

**YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**DESTEK VEKTÖR MAKİNELERİ İLE  
KARAKTER TANIMA**

**Matematik Mühendisi Deniz Adil DEMİRCİ**

**FBE Matematik Mühendisliği Anabilim Dalı Matematik Mühendisliği Programında  
Hazırlanan**

**YÜKSEK LİSANS TEZİ**

**Tez Danışmanı : Yrd. Doç. Dr. Nilgün Güler BAYAZIT (Y.T.Ü.)**

**İSTANBUL, 2007**

# İÇİNDEKİLER

|                                                             | Sayfa |
|-------------------------------------------------------------|-------|
| SİMGE LİSTESİ .....                                         | iv    |
| KISALTMA LİSTESİ .....                                      | v     |
| ŞEKİL LİSTESİ .....                                         | vi    |
| ÇİZELGE LİSTESİ .....                                       | vii   |
| ÖNSÖZ .....                                                 | viii  |
| ÖZET .....                                                  | ix    |
| ABSTRACT .....                                              | x     |
| 1. GİRİŞ .....                                              | 1     |
| 2. DESTEK VEKTÖR MAKİNELERİ .....                           | 3     |
| 2.1 Lineer Destek Vektör Makineleri .....                   | 3     |
| 2.1.1 Lineer Ayrılabilme Durumu .....                       | 3     |
| 2.1.2 Lineer Ayrılamama Durumu .....                        | 6     |
| 2.2 Lineer Olmayan Destek Vektör Makineleri .....           | 8     |
| 2.3 Çok Sınıflı Destek Vektör Makineleri .....              | 9     |
| 3. ÖN İŞLEMLER .....                                        | 11    |
| 3.1 Ölçeklendirme .....                                     | 11    |
| 3.2 Temel Bileşen Analizi .....                             | 11    |
| 3.2.1 Temel Bileşen Analizinin Tanımı .....                 | 11    |
| 3.2.2 Temel Bileşen Analizinin Matematiksel Yapısı .....    | 12    |
| 3.3 Bağımsız Bileşen Analizi .....                          | 13    |
| 3.3.1 Bağımsız Bileşen Analizinin Tanımı .....              | 13    |
| 3.3.2 Bağımsız Bileşen Analizinin Matematiksel Yapısı ..... | 13    |
| 4. UYGULAMALAR .....                                        | 15    |
| 4.1 Veritabanı: MNIST .....                                 | 15    |
| 4.2 Uygulama Platformu ve Ön Hazırlıklar .....              | 17    |
| 4.3 Orijinal Veri ile Yapılan Uygulama .....                | 19    |
| 4.4 Temel Bileşen Analizi ile Yapılan Uygulamalar .....     | 24    |
| 4.5 Bağımsız Bileşen Analizi ile Yapılan Uygulamalar .....  | 30    |
| 4.6 Yöntemlerin Karşılaştırması .....                       | 36    |
| 5. SONUÇLAR .....                                           | 39    |
| KAYNAKLAR .....                                             | 41    |

|               |    |
|---------------|----|
| ÖZGEÇMİŞ..... | 43 |
|---------------|----|

## SİMGE LİSTESİ

|            |                           |
|------------|---------------------------|
| $X$        | Eğitim seti               |
| $w$        | Yüzey normali             |
| $y_i$      | Etiket değerleri          |
| $x_i$      | Veri değerleri            |
| $L_P$      | İlkel Lagrange ifadesi    |
| $L_d$      | Dual Lagrange ifadesi     |
| $\alpha_i$ | i. Lagrange çarpanı       |
| $\xi_i$    | i. zayıflık değişkeni     |
| $\mu_i$    | i. Lagrange parametresi   |
| $K(.)$     | Çekirdek fonksiyonu       |
| $\Phi(.)$  | Dönüşüm fonksiyonu        |
| $E[.]$     | Özdeğer dizisi            |
| $V[.]$     | Özvektör matrisi          |
| $Cov(.)$   | Kovaryans matris          |
| $D[.]$     | Diagonal özdeğer matrisi  |
| $z[.]$     | Ağırlık değerleri matrisi |

## KISALTMA LİSTESİ

|         |                                                               |
|---------|---------------------------------------------------------------|
| BB      | Bağımsız Bileşen                                              |
| BBA     | Bağımsız Bileşen Analizi                                      |
| DV      | Destek Vektör                                                 |
| DVM     | Destek Vektör Makinesi                                        |
| FastICA | Fast Fixed-Point Algorithm for Independent Component Analysis |
| KKT     | Karush-Kuhn-Tucker                                            |
| LIBSVM  | A Library for Support Vector Machines                         |
| MNIST   | Modified National Institute of Standards and Technology       |
| RTF     | Radyal Tabanlı Fonksiyon                                      |
| TB      | Temel Bileşen                                                 |
| TBA     | Temel Bileşen Analizi                                         |

## ŞEKİL LİSTESİ

|            | Sayfa                                                                                             |
|------------|---------------------------------------------------------------------------------------------------|
| Şekil 2.1  | Lineer ayrılabilme durumunda optimum ayırıcı aşırıdüzlem ..... 4                                  |
| Şekil 2.2  | Lineer ayrılamama durumunda optimal ayırıcı aşırıdüzlem ..... 7                                   |
| Şekil 4.1  | MNIST verilerinin genel görünümü ..... 17                                                         |
| Şekil 4.2  | Orijinal veriler ile sistemin eğitilmesi..... 19                                                  |
| Şekil 4.3  | Orijinal veri ile yapılan uygulamada çekirdek fonksiyonlarına göre başarı oranları ..... 21       |
| Şekil 4.4  | Orijinal veri ile yapılan uygulamada çekirdek fonksiyonlarına göre test süreleri..... 21          |
| Şekil 4.5  | Orijinal veri ile yapılan uygulamada çekirdek fonksiyonlarına göre eğitim süreleri..... 22        |
| Şekil 4.6  | TBA uygulanan sistemin çalışması..... 25                                                          |
| Şekil 4.7  | TB sayısına göre başarı değişimi ..... 27                                                         |
| Şekil 4.8  | TB sayısına göre test süresi değişimi ..... 28                                                    |
| Şekil 4.9  | TB sayısına göre eğitim süresi değişimi ..... 28                                                  |
| Şekil 4.10 | BBA uygulanan sistemin çalışması ..... 31                                                         |
| Şekil 4.11 | Farklı sayıda TB ile hesaplanmış, 150 BB için başarı değişimi ..... 33                            |
| Şekil 4.12 | BB sayısına göre başarı değişimi ..... 34                                                         |
| Şekil 4.13 | BB sayısına göre test süresi değişimi..... 34                                                     |
| Şekil 4.14 | BB sayısına göre eğitim süresi değişimi ..... 35                                                  |
| Şekil 4.15 | Kullanılan verinin türüne göre farklı çekirdek fonksiyonlarının başarı değişimleri..... 37        |
| Şekil 4.16 | Kullanılan verinin türüne göre farklı çekirdek fonksiyonlarının test süresi değişimleri..... 37   |
| Şekil 4.17 | Kullanılan verinin türüne göre farklı çekirdek fonksiyonlarının eğitim süresi değişimleri..... 38 |

## ÇİZELGE LİSTESİ

|              | Sayfa                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Çizelge 4.1  | Örneklerin eğitim ve test kümelerindeki dağılımları ..... 16                                                                                 |
| Çizelge 4.2  | Orijinal veri ile elde edilen sonuçlar ..... 20                                                                                              |
| Çizelge 4.3  | Orijinal test kümesinin DVM ile testi aşamasında lineer çekirdek kullanılması durumunda oluşan hataların dağılımı ..... 22                   |
| Çizelge 4.4  | Orijinal test kümesinin DVM ile testi aşamasında 3. dereceden polinom çekirdek kullanılması durumunda oluşan hataların dağılımı ..... 23     |
| Çizelge 4.5  | Orijinal test kümesinin DVM ile testi aşamasında radyal tabanlı fonksiyon çekirdek kullanılması durumunda oluşan hataların dağılımı ..... 23 |
| Çizelge 4.6  | Lineer çekirdek ve TBA kullanılarak elde edilen sonuçlar ..... 26                                                                            |
| Çizelge 4.7  | 3. dereceden polinom çekirdek ve TBA kullanılarak elde edilen sonuçlar ..... 26                                                              |
| Çizelge 4.8  | RTF çekirdek ve TBA kullanılarak elde edilen sonuçlar ..... 26                                                                               |
| Çizelge 4.9  | 150 TB ve lineer çekirdek ile yapılmış olan hataların dağılımı ..... 29                                                                      |
| Çizelge 4.10 | 34 TB ve 3. dereceden polinom çekirdek ile yapılmış olan hataların dağılımı ..... 29                                                         |
| Çizelge 4.11 | 53 TB ve RTF ile yapılmış olan hataların dağılımı ..... 29                                                                                   |
| Çizelge 4.12 | Lineer çekirdek ve BBA kullanılarak elde edilen sonuçlar ..... 32                                                                            |
| Çizelge 4.13 | Polinom çekirdek ve BBA kullanılarak elde edilen sonuçlar ..... 32                                                                           |
| Çizelge 4.14 | RTF çekirdek ve BBA kullanılarak elde edilen sonuçlar ..... 32                                                                               |
| Çizelge 4.15 | 150 BB ve lineer çekirdek ile yapılmış olan hataların dağılımı ..... 35                                                                      |
| Çizelge 4.16 | 53 BB ve 3. dereceden polinom çekirdek ile yapılmış olan hataların dağılımı ..... 36                                                         |
| Çizelge 4.17 | 53 BB ve RTF çekirdek ile yapılmış olan hataların dağılımı ..... 36                                                                          |

## ÖNSÖZ

Çalışmalarımın her aşamasında yardımlarıyla beni destekleyen, eleştiri ve katkıları ile beni yönlendiren, akademik çalışma detaylarını bana öğreten tez danışmanım Sayın Yrd. Doç. Dr. Nilgün Güler Bayazıt'a teşekkür ederim.

Ayrıca her zaman yanımda olacağına inandığım sevgili dostum ve babam Tacettin Demirci'ye, bana yürekten inanan ve hiçbir konuda desteğini esirgemeyen Gonca Dardeniz'e, öğrenciliğim süresince maddi ve manevi desteklerinden dolayı Sayın Melih ve Güngör Sipahioğlu'na, yardımlarından dolayı arkadaşlarım İ. Saygın Topkaya ve İlkay Uluköklü'ye ve bugüne kadar bana emeği geçen bütün hocalarıma teşekkürü bir borç bilirim.

Deniz Adil DEMİRCİ



## ÖZET

Karakter tanıma, görüntü dosyaları içinde yer alan yazı karakterlerinin görüntü işleme ve makine öğrenmesi algoritmaları kullanılarak metin dosyaları halinde elde edilmesi yöntemlerinin genel adıdır.

Tezde karakter tanıma probleminde destek vektör makinelerinin başarısı incelenmiştir. Bunun yanısıra temel bileşen analizi ve bağımsız bileşen analizi yöntemleri ile kullanılan örnek verilerin, daha az bileşen ile ifade edilmeleri durumunda uygulanan sistemin başarısında meydana gelen değişimler incelenmiştir.

Bu amaçla tez içerisinde öncelikle destek vektör makineleri, temel bileşen analizi ve bağımsız bileşen analizi hakkında bilgi verilmiştir. Ardından uygulama aşamasında elde edilen sonuçlar karşılaştırılarak, değerlendirmesi yapılmıştır.

**Anahtar kelimeler:** Karakter Tanıma, Destek Vektör Makineleri, Temel Bileşen Analizi, Bağımsız Bileşen Analizi

## **ABSTRACT**

Character recognition is a general name for converting letter characters in image files into text files by using image processing and machine learning algorithms.

In this thesis the success of support vector machines on the problem of character recognition is studied. In addition, the situation and the success of the system are studied when the samples are expressed with less component then using principal component analysis and independent component analysis.

For this purpose; firstly, information about support vector machines, principal component analysis and independent component analysis are given. After that, the results of the applications are compared and evaluated.

**Keywords:** Character Recognition, Support Vector Machines, Principal Component Analysis, Independent Component Analysis

## 1. GİRİŞ

Günümüz dünyasının getirdiği şartlardan dolayı, insanlar tarafından yapılan işlerin pek çoğu zamandan kazanç sağlamak amacıyla bilgisayarlar ve makineler aracılığıyla yapılmaktadır. Yapılan bu işlemler makine parçalarının robotlar tarafından montajından başlayarak bankaların müşterilerine ait bilgilerinin analizine kadar pek çok farklı alanda olmaktadır. Bu işlemler ile amaçlanan insan gücünden doğabilecek zaman kısıtlaması, hatalı üretim gibi sorunların en az düzeye indirenmesidir. Örüntü tanıma da bu nedenlerden dolayı ortaya çıkmış bir fikirdir. Parmak izi analizi buna basit bir örnek olarak gösterilebilir. Gelişen teknolojinin sağladığı olanaklar sayesinde insanlara ait parmak izi verileri kullanılarak, aranan parmak izinin kime ait olduğu dakikalar içinde bulunabilmektedir. Bu işlemin bir kişi ya da grup tarafından yapılması ise çok uzun bir zaman gerektirmektedir.

Örüntü tanımanın bir alt dalı da karakter tanıma uygulamalarıdır. Karakter tanıma ise kendi içinde el yazısı karakter tanıma ve optik karakter tanıma olarak ikiye ayrılabilir. Karakter tanıma temel olarak resim formatındaki belge ve dökümanların taranarak içindeki yazıların tanınması ve bunların bir metin editörünün kullanabileceği biçime dönüştürülmesi işlemidir. Optik karakter tanıma uygulama olarak el yazısı karakter tanımaya göre daha basit bir uygulamaya sahiptir, çünkü optik karakterler el yazılarına göre daha düzgün bir yapıya sahiptirler. El yazıları ise kimi zaman insanlar tarafından bile yanlış tanınabilirler. Bu açıdan hata yapılma ihtimalleri daha yüksektir.

Karakter tanıma sistemleri kullanım alanları açısından küçümsenemeyecek bir öneme sahiptir. Masaüstü yayıncılık, mahkemeler, vergi ve tahsilat büroları, personel kayıtlarının yönetimi, bankacılıkta özellikle doldurulan formların takibi ve daha birçok alanda kullanılmaktadırlar. Hergün yüzlerce kredi kartı başvurusu yapılan bir bankanın, doldurulan başvuru formlarını bilgisayar sistemine aktarması karakter tanıma yazılımları ile dakikalar içinde yapılabilirken, bu işlemin personel tarafından yapılması saatleri bulacaktır. Bunların dışında günümüzde el yazısı ile işlem yapan elektronik cihazların sayılarında da oldukça büyük bir artış olmuştur ve bu tip cihazlar içinde karakter tanıma yazılımlarının başarısı ve hızı çok önemli bir çıkış nedeni olmuştur.

Karakter tanıma uygulamalarında görüntü işleme ve makine öğrenmesi algoritmaları temel olarak kullanılan yapılardır. Bu tezin içeriğinde de istatistiksel öğrenme metotlarından biri olan Destek Vektör Makineleri (DVM)'nin, karakter tanımadaki başarısı incelenmiştir. Bu amaçla el yazısı rakam karakterlerinden oluşan Modified National Institute of Standards and

Technology (MNIST) veritabanı üç farklı çekirdek yapısına sahip DVM kullanılarak incelenmiş, elde edilen başarı oranı ve çalışma zamanı üzerinde değerlendirmeler yapılmıştır.

DVM ile birlikte ön işlemler kullanılarak kullanılan verilerin boyutlarının küçültülmesi, ve orijinal verilerin içindeki önemli bilgilerin elde edilerek kullanılması ile başarının arttırılacağı düşünülmektedir. Bu amaçla örnek verilerin içinde bulunan önemli bilgilerin elde edilmesi amacı ile Temel Bileşen Analizi (TBA) ve Bağımsız Bileşen Analizi (BBA) yöntemleri ön işlem olarak kullanılmıştır.

Bu bilgiler eşliğinde tezin asıl amacının karakter tanımda DVM kullanımı ve bu uygulamaların başarısında TBA ve BBA'nın etkilerinin araştırılması olacağı söylenebilir.

Tezin ilerleyişi şu şekilde olacaktır:

Tezin ikinci bölümünde uygulama aşamasında kullanılacak olan DVM'nin tanımı yapılacak, ardından matematiksel yapısından bahsedilerek çoklu sınıflandırma işleminin nasıl yapıldığından bahsedilecektir.

Üçüncü bölümde, kullanılan ön işlemler tanıtılacaktır. Bu amaçla öncelikle tüm veriler için kullanılacak olan ölçeklendirme işlemi ve çalışma şekli tanıtılacak, ardından TBA ve BBA'nın tanımları ve çalışma şekilleri ile devam edilecektir.

Dördüncü bölümde, yapılan uygulamalar detaylı olarak anlatılacaktır. Bölümün başlagıcında öncelikle kullanılan verilere ait bilgilerden bahsedilecek, ardından uygulamaların çalıştırıldığı platform bilgileri verilecektir. Ardından sırasıyla orijinal verilerin kullanımıyla gerçekleştirilen uygulamalar, Temel Bileşenlerin (TB) kullanımları ile gerçekleştirilen uygulamalar ve Bağımsız Bileşenlerin (BB) kullanımıyla gerçekleştirilen uygulamalar anlatılacak ve elde edilen sonuçlar zaman ve başarılarına göre belirtilecektir. Yine aynı bölümde kullanılan yöntemlere göre hangi rakamlarda hata yapıldığı ve hataların hangi rakamlara benzetilerek yapıldığı da belirtilecektir. Bölümün sonunda ise yapılan üç farklı uygulamanın karşılaştırması yapılacaktır.

