herhangi başka bir fonksiyona da yakınsamak mümkündür. ReLU fonksiyonunun hem kendisi hem de türevi monoton fonksiyondur. Herhangi bir negatif değer ReLU tarafından sıfır olarak döndürülür. Negatif ekseninde 0 değerlerini alması ağına daha hızlı çalışacağı anlamına gelirken tüm negatif değerlerin 0'a çevrilmesi modelin uygun bir şekilde eğitilmesini engeller. Bu nedenle model negatif değerler ile eğitilemez.

**Formül :**

$$F(Net) = \max(0, x) \quad (10)$$

### Sızıntı (Leaky) ReLU Fonksiyonu



Leaky ReLU negatif değerlerin ReLU’da yok olmasını engellemek için geliştirilmiştir. Bu aktivasyon fonksiyonunda, ReLU’dan farklı olarak, negatif değerler için aktivasyon fonksiyonuna eğim eklenilmesini sağlayan bir değer seçilir. Seçilecek bu değer, sabit küçük (0.001) bir değer olursa aktivasyon fonksiyonu Leaky-ReLU şeklinde isimlendirilir.

**Formül :**

$$F(Net) = \max(0.001x, x) \quad (11)$$

### **Birleştirme(Toplama) Fonksiyonu**

Toplama fonksiyonu bir yapay sinir hücresine ağırlıklarla çarpılarak gelen girdileri toplayarak o hücrenin net girdisini hesaplayan bir fonksiyondur. Bazı durumlarda gelen girdilerin değeri dikkate alınırken bazı durumlarda ise gelen girdilerin sayısı önemli olabilmektedir. Bir problem için en uygun toplama fonksiyonu belirlenirken geliştirilmiş bir yöntem yoktur. Genellikle deneme yanılma yoluyla toplama fonksiyonu belirlenmektedir. Bazen her hücrenin toplama fonksiyonunun aynı olması gerekmez. Bu konulara karar vermek tasarımcıya aittir.

### **Bazı Toplama Fonksiyonları**

Toplam
$Net = \sum_{i=1}^N X_i * W_i$
Çarpım
$Net = \prod_{i=1}^N X_i * W_i$
Maksimum
$Net = \text{Max}(X_i * W_i)$
Minimum
$Net = \text{Min}(X_i * W_i)$

**Toplam Fonksiyonu :** Ağırlık değerleri girdiler ile çarpılır ve bulunan değerler birbirleriyle toplanarak Net girdi hesaplanır.

**Çarpım Fonksiyonu :** Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleriyle çarpılarak Net Girdi Hesaplanır.

**Maksimum Fonksiyonu :** n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en büyüğü Net girdi olarak kabul edilir.

**Minimum Fonksiyonu :** n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en küçüğü Net girdi olarak kabul edilir.



### 1.3 Yapılan Deneysel Çalışma Ayarları

KNN ve YSA performanslarını değerlendirmek için UCI makine öğrenme deposundan 2 veri seti alınmıştır. Veri setlerinin özellikleri tablo 1’de görülmektedir.

Tablo 2. Veri setlerinin özellikleri						
					Etiketlerin Dağılımı	
Veri Seti No	Veri Seti Adı	Toplam Veri Sayısı	Nitelik Sayısı	Etiket Sayısı	Etiket	Sayı
1	Electrical Grid Stability Simulated	6000	13	2	0	3833
					1	2161
2	Avila	3509	10	6	1	1640
					2	105
					3	294
					4	1019
					5	239
					6	212

1 nolu veri setinin 1-2000 arası YSA ve KNN ağırlıklarının bulunmasında, 2001-4000 arası performans değerlendirmesinde ve 4001-6000 arası KNN’de sınıflandırma yapılmak üzere bölünmüştür.

2 nolu veri setinin 1001-2000 arası YSA ve KNN ağırlıklarının bulunmasında, 2001-3509 arası performans değerlendirmesinde ve 1-1000 arası KNN’de sınıflandırma yapılmak üzere bölünmüştür.

