

**T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM
DALI**

**DESTEK VEKTÖR MAKİNALARI KULLANARAK KİŞİSEL
TERMAL KONFOR MODELLEMESİ**

YÜKSEK LİSANS TEZİ

EMRE UNCUOĞLU

DENİZLİ, AĞUSTOS - 2018

T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM
DALI



DESTEK VEKTÖR MAKİNALARI KULLANARAK KİŞİSEL
TERMAL KONFOR MODELLEMESİ

YÜKSEK LİSANS TEZİ

EMRE UNCUOĞLU

DENİZLİ, AĞUSTOS - 2018

KABUL VE ONAY SAYFASI

EMRE UNCUOĞLU tarafından hazırlanan “Destek Vektör Makinaları Kullanarak Kişisel Termal Konfor Modellemesi” adlı tez çalışmasının savunma sınavı 10.08.2018 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği / oy çokluğu ile Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Danışman
Prof. Dr. Serdar İPLİKÇİ



Üye
Doç. Dr. Selami BEYHAN
Pamukkale Üniversitesi



Üye
Dr. Öğr. Üy. Ali İhsan KOCA
Fatih Sultan Mehmet Vakıf Üniversitesi



Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
05/09/2018 tarih ve ...35/11... sayılı kararıyla onaylanmıştır.



Prof. Dr. Uğur YÜCEL

Fen Bilimleri Enstitüsü Müdürü

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bilimsel etięe ve akademik kurallara özenle riayet edildiđini; bu alıřmanın dođrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etięe uygun olarak kaynak gösterildiđini ve alıntı yapılan alıřmalara atfedildiđine beyan ederim.

EMRE UNCUOĐLU

imza:



ÖZET

DESTEK VEKTÖR MAKİNALARI KULLANARAK KİŞİSEL TERMAL KONFOR MODELLEMESİ

YÜKSEK LİSANS TEZİ

EMRE UNCUOĞLU

PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

(TEZ DANIŞMANI: PROF. DR. SERDAR İPLİKÇİ)

DENİZLİ, AĞUSTOS - 2018

Termal konfor, uzun yıllardan beri kullanılan bir terim olmakla beraber 1970'li yıllarda P.O. Fanger tarafından formülize edilmiş olup kişisel ve çevresel faktörlerin etkisiyle değişebilen, bireylerin içinde buldukları ortamdan termal olarak tatmin olma durumu şeklinde tanımlanmaktadır. Bu tezin ana hedefi, termal konfor parametrelerinin doğru ayarlanması sorununa Destek Vektör Makinalarının (Support Vector Machines - SVM) kullanılması ile yenilikçi ve kullanıcı odaklı bir yaklaşım getirmek ve bu sayede bireylerin yaşam kalitesini artırarak termal konforun elde edilmesini sağlamaktır. Bu çalışmanın kapsamı, insan iç mekan konforu ile ilgili güncel teorileri incelemek, SVM hakkındaki güncel teorileri tanımlamak ve SVM ile modellenebilecek termal konfor parametrelerini belirleyen algoritmayı oluşturmaktır. Bu çalışma, termal konfor ve SVM ile ilgili literatür kaynaklardan, termal konfor modellemesinin makine öğrenmesi ile yapıldığı teorilerden ve termal konforun belirlenebilmesi için daha önce yapılmış sistemlerden yararlanmaktadır. Bu teorilerden yola çıkarak tezin ana hedefine ulaşmak için, kullanıcıların termal konfor koşullarını öğrenerek bilgi toplanması, gerekli parametrelerin hesaplanması ve farklı hava koşullarında iç ortamın termal konforunun kullanıcı odaklı hale getirilmesi amaçlanmaktadır. Bu çalışmanın sonucunda çıkan algoritma, ölçüm sistemi ile beraber termal kontrol elemanlarına bağlandığında bireylerin isteğine uygun koşullarda konfor sağlayabilecek bir sistem oluşturabilecektir.

ANAHTAR KELİMELER: Termal Konfor, Destek Vektör Makinaları, Makine Öğrenmesi, Çok Sınıflı Sınıflandırma, PMV.

ABSTRACT

MODELING INDIVIDUAL THERMAL COMFORT USING SUPPORT VECTOR MACHINES

MSC THESIS

EMRE UNCUOĞLU

PAMUKKALE UNIVERSITY INSTITUTE OF SCIENCE

ELECTRICAL AND ELECTRONICS ENGINEERING

(SUPERVISOR: PROF. DR. SERDAR İPLİKÇİ)

DENİZLİ, AUGUST 2018

Thermal comfort is a term that has been used for many years and formulated by P.O. Fanger in the 1970's in order to define a state where individuals are thermally satisfied from the environment in which they have currently been and the state can be changed by the influence of personal and environmental factors. The main objective of this thesis is to bring an innovative and user-oriented approach by using Support Vector Machines (SVM) for the correct adjustment of the thermal comfort parameters and to achieve the thermal comfort by increasing the individuals' quality of life. The scope of this study covers examining current theories about human interior comfort, describing the current theories about SVM as well as forming the algorithm that determines thermal comfort parameters which can be modeled with SVM. This study benefits from the literature on thermal comfort and SVM, the theories of thermal comfort modeling with machine learning and previously built systems for the determination of thermal comfort. Based on these theories, in order to reach the goal of this thesis, it is aimed to obtain the information on the thermal comfort conditions of the users, to calculate the necessary parameters and to make the thermal comfort of the indoor environment user-centered in different weather conditions. The algorithm that comes out as a result of this study will be able to create a system that can provide comfort in the requested conditions for individuals when connected to the thermal control elements together with the measurement system.

KEYWORDS: Thermal Comfort, Support Vector Machines, Machine Learning, Multiclass Classifier, PMV.