Sonuç bölümünde ise yapılan işlemler ve elde edilen sonuçlar özetlenecek ve bu konuda ileride yapılabilecek çalışmalar için olası öneriler belirtilecektir.

## 2. DESTEK VEKTÖR MAKİNELERİ

Vapnik tarafından geliştirilen DVM istatistiksel öğrenme teorisi alanında ortaya çıkmış bir öğrenme metodudur. DVM lineer olmayan örnek uzayını, örneklerin lineer olarak ayrılabilceği bir yüksek boyuta aktararak, farklı örnekler arasındaki maksimum sınırın bulunması esasına dayanır.

DVM’de karşılaşılabilecek iki durum, verilerin lineer olarak ayrılabilcekleri bir yapıda olması veya lineer olarak ayrılamayan yapıda olmasıdır. Ancak, gerçek yaşam problemlerinin büyük çoğunluğu birçok farklı bileşenden oluşan problemlerdir ve lineer olarak ayrılmış bir yapı halinde karşımıza çıkmazlar. Lineer olarak ayrılmış olan veriler arasında direkt olarak maksimum sınırın bulunması oldukça kolaydır, ancak lineer olarak ayrılamayan veriler öncelikle lineer olarak ayrılabilcekleri farklı bir uzaya aktarılmalıdırlar. Bu bölüm altında bu işlemlerin nasıl gerçekleştirildiği açıklanacaktır.

DVM, ikili sınıflandırma yapacak şekilde tasarlanmıştır. Ancak tez içinde yapılan uygulamada 0-9 arası rakamları içeren, on farklı sınıf mevcuttur. Bundan dolayı, bu bölüm başlığı altında DVM ile çoklu sınıflandırmanın nasıl yapılacağından da bahsedilecektir.

### 2.1 Lineer Destek Vektör Makineleri

DVM lineer ve lineer olmayan destek vektör makineleri olmak üzere iki farklı alanda incelenecektir. Lineer DVM’ler de lineer olarak ayrılabilen ve ardından iç içe geçmiş ve lineer olarak ayrılamayan verilerle işlem yapan DVM’ler olarak incelenecektir.

#### 2.1.1 Lineer Ayrılabilme Durumu

Eğitim için kullanılacak  $N$  elemandan oluşan verinin  $\mathcal{X} = \{x_i, y_i\}$ ,  $i = 1, 2, \dots, N$  olduğunu varsayalım. Burada  $y_i \in \{-1, 1\}$  etiket değerleri ve  $x_i \in \mathbb{R}^d$  özellikler vektörüdür. Lineer olarak ayrılabilme durumunda, bu iki değerli veriler direkt olarak bir aşırıdüzlem ile ayrılabilir. Bu aşırıdüzleme ayırıcı aşırıdüzlem adı verilir. DVM’nin amacı bu aşırıdüzlemin iki örnek grubuna eş uzaklıkta olmasını sağlamaktır.

DVM ilk işlem olarak lineer olarak ayrılamayan verileri yüksek boyutlu bir forma dönüştürür. Bu yüksek boyut verinin özelliklerinin boyutundan daha büyük derecede olacaktır. Bu işlemin yapılmasının amacı verilerin bir aşırıdüzlem ile ayrılmasını sağlamaktır. Aşırıdüzlem üzerindeki herhangi bir  $x$  noktası,  $w$  aşırıdüzlemin normali ve  $|b|/\|w\|$  hiperuzayın orijine dik



olduğunu varsayalım. Bu durumda  $A_1$  aşırıdüzleminin orijine dik uzaklığı  $|1-b|/\|w\|$  ve  $A_2$  aşırıdüzleminin orijine uzaklığı  $|-1-b|/\|w\|$  olacaktır. Bu iki aşırıdüzlemin optimal aşırıdüzleme uzaklıkları ise  $1/\|w\|$  kadardır, yani iki örnek kümesi arasındaki uzaklık  $A_1$  ve  $A_2$  aşırıdüzlemlerinin birbirlerine paralel olmalarından dolayı  $2/\|w\|$  kadardır. Burada dikkat edilirse  $A_1$  ve  $A_2$  aşırıdüzlemleri arasında eğitim verilerine ait hiçbir örnek bulunmamaktadır. Bu iki aşırıdüzlem arasındaki maksimum uzaklık ise  $\|w\|$  değerinin minimize edilmesiyle bulunabilir. DVM yöntemiyle yapılmaya çalışılan bu iki aşırıdüzlemin arasındaki uzaklığın (sınırın) maksimum olmasını sağlamaktır. Aralarındaki sınır maksimum olan aşırıdüzlemlerin üzerinde yer alan noktalara ise Destek Vektörleri (DV) adı verilmektedir.

Maksimum sınırın bulunması işlemi;

$$\min \frac{1}{2} \|w\|^2 \quad (2.5)$$

$$y_i(w^T x_i + b) \geq +1, \forall i \quad (2.6)$$

ile ifade edilir. Burada (2.5) çözülecek problem ve (2.6) problemin çözümü sırasında kullanılan koşuldur ve bu ifade ikinci dereceden optimizasyon problemidir. Problemin çözümü için problemin Lagrange formülasyonu yapılır. Bunun yapılmasının iki sebebi vardır; birincisi problemin Lagrange formülasyonu yapılarak Lagrange çarpanlarının hesaplanması daha kolaydır, ikincisi ise problemin lineer olmayan durumu içinde genelleştirilmesi için daha uygundur (Burges, 1998). Problemin Lagrange formülasyonu ise,

$$L_P = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i y_i (w^T x_i + b) + \sum_{i=1}^N \alpha_i \quad (2.7)$$

şeklinde dir. Bu formülasyonda  $\alpha_i \geq 0$  değerleri pozitif Lagrange çarpanları olarak adlandırılır. Ancak (2.7)'de ifade edilen formülasyonun çözülmesi oldukça karmaşıktır. Çözümün bulunması için (2.7)'deki ifade Karush-Kuhn-Tucker (KKT) koşulları kullanılarak dual problemine dönüştürülür. Bu problem için KKT koşulları:

$$\frac{\partial L_P}{\partial w} = 0 \Rightarrow w = \sum_i \alpha_i y_i x_i \quad (2.8)$$

$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_i \alpha_i y_i = 0 \quad (2.9)$$

şeklindedir. Bu koşullar (2.7)'de yerine yazılacak olursa:

$$\begin{aligned}
 L_d &= \frac{1}{2}(w^T w) - w^T \sum_i \alpha_i y_i x_i - b \sum_i \alpha_i y_i + \sum_i \alpha_i \\
 &= -\frac{1}{2}(w^T w) + \sum_i \alpha_i \\
 &= \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j
 \end{aligned} \tag{2.10}$$

$$\alpha_i \geq 0, \forall i \tag{2.11}$$

ifadesi elde edilmiş olur. (2.10) probleminin (2.11) koşulları altında çözümü ikinci dereceden optimizasyon problemi ile gerçekleştirilir. Burada dikkat edilirse her eğitim örneği için bir tane Lagrange çarpanının olduğu görülür. Çözümde elde edilen Lagrange çarpanlarının büyük çoğunluğunun değeri sıfır olacaktır. Geriye kalan  $\alpha_i > 0$  değerli  $x_i$  örnekleri DV'lerdir ve  $A_1$  veya  $A_2$  aşırıdüzlemlerinin üzerinde yer alırlar. Lagrange çarpanı sıfır olan örnekler ise  $A_1$  veya  $A_2$  aşırıdüzlemlerinin arka taraflarında kalan örneklerdir.

### 2.1.2 Lineer Ayrılama Durumu

Bölüm 2.1.1'de belirtilen işlemler ancak eğitim örneklerinin tamamen ayrılabilir olması durumunda çalışmaktadır. Eğer ki, örnekler lineer olarak tamamen ayrılabilir durumda değilse problemin çözümü için pozitif zayıflık değişkenleri,  $\xi_i$ ,  $i=1,2,...,N$  kullanılır (Cortes ve Vapnik, 1995). (2.2) ve (2.3)'deki koşullar bu zayıflık değişkenleri ile yeniden tanımlanacak olursa, yeni ifadeler:

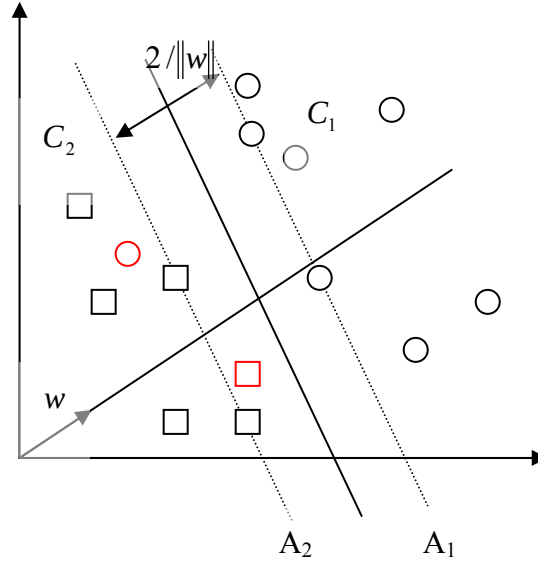
$$y_i = +1 \text{ için, } w^T x_i + b \geq +1 - \xi_i \tag{2.12}$$

$$y_i = -1 \text{ için, } w^T x_i + b \leq -1 + \xi_i \tag{2.13}$$

$$\xi_i \geq 0, \forall i \tag{2.14}$$

şeklinde olacaktır.  $\xi_i = 0$  olması durumunda  $x_i$  örneği doğru sınıflandırılmış,  $0 < \xi_i < 1$  olması durumunda  $x$  örneği doğru sınıflandırılmış ancak  $A_1$  ve  $A_2$  aşırıdüzlemleri arasında yer alıyor,  $\xi_i \geq 1$  ise yanlış sınıflandırılmış demektir (Alpaydın, 2004).





Şekil 2.2 Lineer ayırlamama durumunda optimal ayırıcı aşırıdüzlem

Lineer olarak ayırlamama durumunda sistemin ezberlemesini, yani eğitim verisi içinde olası her durum için bir çözüm üretmemesi için sisteme bir  $C$  üst sınırı eklenir. Bu üst sınır Lagrange çarpanlarının alabilecekleri maksimum değeri göstermektedir. Bu şekilde Lagrange çarpanlarının  $0 \leq \alpha_i \leq C$  aralığında kalması sağlanmaktadır. Bu bilgilere göre Lagrange formülasyonu yeniden şu şekilde ifade edilecektir:

$$L_P = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i - \sum_i \alpha_i \{y_i (w^T x_i + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i \quad (2.15)$$

Yukarıdaki formülasyonda  $\mu_i$ ,  $\xi_i$ 'nin pozitif olmasını garanti etmek için kullanılmış olan Lagrange parametreleridir. Bu Lagrange formülasyonunda çözülmesi zor olduğundan dolayı lineer ayrılabilir örneklerde olduğu gibi dual problemine dönüştürülmektedir. Bu probleme de KKT şartları uygulanırsa:

$$\frac{\partial L_P}{\partial w} = w - \sum_i \alpha_i y_i x_i = 0 \quad (2.16)$$

$$\frac{\partial L_P}{\partial b} = -\sum_i \alpha_i y_i = 0 \quad (2.17)$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad (2.18)$$

ifadeleri elde edilir. Bu ifadeler (2.15)'te yerlerine yazılırsa:

$$L_d = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (2.19)$$

$$0 \leq \alpha_i \leq C, \forall i \quad (2.20)$$

elde edilir. Bu problemin çözümünde  $0 < \alpha_i < C$  aralığında yer alan Lagrange çarpanlarına karşılık gelen  $x_i$  değerleri DV'lerdir.

## 2.2 Lineer Olmayan Destek Vektör Makineleri

Önceki bölümlerde bahsedilmiş olan problemler örneklerin lineer olarak ayrılmasında problem olmaması yada lineer olmayan bir dağılıma sahipse zayıflık parametreleri ile problemin çözümünü aramaktaydı. Lineer olmayan problemlerde çözümü bulmanın bir başka yolu da çekirdek fonksiyonları ile örneklerin öncelikle daha yüksek boyutlu ve lineer olarak ayrılabilirleri bir uzaya taşınıp çözümün bu yeni uzayda aranmasıdır. Örnekleri farklı bir  $H$  Euclid uzayına taşıyan  $\Phi$  fonksiyonunu göz önüne alalım.

$$\Phi : \mathcal{R}^d \mapsto H \quad (2.21)$$

DVM'nin eğitim algoritması sadece  $H$  uzayındaki verilerin  $\Phi(x_i) \cdot \Phi(x_j)$  iç çarpımlarına bağlı olacaktır. Bu çarpımı  $K$  ile gösterirsek:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2.22)$$

olarak ifade edilir.  $K$  fonksiyonu, çekirdek çekirdek fonksiyonu olarak adlandırılmaktadır. Test aşamasında ise sistemin test örneğinin alacağı değer:

$$f(x) = \sum_{i=1}^{N_s} \alpha_i y_i \Phi(s_i) \cdot \Phi(x) + b = \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x) + b \quad (2.23)$$

fonksiyonun işareti ile belirlenir. Bu fonksiyonda  $N_s$  DV'lerin sayısı  $s_i$  ise DV'lerdir.

Tezin uygulama aşamasında lineer DVM'nin yanında iki farklı çekirdek fonksiyonu da kullanılarak sistemin başarısı ölçülecektir. Uygulamada kullanılacak olan çekirdek fonksiyonları:

$$\text{Polinom fonksiyonu: } K(x_i, x_j) = (\gamma(x_i^T \cdot x_j))^d \quad (2.24)$$

$$\text{Radyal Tabanlı Fonksiyon (RTF): } K(x_i, x_j) = \exp\left(-\gamma \|x_i^T - x_j\|^2\right) \quad (2.25)$$

şeklindedir. Fonksiyonlar içindeki  $\gamma$  değerleri  $1/N$  olarak kullanılmıştır. Polinom fonksiyonundaki  $d$  ise polinomun derecesidir.

### 2.3 Çok Sınıflı Destek Vektör Makineleri

Şu ana kadar anlatılmış olan DVM'ler, örneklerin sadece iki farklı sınıftan oluşması durumunda çalışmaktadır. Sınıf sayısının ikiden fazla olması durumunda sınıflandırılmanın nasıl yapılacağından ise bahsedilmemiştir. Bu başlık içinde sınıf sayısının ikiden fazla olması durumunda sistemin çoklu sınıflandırmayı nasıl yaptığından bahsedilecektir.

DVM'nin, çok sınıfı ayırması aşamasında temel olarak kullanılan iki yaklaşım vardır. Bunlardan ilki direkt olarak DVM için kullanılan Lagrange fonksiyonunun çok sınıfla işlem yapacak hale getirilmesidir. Ancak sınıf sayısı arttıkça, bir fonksiyon kullanımından dolayı hataların çok arttığı görüldüğünden bu yöntem tercih edilmemektedir.

İkinci yaklaşımda ise DVM'nin, sınıf sayısına bağlı olarak ikili sınıflandırmalar yapacak şekilde çalıştırılmasıdır. Bu amaçla kullanılan bazı yöntemler, bire karşı bir, bire karşı hepsi ve döngüsel olmayan yönlü graftır.

Bire karşı bir yönteminde sınıf sayısındaki her örnek kümesi diğer örnekler kümeleriyle ayrı ayrı eğitilir. Yani  $S$  farklı sınıf olması durumunda  $S(S-1)/2$  tane eğitim işlemi yapılır. Ardından test aşamasında gelen örneğin bu sınıflardan hangisine ait olduğu, bütün eğitim işlemlerinin sonucunda bulunan DV'ler ile kıyaslanarak ait olduğu sınıf bulunur.

Bire karşı hepsi yönteminde her örnek kümesi, geri kalan tüm örnekler bir kümeye aitmiş gibi varsayılarak eğitilir. Yani  $S$  farklı sınıf olması durumunda  $S$  tane eğitim işlemi yapılır. Test aşamasında ise gelen örneğin ait olduğu sınıf bu eğitim verilerinde elde edilen DV'ler ile kıyaslanarak bulunur.

Döngüsel olmayan yönlü graf yönteminde ise işleyiş bire karşı bir yönteminde olduğu gibidir. Eğitim aşaması tamamen aynı şekilde yapılır. Yöntemin farklılığı ise test aşamasında olmaktadır. Bu yöntemde test yapılırken bütün eğitim işlemleri kontrol edilerek gerçekleştirilmez. Bunun yerine örneğin ait olmadığı düşünülen eğitim örnekleri elenerek ilerlenir.

Tezin uygulama aşamasında eğitilen sistemin testinde bire karşı bir yöntemi kullanılmıştır. El yazısı rakamların tanınması aşamasında 10 farklı sınıfın sınıflandırma işlemi yapılmıştır ve bu amaçla 45 eğitim işlemi gerçekleştirilmiştir. Örnek sınıflar ile sırayla karşılaşıldığı

varsayılırsa oluşturulan eğitim verileri şu şekilde olmaktadır:

Eğitim verileri içinde sınıf değeri  $y_i = 0$  olan örneklerle,  $y_i = 1$  olan örnekler kullanılarak 0 ve 1 değerleri için eğitim işlemi gerçekleştirilir ve bu iki rakamı ayıran DV'ler hesaplanır. Ardından  $y_i = 0$  örnekleri ile  $y_i = 2$  olan örnekler karşılaştırılır. Bu şekilde devam edilir ve son olarak  $y_i = 8$  sınıfı ile  $y_i = 9$  karşılaştırılarak DV'leri hesaplanır.