KNN ve YSA ağırlıklarının bulunmasında Simbiyotik Organizmalar Arama Algoritması kullanılmıştır. Maksimum iterasyon sayısı  $(1.000 * \text{problem boyutu}) + 80.000$  olarak belirlenmiştir. Hazırlanan modeller 4 kez çözümlenmiştir ve en düşük hataya sahip model seçilmiştir.

#### 1.3.4 Avila Veri Seti için Yapılan Deneysel Çalışmalar

Baz Knn algoritması için belirlenen 3 komşu sayısı sayısı ve Öklid uzaklığı için sınıflandırma performansı %63.7’dir. Aynı modelin sezgisel algoritma ile geliştirilmiş halinde ise başarı oranı %91.1’e çıkartılmıştır. Sınıflandırma performansını daha da iyileştirebilmek için oylama türü ve uzaklık metrikleri sırasıyla ağırlıklı oylama ve manhattan uzaklığı ile değiştirilmiştir. Veri setindeki bağımsız değişkenlerin sınıflandırmadaki etki oranları tablo 3’ de verilmiştir. Elde edilen başarı oranı Şekil 2’ de verilmiştir.

Tablo:3 Sınıflandırmadaki ağırlık değerleri

Ağırlıklar				
$A_1 = 0.5054$	$A_2 = 0.0740$	$A_3 = 0.0081$	$A_4 = 0.3384$	$A_5 = 0.1306$
$A_6 = 0$	$A_7 = 0$	$A_8 = 0$	$A_9 = 0$	$A_{10} = 0.0083$

Avila Confusion Matrix							
Output Class	1	2	3	4	5	6	
	454 45.4%	0 0.0%	0 0.0%	2 0.2%	0 0.0%	2 0.2%	99.1% 0.9%
	1 0.1%	31 3.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.9% 3.1%
	0 0.0%	0 0.0%	85 8.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	7 0.7%	0 0.0%	1 0.1%	297 29.7%	0 0.0%	1 0.1%	97.1% 2.9%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	64 6.4%	3 0.3%	95.5% 4.5%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	52 5.2%	100% 0.0%
Target Class							
1	2	3	4	5	6		
98.3% 1.7%	100% 0.0%	98.8% 1.2%	99.3% 0.7%	100% 0.0%	89.7% 10.3%	98.3% 1.7%	

Şekil 2. KNN Modelinin Sınıflandırma Performansı

YSA için ise tek çıkışlı yaklaşımında sırasıyla 1. Ve 2. Gizli katmanları 2-6, 3-3, 3-5 ve 5-4 olacak şekilde 4 farklı model oluşturulmuştur. Ağırlıkların bulunmasında en küçük hatayı veren 2-6'lı model olmuştur. 494 hatalı sınıflandırma yapan ysa modeli tablo 4'de verilmiştir. Şekil 3'te ise doğrulama performansı verilmiştir.