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİL LİSTESİ	v
TABLO LİSTESİ	vi
SEMBOL LİSTESİ	vii
ÖNSÖZ	ix
1. GİRİŞ	1
1.1 Genel Amaçlar	2
1.2 Tezin Yapısı	3
2. TERMAL KONFOR	4
2.1 Termal Konfor Nedir?	4
2.2 Vücut Sıcaklığı Nasıl Düzenlenir?	4
2.3 İnsan Termal Ortamı Nasıl Değerlendirir?	5
2.4 Termal Konfor için Başlıca Koşullar	6
2.5 Konfor Denklemi	7
2.6 Metabolik Hız “met” Tahmini	10
2.7 Giysi Yalıtımı “clo” Değeri Hesaplamaları	11
2.8 Ne Ölçülmeli?	12
2.9 Ortalama Radyan Sıcaklık Nedir ve Nasıl Ölçülür?	13
2.10 Termal Konfor Nasıl Oluşturulur	14
2.11 PMV ve PPD Ölçekleri	15
2.12 Bir İşyerinde Termal Ölçüm Nasıl Yapılır?	17
3. DAHA ÖNCEKİ YAKLAŞIMLAR	18
3.1 PMV Üzerine Yapılmış Çalışmalar	18
3.2 ASHRAE RP-884 Adaptif Model Projesi	19
3.3 Yapay Sinir Ağları ile Yapılmış Çalışmalar	20
3.4 SVM ile Yapılmış Çalışmalar	22
4. DESTEK VEKTÖR MAKİNALARI	27
4.1 SVM Tarihçesi	27
4.2 SVM Uygulamaları	28
4.3 SVM'nin Diğer Sistemler ile Karşılaştırılması	28
4.4 Doğrusal Sınıflandırma	29
4.4.1 Sert Marjinli Sınıflandırıcı	33
4.4.2 Esnek Marjinli Sınıflandırıcı	34
4.5 Karesel Programlama	35
4.6 Doğrusal Olmayan Sınıflandırma	37
4.6.1 Lagrange Dönüşümü	38
4.6.2 Lagrange Dönüşümü SVM Uygulaması	39
4.6.3 Dual uzay, Esnek Marjinli Sınıflandırıcı	41
4.6.4 Kernel (Çekirdek) Fonksiyonu	46
4.7 Çok-Sınıflı SVM	50

5. SVM İLE TERMAL KONFOR MODELLEMESİ	53
5.1 Veri Setinin Seçilmesi	54
5.2 PMV ile Hesaplanması	56
5.3 SVM Modeli Oluşturulması	57
5.4 Karşılaştırma	61
5.5 Gerçek Ortam Karşılaştırması	67
6. SONUÇ VE ÖNERİLER	73
7. TEŞEKKÜR.....	75
8. KAYNAKLAR.....	76
9. EKLER.....	82
EK A.1 PMV Hesaplama Programı	82
EK A.2 Rastgele Çok Boyutlu PMV Verisi Üretici	83
EK A.3 Düzenli Çok Boyutlu PMV Verisi Üretici	84
EK A.4 Ana Program	86
EK A.5 Destek Vektörünü Eğitme ve Test Etme Programı	87
EK A.6 Destek Vektörü Sınıflandırıcısı Eğitme Programı	93
EK A.7 Destek Vektörü Sınıflandırıcısı Programı	94
EK A.8 Kernel Fonksiyonu	94
EK A.9 Öğrenilmiş bir SVM Kullanarak Veri Kümesi Çizme Programı	94
10. ÖZGEÇMİŞ.....	97

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: PMV-PPD grafiği.....	16
Şekil 3.1: ASHRAE RP-884 çalışmasının dünya üzerinde yapıldığı bölgeler.	19
Şekil 3.2: Yapay sinir ağı ile sıcaklık kontrolü yapılarak PMV hatasını minimum yapmaya yönelik sistem.....	22
Şekil 4.1: Örnek veri kümesi. Mavi kareler ' $y_i = -1$ ', kırmızı yuvarlaklar ' $y_i = +1$ ' olarak işaretlenmiş verileri gösterir.....	30
Şekil 4.2: İki veri kümesinin ayrabilecek sonsuz adet hiper-düzlemlerden bazıları.....	31
Şekil 4.3: Maksimum marjlinli lineer sınıflandırıcı.....	33
Şekil 4.4: Lineer olarak ayrılabilir ve ayrılamaz olan veri kümelerinin esnek marjlinli sınıflandırıcı kullanılarak farklı C değerlerinde sınıflandırılması, (a,b) $C = \infty$, (c,d) $C = 10$, (e,f) $C = 1$, (g,h) $C = 0.1$, (i,j) $C = 0.05$, (k,l) $C = 0.03$, (m,n) $C = 0.01$	45
Şekil 4.5: $C = 30$ iken farklı σ değerlerinde iki farklı veri kümesinde SVM sınıflandırması (a,b) $\sigma = 30$, (c,d) $\sigma = 10$, (e,f) $\sigma = 5$, (g,h) $\sigma = 3$, (i,j) $\sigma = 1$, (k,l) $\sigma = 0.5$	50
Şekil 5.1: Optimum " σ " değeri tarama algoritmasının çalışması.....	59
Şekil 5.2: İterasyonlara göre σ değerinin taranması.....	60
Şekil 5.3: $va = 0.3$, $HR = \%40$, $met = 1$, $clo = 0.5$ için test grafiği.....	63
Şekil 5.4: $va = 0.25$, $HR = \%45$, $met = 1.1$, $clo = 0.625$ için test grafiği..	64
Şekil 5.5: $va = 0.2$, $HR = \%50$, $met = 1.2$, $clo = 0.75$ için test grafiği.....	65
Şekil 5.6: $va = 0.15$, $HR = \%55$, $met = 1.3$, $clo = 0.875$ için test grafiği ..	66
Şekil 5.7: $va = 0.1$, $HR = \%60$, $met = 1.4$, $clo = 1$ için test grafiği.....	67
Şekil 5.8: RP-884 Atina çalışmalarının modellenmesi,.....	69
Şekil 5.9: RP-884 projesini iklim bölgeleri.....	70
Şekil 5.10: San Francisco Bay Area bölgesi HVAC sistemli yaz verileri.....	71