### 3. ÖN İŞLEMLER

Bu bölüm başlığı altında tezin uygulama aşamasında kullanılmış olan ön işlemler ve bunların matematiksel yapılarından bahsedilecektir. Ön işlemler kurulan sistemin başarısının ve hızının artırılması için yapılan ön çalışmalar olarak tanımlanabilirler. Tezin uygulama aşamasında kullanılan üç ön işlem süreci olan ölçeklendirme, TBA ve BBA sırası ile ele alınacaktır.

#### 3.1 Ölçeklendirme

Ölçeklendirme, veri kümesi içindeki değerlerin değer aralıklarının değiştirilmesi işlemidir. Makine öğrenmesi çalışmalarında elde edilen sonuçlara bakılarak ölçeklendirmenin başarı ve hız artırma konularında son derece önemli bir yeri olduğu görülmektedir (LeCun, vd.,1998). Tezin uygulama aşamasında kullanılmış olan ölçeklendirme işleminin temel çalışma prensibi aşağıdaki gibidir:

- Ölçeklendirme yapılacak değer aralığı belirlenir. [taban,tavan]
- Veri kümesinin her sütunu için minimum ve maksimum değerleri bulunur.
- Herbir sütunun içindeki elemanların yeni değerleri hesaplanır.

Yeni değerler ise,

$$yeni\ deger = taban + (tavan - taban) \times \frac{eski\ deger - minimum}{maksimum - minimum} \quad (3.1)$$

yöntemiyle hesaplanır.

Burada karşılaşılabilecek özel bir durum olan maksimum ve minimum değerlerin eşit olması halinde, ki bu durum tezde yapılmış olan uygulamada sadece minimum ve maksimum değerlerin ikisinin birden sıfır olması halinde gerçekleşmiştir, yeni değer sıfır olarak kabul edilerek çözülmüştür.

#### 3.2 Temel Bileşen Analizi

##### 3.2.1 Temel Bileşen Analizinin Tanımı

TBA birbiriyle bağlantılı çok sayıda değişkeni bulunan bir veri kümesinin boyutunu düşürmek amacıyla ortaya çıkmış bir yöntemdir (Jolliffe, 2002). Bu boyut düşürme işlemi yapılırken, veri kümesi içerisinde olabildiğince çok varyasyonun kalması sağlanmaya çalışılır. Böylece orijinal veri kümesine göre daha düşük boyutlu bir veri kümesi ile orijinal kümenin

temsil edilmesi sağlanır. TBA, bu özelliğinden dolayı uzun zamanlardır örüntü tanıma alanında sıklıkla kullanılmaktadır.

TBA veri kümesinin karakteristiğini bularak boyutlarının düşürülmesini sağlayan bir yöntemdir. Bu işlemi, veri kümesine ait varyansı hesaplayarak yapmaktadır. Veri kümesinin varyansını hesaplamak için öncelikle örneklerle ait değerlerle oluşturulan bir matrisin kovaryans matrisini hesaplayıp, bu kovaryans matrise ait özdeğer ve özvektörleri hesaplar. Hesaplanan özdeğerlerin yüksek sayısal değere sahip olan elemanları veriyi daha iyi temsil eden özvektörleri temsil etmektedir, sayısal değer düştükçe ise verinin karakteristiğini daha az belirleyen özvektörlere ulaşılır. Özdeğerin değerinin sıfır olması ise o özdeğere karşılık gelen özvektörün diğer özvektörlerin bileşenleri olarak ifade edilebileceğini, yani lineer bağımlı olduğunu gösterir.

Özdeğerler ve özvektörler elde edildikten sonra ise en yüksek değere sahip olan özvektörlerden kaç tanesinin kullanımıyla yeni verilerin elde edilmesi gerektiği deneme yanılma yoluyla bulunabilmektedir. Uygulama aşamasındaki sonuçlarda da görüleceği gibi kullanılan TB sayısı arttıkça kurulan sistemin başarısı bir süre artmakta ancak daha sonra düşmeye başlamaktadır. Bundan dolayı TB'ler hesaplandıktan sonra başarıyı maksimum yapan TB sayısının bulunması için örnek veriler değişik bileşen sayıları ile test edilerek optimum TB sayısı bulunmaktadır.

### 3.2.2 Temel Bileşen Analizinin Matematiksel Yapısı

TBA'nın ilk adımında kullanılan örneklerin her bir sınıfının ortalamaları hesaplanır.  $x_i$ ,  $i=1,2,...,N$  örnek veriler ve her bir  $x_i$  örneği için  $x_i = [x_i^1 \ x_i^2 \ \dots \ x_i^M]^T$  olmak üzere, ortalama:

$$m = \frac{1}{N} \sum_{i=1}^N x_i = \frac{1}{N} \begin{bmatrix} x_1^1 + x_2^1 + \dots + x_N^1 \\ \vdots \\ x_1^M + x_2^M + \dots + x_N^M \end{bmatrix} \quad (3.2)$$

ile hesaplanır. Ardından hesaplanan ortalama  $x = [x_1 \ x_2 \ \dots \ x_N]$  matrisinden çıkarılarak ortalaması sıfır olan bir veri kümesi elde edilir.

$$\bar{x}_i = x_i - m_i \quad (3.3)$$

$$\bar{x} = [\bar{x}_1 \ \bar{x}_2 \ \dots \ \bar{x}_N]_{M \times N} \quad (3.4)$$

Ardından (3.4)'te ifade edilen  $\bar{x}$  matrisi transpozesi ile çapılarak kovaryans matrisi hesaplanır.

$$\text{Cov}(\bar{x}\bar{x}^T) \quad (3.5)$$

Hesaplanan kovaryans matrisine ait özdeğerler ve bu özdeğerlere karşılık gelen özvektörler hesaplanır. Bulunan bu özvektörler aranan TB'lerdir. Herhangi bir özvektörün hesaplanması için kullanılan özdeğerinin sayısal değeri ise özvektörün elimizdeki örnek verileri ne kadar iyi temsil ettiğini göstermektedir. Bundan dolayı hesaplanan özvektörler kendilerine karşılık gelen özdeğerlere göre büyükten küçüğe doğru sıralanır. Hesaplanan ve büyükten küçüğe doğru sıralanan özdeğer dizisi  $E = [E_1 E_2 \dots E_M]$  ve özdeğerlerin değerine göre sıralanmış özvektör matrisi  $V = [V_1 V_2 \dots V_M]$ ,  $\forall V_i = [V_i^1 V_i^2 \dots V_i^M]^T$  olmak üzere, yeni örnek kümesi:

$$\tilde{x} = \bar{x} \tilde{V} \quad (3.6)$$

ile bulunur. Burada  $\tilde{V}$  seçilmiş olan ilk  $t$ ,  $t \leq M$  özdeğere karşılık gelen özvektör matrisini göstermektedir.

### 3.3 Bağımsız Bileşen Analizi

#### 3.3.1 Bağımsız Bileşen Analizinin Tanımı

BBA çok boyutlu bir vektörü mümkün olduğunca istatistiksel olarak bağımsız bileşenlere dönüştüren bir yöntemdir (Hyvärinen, 1999). TBA'dan farklı olarak BBA ile elde edilen BB'lerin varyansları ve sıralamaları belli değildir. Burada önemli olan bir nokta da BB'lerin dağılımının istatistiksel olarak Gauss olmayan bir dağılıma sahip olmalarıdır.

#### 3.3.2 Bağımsız Bileşen Analizinin Matematiksel Yapısı

BBA işleminden önce TBA'da olduğu gibi örneklerle ait sınıfların ortalaması hesaplanır.  $x_i$ ,  $i=1,2,\dots,N$  örnek veriler ve her bir  $x_i$  örneği için  $x_i = [x_i^1 \ x_i^2 \ \dots \ x_i^M]^T$  olmak üzere, ortalama:

$$m = \frac{1}{N} \sum_{i=1}^N x_i = \frac{1}{N} \begin{bmatrix} x_1^1 + x_2^1 + \dots + x_N^1 \\ \vdots \\ x_1^M + x_2^M + \dots + x_N^M \end{bmatrix} \quad (3.7)$$

ile hesaplanır. Ardından hesaplanan ortalama  $x = [x_1 x_2 \dots x_N]$  matrisinden çıkarılarak ortalaması sıfır olan bir veri kümesi elde edilir.

$$\bar{x}_i = x_i - m_i \quad (3.8)$$

$$\bar{x} = [\bar{x}_1 \bar{x}_2 \cdots \bar{x}_N]_{M \times N} \quad (3.9)$$

İkinci önişlem olarak ortalaması sıfır olan yeni matrin bileşenlerinin birbirlerinden bağımsız ve varyanslarının eşit olması sağlanır. Elde edilen bu özellikteki yeni matrisin  $\hat{x}$  ile gösterildiğini varsayarsak, bu özellikleri sağlaması için,  $\hat{x}$ 'in kovaryans matrisinin birim matrise eşit olması gerekmektedir.

$$\text{Cov}(\hat{x}\hat{x}^T) = I \quad (3.10)$$

Bu eşitliği sağlayan  $\hat{x}$  matrisi ise özdeğer dekompozisyonu yöntemi ile bulunur.

$$\hat{x} = VD^{-1/2}V^T\bar{x} \quad (3.11)$$

Bu ifadede  $V = [V_1 V_2 \cdots V_M]$ ,  $\forall V_i = [V_i^1 V_i^2 \cdots V_i^M]^T$  olmak üzere özvektör matrisini,  $D = \text{diag}[E_1 E_2 \cdots E_M]$  diagonal özdeğer matrisini göstermektedir.

Bu ön işlemlerin ardından kayan nokta algoritması ile çalışan BBA işlemine geçilir. Kayan nokta algoritmasının BBA'ya uygulanması Hyvärinen tarafından gerçekleştirilmiştir ve analizin daha hızlı bir şekilde yapılmasını sağlamaktadır.  $\hat{x} = [\hat{x}_1 \hat{x}_2 \cdots \hat{x}_M]$ 'in her bir elemanı için rastgele  $z$  başlangıç vektörleri seçilir. Ardından yeni  $z$  ağırlık vektörü:

$$z_i \leftarrow \text{var}\{\hat{x}_i g(z_i^T \hat{x}_i)\} - \text{var}\{g'(z_i^T \hat{x}_i)\} \hat{x}_i \quad (3.12)$$

ile belirlenir. Burada  $\text{var}$  varyansı,  $g$  ise dağılımın simetrik olması için kullanılan bir fonksiyondur. Uygulama aşamasında  $g(t) = t \times \exp(-t^2 / 2)$  olarak kullanılmıştır. Bu durumda:

$$g'(t) = (1 - t^2) \exp(-t^2 / 2) \quad (3.13)$$

olacaktır. Yeni  $z$  değeri hesaplandıktan sonra normuna bölünerek birim normda elde edilir.

$$z_i \leftarrow z_i / \|z_i\| \quad (3.14)$$

Eğer  $z$  değeri yakınsamıyorsa (3.12) işlemine geri dönülür. Son olarak  $z$  matrisi elde edildiğinde  $x$  ile çarpılarak yeni örnek veri kümesi elde edilir.



#### 4. UYGULAMALAR

Bu bölümde, öncelikle tezde kullanılan veri tanıtılıp, veri üzerinde yapılan işlemler ayrıntılı olarak açıklanacaktır. Yapılan uygulama, öncelikle verinin üzerinde sadece ölçeklendirme yapılarak sistemin eğitilmesi ve test edilmesi ile başlayacak, ardından TBA ve BBA yöntemleri ile elde edilen yeni veriler ile yapılmış olan eğitim ve test işlemlerinin açıklanması ile devam edecektir. Bölümün son kısmında ise elde edilen sonuçlar karşılaştırılacak ve sonuç bölümüne geçilmeden önce tablolar halinde sunulacaktır.

Uygulama aşamasında TBA veya BBA kullanılarak sistemin orijinal verilere göre daha yüksek başarı elde edeceği düşünülmektedir. Sistemin başarısının ölçümünde, sistemin çalışma süresi ve elde edilen başarı oranları kriter olarak alınacak, bunların yorumlanması ise sonuç bölümünde yapılacaktır.

##### 4.1 Veritabanı: MNIST

Tezin uygulama aşaması için MNIST verileri örnek veritabanı olarak alınmıştır. MNIST veritabanı, NIST veritabanının modifiye edilmiş halidir. MNIST veritabanı Yann LeCun tarafından hazırlanmış ve <http://yann.lecun.com/exdb/mnist/> adresinden temin edilmiştir. Veritabanının oluşturulmasına ait bilgiler (LeCun vd., 1998) makalesinden alınmıştır.

Orijinal NIST veritabanı SD-3 eğitim kümesi ve SD-1 isimli test kümesinden oluşmaktadır. Ancak SD-3 kümesi, SD-1'e göre çok daha temiz ve tanınması kolay olan rakamlardan oluşmaktadır. Bunun sebebi SD-3 veri kümesinin Amerikan nüfus bürosu çalışanlarından, SD-1'in ise lise öğrencilerinden alınmış örneklerinden oluşturulmuş olmasıdır. Bu farklılığı ortadan kaldırmak için bu iki veri kümesinin karışımından elde edilen MNIST veri kümesi oluşturulmuştur.

SD-1 500 farklı kişi tarafından yazılmış 58.527 rakam görüntüsünden oluşmaktadır. SD-3 verilerinde tüm yazarların girmiş olduğu veriler sıra ile kaydedilmiş olmasına karşın, SD-1'in verileri karışık bir düzen halinde oluşturulmuştur. Ancak SD-1'i oluşturan kişilerin bilgilerine ulaşıp, SD-1'de kişilere göre ayrılmıştır ve ardından SD-1'in ilk 250 kişisine ait bilgiler eğitim ve sonraki 250 kişisine ait bilgiler test verisi olarak ayrılmıştır. Böylece yaklaşık olarak eğitim ve test aşaması için 30.000'er örnekli iki küme elde edilmiştir. Bunun ardından yeni eğitim kümesi, SD-3'ün ilk elemanından itibaren yeterince örnek alınarak 60.000 rakamdan oluşacak şekilde düzenlenmiştir. Aynı şekilde test kümesi de SD-3'ün 35.000. elemanından itibaren ilaveler yapılarak 60.000 eleman içerecek şekilde oluşturulmuştur. Ancak denemeler

için test kümesi 10.000 eleman (5.000 SD-1 ve 5.000 SD-3) olarak kullanılacak şekilde düzenlenmiştir. Oluşturulmuş olan bu yeni veri kümesi modifiye NIST veya MNIST olarak tanımlanmaktadır.

Orijinal siyah-beyaz görüntüler büyüklük oranları korunarak 20x20 piksel boyutlarına normalize edilmiştir. Sonuçta elde edilen görüntüler, normalizasyonda kullanılan görüntü interpolasyonundan dolayı gri seviye görüntü değerlerine sahip olmuşlardır. Ardından elde edilen görüntüler piksel değerlerinin ağırlık merkezi hesaplanarak 28x28 boyutunda görüntüler olarak kaydedilmiştir.

Bu tez içinde yapılan uygulamada bu iki veri kümesi eğitim ve test kümeleri olarak alınarak, bundan sonraki adımlarda yapılan işlemlerde kullanılmışlardır. Buradan itibaren eğitim kümesi ile MNIST veritabanının 60.000 elemanlık eğitim kümesi ve test kümesi ile MNIST veritabanının 10.000 elemanlık test kümesi ifade edilecektir. Eğitim ve test kümelerindeki elemanların dağılımı ise aşağıdaki gibidir:

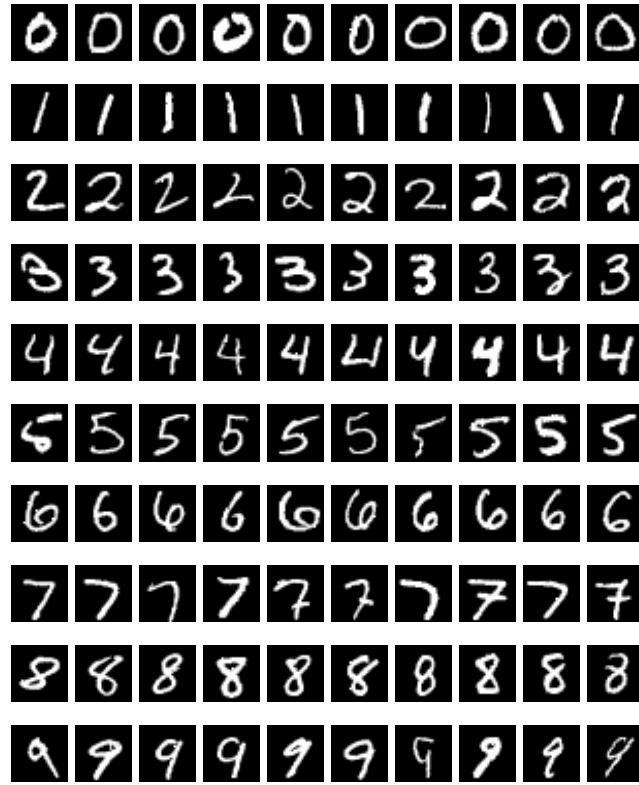
Çizelge 4.1 Örneklerin eğitim ve test kümelerindeki dağılımları

| <b>Rakam</b>               | <b>Eğitim kümesi<br/>örnek sayısı</b> | <b>Test kümesi<br/>örnek sayısı</b> |
|----------------------------|---------------------------------------|-------------------------------------|
| 0                          | 5.923                                 | 980                                 |
| 1                          | 6.742                                 | 1.135                               |
| 2                          | 5.958                                 | 1.032                               |
| 3                          | 6.131                                 | 1.010                               |
| 4                          | 5.842                                 | 982                                 |
| 5                          | 5.421                                 | 892                                 |
| 6                          | 5.918                                 | 958                                 |
| 7                          | 6.265                                 | 1.028                               |
| 8                          | 5.851                                 | 974                                 |
| 9                          | 5.949                                 | 1.009                               |
| <b><math>\Sigma</math></b> | <b>60.000</b>                         | <b>10.000</b>                       |

Çizelge 4.1'e bakarak eğitim ve test kümelerindeki rakam örneklerinin, birbirlerine yaklaşık olarak eşit oldukları söylenebilir. Herbir rakamın öğrenilmesi için yaklaşık olarak eşit miktarda örnek kullanılmasından dolayı, kurulacak sistemin herbir rakam için eş miktarda örnekle eğitilmesi ve böylece eğitim ve test aşamasında sistemin dengede olmasının bekleneceği sonucu çıkarılır.