Tablo 4. YSA Modeli

Katmanlar	Nöron	Ağırlıklar			Aktivasyon Fonksiyonları	Birleştirme Fonksiyonları
1. Gizli Katman	1	$A_1 = -1$	$A_2 = 0.7115$	$A_3 = 0.6908$	Relu	Toplam
		$A_4 = -0.0150$	$A_5 = 0.1719$	$A_6 = 0.3562$		
		$A_7 = -0.5071$	$A_8 = -0.5257$	$A_9 = 0.6745$		
		$A_{10} = -0.7367$				
	2	$A_1 = 0.4063$	$A_2 = -0.9420$	$A_3 = -0.3405$	Sigmoid	Çarpım
		$A_4 = 0.2612$	$A_5 = 0.8698$	$A_6 = -0.6731$		
		$A_7 = 1$	$A_8 = 0.9769$	$A_9 = -0.9936$		
		$A_{10} = -0.2967$				
2. Gizli Katman	1	$A_1 = 0.9742$	$A_2 = -0.1817$		Sinüs	Toplam
	2	$A_1 = -0.2625$	$A_2 = -0.9720$		Sinüs	Max
	3	$A_1 = -0.0878$	$A_2 = 0.0763$		Relu	Toplam
	4	$A_1 = 0.9997$	$A_2 = 0.4565$		Sinüs	Max
	5	$A_1 = -0.9978$	$A_2 = -0.1167$		Leaky Relu	Min
	6	$A_1 = 0.5574$	$A_2 = 0.4346$		Leaky Relu	Toplam
Çıkış Katman	1	$A_1 = -0.9481$	$A_2 = 0.4646$	$A_3 = 0.8436$	Sinüs	Max
		$A_4 = 1$	$A_5 = -0.9430$	$A_6 = 0.6355$		

Avila Confusion Matrix								
Output Class	1	696 46.1%	19 1.3%	42 2.8%	398 26.4%	63 4.2%	35 2.3%	55.5% 44.5%
	2	18 1.2%	18 1.2%	35 2.3%	23 1.5%	33 2.2%	45 3.0%	10.5% 89.5%
	3	1 0.1%	11 0.7%	43 2.8%	1 0.1%	5 0.3%	12 0.8%	58.9% 41.1%
	4	0 0.0%	0 0.0%	6 0.4%	0 0.0%	2 0.1%	3 0.2%	0.0% 100%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
Target Class								
1	2	3	4	5	6			
97.3% 2.7%	37.5% 62.5%	34.1% 65.9%	0.0% 100%	0.0% 100%	0.0% 100%	50.2% 49.8%		

Şekil 3. Tek Çıkışlı 2-6 YSA Modelinin Sınıflandırma Performansı

Tek çıkışlı modeldeki hata oranını düşürmek için, çıkış katmanındaki nöron sayısı 1 yerine, etiket sayısı kadar yapılmıştır. Bu yaklaşımda sırasıyla 1. Ve 2. Gizli katmanları 2-6, 3-3, 3-5 ve 5-4 olacak şekilde 4 farklı model oluşturulmuştur. Ağırlıkların bulunmasında en küçük hatayı veren 3-3'ü model olmuştur. 394 hatalı sınıflandırma yapan ysa modeli tablo 5'de verilmiştir. Şekil 4'te ise doğrulama performansı verilmiştir.

Tablo 5. YSA Modeli

Katmanlar	Nöron	Ağırlıklar			Aktivasyon Fonksiyonları	Birleştirme Fonksiyonları
1. Gizli Katman	1	$A_1 = -0.6486$	$A_2 = -0.6091$	$A_3 = -0.5010$	Hiperbolik Tanjant	Toplam
		$A_4 = -0.9592$	$A_5 = 0.9996$	$A_6 = -0.9928$		
		$A_7 = -0.5777$	$A_8 = 0.2214$	$A_9 = -0.5626$		
		$A_{10} = 0.5417$				
	2	$A_1 = 0.3963$	$A_2 = 0.8691$	$A_3 = 0.2525$	Sigmoid	Çarpım
		$A_4 = 0.5361$	$A_5 = 0.1367$	$A_6 = -0.7476$		
		$A_7 = -0.7828$	$A_8 = -0.9457$	$A_9 = -0.8437$		
		$A_{10} = 0.7480$				
	3	$A_1 = 0.7983$	$A_2 = -0.1171$	$A_3 = -0.4140$	Sigmoid	Toplam
		$A_4 = -0.2794$	$A_5 = -0.6979$	$A_6 = 0.6022$		
		$A_7 = -0.6254$	$A_8 = 0.1287$	$A_9 = -0.3304$		
		$A_{10} = -0.7812$				
2. Gizli Katman	1	$A_1 = -0.1895$	$A_2 = -0.6979$	$A_3 = 0.9998$	Sinüs	Toplam
	2	$A_1 = 0.9291$	$A_2 = 0.1766$	$A_3 = -0.8691$	Relu	Toplam
	3	$A_1 = 0.4399$	$A_2 = 1$	$A_3 = -0.5450$	Sinüs	Toplam
Çıkış Katman	1	$A_1 = 0.5857$	$A_2 = -0.3979$	$A_3 = -0.7955$	Sigmoid	Toplam
	2	$A_1 = 0.5107$	$A_2 = 0.2086$	$A_3 = -0.7082$	Sinüs	Toplam
	3	$A_1 = -0.9992$	$A_2 = 0.4199$	$A_3 = -0.0927$	Sigmoid	Toplam
	4	$A_1 = 0.1132$	$A_2 = 1$	$A_3 = -0.9956$	Sigmoid	Max
	5	$A_1 = -0.1682$	$A_2 = -0.6699$	$A_3 = 0.4172$	Sigmoid	Max
	6	$A_1 = -0.7640$	$A_2 = -0.9883$	$A_3 = -0.5763$	Sinüs	Toplam