TABLO LİSTESİ

Sayfa

Tablo 2.1: Bazı Aktivitelere göre metabolik hız değerleri.....	10
Tablo 2.2: Bazı kıyafetlere göre giysi izolasyonu değerleri.....	11
Tablo 2.3: Bazı kıyafet parçalarının clo değerleri.....	12
Tablo 3.1: Çalışma için doğru tahmin oranları	26
Tablo 4.1: Çok sınıflı SVM: sınıfların SVM'lere ayrılması	51
Tablo 4.2: Çok sınıflı SVM: SVM sonuçlarına göre sınıfların belirlenmesi ...	51
Tablo 4.3: Çok sınıflı SVM: 3 SVM'li sınıflandırma örneği	52
Tablo 5.1: Veri sınıflarına göre SVM tahminleri.....	69
Tablo 5.2: Veri sınıflarına göre SVM tahminleri.....	71
Tablo 5.3: Veri sınıflarına göre SVM tahminleri.....	72

SEMBOL LİSTESİ

SVM	:	Support Vector Machines, Destek Vektör Makinaları
M	:	Kimyasal enerjinin vücut içindeki aerobik ve anaerobik aktivitelerle ısı ve mekanik işlere dönüşüm oranı [W/m^2]
W	:	Etkin mekanik güç [W/m^2]
H	:	Kuru Isı Kaybı. Konveksiyon, radyasyon ve iletim yoluyla vücut yüzeyinden ısı kaybı [W/m^2]
E	:	Ciltte buharlaşma yoluyla ısı değişimi [W/m^2]
E_c	:	Birey termal konforlu hissettiğinde, ciltte buharlaşma yoluyla ısı değişimi [W/m^2]
C_{res}	:	Solunum yoluyla konvektif ısı değişimi. [W/m^2]
E_{res}	:	Solunum yoluyla evaporatif ısı değişimi. [W/m^2]
\bar{t}_{sk}	:	Ortalama cilt sıcaklığı [$^{\circ}C$]
p_a	:	Nem oranı. Havada kısmi su buharı basıncı. [Pa]
E_{sw}	:	Terin buharlaşması ile oluşan Evaporatif ısı kaybı. [W/m^2]
t_a	:	Hava Sıcaklığı [$^{\circ}C$]
f_{cl}	:	Kıyafet yüzey alanı faktörü
t_{cl}	:	Kıyafet yüzey sıcaklığı [$^{\circ}C$]
\bar{t}_r	:	Ortalama radyan sıcaklık [$^{\circ}C$]
h_c	:	Konvektif ısı transfer katsayısı. [$W/m^2 \cdot K$]
I_{cl}	:	Kıyafet izolasyonu [$m^2 \cdot K/W$]
v_{ar}	:	Bağıl hava hızı [m/s]
met	:	Metabolik hız
clo	:	Kıyafet izolasyonu
HR	:	Hava Nemi [%]
t_i	:	i' nci Yüzeyin Sıcaklığı [$^{\circ}C$]
F_{p-i}	:	i' inci yüzey ile insan arasındaki açı faktörü [$^{\circ}C$] $\sum F_{p-i} = 1$
t_g	:	Globe Sıcaklığı [$^{\circ}C$]
ϵ_g	:	Globe Emisivitesi
D	:	Globe Çapı
t_{co}	:	Konforlu Sıcaklık, Eşdeğer Sıcaklık
PMV	:	Predicted Mean Vote - Öngörülen Ortalama Oylama
PPD	:	Predicted Percentage of Dissatisfied - Öngörülümüş Memnuniyetsizlik Yüzdesi
HVAC	:	Heating, Ventilating and Air Conditioning - Isıtma, havalandırma ve hava şartlandırma sistemi
PI	:	Proportional-Integral - Oransal İntegral kontrolcü
LS-SVM	:	Least Square SVM - En küçük kareler SVM
C	:	Lagrange parametrelerinin üst limit katsayısı
σ	:	Gaussian Radial Basis Function için genişlik parametresi
RBF	:	(Gaussian) Radial Basis Function
NN, ANN	:	(Astificial) Neural Network, Yapay Sinir Ağı
QSAR	:	Quantitative Structure-Activity Relationship
n	:	Tasarım değişkenleri sayısı
N_{eq}	:	Eşitlik kısıtları sayısı
N_{ie}	:	Eşitsizlik kısıtları sayısı
x	:	n x 1 boyutlu tasarım değişkenleri vektörü

H	:	$n \times n$ boyutlu simetrik kare matris, Hessian matrisi
g	:	$n \times 1$ boyutlu vektör
A_{eq}	:	$N_{eq} \times n$ boyutlu eşitlik kısıtları matrisi
b_{eq}	:	$N_{eq} \times 1$ boyutlu eşitlik kısıtları vektörü
A_{ie}	:	$N_{ie} \times n$ boyutlu eşitsizlik kısıtları matrisi
b_{ie}	:	$N_{ie} \times 1$ boyutlu eşitsizlik kısıtları vektörü
L	:	Lagrange dönüşümü
α, β	:	Lagrange çarpanları
K(x, y)	:	Kernel (Çekirdek) Fonksiyonu
#SVM	:	SVM sayısı
#CLASS	:	Sınıf Sayısı

ÖNSÖZ

Termal konfor insanların çalışma ve yaşam ortamlarında en çok etkilendikleri parametrelerden biridir. Termal konfor konusu tamamen kişiye özel olduğu için genel sistemlerden çok öğrenen akıllı algoritmalara ihtiyaç duyulduğu anlaşılmıştır. Bu soruna cevap olarak MİR ARAŞTIRMA ve GELİŞTİRME A.Ş. tarafından yürütülmekte olan ve koordinatörlüğünü yapmakta olduğum Comfostat projesi ortaya atılmıştır. Hali hazırda yürümekte olan ve EUREKA destekli bu projenin önemli bir ayağını termal konforun belirlenmesi oluşturur. Bu tez bu sorun karşısında önemli bir adım atarak bu proje adına yapılacak olan sonraki çalışmalar için temel niteliği taşımaktadır.

Tez çalışmamın yürütülmesi sırasında deneyim, bilgi birikimi ve desteğini hiçbir zaman esirgemeyen değerli danışman hocam sayın Prof. Dr. Serdar İplikçi'ye teşekkür ve saygılarımı sunarım. Kendisinin bana ayırdığı zaman ve sabırdan dolayı minnettarım.