Eğitim ve test kümelerinin her bir elemanı yukarıda da belirtildiği gibi 28x28 piksel boyutlarındadır ve gri seviye görüntülerden, yani piksel değerleri [0,255] olan değerlerden

oluşmaktadır. MNIST veritabanı içinde bulunan görüntüler Şekil 4.1’deki gibidir:



Şekil 4.1 MNIST verilerinin genel görünümü

Şekil 4.1’de görüldüğü gibi MNIST veritabanı içinde bulunan rakamlar arka fon siyah ve rakamlar beyaz olacak şekilde kaydedilmiştir. Uygulama içinde bu yapı aynen kullanılacaktır. DVM’nin eğitim ve test aşamalarının hızlı çalışabilmesi için az sayıda piksel değerinin girdi olarak verilmesi gerekmektedir. DVM’nin çalıştırılması aşamasında kullanılan verilerde değeri sıfır olan elemanlar sisteme dahil edilmez. Dijital görüntülerde sıfır piksel değeri siyah rengine karşılık geldiğinden, arka planın siyah olması, sistemin girdi değerleri belirlenirken arka planın sisteme dahil edilmemesi avantajını sağlamaktadır. Bu sayede 28x28 piksel boyutundaki, yani 784 piksel değeri taşıyan görüntülerin daha az girdi değeri ile ifade edilmesi sağlanır ve bu şekilde zaman konusunda büyük bir tasarruf sağlanmış olur.

#### 4.2 Uygulama Platformu ve Ön Hazırlıklar

Bu bölümde uygulamaların çalıştırılması için kurulmuş olan sistem hakkında bilgi verilecektir. Hazırlanan uygulamalar DVM’nin çalışması sırasında hesaplanan sürenin farklılık göstermemesi açısından tek bir bilgisayar üzerinde yapılmıştır. Sadece BBA’nın yapılması için farklı bir bilgisayar kullanılmıştır.

Hazırlanan program BBA hariç tamamen C programlama dili kullanılarak yazılmıştır. DVM aşaması için hazır bir uygulama olan LIBSVM (A Library for Support Vector Machines) üzerinde bazı değişiklikler yapılarak kullanılmıştır. BBA’da yine hazır bir uygulama olan FastICA (Fast Fixed-Point Algorithm for Independent Component Analysis) ile gerçekleştirilmiştir. FastICA Matlab ile yazılmış olduğundan dolayı farklı bir bilgisayar gereksinimi duyulmuştur.

Gerçekleştirilen uygulamanın tamamı MS Windows XP Pro işletim sistemi üzerinde çalıştırılmıştır. FastICA uygulaması için kullanılan sistem Intel Pentium 4, 2.8 Ghz işlemcili ve 768 Mb bellek içeren bir bilgisayardır. Uygulamanın diğer kısmında kullanılan sistem ise, AMD 64 bit 3 Ghz işlemcili, 2 Gb bellek bulunduran bir donanımda gerçekleştirilmiştir.

Sistemin uygulanmasında ön hazırlık işlemi olarak MNIST veritabanı üzerinde değişiklik yapılmıştır. MNIST veritabanı dört parça halinde temin edilmiştir. Bunlar eğitim verilerinin tutulduğu 9.912.422 byte büyüklüğündeki görüntü dosyası, test verilerinin tutulduğu 1.648.877 byte büyüklüğündeki görüntü dosyası, eğitim verilerinin etiket bilgilerinin tutulduğu 28.881 byte büyüklüğündeki etiket dosyası ve test verilerinin etiket bilgilerinin tutulduğu 4.542 byte büyüklüğündeki etiket dosyasıdır. Görüntü dosyaları, eğitim kümesi için 60.000 rakamın herbiri için 784 piksel değerin tutulduğu, test kümesi için de 10.000 rakamın 784 piksel değerinin tutulduğu dosyalardır. Etiket dosyaları ise her 784 piksellik elemana karşılık gelen rakamın değerini saklayan dosyalardır. Kurulan sistemde öncelikle, eğitim kümesine ait etiket ve piksel değerleri bir araya gelecek şekilde tek bir eğitim veritabanı ve test kümesine ait etiket ve piksel değerleri bir araya gelecek şekilde bir test veritabanı oluşturulmuştur ve uygulamada yapılacak işlemler bu iki veritabanı ile gerçekleştirilmiştir. Hazırlanan dosyalarda her satırda ayrı bir örnek için, görüntü etiketi ve piksel değerleri yer alacak şekilde kayıt yapılmıştır. Oluşturulan dosyaların yapısı aşağıdaki gibidir:

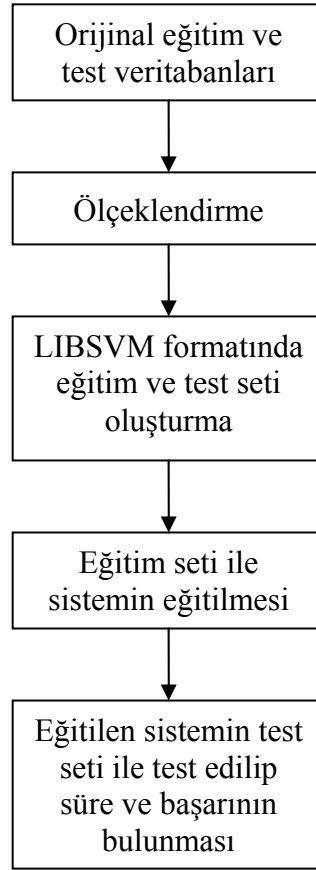
<1.rakam değeri> <1.piksel değeri> <2.piksel değeri> ... <784.piksel değeri>

<2.rakam değeri> <1.piksel değeri> <2.piksel değeri> ... <784.piksel değeri>

Bu şekilde eğitim veritabanı 60.000 satır ve 785 sütun, test veritabanı ise 10.000 satır ve 785 sütundan oluşturulmuştur.

### 4.3 Orijinal Veri ile Yapılan Uygulama

Sistemin başarısını ölçmek için ilk olarak orijinal veriler kullanılarak sistemin eğitilmesi ve test edilmesi gerçekleştirilmiştir. Kurulan sistemin çalışması Şekil 4.2’de gösterildiği gibidir.



Şekil 4.2 Orijinal veriler ile sistemin eğitilmesi

Orijinal veriler ile yapılan uygulama gerçekleştirilmeden önce sistemin girdileri üzerinde Bölüm 3.1’de belirtilmiş olan ölçeklendirme işlemi yapılmıştır. Ön hazırlıklar bölümünde elde edilen eğitim ve test veritabanlarının piksel değerleri  $[0,255]$  gri seviye değerlerine sahipti; DVM’nin çalışması sırasında karşılaşılabilecek iki büyük problemten kurtulmak amacıyla bu değerler DVM ile eğitim ve test süreçlerinden önce  $[0,1]$  aralığına indirgenmiştir.  $[0,255]$  piksel değerlerinin kullanılması durumunda karşılaşılabilecek sorunlar:

- Verilerin boyutu büyük olacağından hesaplamalarda işlemlerin hatalı sonuçlar üretme ihtimali çok yüksektir.
- Veriler ölçeklendirilmeden yapılan denemede işlem süresinin çok fazla uzadığı gözlemlenmiştir.

Bu sebeplerden dolayı verilerin ölçeklendirilmesinin gerektiği görülmüştür. Uygulamada ölçeklendirme için öncelikle  $[-1,1]$  aralığı denenmiştir, ancak süre konusunda yine çok yavaş olduğundan dolayı sonradan  $[0,1]$  aralığı denenip, ölçeklendirme bu şekilde yapılmıştır.

Ölçeklendirme işlemi için eğitim ve test veritabanları birlikte kontrol edilerek herbir piksel için minimum ve maksimum değerler hesaplanmıştır ve ölçeklendirme işlemi iki veritabanından elde edilen bu sonuçlar beraber kullanılarak yapılmıştır. Bu şekilde oluşturulan minimum, maksimum değer bilgileri ile eğitim ve test veritabanlarındaki değerler  $[0,1]$  aralığına indirgenmiştir.

Ölçeklendirme işleminin ardından elde edilen yeni eğitim ve test veritabanları LIBSVM'nin girdi formatına dönüştürülmüştür. Bu aşamada piksel değeri sıfır olan noktalar girdi formatı oluşturulurken sisteme dahil edilmemiştir. LIBSVM'nin girdi formatı;

<rakam değeri> <1: 1. piksel değeri> <2: 2. piksel değeri> ... <784: piksel değeri>

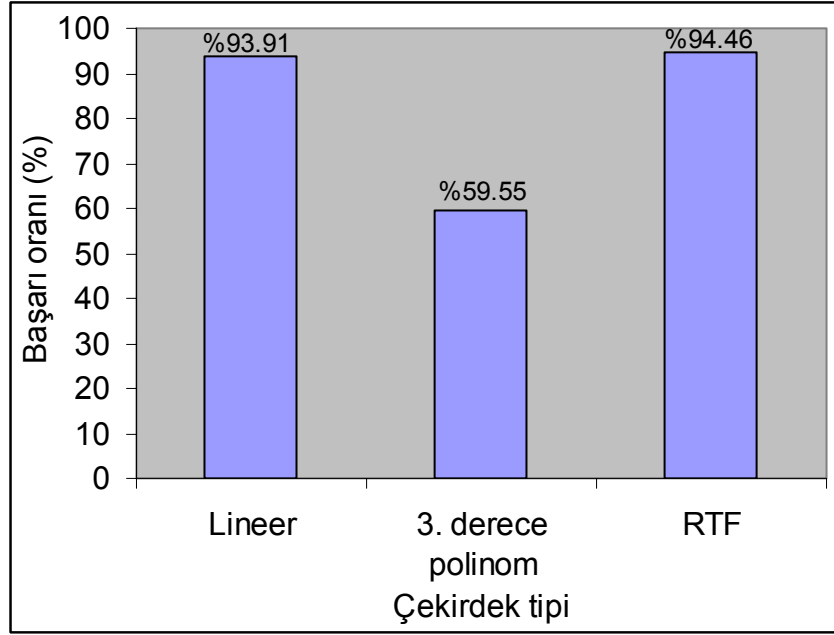
şeklindedir.

Sistemin girdi dosyalarının hazırlanmasının ardından LIBSVM ile üç farklı çekirdek fonksiyonu kullanılarak eğitim ve test işlemi gerçekleştirilmiştir. Elde edilen sonuçlar aşağıdaki çizelgede görülmektedir.

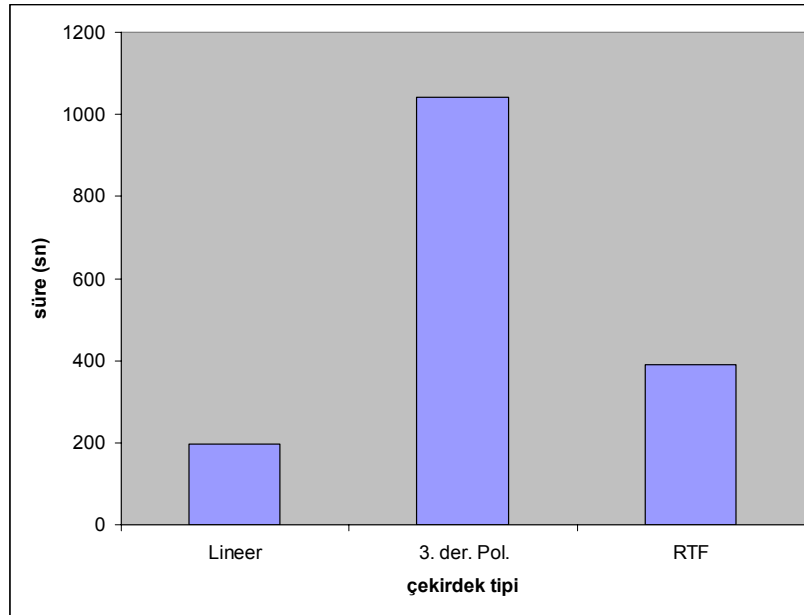
Çizelge 4.2 Orijinal veri ile elde edilen sonuçlar

| Çekirdek Tipi     | Eğitim süresi (sn) | Test süresi (sn) | Başarı oranı (%) |
|-------------------|--------------------|------------------|------------------|
| Lineer            | 929,45             | 195,33           | 93,91            |
| 3. derece polinom | 10.149,36          | 1.040,16         | 59,55            |
| RTF               | 1.130,03           | 388,84           | 94,46            |

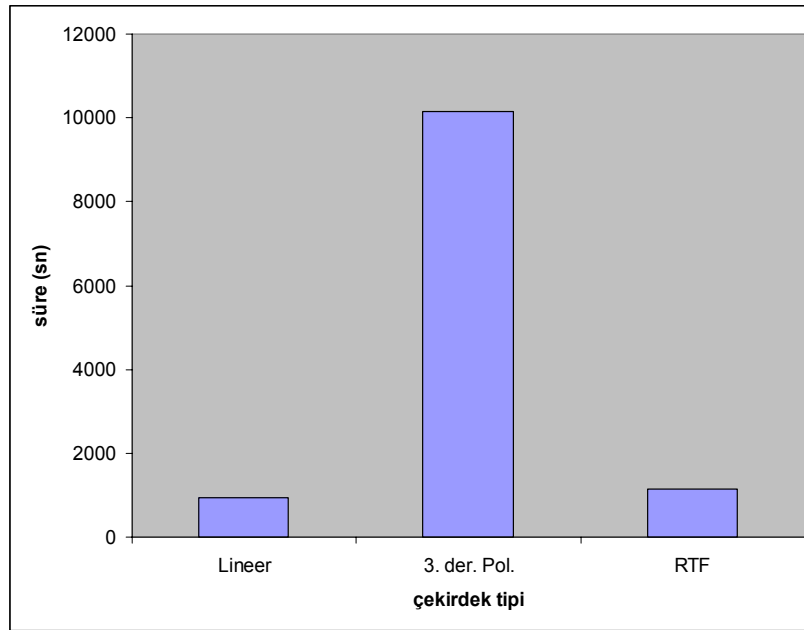
Uygulamanın başarısının belirlenmesinde en önemli iki kriter kurulan sistemin çalışması sonucu elde edilen başarıdır oranı ve sistemin çalışma süresidir. Çizelge 4.2'de belirtilen sonuçlar, başarı, test ve eğitim zamanlarına göre histogramlarla gösterilmiştir (Şekil 4.3, 4.4 ve 4.5).



Şekil 4.3 Orijinal veri ile yapılan uygulamada çekirdek fonksiyonlarına göre başarı oranları



Şekil 4.4 Orijinal veri ile yapılan uygulamada çekirdek fonksiyonlarına göre test süreleri



Şekil 4.5 Orijinal veri ile yapılan uygulamada çekirdek fonksiyonlarına göre eğitim süreleri

Aşağıdaki çizelgelerde DVM uygulaması sırasında oluşan hataların rakamlara göre dağılımları, kullanılan çekirdek fonksiyonlarına göre ayrı ayrı gösterilmektedir. Çizelgelere rakamlardaki toplam hataların yanısıra, tanıma sırasında orijinal rakamın hangi rakamlara benzetilerek hata yapıldığı da eklenmiştir.