Avila Confusion Matrix								
Output Class	1	555 36.8%	18 1.2%	18 1.2%	201 13.3%	69 4.6%	49 3.2%	61.0% 39.0%
	2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	3	6 0.4%	24 1.6%	86 5.7%	14 0.9%	4 0.3%	6 0.4%	61.4% 38.6%
	4	154 10.2%	6 0.4%	22 1.5%	207 13.7%	30 2.0%	40 2.7%	45.1% 54.9%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
		77.6% 22.4%	0.0% 100%	68.3% 31.7%	49.1% 50.9%	0.0% 100%	0.0% 100%	56.2% 43.8%
		Target Class						

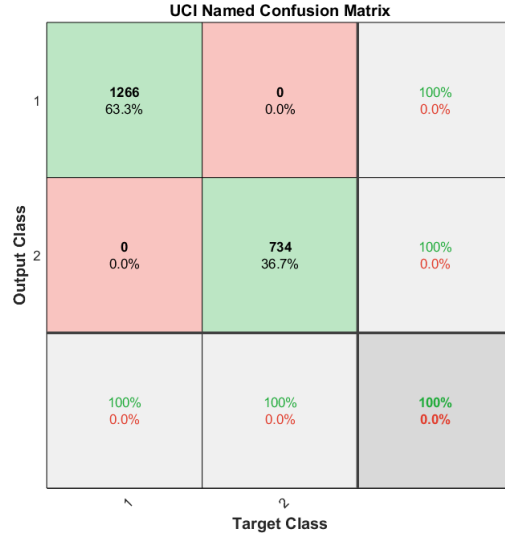
Şekil 4. Multi Çıkışlı 3-3 YSA Modelinin Sınıflandırma Performansı

### 1.3.5 Electrical Grid Stability Simulated Veri Seti için Yapılan Deneysel Çalışmalar

Baz Knn algoritması için belirlenen 7 komşu sayısı sayısı ve Öklid uzaklığı için sınıflandırma performansı %90.1'dir. Aynı modelin sezgisel algoritma ile geliştirilmiş halinde ise başarı oranı %99.9'e çıkartılmıştır. Veri setindeki bağımsız değişkenlerin sınıflandırmadaki etki oranları tablo 6 de verilmiştir. Elde edilen başarı oranı Şekil 5'de verilmiştir.