Bu çalışma sırasında bana gerekli kolaylıkları göstermekten çekinmeyen çalışmakta olduğum MİR ARAŞTIRMA ve GELİŞTİRME A.Ş.'ye mesai arkadaşlarıma ve yöneticilerime desteklerinden dolayı içten teşekkürlerimi sunuyorum.

Ayrıca çalışmalarım boyunca yardımını hiç esirgemeyen, manevi desteğini her an yanımda hissettiğim sevgili eşim Yasemin'e, bu günlere gelmemde çok büyük payı olan, maddi manevi her türlü desteği olan, her zaman arkamda yer alan anneme ve babama, başta Gürcan olmak üzere tüm dostlarıma da hep yanımda oldukları için sonsuz teşekkürlerimi iletiyorum.

1. GİRİŞ

Günümüzde, gelişmiş ülkelerde yaşayan insanlar, zamanlarının %90'ından fazlasını kapalı mekanlarda geçirmektedirler (Frontczak ve Wargocki 2011). Rahat iç ortam koşulları, hem iş hem de özel hayatımızda sağlığımızı, refahımızı ve verimliliğimizi önemli ölçüde arttırmaktadır. Öte yandan, iç mekan konforu sağlamak önemli miktarda enerji tüketimini de beraberinde getirmekte ve aslında binalar dünyanın toplam enerjisinden büyük oranda pay almaktadırlar. Örneğin, İsviçre'de enerji tüketiminin yaklaşık %50'si binalara atfedilir (SFOE 2013). 2016 yılı Enerji ve Tabii Kaynaklar Bakanlığı (ETKB) verilerine göre, ülkemizde tüketilen toplam enerjinin %34'ü konutlar ve ticari binalardan oluşan bina sektöründen kaynaklanmaktadır (ETKB 2017).

İç mekan konforunu sağlamak ve artırmak için günümüzün yapıları giderek artan şekilde bina otomasyon sistemlerini kullanılmaktadır. Bu sistemler ise havalandırma, ısıtma, aydınlatma ve klima için optimize edilmiş tasarımlardan meydana gelmektedir. Kullanılan bu tasarımlar, oda büyüklüğü veya ortam sıcaklığı gibi standart ve istatistiksel değerlere dayandırılmaktadır (Ku ve diğ. 2004). Örneğin, bu binalarda termal konfor, basit kuru hava sıcaklığı ayarları ile kontrol edilir. Ek olarak, sıcaklık sensörleri her zaman mahal sakinlerinin yakınında olması gereken yerlere yerleştirilmez. Tüm bu sistemler binayı kullanıcı odaklı olmayan ortalama değerlere göre düzenler. Bu durumda kullanıcılar kendi termal konforlarını sağlamak için sistemleri daha fazla enerji tüketecek şekilde yönlendirirler. Gelişmiş algılama ve kontrol sistemlerinin ortaya çıkması, binaların daha önce hiç olmadığı kadar verimli ve ihtiyaç sahiplerine daha duyarlı olmasını sağlamaktadır.

Yeni teknolojilerin sunduğu fırsatlar, insanlar ile binalar arasındaki bağı daha da güçlendirebilmektedir. İnsanların binalar içerisinde çok fazla zaman harcadıklarını düşünürsek, yeni interaktif teknolojileri kullanarak yaşam kalitesini nasıl artırabilir ve en iyi kullanıcı deneyimini sağlayarak termal konforu nasıl sağlayabiliriz?

Bu tez bu soruna teknolojik bir yaklaşım getirmeyi amaçlamaktadır. Kullanıcılardan bilgi toplayarak ve bu bilgileri işleyerek farklı dış hava koşullarında

dahi iç ortamın termal konforunun kullanıcı odaklı olmasını sağlayacak parametreleri öğrenerek termal konfor bölgesini bulmayı hedeflemektedir.

1.1 Genel Amaçlar

Bu tez, bireylerin termal konfor koşullarını öğrenerek yaşam mahallini termal konfor koşullarına getirmek için gerekli olan parametreleri hesaplamayı amaçlamaktadır. Bu hesaplama sayesinde termal konfor şartlarının sağlanabilmesi için kontrol edilebilen parametrelerin değerleri belirlenebilecektir.

İlk yapılacak iş, insan iç mekan konforu ile ilgili mevcut çalışmaları anlamak ve sıralamaktır. Bu çalışmaların sonuçlarına bakarak termal konfor probleminin şu zamana kadar nasıl çözüldüğünü anlayacağız, bu çözümün pozitif ve negatif yönlerini inceleyeceğiz.

Bu tezin amacı termal konfor parametrelerinin doğru ayarlanması sorununa Destek Vektör Makineleri (SVM) kullanarak yenilikçi ve kullanıcı odaklı bir yaklaşım getirmektir.

Bu çalışmanın sonucunda çıkan algoritma, termal ölçüm sistemi ile beraber termal kontrol elemanlarına bağlandığında kullanıcının isteğine uygun koşullarda konfor sağlayabilecek bir sistem oluşturulabilecektir.

Özetle, bu çalışmanın üç ana hedefi aşağıda gibi açıklanabilir:

- İnsan iç mekan konforu hakkındaki güncel teorileri anlamak, ne yapmamız gerektiğini tanımlamak
- Destek Vektör Makinaları hakkındaki güncel teorileri anlamak, ne yapmamız gerektiğini tanımlamak
- Destek Vektör Makinası ile modellenebilecek termal konfor parametrelerini belirleyen algoritma oluşturmak.

Sonuç olarak, bu proje, kullanıcıların çalışma alanlarında kişisel konfor profilleri ve optimum konfor koşullarına sahip olmalarını minimum enerji harcayarak sağlayacak daha kullanıcı merkezli bir projenin ilk aşamasıdır.

1.2 Tezin Yapısı

Bu tezde, hedeflerimize ulaşmak için takip ettiğimiz prosedür dört bölümden oluşmaktadır. Birinci bölüm, termal konfor hakkındaki literatür taramasını içermekte ve kullanılan araçlardan bahsetmektedir. İkinci bölümde termal konforun belirlenebilmesi için daha önce yapılmış adaptif, öğrenen sistemler üzerine literatür taraması yer almaktadır. Üçüncü bölümde destek vektör makinaları sınıflandırma algoritmasının nasıl çalıştığı incelenecektir. Dördüncü bölümde ise Destek Vektör Makinaları ile ilk bölümündeki teorilere dayanarak ve ikinci bölümdeki çalışmalardan yola çıkarak termal konfor koşullarını belirlemek için geliştirilmiş sistemi anlatılmaktadır.