Çizelge 4.3 Orijinal test kümesinin DVM ile testi aşamasında lineer çekirdek kullanılması durumunda oluşan hataların dağılımı

|                  |   | HATALAR |    |    |    |    |     |    |    |    |    |     |
|------------------|---|---------|----|----|----|----|-----|----|----|----|----|-----|
|                  |   | 0       | 1  | 2  | 3  | 4  | 5   | 6  | 7  | 8  | 9  | Σ   |
| ORJİNAL RAKAMLAR | 0 | -       | 0  | 4  | 1  | 1  | 14  | 5  | 1  | 1  | 0  | 27  |
|                  | 1 | 0       | -  | 2  | 3  | 0  | 1   | 2  | 1  | 5  | 0  | 14  |
|                  | 2 | 8       | 7  | -  | 8  | 6  | 8   | 8  | 6  | 14 | 2  | 67  |
|                  | 3 | 3       | 1  | 13 | -  | 3  | 23  | 1  | 7  | 13 | 6  | 70  |
|                  | 4 | 0       | 1  | 7  | 0  | -  | 1   | 5  | 0  | 3  | 19 | 36  |
|                  | 5 | 9       | 3  | 1  | 30 | 3  | -   | 12 | 0  | 22 | 3  | 83  |
|                  | 6 | 10      | 4  | 12 | 1  | 6  | 19  | -  | 1  | 2  | 0  | 55  |
|                  | 7 | 2       | 9  | 16 | 9  | 8  | 1   | 0  | -  | 3  | 28 | 76  |
|                  | 8 | 6       | 5  | 7  | 19 | 9  | 29  | 5  | 6  | -  | 16 | 102 |
|                  | 9 | 5       | 7  | 3  | 6  | 31 | 7   | 0  | 18 | 2  | -  | 79  |
| Σ                |   | 43      | 37 | 65 | 77 | 67 | 103 | 38 | 40 | 65 | 74 | 609 |



Çizelge 4.4 Orijinal test kümesinin DVM ile testi aşamasında 3. dereceden polinom çekirdek kullanılması durumunda oluşan hataların dağılımı

|                  |   | HATALAR |      |    |     |    |    |    |    |    |     |          |
|------------------|---|---------|------|----|-----|----|----|----|----|----|-----|----------|
|                  |   | 0       | 1    | 2  | 3   | 4  | 5  | 6  | 7  | 8  | 9   | $\Sigma$ |
| ORJİNAL RAKAMLAR | 0 | -       | 222  | 4  | 7   | 2  | 52 | 14 | 2  | 1  | 2   | 306      |
|                  | 1 | 0       | -    | 1  | 1   | 0  | 0  | 1  | 0  | 1  | 0   | 4        |
|                  | 2 | 8       | 434  | -  | 12  | 7  | 2  | 3  | 8  | 13 | 1   | 488      |
|                  | 3 | 0       | 366  | 12 | -   | 0  | 5  | 1  | 7  | 10 | 5   | 406      |
|                  | 4 | 1       | 445  | 0  | 0   | -  | 0  | 4  | 1  | 0  | 78  | 529      |
|                  | 5 | 2       | 506  | 3  | 40  | 6  | -  | 10 | 3  | 0  | 6   | 576      |
|                  | 6 | 3       | 314  | 2  | 0   | 6  | 4  | -  | 0  | 0  | 0   | 329      |
|                  | 7 | 1       | 445  | 4  | 0   | 2  | 0  | 0  | -  | 3  | 4   | 459      |
|                  | 8 | 6       | 358  | 3  | 39  | 7  | 20 | 5  | 18 | -  | 29  | 485      |
|                  | 9 | 9       | 390  | 1  | 4   | 12 | 1  | 1  | 41 | 4  | -   | 463      |
| $\Sigma$         |   | 30      | 3480 | 30 | 103 | 42 | 84 | 39 | 80 | 32 | 125 | 4045     |

Çizelge 4.5 Orijinal test kümesinin DVM ile testi aşamasında radyal tabanlı fonksiyon çekirdek kullanılması durumunda oluşan hataların dağılımı

|                  |   | HATALAR |    |    |    |    |    |    |    |    |    |          |
|------------------|---|---------|----|----|----|----|----|----|----|----|----|----------|
|                  |   | 0       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | $\Sigma$ |
| ORJİNAL RAKAMLAR | 0 | -       | 0  | 2  | 0  | 0  | 6  | 3  | 1  | 1  | 0  | 13       |
|                  | 1 | 0       | -  | 2  | 2  | 0  | 1  | 4  | 0  | 4  | 1  | 14       |
|                  | 2 | 8       | 1  | -  | 9  | 11 | 5  | 12 | 11 | 15 | 3  | 75       |
|                  | 3 | 2       | 0  | 13 | -  | 1  | 19 | 1  | 9  | 12 | 5  | 62       |
|                  | 4 | 1       | 0  | 5  | 0  | -  | 1  | 7  | 2  | 2  | 25 | 43       |
|                  | 5 | 7       | 3  | 4  | 31 | 4  | -  | 13 | 2  | 11 | 5  | 80       |
|                  | 6 | 9       | 3  | 4  | 1  | 5  | 9  | -  | 0  | 1  | 0  | 32       |
|                  | 7 | 2       | 13 | 22 | 3  | 7  | 1  | 0  | -  | 4  | 20 | 72       |
|                  | 8 | 4       | 7  | 7  | 12 | 8  | 24 | 10 | 5  | -  | 5  | 82       |
|                  | 9 | 6       | 7  | 0  | 10 | 32 | 9  | 1  | 11 | 5  | -  | 81       |
| $\Sigma$         |   | 39      | 34 | 59 | 68 | 68 | 75 | 51 | 41 | 55 | 64 | 554      |

Orijinal veri ile yapılmış olan testlerde çekirdek fonksiyonlarının farklı rakamların tanınmasında birbirlerinden farklı başarılar elde ettiği görülmektedir. Lineer ve radyal tabanlı çekirdek fonksiyonu kullanımlarında hatalar, düzgün bir dağılıma sahip olmalarına karşın, 3. dereceden polinom kullanımında yapılan hataların özellikle 1 rakamına benzetme sonucunda olduğu görülmektedir.

#### 4.4 Temel Bileşen Analizi ile Yapılan Uygulamalar

TBA'nın uygulanmasındaki ilk adım, eğitim kümesi kullanılarak bu kümeye ait TB'lerin hesaplanmasıdır. Bu aşamada, MNIST değer ve etiket verilerinin birleştirilmesinden oluşturulmuş olan eğitim verilerindeki etiket kısımlarının kullanılmaması gerekmektedir. Çünkü, etiket değerleri sistemde diğer sütunların ifade ettiği rakamı belirttiğinden, TBA içinde hesaplanmaz. Bundan dolayı yapılacak olan TBA için 60.000 satır ve 784 sütundan oluşan bir matris oluşturulmuştur.

TBA için kullanılacak matrise ait değerler alındıktan sonra kovaryans matrisin hesaplanması aşamasına geçilir. Bunun için, 60.000 satır ve 784 sütundan oluşan değer matrisinin transpozesi hesaplanır. Daha sonra matrisin transpozesi, kendisi ile çarpılarak 784 satır ve 784 sütundan oluşan kovaryans matris hesaplanır. Bu işlem yapılırken aynı zamanda sütunların toplamaları hesaplanır. Ardından hesaplanan ortalamalar kovaryans matris hesaplamasında kullanılarak sonucun çok büyük olması engellenir.

Kovaryans matriste hesaplandıktan sonra TB'ler hesaplanır. Kullandığımız örneklerin TB'leri hesaplanmış olan kovaryans matrisin özvektörleridir. Kovaryans matrisin özdeğer ve özvektörleri sayısal bir yöntem olan Jacobi metoduyla bulunmuştur.

TBA'nın son adımı olarak hesaplanan özdeğer ve özvektörler, özdeğerlere göre büyükten küçüğe doğru sıralanarak kaydedilmiştir. Verilerin tanımlanmasında özdeğeri büyük olan TB'ler yani özvektörler, özdeğeri küçük olanlara göre veriyi daha iyi bir şekilde ifade etmektedir.

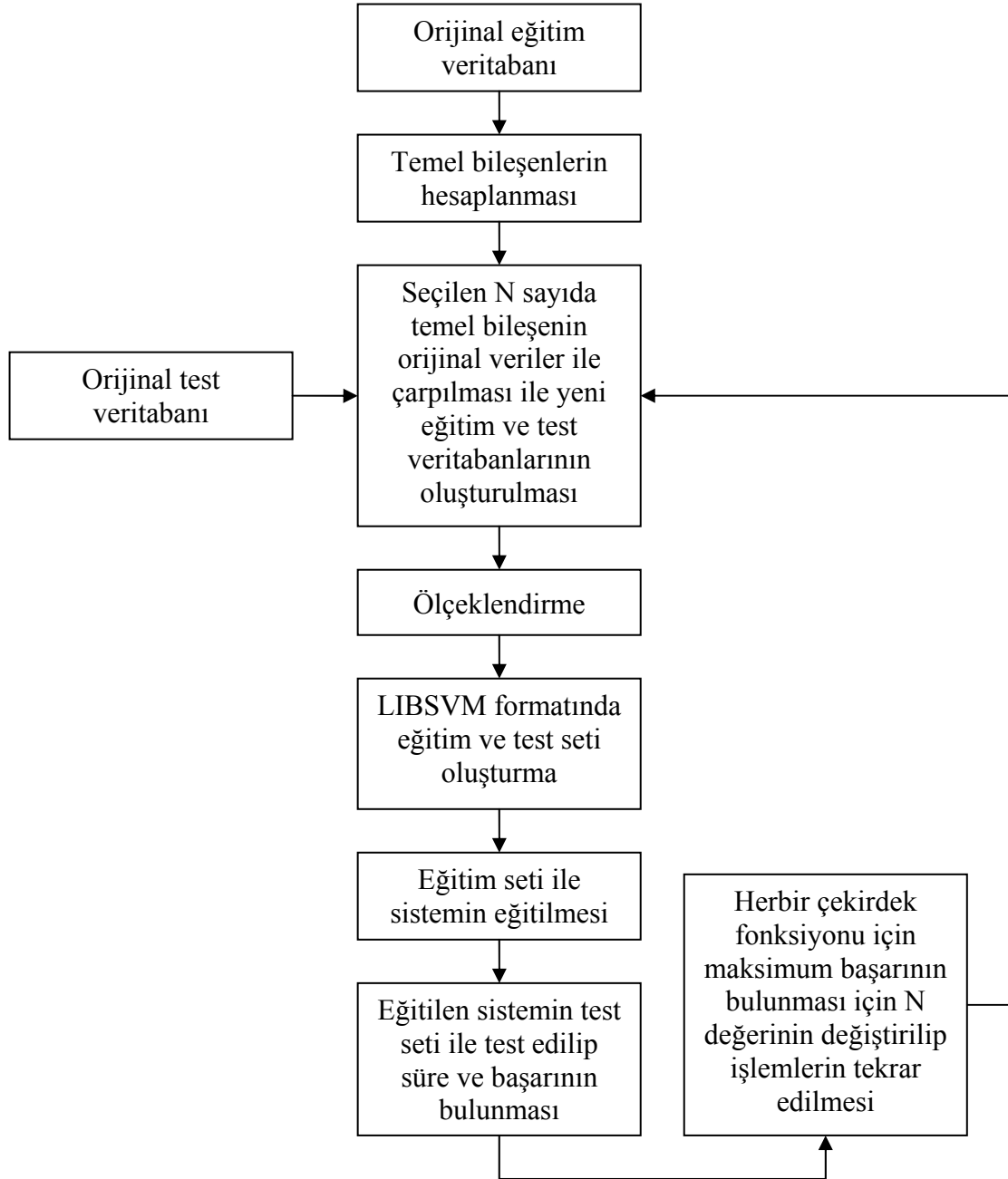
Sistemin öğrenmesinde yüksek başarı elde etmek için seçilecek olan TB sayısı deneme yanılma yoluyla bulunur. Çünkü, kullanılan TB sayısı arttıkça başarı bir süre artar, ancak belli bir artımdan sonra TB'ler kullanılarak oluşturulan yeni eğitim kümesinin başarısı düşmeye başlar. Bu durumda yapılacak olan sistemin başarısını en yüksek yapan TB sayısının bulunmasıdır. Bunun için eğitim ve test kümeleri belirli sayılarda TB'ler kullanılarak yeniden oluşturulur ve elde ettikleri başarıya göre başarının maksimum olduğu nokta bulunmaya çalışılır.

Herhangi bir sayıda TB kullanılarak, yeni eğitim ve test verilerinin hesaplanması şu şekilde olacaktır:

- Eğitim ve test veritabanları seçilmiş olan sayıda TB ile çarpılarak yeni eğitim ve test veritabanları elde edilir.

- Elde edilmiş olan yeni veritabanları DVM'lerin hızlı çalışması açısından [0,1] aralığına indirgenir.
- Ölçeklendirme işleminden geçmiş olan veritabanları LIBSVM yazılımı için girdi verisi haline getirilir. Burada orijinal verilerde olduğu gibi değeri sıfır olan birimler varsa, girdi içinde yer almaz.

TBA ile yapılan uygulamanın çalışması Şekil 4.6'da olduğu gibidir.



Şekil 4.6 TBA uygulanan sistemin çalışması

TBA ile elde edilen veritabanlarıyla yapılan testlerin sonuçları aşağıdaki gibidir.

Çizelge 4.6 Lineer çekirdek ve TBA kullanılarak elde edilen sonuçlar

| TB sayısı  | Eğitim süresi (sn) | Test süresi (sn) | Başarı (%)   |
|------------|--------------------|------------------|--------------|
| 1          | 623,59             | 36,20            | 31,32        |
| 3          | 447,78             | 53,58            | 49,28        |
| 7          | 247,97             | 32,23            | 79,93        |
| 16         | 217,94             | 49,02            | 89,22        |
| 34         | 235,61             | 49,72            | 92,89        |
| 53         | 207,03             | 64,23            | 93,51        |
| 100        | 436,98             | 106,38           | 94,30        |
| 125        | 506,44             | 130,75           | 94,31        |
| <b>150</b> | <b>589,58</b>      | <b>280,24</b>    | <b>94,48</b> |
| 175        | 680,53             | 178,77           | 94,42        |
| 199        | 762,00             | 210,20           | 94,36        |
| 250        | 958,01             | 254,78           | 94,20        |

Çizelge 4.7 3. dereceden polinom çekirdek ve TBA kullanılarak elde edilen sonuçlar

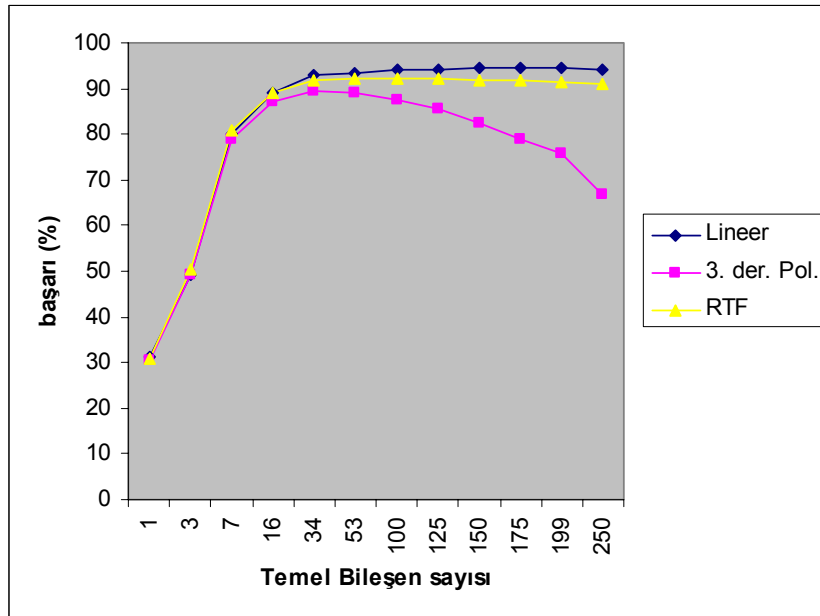
| TB sayısı | Eğitim süresi (sn) | Test süresi (sn) | Başarı (%)   |
|-----------|--------------------|------------------|--------------|
| 1         | 749,06             | 48,73            | 30,46        |
| 3         | 624,02             | 69,56            | 49,26        |
| 7         | 413,89             | 55,47            | 78,88        |
| 16        | 699,84             | 120,75           | 87,03        |
| <b>34</b> | <b>1339,00</b>     | <b>161,22</b>    | <b>89,33</b> |
| 53        | 2118,84            | 239,89           | 89,09        |
| 100       | 4409,98            | 444,26           | 87,43        |
| 125       | 5771,13            | 561,30           | 85,39        |
| 150       | 8050,38            | 703,97           | 82,27        |
| 175       | 8412,58            | 784,42           | 78,85        |
| 199       | 9796,00            | 904,66           | 75,94        |
| 250       | 12127,88           | 1082,17          | 66,63        |

Çizelge 4.8 RTF çekirdek ve TBA kullanılarak elde edilen sonuçlar

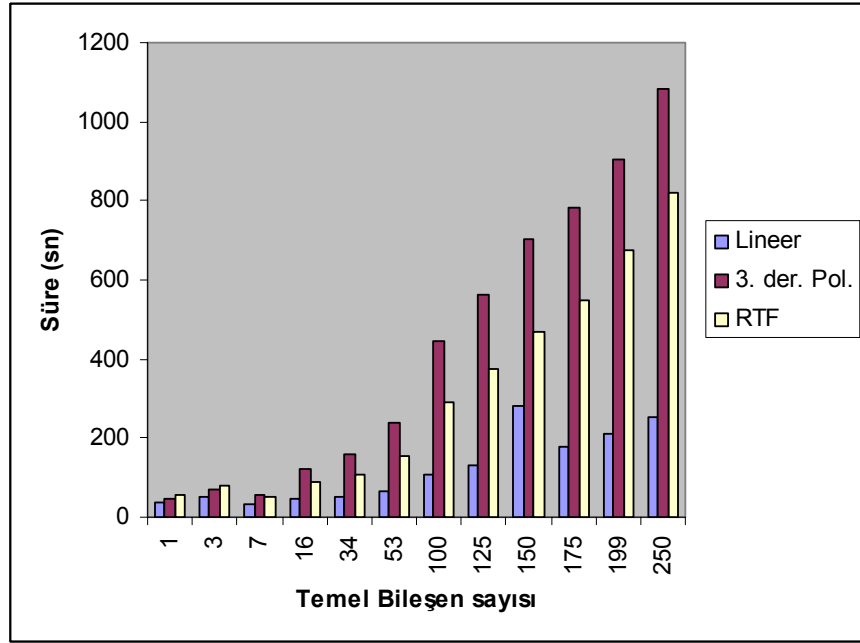
| TB sayısı | Eğitim süresi (sn) | Test süresi (sn) | Başarı (%)   |
|-----------|--------------------|------------------|--------------|
| 1         | 882,34             | 57,95            | 30,85        |
| 3         | 535,11             | 81,97            | 50,50        |
| 7         | 292,25             | 52,81            | 80,86        |
| 16        | 422,34             | 89,66            | 89,13        |
| 34        | 548,36             | 105,52           | 91,77        |
| <b>53</b> | <b>802,34</b>      | <b>152,72</b>    | <b>92,34</b> |
| 100       | 1601,44            | 290,14           | 92,26        |
| 125       | 2137,38            | 373,20           | 92,19        |
| 150       | 5193,77            | 467,58           | 91,87        |
| 175       | 3319,91            | 546,25           | 91,75        |
| 199       | 4031,26            | 673,22           | 91,41        |
| 250       | 5229,27            | 821,58           | 91,15        |

Çizelge 4.6, 4.7 ve 4.8'e bakılacak olursa TBA kullanılarak sistemin eğitilmesinde farklı çekirdek fonksiyonları için farklı sayıda TB ile en yüksek başarı değerinin elde edildiği görülecektir. Orijinal veritabanı kullanımı ile elde edilen sonuçlar ve TBA yapılarak elde edilen sonuçlar karşılaştırılacak olursa, lineer ve RTF çekirdeklerinde elde edilen sonuçların, orijinal yöntemle göre  $\pm \%1 - 2$ 'lik bir başarı fark oluşturduğu, polinom fonksiyonunda ise  $\%30$ 'a yaklaşan bir başarı artışı sağladığı görülmektedir. Aynı şekilde, orijinal veritabanının kullanımına göre elde edilen en yüksek başarıda  $\%0,02$ 'lik bir başarı artışı sağlanmıştır. En yüksek başarının elde edilmesinde ise, test aşamasında geçen süre yaklaşık 100 saniye civarında azalmıştır. Burada bahsedilen süre sadece LIBSVM'nin test için kullandığı süredir. Bunun yanısıra TBA için harcanan sürede sisteme dahil edilmelidir. TBA'nın yapılması, sistemin kullanılabilir hale getirilmesinden önce, bir defaya mahsus olmak üzere yapılacağından, eğitim aşaması için bir defalık zaman maliyeti getirecektir. Test aşaması için ise, test veritabanının TB'ler ile çarpılıp yeni test veritabanının hesaplanması yaklaşık olarak 80 saniye sürmektedir ki; bu durumda dahi orijinal test süresine göre 20 saniyelik bir performans artışı söz konusu olmaktadır. Bu sebeplerden dolayı, orijinal veri yerine TB'ler ile oluşturulmuş verilerin kullanıldığı bir sistemin kurulması hem hız hem de başarı açısından daha uygun olacaktır.