Tablo 6. KNN Modeli

Ağırlıklar				
$A_1 = 0$	$A_2 = 0$	$A_3 = 0$	$A_4 = 0$	$A_5 = 0$
$A_6 = 0$	$A_7 = 0$	$A_8 = 0$	$A_9 = 0$	$A_{10} = 0.0865$

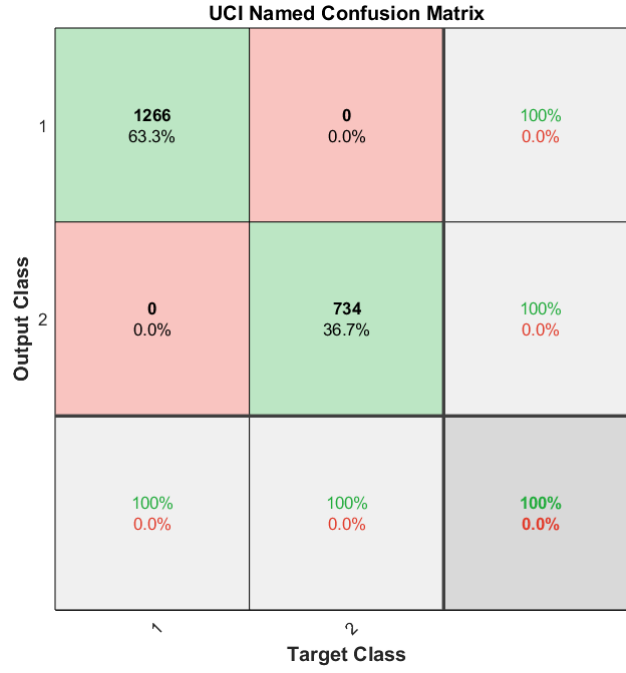


Şekil 5. KNN Modelinin Sınıflandırma Performansı

Bu veri setinde ön çalışma için oluşturulan Ysa modelinde ağırlıkların bulunmasında 100% sınıflandırma başarısı elde edildiğinden dolayı başka modeller oluşturulmamıştır. Oluşturulan model Şekil 6’da verilmiştir.

Tablo 7. YSA Modeli

Katmanlar	Nöron	Ağırlıklar	Aktivasyon Fonksiyonları	Birleştirme Fonksiyonları
1. Gizli Katman	1	$A_1 = -0.4743$	Leaky Relu	Min
		$A_2 = 0.7167$		
		$A_3 = 0.7549$		
		$A_4 = 0.1788$		
		$A_5 = 0.9942$		
	2	$A_6 = -0.1238$	Sinüs	Çarpım
		$A_7 = -0.6329$		
		$A_8 = 0.9706$		
		$A_9 = 0.7560$		
		$A_{10} = -0.1720$		
	3	$A_{11} = -0.2700$	Hiperbolik Tanjant	Çarpım
		$A_{12} = 0.1803$		
		$A_{13} = -0.0390$		
		$A_1 = 0.9583$		
		$A_2 = -0.5886$		
	4	$A_3 = 0.3329$	Eşik Değer	Min
		$A_4 = 0.3451$		
		$A_5 = 0.1357$		
		$A_6 = -0.2106$		
		$A_7 = 0.9174$		
	5	$A_8 = 0.9851$	Leaky Relu	Max
		$A_9 = 0.1708$		
		$A_{10} = 0.1259$		
		$A_{11} = -0.5872$		
		$A_{12} = 0.4105$		
2. Gizli Katman	1	$A_1 = -0.2376$	Sigmoid	Max
		$A_2 = -0.9279$		
	2	$A_3 = -0.1653$	Sinüs	Toplam
		$A_4 = 0.4282$		
	3	$A_5 = 0.5861$	Sinüs	Çarpım
		$A_6 = 0.9561$		
	4	$A_7 = 1.0000$	Sigmoid	Min
		$A_8 = -0.8052$		
Çıkış Katman	1	$A_9 = -0.1032$	Eşik Değer	Toplam
		$A_{10} = 0.0583$		
	2	$A_{11} = 0.4980$	Leaky Relu	Max
		$A_{12} = -0.5675$		
	3	$A_{13} = 0.1785$	Sinüs	Toplam
		$A_1 = 0.1232$		
	4	$A_2 = 1.0000$	Sigmoid	Min
		$A_3 = 0.0506$		
	5	$A_4 = -0.5537$	Leaky Relu	Max
		$A_5 = -0.4689$		
	6	$A_6 = 0.1405$	Sinüs	Toplam
		$A_7 = 0.9488$		
	7	$A_8 = -0.7889$	Sigmoid	Min
		$A_9 = -0.7788$		
	8	$A_{10} = 0.4530$	Leaky Relu	Max
		$A_{11} = -0.0546$		
	9	$A_{12} = 0.8337$	Sinüs	Toplam
		$A_{13} = 0.3423$		
	10	$A_1 = -0.1931$	Sigmoid	Min
		$A_2 = -0.2716$		
	11	$A_3 = 0.0823$	Leaky Relu	Max
		$A_4 = 0.0042$		
	12	$A_5 = -0.3358$	Sinüs	Toplam
		$A_6 = -0.3611$		
	13	$A_7 = 0.3664$	Sigmoid	Min
		$A_8 = -0.9996$		
	14	$A_{10} = -0.4157$	Leaky Relu	Max
		$A_{11} = 0.0213$		
	15	$A_{12} = -0.0135$	Sinüs	Toplam
		$A_{13} = 0.9162$		