En son olarak yapılmış olan çalışmanın sonuçlarından, hedeflerinden ve gelecekte yapılması planlanan çalışmalardan bahsedilecektir.

2. TERMAL KONFOR

2.1 Termal Konfor Nedir?

Eski zamanlardan beri insanlar her zaman termal olarak rahat bir ortam yaratmak için çabaladı. Bu durum antik çağlardan günümüze kadar dünyadaki geleneksel yapılarda rahatlıkla görülebilir. Bugün, termal olarak konforlu bir ortam yaratmak, binaları tasarlarken göz önünde bulundurulması gereken en önemli parametrelerden biridir. (Innova 2002)

Ama termal konfor tam olarak nedir? ISO 7730 standardında “Termal ortamlarla ilgili memnuniyeti ifade eden zihnin durumu” olarak tanımlanmıştır (2005). Çoğu insan üzerinde hemfikir olduğu ancak fiziksel parametrelere kolayca dönüştürülmeyen bir tanım.

Örneğin günlük kıyafetlerimiz ile evimizde sakince yemeğimizi yerken de kışlık kıyafetlerimiz ile dışarıda karlı ve soğuk bir havada spor yaparken de tamamen farklı termal ortamlarda olmamıza rağmen termal olarak rahat olma olasılığımız yüksektir. Bu bize, termal konforun, hava sıcaklığı gibi sadece bir değil, birçok fiziksel parametrenin fonksiyonu olduğunu gösterir.

Çalışma ortamımızın genel konforunu değerlendirirken termal konforun yanı sıra hava kalitesi, ışık ve gürültü seviyesi gibi faktörler beraber etkili olmaktadır. Ancak termal konfor, bu parametrelerin en önemlisi ve en karmaşık olanıdır.

2.2 Vücut Sıcaklığı Nasıl Düzenlenir?

İnsan, kendi vücudunun iç sıcaklığının yaklaşık 37°C'de tutulmasını sağlayan çok etkili bir sıcaklık kontrol sistemine sahiptir (Prek 2005).

Vücut çok ısındığında, iki süreç başlatılır: önce vazodilatasyon (Damar Genişlemesi) olur, ciltten kan akışı artar ve daha sonra deri terlemeye başlar.

Terleme etkili bir soğutma aracıdır çünkü terin buharlaşması için gerekli olan enerji deriden alınır. Merkez vücut sıcaklığında 0.1°C'lik artışlar dahi ter üretimini başlatabilir ve bu sayede vücudun ısı kaybı 4 kata kadar arttırılmış olur.

Vücut sıcaklığının normalin altında olması durumunda, bedenin ilk tepkisi vazokonstriksiyon (damar büzülmesi) olacaktır. Bu durumda kan damarlarının daralması sebebiyle kan akışını azaltır. İkinci tepkisi, titreyebilecek kasları harekete geçirerek vücut içi ısı üretimini arttırmaktır. Titreyerek üretilen ısı vücut sıcaklığının artması konusunda çok etkilidir ve çok kısa sürede sıcaklık değerleri normale dönebilir.

Vücut sıcaklığını düzenleyen kontrol sistemi oldukça karmaşıktır ve üzerinde yapılan birçok çalışmaya rağmen tam olarak nasıl çalıştığı gün yüzüne çıkartılamamıştır. Ancak bu kontrol sisteminin veri toplayan en önemli iki sensörü bilinmektedir. Bu sensörlerden biri hipotalamusta diğeri deri yüzeyinde bulunmaktadır. Hipotalamus sensörü, insan bedeninin iç sıcaklığı 37 °C'yi aştığında soğutma işlevini harekete geçiren bir detektördür. Deri yüzeyindeki sensörler ise insan bedeninin iç sıcaklığını ölçer. Bu sıcaklık 34 °C'nin altına düştüğünde harekete geçerek vücudun soğumaya karşı savunmasını başlatan detektörlerdir. (Hall ve Guyton 2007).

Yukarıda bahsedilen iki sensörün çıkışı aynı zamanda sinyal verirse, insan beyni vücudun bu sensörlere bağlı olan savunma mekanizmalarından biri ya da ikisini birden tetikleyecektir.

2.3 İnsan Termal Ortamı Nasıl Değerlendirir?

İnsan, çevre şartlarında herhangi bir termal konforsuzluk yoksa ortamı termal olarak rahat ve konforlu olarak algılar. İlk konfor koşulu ise termal nötrlüktür, bu da bir insanın ne sıcak ne de çok soğuk olduğunu hissetmesi anlamına gelmektedir.

Cilt sıcaklığı 34 °C'nin altına düştüğünde, soğuk algılayıcılarımız beynimize impuls göndermeye başlar ve sıcaklık düşmeye devam ettikçe, impuls sayısı da artar.

İmpuls sayısı aynı zamanda cilt sıcaklığının ne kadar hızlı düştüğünün bir göstergesidir - hızlı sıcaklık düşüşleri çok impuls gönderilmesine yol açar.

Benzer şekilde, hipotalamustaki ısı sensörü, sıcaklık 37 °C'yi aştığında impuls gönderir ve sıcaklık arttıkça, impuls sayısı da artar. Termal ortamın değerlendirilmesinde, bu iki sensör sisteminden gelen sinyallerin temel oluşturduğuna inanılmaktadır.

Beynin sinyalleri yorumlamasının, ipin bir ucundan soğuk darbelerin ve diğerinden de sıcak darbelerin çektiği bir güç yarışı gibi olduğu varsayılmaktadır. Her iki taraftaki sinyaller aynı büyüklükte ise, termal olarak nötr hissedersiniz, aksi takdirde çok sıcak veya çok soğuk hissedersiniz. Termal olarak nötr durumda olan ve tamamen rahatlamış bir kişi, ne ısıyı ne de soğuk algılayıcıları aktive edeceği için özel bir durum oluşturur.