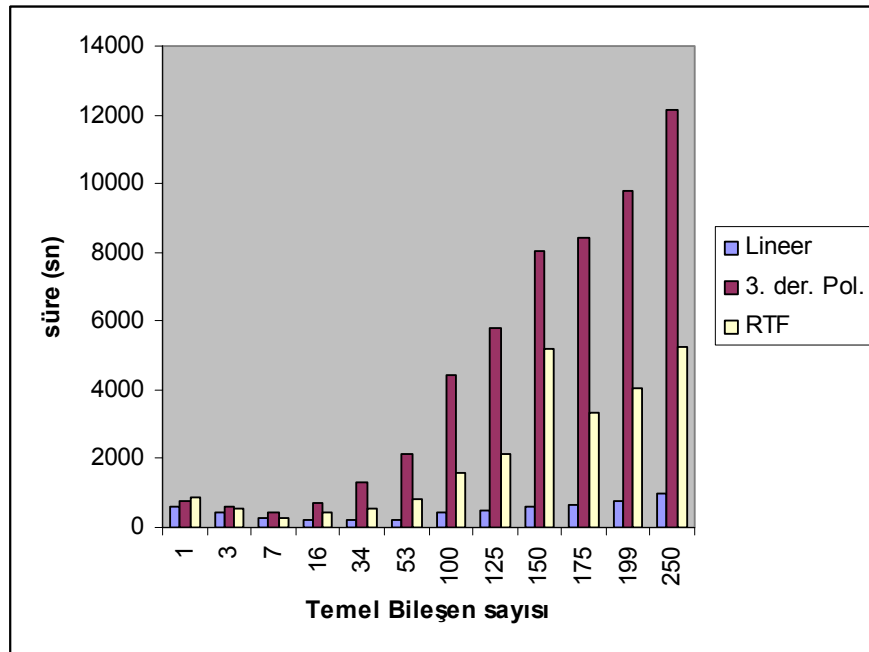
Yukarıdaki çizelgelerde belirtilen uygulama sonuçlarının TB sayılarına göre başarı ve süre histogramları, üç farklı çekirdek fonksiyonuna göre aşağıda belirtilmiştir.



Şekil 4.7 TB sayısına göre başarı değişimi



Şekil 4.8 TB sayısına göre test süresi değişimi



Şekil 4.9 TB sayısına göre eğitim süresi değişimi

Çizelge 4.9, 4.10 ve 4.11’de herbir çekirdek yöntemi için en yüksek başarımın elde edildiği TB’lerde yapılan hataların dağılımı incelenebilir.

Çizelge 4.9 150 TB ve lineer çekirdek ile yapılmış olan hataların dağılımı

|                  |   | HATALAR |    |    |    |    |    |    |    |    |    |          |
|------------------|---|---------|----|----|----|----|----|----|----|----|----|----------|
|                  |   | 0       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | $\Sigma$ |
| ORJINAL RAKAMLAR | 0 | -       | 0  | 1  | 1  | 1  | 7  | 3  | 2  | 1  | 0  | 16       |
|                  | 1 | 0       | -  | 2  | 2  | 1  | 2  | 3  | 2  | 3  | 0  | 15       |
|                  | 2 | 7       | 4  | -  | 7  | 10 | 5  | 10 | 8  | 14 | 4  | 69       |
|                  | 3 | 3       | 0  | 17 | -  | 1  | 18 | 1  | 10 | 14 | 6  | 70       |
|                  | 4 | 1       | 0  | 4  | 0  | -  | 1  | 6  | 2  | 2  | 23 | 39       |
|                  | 5 | 9       | 2  | 2  | 31 | 2  | -  | 9  | 1  | 17 | 4  | 77       |
|                  | 6 | 5       | 2  | 9  | 1  | 4  | 15 | -  | 1  | 0  | 0  | 37       |
|                  | 7 | 0       | 9  | 20 | 5  | 12 | 1  | 0  | -  | 1  | 19 | 67       |
|                  | 8 | 5       | 4  | 5  | 17 | 6  | 30 | 9  | 8  | -  | 7  | 91       |
|                  | 9 | 5       | 5  | 0  | 8  | 27 | 6  | 0  | 15 | 5  | -  | 71       |
| $\Sigma$         |   | 35      | 26 | 60 | 72 | 64 | 85 | 41 | 49 | 57 | 63 | 552      |

Çizelge 4.10 34 TB ve 3. dereceden polinom çekirdek ile yapılmış olan hataların dağılımı

|                  |   | HATALAR |     |    |     |     |     |     |    |     |     |          |
|------------------|---|---------|-----|----|-----|-----|-----|-----|----|-----|-----|----------|
|                  |   | 0       | 1   | 2  | 3   | 4   | 5   | 6   | 7  | 8   | 9   | $\Sigma$ |
| ORJINAL RAKAMLAR | 0 | -       | 0   | 0  | 1   | 1   | 22  | 15  | 1  | 4   | 0   | 44       |
|                  | 1 | 0       | -   | 1  | 4   | 0   | 4   | 5   | 0  | 14  | 0   | 28       |
|                  | 2 | 8       | 23  | -  | 18  | 20  | 7   | 20  | 20 | 40  | 8   | 164      |
|                  | 3 | 4       | 6   | 17 | -   | 0   | 33  | 8   | 14 | 27  | 14  | 123      |
|                  | 4 | 1       | 5   | 3  | 0   | -   | 2   | 11  | 2  | 6   | 78  | 108      |
|                  | 5 | 6       | 12  | 3  | 46  | 11  | -   | 26  | 6  | 18  | 7   | 135      |
|                  | 6 | 8       | 4   | 5  | 1   | 14  | 22  | -   | 1  | 1   | 0   | 56       |
|                  | 7 | 1       | 35  | 16 | 2   | 13  | 2   | 2   | -  | 4   | 35  | 110      |
|                  | 8 | 7       | 18  | 6  | 25  | 9   | 47  | 18  | 6  | -   | 23  | 159      |
|                  | 9 | 8       | 10  | 2  | 9   | 56  | 13  | 1   | 25 | 16  | -   | 140      |
| $\Sigma$         |   | 43      | 113 | 53 | 106 | 124 | 152 | 106 | 75 | 130 | 165 | 1067     |

Çizelge 4.11 53 TB ve RTF ile yapılmış olan hataların dağılımı

|                  |   | HATALAR |    |    |    |     |     |    |    |    |     |          |
|------------------|---|---------|----|----|----|-----|-----|----|----|----|-----|----------|
|                  |   | 0       | 1  | 2  | 3  | 4   | 5   | 6  | 7  | 8  | 9   | $\Sigma$ |
| ORJINAL RAKAMLAR | 0 | -       | 0  | 0  | 1  | 2   | 7   | 5  | 2  | 2  | 0   | 19       |
|                  | 1 | 0       | -  | 2  | 2  | 1   | 3   | 4  | 0  | 8  | 0   | 20       |
|                  | 2 | 6       | 5  | -  | 11 | 19  | 9   | 14 | 16 | 31 | 9   | 120      |
|                  | 3 | 3       | 0  | 21 | -  | 0   | 24  | 3  | 12 | 20 | 6   | 89       |
|                  | 4 | 1       | 0  | 2  | 0  | -   | 3   | 9  | 3  | 3  | 44  | 65       |
|                  | 5 | 7       | 7  | 5  | 36 | 10  | -   | 16 | 3  | 13 | 5   | 102      |
|                  | 6 | 8       | 3  | 7  | 1  | 12  | 14  | -  | 1  | 1  | 0   | 47       |
|                  | 7 | 2       | 20 | 18 | 5  | 12  | 3   | 0  | -  | 3  | 25  | 88       |
|                  | 8 | 4       | 9  | 6  | 14 | 8   | 36  | 14 | 7  | -  | 13  | 111      |
|                  | 9 | 7       | 7  | 1  | 9  | 41  | 11  | 1  | 20 | 8  | -   | 105      |
| $\Sigma$         |   | 38      | 51 | 62 | 79 | 105 | 110 | 66 | 64 | 89 | 102 | 766      |

#### 4.5 Bağımsız Bileşen Analizi ile Yapılan Uygulamalar

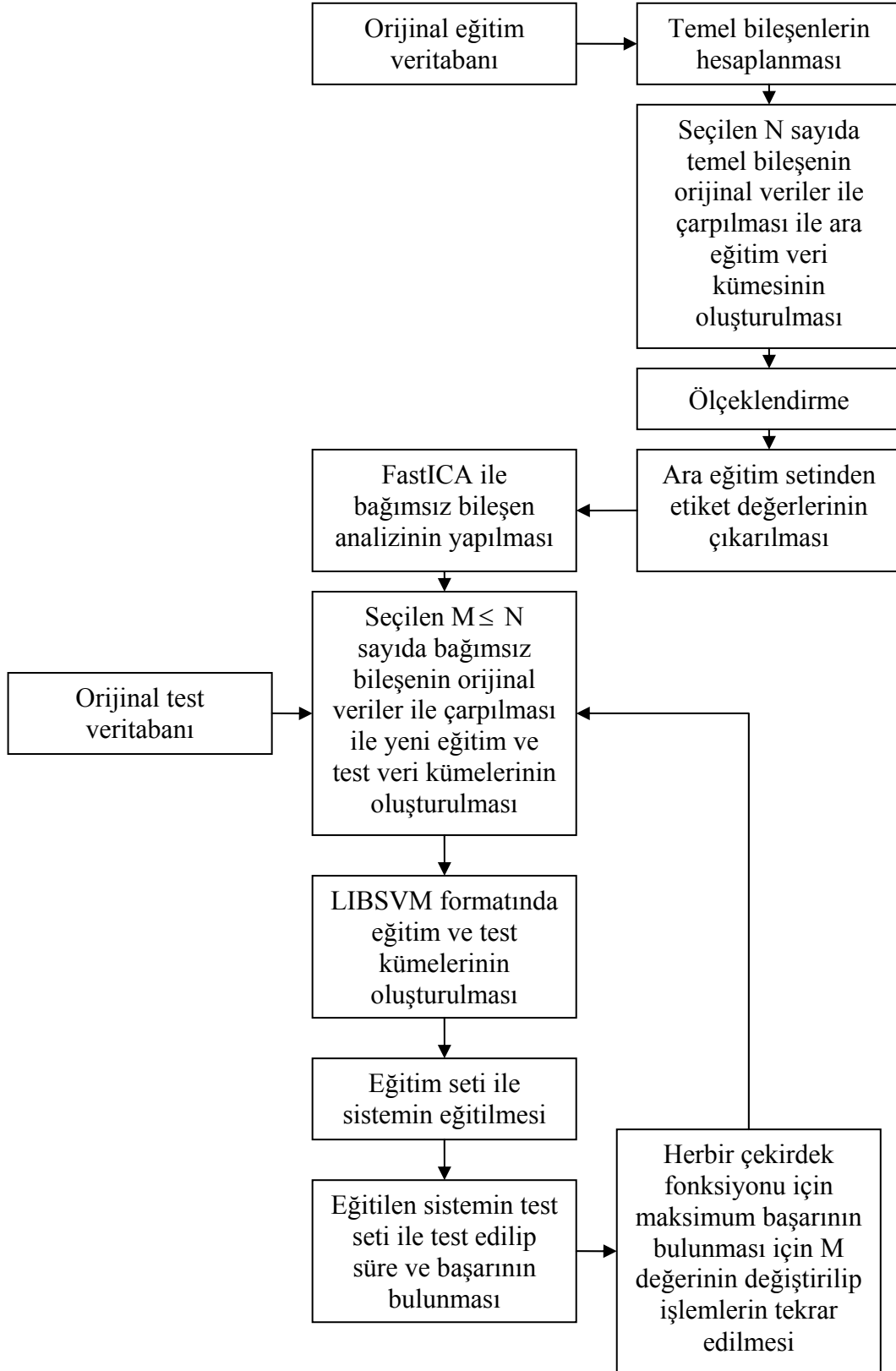
TBA ile sistem başarısının ölçülmesinin ardından, sistemin BBA ile başarısının hesaplanması aşamasına geçilmiştir. Bu aşamada yapılacak olan orijinal eğitim veritabanının BB'lerin hesaplanmasıdır. Ancak uygulama sırasında BB'leri hesaplamak için kullanılmış olan FastICA yazılımı orijinal eğitim veritabanının boyutunun büyük olmasından dolayı BB'leri hesaplayamamıştır. Bundan dolayı BB'leri bulabilmek için orijinal veritabanı yerine TB'leri hesaplanmış olan ve bir önceki aşamada kullanılmış olan veritabanları FastICA programı için girdi verileri olarak kullanılmıştır.

BB'lerin hesaplanmasında karşılaşılan ilk problem, TB'lerin sonucuna göre oluşturulacak olan ara katman veritabanında kaç sütun olacağı, dolayısıyla kaç TB'nin seçileceği problemidir. Uygulama aşamasında ilk olarak TBA için seçilen en yüksek eleman sayısı olan 250 TB ile oluşturulmuş, 60.000 satır ve 250 sütunluk eğitim veritabanı için BB hesaplaması yapılmıştır. Ardından herbir çekirdek fonksiyonu için en yüksek başarıyı sağlamış olan 150, 34 ve 53 elemanlı TB'lerle oluşturulmuş eğitim veritabanları kullanılarak, bu verilere ait BB'ler hesaplanmıştır. Oluşturulan ara katman veritabanları  $[0,1]$  aralığına ölçeklendirilmiştir. Ardından, TB'lerle oluşturulan ve  $[0,1]$  arasına ölçeklendirilen verilerin FastICA programında kullanılmasından önce, bir ön işlem olarak verilerin önündeki etiket değerleri, BB'lerin hesabında girdi değeri olarak bulunmaması için kaldırılmıştır.

BB'ler hesaplandıktan sonra uygulamada karşılaşılan ikinci problem hesaplanan BB'lerden kaç tanesi ile yeni eğitim ve test veri kümelerinin oluşturulacağıdır. Bu aşamada başarıyı en fazla yapacak olan bileşen sayısının bulunması için, elde edilen BB'ler, TB'lerde olduğu gibi deneme yanılma yoluyla başarısı ölçülerek bulunur.

Herbir sonuç için öncelikle eğitim ve test veri kümeleri, seçilen sayıda BB ile çarpılarak yeni eğitim ve test veri kümeleri elde edilir. Ardından DVM'nin hesaplama aşamasında çok büyük rakamlar oluşmaması açısından elde edilen yeni veritabanları  $[0,1]$  değerleri arasına ölçeklendirilir. Sistem daha önceki uygulamalarda olduğu gibi üç farklı çekirdek fonksiyonu ile çalıştırılarak başarı ve zaman değerleri ölçülür. BB'ler için sistemin çalışması Şekil 4.10'da olduğu gibidir.





Şekil 4.10 BBA uygulanan sistemin çalışması

Sistemin çalışması sonucu üç çekirdek fonksiyonu için elde edilen sonuçlar aşağıdaki gibidir.