Şekil 6. Tek Çıkışlı 5-4 YSA Modelinin Sınıflandırma Performansı

#### 1.4 Sonuç

Electrical Grid Stability Simulated veri setinde Sezgisel KNN VE Sezgisel YSA genel olarak performans açısından benzer sonuçlar vermiştir. Avila veri setinde ise Sezgisel KNN in sınıflandırma performansı %40.1 lik daha fazla doğruluk üstünlüğü göstermiştir. Veri setlerinin homojenlik durumu göz önüne alındığında Avila veri seti yeterince homojen değildir. Homojen olmayan veri setinde Sezgisel YSA'nın doğruluk oranında büyük ölçüde azalma vardır. Ancak Sezgisel KNN aynı durumlar göz önüne alındığında başarı oranını koruyabilmiştir. Veri setinin homojenlik durumunun Sezgisel YSA'nın sınıflandırma performansı üzerindeki olumsuz etkisi bariz görülmektedir. Fakat Sezgisel KNN de böyle bir etki görülmemiştir. Homojen olmayan veri setlerinin ysa sınıflandırma performansı üzerindeki etkisinin daha iyi araştırılabilmesi için homojen olmayan farklı veri setlerinde deneysel çalışmalar yapılabilir. Sezgisel KNN'de veri setlerindeki bağımsızlık değişkenlerin sınıflandırma etki oranları incelendiğinde etki derecesi çok düşük nitelikler bulunmaktadır. Hata oranında çok büyük bir kötüleşme yapmayacak optimum bir eşik değerin belirlenmesi durumunda bu eşik değerden küçük olan nitelikler çıkartılarak boyut azaltma işlemi yapılır. Bu işlemin ardından çoklu sezgisel Knn algoritmasının sınıflandırma performansı artarken problemin boyutunun küçülmesinden dolayı sınıflandırmadaki işlem adımlarında azalma olur dolayısıyla yapılan işlem hızlandırılmış olur.

## 1.5 Kaynakça

<https://medium.com/@ayyucekizrak/derin-öğrenme-için-aktivasyon-fonksiyonlarının-karşılaştırılması-cee17fd1d9cd>

<http://bilgisayarkavramlari.sadievrenseker.com/2013/03/31/siniflandirma-classification/>

<https://medium.com/@k.ulgen90/makine-öğrenimi-bölüm-2-6d6d120a18e1>

<http://bilgisayarkavramlari.sadievrenseker.com/2008/11/17/knn-k-nearest-neighborhood-en-yakin-k-komsu/>

<https://archive.ics.uci.edu/ml/datasets/Electrical+Grid+Stability+Simulated+Data+>

<https://archive.ics.uci.edu/ml/datasets/Avila>