Vücudun merkez sıcaklığını değiştirmek biraz zaman alır; Bu nedenle, sıcaklık sensöründen gelen sinyal, soğuk algılayıcılardan gelen sinyallere kıyasla çok yavaş değişir.

2.4 Termal Konfor için Başlıca Koşullar

Termal konforu korumak için iki koşulun yerine getirilmesi gerekmektedir. Birincisi, cilt sıcaklığının ve vücudun çekirdek sıcaklığının kombinasyonunun bir termal nötrlük hissi sağlamasıdır. İkincisi, vücudun enerji dengesinin gerçekleşmesidir: Metabolizmanın ürettiği ısı, vücuttan atılan ısı miktarına eşit olmalıdır. Bu parametreler yani cilt ısı, merkez vücut ısı ve aktivite arasındaki ilişki, termal olarak nötr bir his ile sonuçlanan, çok sayıda deneyle belirlenmiştir. Bu deneyler sırasında, vücudun merkez sıcaklığı, deri sıcaklığı ve üretilen ter miktarı çeşitli etkinlik seviyelerinde ölçülürken, denekler termal olarak konfordadır.

Ter üretimi merkez vücut sıcaklığı yerine ölçülebilecek bir parametre olarak seçilmiştir. Ter üretimi merkez vücut ve cilt sıcaklığının bir fonksiyonu olduğundan, prensipte termal duyum modelinde hiçbir şey değiştirmez.

Bir insan için enerji dengesini kontrol eden denklemler aşağıdaki gibidir:

Bedenin Isıl Denge Denklemi şu şekildedir:

$$M - W = H + E + C_{res} + E_{res}$$

Kişi termal olarak konforlu hissettiğinde ise denklem şu hali alır:

$$M - W = H + E_c + C_{res} + E_{res}$$

Burada:

M : Metabolik hız, Kimyasal enerjinin vücut içindeki aerobik ve anaerobik aktivitelerle ısı ve mekanik işlere dönüşüm oranı [W/m^2]

W : Etkin mekanik güç [W/m^2]

H : Kuru Isı Kaybı. Konveksiyon, radyasyon ve iletim yoluyla vücut yüzeyinden ısı kaybı [W/m^2]

E : Ciltte buharlaşma yoluyla ısı değişimi [W/m^2]

E_c : Birey termal konforlu hissettiğinde, ciltte buharlaşma yoluyla ısı değişimi [W/m^2]

C_{res} : Solunum yoluyla konvektif ısı değişimi. [W/m^2]

E_{res} : Solunum yoluyla evaporatif ısı değişimi. [W/m^2]

2.5 Konfor Denklemi

Konfor Denklemine elde etmek için konforlu cilt sıcaklığı ve ter üretimi denklemi, vücudun enerji dengesi denklemi ile birleştirilebilir. Bu denklem, ölçülebilir fiziksel parametreler ile "ortalama" kişi tarafından termal olarak nötr hissedilen ortam arasındaki bağlantıyı sağlamaktadır.

Konfor denklemi bize, fiziksel parametrelerin ölçülmesiyle, ortam termal konforun hangi koşullar altında sunulabileceğini değerlendirmemize olanak sağlayan bir araç sağlar. P.O. Fanger tarafından elde edilen konfor denklemi manuel aritmetik

için çok karmaşıktır ve normalde bir bilgisayar kullanılarak çözülmektedir. Bu denklemin hesaplama detaylarından ileride bahsedilecektir (Fanger 1970).

Denklem, bir kişinin bulunduğu ortamdaki yüzeylerin sıcaklığının, termal duyumda büyük bir etkisi olduğunu ortaya koymaktadır. Ortamdaki yüzey sıcaklığındaki (tavan, zemin, duvarlar) 1 °C'lik bir değişimin birçok durumda hava sıcaklıklarında 1°C'lik bir değişiklik olarak kişilerin termal hissi üzerinde büyük bir etkisi olabilir. Ayrıca, konfor denklemi, nem seviyesinin sadece termal duyumda ılımlı bir etkiye sahip olduğunu ortaya koymaktadır.

Pratikte, Konfor Denkleminin hangi giriş parametrelerini gerektirdiğini bilmek önemlidir. Bunlar:

- Kişilerin aktivite ve giyim seviyelerini veren 2 adet tablo değeri. (clo ve met değerleri).
- İşyerinde termal ortamı tanımlayan 2-4 adet ölçülen parametre.

Konfor denklemi daha önce bahsedildiği gibi şu şekildedir:

$$M - W = H + E_c + C_{res} + E_{res}$$

Bu denklemde hesaplanacak parametreler şu denklemlerle hesaplanabilir:

$$E = 3.05 \cdot 10^{-3} (256 \cdot \bar{t}_{sk} - 3373 - p_a) + E_{sw}$$

$$E_c = 3.05 \cdot 10^{-3} [5733 - 6.99 \cdot (M - W) - p_a] + 0.43 \cdot (M - W - 58.15)$$

$$C_{res} = 0.0014 \cdot M \cdot (24 - t_a)$$

$$E_{res} = 1.72 \cdot 10^{-5} \cdot M \cdot (5867 - p_a)$$

Burada:

\bar{t}_{sk} : Ortalama cilt sıcaklığı [°C]

p_a : Nem oranı. Havadaki kısmi su buharı basıncı [Pa]

E_{sw} : Terin buharlaşması ile oluşan Evaporatif ısı kaybı. [W/m^2]

t_a : Hava Sıcaklığı [$^{\circ}C$]

Kuru ısı kaybı ise şu şekilde hesaplanır:

$$H = 3.96 \cdot 10^{-8} \cdot f_{cl} \cdot [(t_{cl} + 273)^4 - (\bar{t}_r + 273)^4] - f_{cl} \cdot h_c \cdot (t_{cl} - t_a)$$

$$t_{cl} = 35.7 - 0.028 \cdot (M - W) - I_{cl} \cdot \{3.96 \cdot 10^{-8} \cdot f_{cl} \cdot [(t_{cl} + 273)^4 - (\bar{t}_r + 273)^4] - f_{cl} \cdot h_c \cdot (t_{cl} - t_a)\}$$

$$h_c = \begin{cases} 2.38 \cdot |t_{cl} - t_a|^{0.25}, & 2.38 \cdot |t_{cl} - t_a|^{0.25} > 12.1\sqrt{v_{ar}} \\ 12.1\sqrt{v_{ar}}, & 2.38 \cdot |t_{cl} - t_a|^{0.25} < 12.1\sqrt{v_{ar}} \end{cases}$$

$$f_{cl} = \begin{cases} 1.00 + 1.290I_{cl}, & I_{cl} \leq 0.078m^2 \cdot K/W \\ 1.05 + 0.645I_{cl}, & I_{cl} > 0.078m^2 \cdot K/W \end{cases}$$