Çizelge 4.12 Lineer çekirdek ve BBA kullanılarak elde edilen sonuçlar

| TB sayısı  | BB sayısı  | Eğitim süresi (sn) | Test süresi (sn) | Başarı (%)   |
|------------|------------|--------------------|------------------|--------------|
| 34         | 34         | 231,33             | 48,16            | 92,90        |
| 53         | 53         | 277,78             | 64,66            | 93,53        |
| 150        | 50         | 947,61             | 144,20           | 76,03        |
| 150        | 100        | 593,02             | 139,69           | 92,02        |
| 150        | 125        | 559,16             | 140,41           | 93,80        |
| <b>150</b> | <b>150</b> | <b>610,41</b>      | <b>159,36</b>    | <b>94,49</b> |
| 250        | 125        | 791,64             | 191,67           | 90,03        |
| 250        | 150        | 756,01             | 190,94           | 91,95        |
| 250        | 175        | 788,94             | 204,92           | 92,91        |
| 250        | 200        | 828,70             | 225,36           | 93,66        |
| 250        | 225        | 889,33             | 238,42           | 94,14        |
| 250        | 250        | 959,99             | 256,20           | 94,15        |

Çizelge 4.13 Polinom çekirdek ve BBA kullanılarak elde edilen sonuçlar

| TB sayısı | BB sayısı | Eğitim süresi (sn) | Test süresi (sn) | Başarı (%)   |
|-----------|-----------|--------------------|------------------|--------------|
| 34        | 34        | 1350,34            | 157,13           | 88,64        |
| <b>53</b> | <b>53</b> | <b>2160,00</b>     | <b>239,95</b>    | <b>88,78</b> |
| 150       | 50        | 2932,61            | 258,74           | 67,38        |
| 150       | 100       | 5326,59            | 472,05           | 80,48        |
| 150       | 125       | 5997,53            | 558,84           | 81,72        |
| 150       | 150       | 7533,92            | 683,14           | 79,98        |
| 250       | 125       | 6383,78            | 596,33           | 73,12        |
| 250       | 150       | 7484,06            | 669,38           | 73,37        |
| 250       | 175       | 8653,61            | 787,99           | 72,91        |
| 250       | 200       | 9785,97            | 874,22           | 71,98        |
| 250       | 225       | 10974,06           | 994,34           | 69,22        |
| 250       | 250       | 12283,86           | 1081,75          | 64,60        |

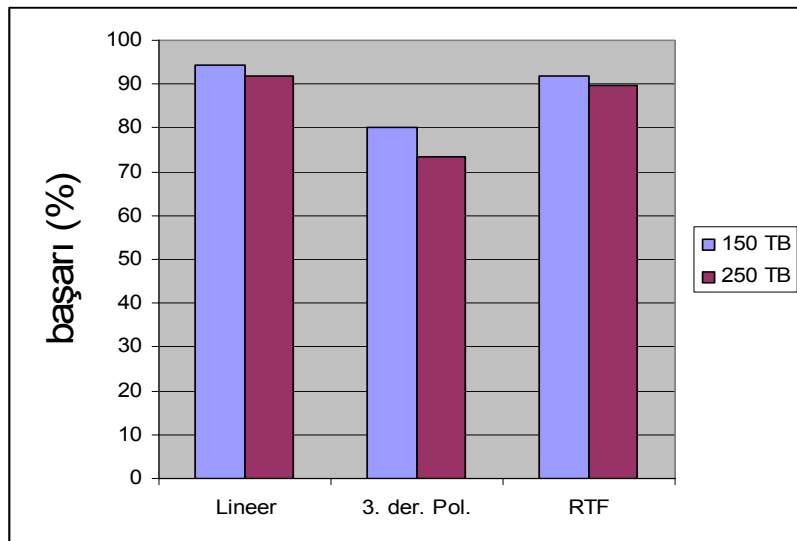
Çizelge 4.14 RTF çekirdek ve BBA kullanılarak elde edilen sonuçlar

| TB sayısı | BB sayısı | Eğitim süresi (sn) | Test süresi (sn) | Başarı (%)   |
|-----------|-----------|--------------------|------------------|--------------|
| 34        | 34        | 564,19             | 102,27           | 91,72        |
| <b>53</b> | <b>53</b> | <b>814,86</b>      | <b>154,67</b>    | <b>92,32</b> |
| 150       | 50        | 1567,55            | 225,05           | 74,96        |
| 150       | 100       | 2047,41            | 337,67           | 89,51        |
| 150       | 125       | 2261,89            | 385,13           | 91,36        |
| 150       | 150       | 2805,33            | 467,56           | 91,82        |
| 250       | 125       | 2762,26            | 452,67           | 87,76        |
| 250       | 150       | 3093,81            | 505,11           | 89,70        |
| 250       | 175       | 3531,39            | 589,09           | 90,40        |
| 250       | 200       | 4023,69            | 649,02           | 90,95        |
| 250       | 225       | 4611,86            | 751,98           | 91,15        |
| 250       | 250       | 5193,77            | 815,50           | 91,04        |

BBA kullanılarak yapılan uygulamaların sonuçları incelendiğinde, oluşturulan sistemin TB'lere oranla başarıyı  $\pm \%0.5$  oranında değiştirdiği görülmektedir. Bununla beraber polinom ve RTF kullanan çekirdeklerde başarının ufak bir miktarda düşmesinin yanı sıra test süresinin de arttığı görülmektedir. Sadece lineer çekirdek için başarı  $\%0.01$  artarken test süresi 130 saniye civarında düşmektedir. Bu süre sadece LIBSVM'nin test aşamasında geçen süredir. Bu süreye orijinal test verilerinin BBA öncesindeki önışlem süresinin ve BB'ler ile çarpılıp, ölçeklendirilmesi sürelerinin de eklenmesi gerekmektedir. Bu sürelerde eklendiğinde, test aşamasının, TB'ler ile yapılan teste göre yaklaşık olarak 15 saniye civarında daha avantajlı hale geldiği görülmüştür.

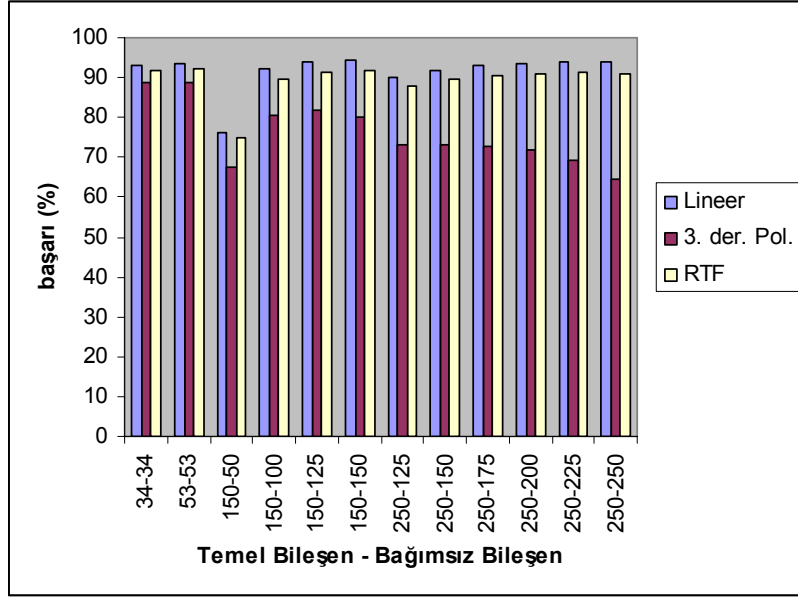
Sistemin eğitim aşamasında ise BB'lerin hesaplanması için geçen sürenin oldukça yüksek olduğu saptanmıştır. 150 TB ile oluşturulan ara eğitim verisinden BB'lerin hesaplanması, kullanılan donanımda 3 saat civarındayken, 250 TB ile oluşturulan ara eğitim verisinin BB'lerinin hesaplanması, yaklaşık olarak 20 saat civarında sürmüştür. Ancak bu sadece ilk kurulum maliyeti olduğundan göz ardı edilebilecek bir süredir.

BB'lerin hesaplanmasında karşılaşılan en büyük zorluk, hesaplama için, öncelikle TBA'e ihtiyaç duymasıdır. Bu noktada BB hesabı için, kullanılması gereken optimum TB sayısının ne olduğu problemi karşımıza çıkar. Bu problemin çözümü ise ancak deneme yanılma ile bulunabilir. Çünkü sonuçların yer aldığı çizelgelere bakarsak, hem 250 TB, hem de 150 TB kullanılarak hesaplanan BB'lerden, 150 tanesi kullanılarak sistem çalıştırıldığında, elde edilen başarıların farklı olduğu görülür.

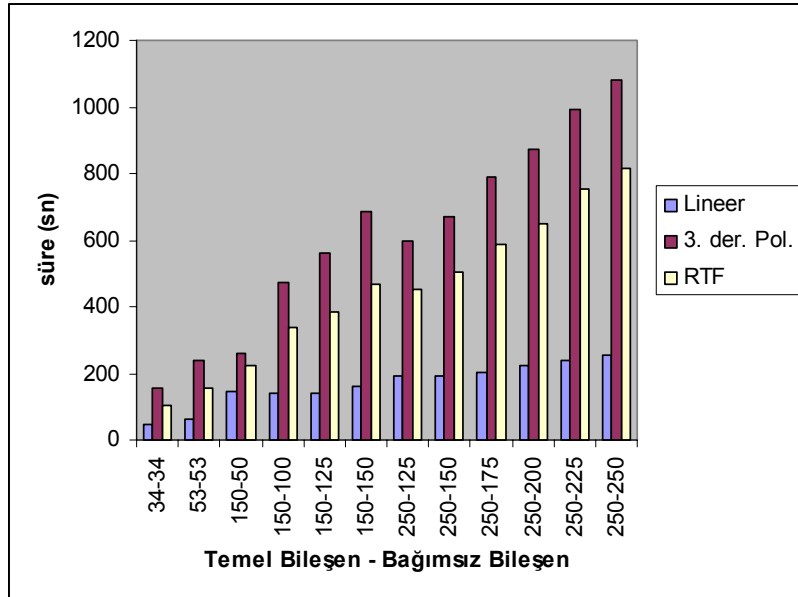


Şekil 4.11 Farklı sayıda TB ile hesaplanmış, 150 BB için başarı değişimi

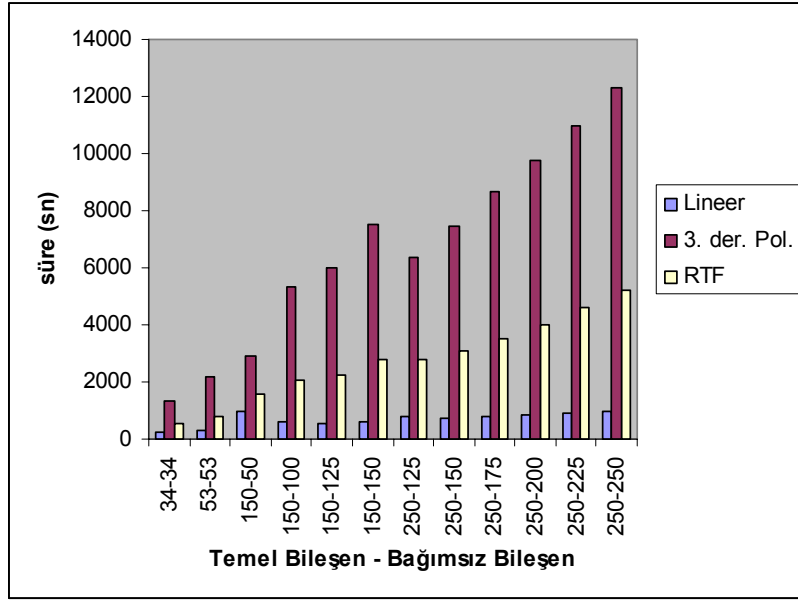
Çekirdek fonksiyonlarına göre başarı ve süre değişimi histogramları aşağıdaki gibi elde edilmiştir.



Şekil 4.12 BB sayısına göre başarı değişimi



Şekil 4.13 BB sayısına göre test süresi değişimi



Şekil 4.14 BB sayısına göre eğitim süresi değişimi

Kullanılan BB sayısı, TB'lere göre elde edildiği için Şekil 4.12, 4.13 ve 4.14'teki histogramların yatay eksenlerinde, BB'lerin elde edilmesinde kullanılan ara eğitim veri kümesinin kaç TB'den oluşturulduğu da ifade edilmektedir.

Çizelge 4.15, 4.16 ve 4.17'de herbir çekirdek fonksiyonu için en yüksek başarımın elde edildiği BB'lerde oluşan hatalar gösterilmektedir.

Çizelge 4.15 150 BB ve lineer çekirdek ile yapılmış olan hataların dağılımı

|                  |   | HATALAR |    |    |    |    |    |    |    |    |    |     |
|------------------|---|---------|----|----|----|----|----|----|----|----|----|-----|
|                  |   | 0       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | Σ   |
| ORJINAL RAKAMLAR | 0 | -       | 0  | 1  | 1  | 1  | 7  | 3  | 1  | 1  | 0  | 15  |
|                  | 1 | 0       | -  | 2  | 2  | 0  | 2  | 3  | 2  | 3  | 0  | 14  |
|                  | 2 | 7       | 3  | -  | 7  | 10 | 5  | 9  | 8  | 13 | 4  | 66  |
|                  | 3 | 3       | 1  | 17 | -  | 1  | 19 | 1  | 10 | 14 | 6  | 72  |
|                  | 4 | 1       | 0  | 3  | 0  | -  | 1  | 6  | 2  | 2  | 23 | 38  |
|                  | 5 | 8       | 2  | 2  | 32 | 2  | -  | 10 | 1  | 17 | 4  | 78  |
|                  | 6 | 5       | 2  | 9  | 1  | 4  | 15 | -  | 1  | 0  | 0  | 37  |
|                  | 7 | 0       | 10 | 19 | 5  | 11 | 2  | 0  | -  | 1  | 19 | 67  |
|                  | 8 | 5       | 5  | 5  | 17 | 6  | 29 | 10 | 8  | -  | 8  | 93  |
|                  | 9 | 6       | 5  | 0  | 8  | 26 | 6  | 0  | 15 | 5  | -  | 71  |
| Σ                |   | 35      | 28 | 58 | 73 | 61 | 86 | 42 | 48 | 56 | 64 | 551 |

Çizelge 4.16 53 BB ve 3. dereceden polinom çekirdek ile yapılmış olan hataların dağılımı

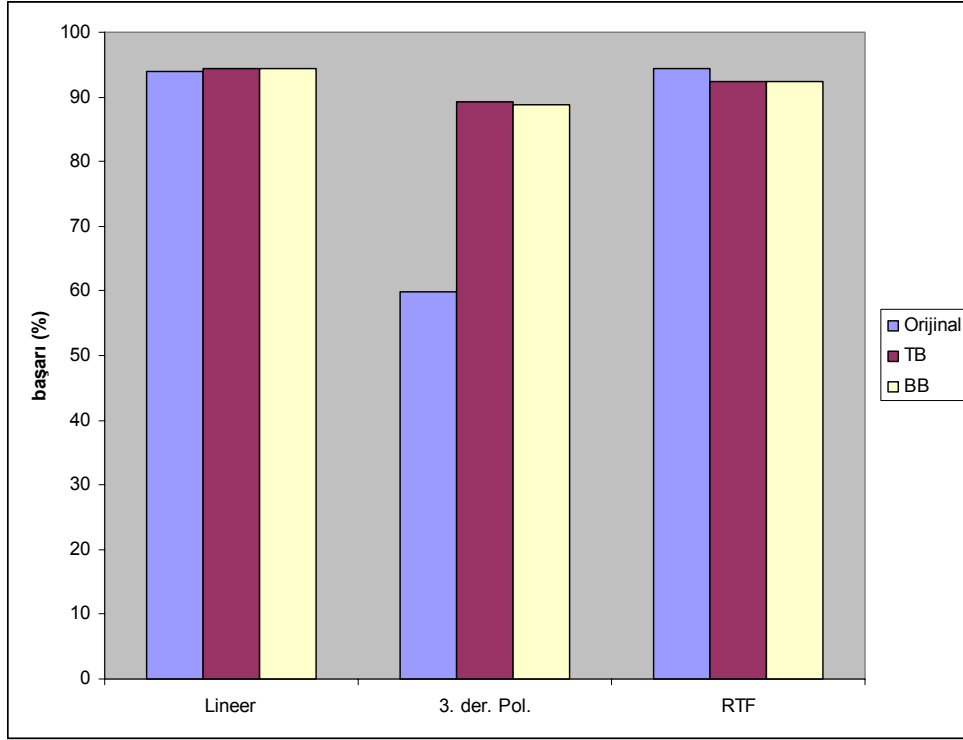
|                  |   | HATALAR |     |    |     |     |     |    |     |    |     |          |
|------------------|---|---------|-----|----|-----|-----|-----|----|-----|----|-----|----------|
|                  |   | 0       | 1   | 2  | 3   | 4   | 5   | 6  | 7   | 8  | 9   | $\Sigma$ |
| ORJINAL RAKAMLAR | 0 | -       | 0   | 0  | 3   | 0   | 10  | 11 | 2   | 2  | 0   | 28       |
|                  | 1 | 0       | -   | 1  | 3   | 0   | 2   | 5  | 0   | 8  | 1   | 20       |
|                  | 2 | 10      | 48  | -  | 26  | 20  | 7   | 12 | 31  | 36 | 9   | 199      |
|                  | 3 | 9       | 8   | 17 | -   | 1   | 25  | 7  | 17  | 20 | 13  | 117      |
|                  | 4 | 1       | 7   | 2  | 0   | -   | 2   | 9  | 1   | 3  | 82  | 107      |
|                  | 5 | 8       | 12  | 4  | 51  | 19  | -   | 21 | 5   | 12 | 8   | 140      |
|                  | 6 | 10      | 5   | 6  | 1   | 15  | 26  | -  | 4   | 3  | 0   | 70       |
|                  | 7 | 1       | 45  | 16 | 4   | 12  | 5   | 1  | -   | 1  | 33  | 118      |
|                  | 8 | 4       | 30  | 4  | 26  | 13  | 52  | 18 | 5   | -  | 33  | 185      |
|                  | 9 | 6       | 10  | 4  | 13  | 42  | 13  | 2  | 42  | 6  | -   | 138      |
| $\Sigma$         |   | 49      | 165 | 54 | 127 | 122 | 142 | 86 | 107 | 91 | 179 | 1122     |