Burada:

f_{cl} : Kıyafet yüzey alanı faktörü

t_{cl} : Kıyafet yüzey sıcaklığı [$^{\circ}C$]

\bar{t}_r : Ortalama radyan sıcaklık [$^{\circ}C$]

h_c : Konvektif ısı transfer katsayısı. [$W/m^2 \cdot K$]

I_{cl} : Kıyafet izolasyonu [$m^2 \cdot K/W$]

v_{ar} : Bağıl hava hızı [m/s]

Bu denklemlerde t_{cl} hesaplamasına dikkat edilirse kendi içerisinde t_{cl} 'nin 1. ve 4. kuvvetini barındırır. Bu formül ISO7730 da da belirtildiği gibi iteratif olarak çözülür. Bu işlemleri elle yapmak özellikle bu sebeple neredeyse imkansız hale gelmektedir. PMW hesabını yapan örnek programını “EK A.1 PMV Hesaplama Programı” bölümünde bulabilirsiniz.

2.6 Metabolik Hız “met” Tahmini

Metabolizma, vücudun motorudur ve metabolizmanın saldıđı enerji miktarı, kas aktivitesinin miktarına bađlıdır. Normal olarak, tüm kas aktivitesi vücutta ısıya dönüştürülür, ancak sert fiziksel çalışma sırasında bu oran %75'e düşebilir. Örneđin, kiři bir dađa çıktığında, kullanılan enerjinin bir kısmı vücutta potansiyel enerji biçiminde depolanır.

Geleneksel olarak, metabolizma hızı met (Metabolic Rate) (1 met = 58.15 W/m^2 vücut yüzeyi) olarak ölçülür. Normal bir yetişkinin 1.7 m^2 'lik bir yüzey alanı vardır ve 1 met'lik bir aktivite seviyesine sahip olan termal konforda bir kiři yaklaşık 100W'lık bir ısı kaybına sahip olmaktadır. Metabolizmamız biz uyurken en düşük seviyede (0.8 met), ve spor aktivitelerinde ise en yüksek seviyededir, 10 met'e kadar ulaşabilir. Aşağıdaki tabloda farklı aktiviteler için birkaç metabolik değeri gösterilmiştir.

Tablo 2.1: Bazı Aktivitelere göre metabolik hız değeri

Aktivite	Metabolik Hız	
	W/m^2	met
Uzanarak dinlenme	46	0.8
Oturarak Dinlenme	58	1.0
Oturarak yapılan Aktivite (Ofis, Konut, Okul, Laboratuvar)	70	1.2
Ayakta Hafif Aktivite (alışveriş, laboratuvar, hafif endüstri)	93	1.6
Ayakta, orta aktivite (tezgâhtar, ev işleri, makine işleri)	116	2.0
Düz bir zeminde yürümek		
2 km/s	110	1.9
3 km/s	140	2.4
4 km/s	165	2.8
5 km/s	200	3.4

Bir ofiste otururken normal çalışmaya karşılık gelen 1.2 met, hesaplamalarda yaygın olarak kullanılan bir değerdir. Ev işlerinin 2.5 - 2.9 arası met değeri ile nispeten sıkı çalışılan bir ortam olması ilginçtir.

Bir bireyin metabolik hızını değerlendirirken, kişinin son bir saat içinde gerçekleştirdiği faaliyetler için ortalama bir değer kullanmak önemlidir. Bunun nedeni, vücudun ısı kapasitesinin, yaklaşık bir saatlik aktivite seviyesini hatırlamasıdır.

2.7 Giysi Yalıtımı “clo” Değeri Hesaplamaları

Giysiler vücudun ısı kaybını azaltır. Bu nedenle, giysiler yalıtım değerine göre sınıflandırılmaktadır. Giysinin yalıtımını ölçmek için normal olarak kullanılan ünite clo (Clothing kelimesinden gelmektedir) ünitesidir, ancak daha teknik bir birim olan $m^2\text{C}/W$ da sıklıkla kullanılmaktadır ($1 \text{ clo} = 0.155 \text{ m}^2\text{C}/W$).

clo ölçeği, çıplak bir kişi için 0.0 değerinde ve tipik bir iş kıyafeti giyen bir kişi için ise 1.0 olacak şekilde tasarlanmıştır. Bazı günlük ve ofis kıyafet kombinasyonlarının clo değerleri aşağıdaki tabloda gösterilmiştir.

Tablo 2.2: Bazı kıyafetlere göre giysi izolasyonu değerleri

	Kıyafet izolasyonu	
	$m^2 \cdot K/W$	clo
İç Çamaşırı, kısa kollu gömlek, hafif pantolon, hafif çorap, ayakkabı	0.080	0.5
İç Çamaşırı, işçi tulumu, çorap, ayakkabı	0.110	0.7
İç Çamaşırı, gömlek, işçi tulumu, çorap, ayakkabı	0.125	0.8
İç Çamaşırı, gömlek, pantolon, önlük, çorap, ayakkabı	0.140	0.9
İç Çamaşırı, gömlek, pantolon, ceket, çorap, ayakkabı	0.155	1.0

Kişinin giydiği giysi parçaları ve bu parçaların ayrı ayrı clo değerleri biliniyorsa, toplam clo değeri bu parçaların değerleri toplanarak hesaplanabilir. Örneğin bazı kıyafet parçalarının yaklaşık clo değerleri aşağıdaki tabloda verilmiştir.