Çizelge 4.17 53 BB ve RTF çekirdek ile yapılmış olan hataların dağılımı

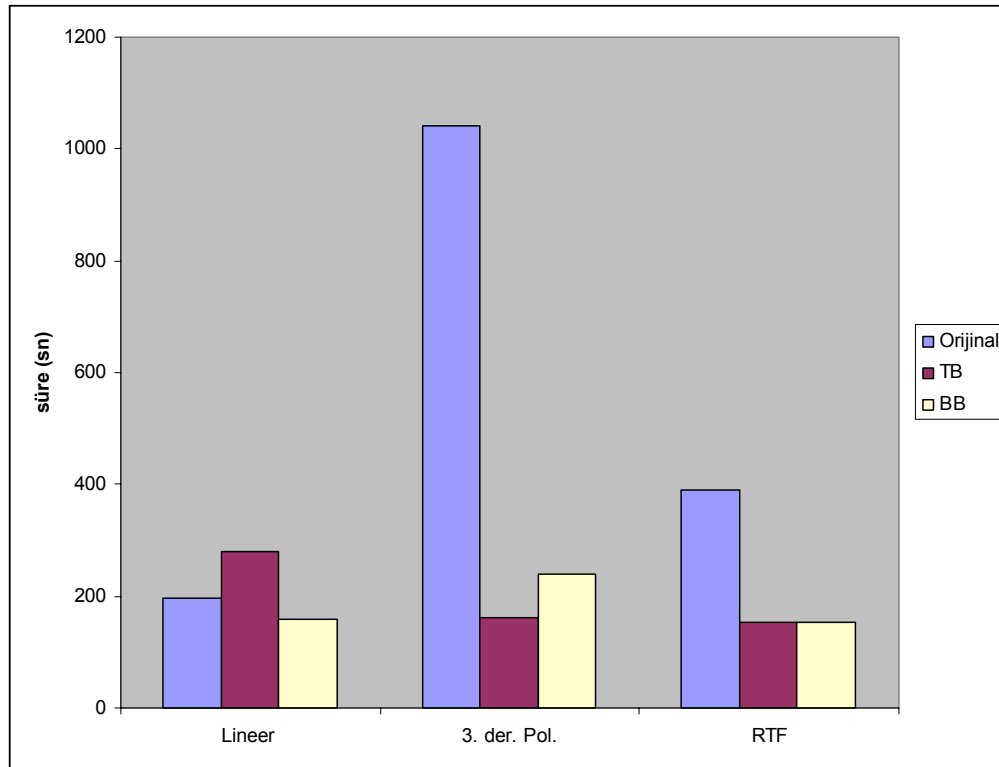
|                  |   | HATALAR |    |    |    |     |     |    |    |    |     |          |
|------------------|---|---------|----|----|----|-----|-----|----|----|----|-----|----------|
|                  |   | 0       | 1  | 2  | 3  | 4   | 5   | 6  | 7  | 8  | 9   | $\Sigma$ |
| ORJINAL RAKAMLAR | 0 | -       | 0  | 0  | 1  | 1   | 7   | 5  | 2  | 2  | 0   | 18       |
|                  | 1 | 0       | -  | 1  | 3  | 1   | 3   | 4  | 0  | 10 | 0   | 22       |
|                  | 2 | 6       | 6  | -  | 12 | 19  | 8   | 14 | 16 | 31 | 9   | 121      |
|                  | 3 | 3       | 0  | 21 | -  | 1   | 27  | 2  | 11 | 20 | 5   | 90       |
|                  | 4 | 1       | 0  | 2  | 0  | -   | 3   | 9  | 2  | 3  | 42  | 62       |
|                  | 5 | 7       | 7  | 5  | 37 | 10  | -   | 13 | 3  | 14 | 5   | 101      |
|                  | 6 | 9       | 3  | 8  | 0  | 12  | 14  | -  | 2  | 2  | 0   | 50       |
|                  | 7 | 2       | 20 | 19 | 4  | 10  | 3   | 0  | -  | 3  | 31  | 92       |
|                  | 8 | 5       | 9  | 5  | 14 | 8   | 35  | 14 | 7  | -  | 12  | 109      |
|                  | 9 | 7       | 7  | 1  | 10 | 42  | 10  | 1  | 20 | 5  | -   | 103      |
| $\Sigma$         |   | 40      | 52 | 62 | 81 | 104 | 110 | 62 | 63 | 90 | 104 | 768      |

#### 4.6 Yöntemlerin Karşılaştırması

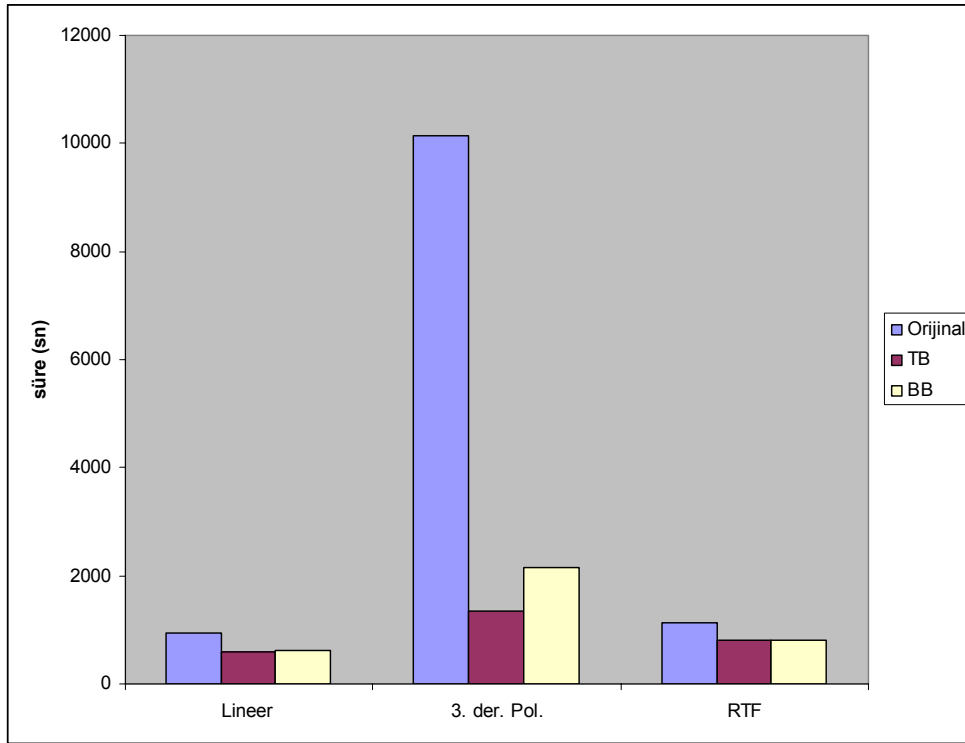
Bu bölüm altında uygulama aşamasında yapılmış olan, el yazısı tanıma amacıyla kurulan sistemin karşılaştırması yapılmaktadır. Yapılan uygulamanın öncesinde orijinal verinin direkt olarak kullanılması yerine TB veya BB'leri kullanılarak başarının daha yüksek olacağı varsayılmıştı. Yapılan uygulama ile başlangıçta yapılan varsayımın doğru olduğu görülmektedir. Başarı oranları ve sürelerdeki değişim Şekil 4.15, 4.16 ve 4.17'de incelenebilir.



Şekil 4.15 Kullanılan verinin türüne göre farklı çekirdek fonksiyonlarının başarı değışimleri



Şekil 4.16 Kullanılan verinin türüne göre farklı çekirdek fonksiyonlarının test süresi değışimleri



Şekil 4.17 Kullanılan verinin türüne göre farklı çekirdek fonksiyonlarının eğitim süresi değişimleri

Yukarıdaki şekillerde herbir yöntemin en başarılı sonuçlar ürettiği TB ve BB sonuçları kullanılmıştır. Sonuçlar incelendiğinde BB'lerin kullanılması durumunda lineer DVM'nin başarıda en yüksek sonucu elde ettiği görülmektedir. Bununla beraber, bu eğitim ve test verileri için lineer DVM'ler eğitim sürecinde en kısa süreye, test aşamasında ise oldukça düşük bir süreye sahip olmuştur. Polinom fonksiyonu kullanan DVM'lerin elde ettiği sonuçlar, lineer DVM'lere oranla düşük olmasına karşı, TB'ler ve BB'lerin kullanılması durumunda %29,38'lik başarı artışı sağlamıştır. RTF'ye sahip DVM'lerde ise TB'ler ve BB'ler kullanılması durumunda başarı %2 civarında düşmüştür, ancak test süresinde %40'lık bir artış olmuştur.

Yapılan bu çalışmaya başlarken temel düşünce, TB ve BB'lerin DVM'lerde başarıyı arttıracak yönündeydi. Elde edilen sonuçlar bunu destekler nitelikte olmuştur. Özellikle polinom çekirdekli DVM'lerde başarının %29,38 gibi bir derecede arttığı, aynı zamanda test süresinin %86,8 oranında düştüğü tespit edilmiştir. Sadece RTF kullanılan DVM'lerin başarısında düşme olmuştur. Sonuçlara bakarak, DVM ile karakter tanımda TB ve BB'lerin mutlaka kullanılması gerektiği sonucuna varılmaktadır. Karakter tanıma alanında en yüksek başarıyı elde eden yöntemin kullanılacağı varsayılırsa, TBA veya BBA'nın lineer fonksiyon için kullanılması gerektiği söylenebilir.



## 5. SONUÇLAR

Karakter tanıma, günümüzde birçok alanda önemini arttırmış bir konudur. El yazısı ile doldurulmuş formların bilgisayarlara aktarılmasından, el yazısı kullanılarak işlem yapılan cihazlara kadar pek çok alanda uygulamaları kullanılmaktadır.

Tez kapsamında yapılan araştırmalarda kullanılan veriler, uygulanan metodlar, elde edilen sonuçlar ve değerlendirilmeleri hakkında aşağıdakiler söylenebilir:

1. Tez kapsamında, karakter tanıma uygulaması alanında araştırma yapabilmek için, el yazısı rakamlardan oluşan bir veritabanı üzerinde denemeler yapılmıştır.
2. Ele alınan örnek veritabanının bilgisayar tarafından öğrenilmesi aşamasında DVM yöntemi kullanılmıştır. DVM ile yapılan sınıflandırma işleminde lineer destek vektör makineleri ile birlikte polinom ve RTF çekirdekleri de kullanılmıştır.
3. DVM ile tanıma işlemi yapılırken verilerin  $[0,1]$  aralığında ölçeklendirilmesinin başarıyı arttırdığı gözlemlenmiştir.
4. TBA ve BBA yöntemleri kullanılarak verilerin boyutlarının ufaltılması ile başarının orijinal verilere göre daha yüksek olduğu görülmüştür.
5. Yine TBA ve BBA yöntemlerinin kurulan sistemin çalışması sonucu yapılan testlerde test süresini büyük ölçüde düşürdüğü gözlemlenmiştir. Eğitim aşamasında ise bu yöntemlerin hesaplanabilmesi yüzünden artma olmaktadır. Ancak eğitimin bir kereye mahsus olmak üzere yapılmasından dolayı, kurulum maliyeti olarak kabul edilmiştir.
6. Kullanılan veriler birçok kişiye ait el yazısı karakterlerden oluştuğundan dolayı, TBA yönteminin hafızalama başarısı BBA'ya göre daha yüksek olduğundan dolayı BBA'ya göre genel olarak daha başarılı olduğu, BB'lerin ya TBA'ya göre daha düşük başarıya sahip olduğu ya da sadece lineer çekirdek için en çok %0,01 oranında arttırabildiği görülmüştür. Veriler tek kişiye ait örneklerden oluşmuş olsaydı, BBA'nın başarısının daha yüksek olacağı düşünülmektedir.
7. Kullanılan çekirdek fonksiyonlarına göre bakıldığında lineer DVM ve RTF çekirdekli DVM'de başarıda temel bileşen ve bağımsız bileşenler  $\pm 1 - 2$  'lik başarı değişimleri yaparken polinom çekirdekli DVM'de başarı %30 yakın bir miktarda arttırılmıştır.
8. Karakter tanıma açısından optimal yöntemin lineer DVM ile BB'lerin kullanılması sonucunda olduğu gözlemlenmiştir. 5. maddede de belirtildiği gibi BBA uygulanması

durumunda eğitim süresinde süre artışı olmuştur, ancak test süresi orijinal örnekler ile yapılan teste göre oldukça kısalmıştır. Bundan dolayı ilk kurulum maliyeti önemsenmemiştir.

9. Tez içersinde tanıma hatalarının kullanılan örneklerdeki yazım bozukluğundan dolayı olduğu görülmüştür. Örneğin;



esasinda 7 rakamına ait örnek iken test sonucunda değeri 1 olarak bulunmuştur.



esasinda 5 rakamına ait örnek iken test sonucunda değeri 3 olarak bulunmuştur.



esasinda 6 rakamına ait örnek iken test sonucunda değeri 0 olarak bulunmuştur.

Ayrıca, ileride bu konu ile yapılan çalışmalarda orijinal veri ve/veya TB/BB uygulanmış orijinal veri kullanılması yerine, bu verilerin fourier transformasyonu veya başka bir dönüşüm yöntemi ile iki boyutlu görüntü verisinden tek boyutlu sinyale dönüştürülmesi durumunda başarının arttırılabileceği düşünülmektedir.

**KAYNAKLAR**

- Abe S., (2005), Support Vector Machines for Pattern Classification, Springer, London
- Alpaydın E., (2004), Introduction to Machine Learning, MIT Press, Massachusetts.
- Atkinson K. E., (1989), An Introduction to Numerical Analysis, Second Edition, John Wiley & Sons Inc., New York.
- Burges C. J. C., (1998), "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, 2: 121-167.
- Chang C. C. ve Lin C. J., (2001), "LIBSVM: A Library For Support Vector Machines".
- Cristianini N. ve Shawe-Taylor J., (2000), An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, New York.
- Dong J. X., Krzyzak A. ve Suen C. Y., (2005), "An Improved Handwritten Chinese Character Recognition System Using Support Vector Machine", Pattern Recognition Letters, 26:1849-1856.
- Duda R. O., Hart P. E. ve Stork D. G., (2001), Pattern Classification, Second Edition, John Wiley & Sons Inc., Canada
- Fortuna J. ve Capson D., (2004), "Improved Support Vector Classification Using PCA and ICA Feature Space Modification", Pattern Recognition, 37:1117-1129.
- Herbrich R., (2002), Learning Kernel Classifiers Theory and Algorithms, MIT Press, Massachusetts.
- Hyvärinen A., Karhunen J. ve Oja E., (2001), Independent Component Analysis, John Wiley & Sons Inc., New York.
- Joachims T., (2002), Learning to Classify Text Using Support Vector Machines, Kluwer Academic Publishers, Boston.
- Jolliffe I. T., (2002), Principal Component Analysis, Second Edition, Springer – Verlag, New York.
- Kincaid D. ve Cheney W., (2002), Numerical Analysis: Mathematics of Scientific Computing, Third Edition, Brooks / Cole, Texas.
- LeCun Y., Bottou L., Bengio Y. ve Haffner P., (1998), "Gradient-based Learning Applied to Document Recognition", Proceeding of the IEEE, 86(11):2278-2324
- Lee, T.-W., (1998), Independent Component Analysis: Theory and Applications, Kluwer Academic Publishers, Boston.
- Lv H., Wang W., Wang C. ve Zhou Q., (2005), "Off-line Chinese Signature Verification Based on Support Vector Machines", Pattern Recognition Letters, 26:2390-2399.
- Schölkopf B., Burges J. C. C. ve Smola A. J., (1999), Advances in Kernel Methods: Support Vector Learning, MIT Press, Massachusetts.
- Stone J. V., (2004), Independent Component Analysis: A Tutorial Introduction, MIT Press, Cambridge.
- Tsai C. F., (2005), "Training Support Vector Machines Based on Stacked Generalization for

Image Classification”, *Neurocomputing*, 64:497-503.

Vapnik V. N., (1998), *Statistical Learning Theory*, John Wiley & Sons Inc., New York.

Vapnik V. N., (2000), *The Nature of Statistical Learning Theory*, Second Edition, Springer Science, New York.

**ÖZGEÇMİŞ**

Doğum tarihi 11.05.1981

Doğum yeri Bad Pyrmont, Almanya

Lise 1995-1988 Yeşilköy 50. Yıl Lisesi

Lisans 2000-2004 Yıldız Teknik Üniversitesi Kimya - Metalurji Fak.  
Matematik Mühendisliği Bölümü

Yüksek Lisans 2004-2007 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü  
Matematik Müh. Anabilim Dalı

**Çalıştığı kurumlar**

2004-2005 Tamek Holding

2005-Devam ediyor YTÜ Fen Bilimleri Enstitüsü Araştırma Görevlisi