Tablo 2.3: Bazı kıyafet parçalarının clo değerleri

	Kıyafet izolasyonu	
	$m^2 \cdot K/W$	clo
İç Çamaşırı	0.003 - 0.016	0.02 - 0.1
Gömlek	0.009 - 0.053	0.06 - 0.34
Pantolon	0.009 - 0.043	0.06 - 0.28
Kazak	0.019 - 0.057	0.12 - 0.37
Ceket	0.020 - 0.047	0.13 - 0.3

clo değerini hesaplama yoluyla elde etmek normalde yeterli bir doğruluk sağlamaktadır. Kesin değerler gerekiyorsa, ısıl manken kullanarak clo değerini ölçmek daha uygundur (Pamuk 2008).

clo değerlerini hesaplarken, döşemeli koltukların, araba koltuklarının ve yatakların vücuttan gelen ısı kaybını da azalttığını hatırlamak önemlidir ve bu nedenle, bunlar genel hesaplamada yer almalıdır.

2.8 Ne Ölçülmeli?

Termal iç ortam iklimini ölçerken, insanın oda sıcaklığını değil vücudun enerji kaybını hissettiğini hatırlamak önemlidir. Ölçülmesi gereken parametreler enerji kaybını etkileyen değerlerdir. Bunlar:

- t_a Hava Sıcaklığı [$^{\circ}C$]
- \bar{t}_r Ortalama Radyan Sıcaklık [$^{\circ}C$]
- v_a Hava Hızı [m/s]
- HR Hava Nemi [%]

Bu parametrelerin enerji kaybı üzerindeki etkisi eşit değildir, ancak bunlardan sadece birini ölçmek yeterli değildir. Örneğin, Ortalama Radyan Sıcaklık enerji kaybı söz konusu olduğunda hava sıcaklığı kadar büyük bir etkiye sahiptir.

2.9 Ortalama Radyan Sıcaklık Nedir ve Nasıl Ölçülür?

Radyan sıcaklık, bir yüzeyin radyasyon yoluyla yaydığı ya da soğurduğu enerji ile alakalıdır.

Bir ortamın Ortalama Radyan Sıcaklığı şöyle hesaplanır: hayali ve duvarları siyah bir odanın tüm duvarlarının eşit sıcaklıkta olduğunu düşünelim; Gerçek ortamın radyasyon yoluyla yaydığı ısının hayali ortamın yaydığı ısıya eşit olması için hayali odanın duvarlarının olması gereken sıcaklığa ortalama radyan sıcaklık denir.

Ortalama Radyan Sıcaklığın hesaplanması için denklem şöyledir:

$$\bar{t}_r = \sqrt[4]{\sum_n F_{p-i} (t_i + 273)^4} - 273$$

Burada:

\bar{t}_r : Ortalama Radyan Sıcaklık [°C]

t_i : i'nci Yüzeyin Sıcaklığı [°C]

F_{p-i} : i'inci yüzey ile insan arasındaki açı faktörü [°C] $\sum F_{p-i} = 1$

(ISO 7726 2001)

Odadaki tüm yüzeylerin sıcaklık değerlerinin ölçülmesi çok zaman alıcıdır ve daha fazla zaman alıcı olan şey ise ilgili açı faktörlerinin hesaplanmasıdır. Bu nedenle, mümkün olduğunda Ortalama Radyan Sıcaklığın kullanılması önerilir.

Ortalama Radyan Sıcaklığın hesaplanması için ikinci bir yöntem de Globe Sıcaklığını, kuru hava Sıcaklığını ve hava hızını ölçmektir. Globe Sıcaklık Sensörü yüksek emisyon değerine sahip siyah bakır bir kürenin içine, merkezine termometre yerleştirilmesi ile oluşturulur. Bu küre içinde okunan sıcaklık t_g olarak adlandırılır ve alttaki formül kullanılarak ortalama radyan sıcaklık elde edilebilir:

$$\bar{t}_r = \left[(t_g + 273)^4 + \frac{1,1 \cdot 10^8 \cdot v_a^{0,6}}{\varepsilon_g \cdot D^{0,4}} \cdot (t_g - t_a) \right]^{1/4} - 273$$

Burada

t_g Globe Sıcaklığı [°C]

t_a Kuru Hava Sıcaklığı [°C]

ε_g Globe Emisivitesi (birimsiz)

D Globe Çapı (Standart olarak 15cm)

Bununla birlikte ortalama radyan sıcaklık ölçümünün kalitesi hem Globe ile bir odadaki yüzeyler arasındaki açı faktörlerinin bir kişi ile aynı yüzeyler arasındaki açı faktörlerinden farklı olması hem de Globe'un konvektif ısı transfer katsayısının belirsizliğinden dolayı etkilenebilir ve gerçek ölçümden bir miktar farklı olabilir.

2.10 Termal Konfor Nasıl Oluşturulur

Bir işyerini değerlendirirken, sıklıkla bir kişinin termal olarak konforlu hissettiği Eşdeğer Sıcaklık olarak tanımlanan t_{co} değerinden söz edilir. Konforlu nemden nadiren bahsederiz, bu kısmen insanın havadaki nemi hissetme zorluğundan ve kısmen de nemin, termal rahatlığa yakın koşullardaki bir kişinin ısı değişiminde az bir etkiye sahip olmasından kaynaklanmaktadır.

Belirli bir ortamdaki konfor sıcaklığı konfor denkleminde hesaplanabilir (bkz. 2.5 Konfor Denklemi). Bu denklemlere bakıldığında termal konfor elde etmek için farklı parametrelerin belirli koşullara gelmesi gerektiği rahatlıkla görülebilir.

Örneğin buzhane çalışan bir işçi için termal konfor koşulları şu şekilde olabilir:

clo=1.7, met=2.5, Eşdeğer Sıcaklık: 6 C°.

Dışarıda çalışan bir araba tamircisi için:

$clo=0.8$, $met=2.2$, Eşdeğer Sıcaklık: 18 C° .

Ya da ofiste çalışan bir hanım için:

$clo=0.5$, $met=1.2$, Eşdeğer Sıcaklık: 24.5C° .

Eğer bir odada farklı kıyafetler giyen, farklı aktiviteler gerçekleştiren çok sayıda insan varsa herkese termal konfor sağlayan bir ortam yaratmak zor olabilir. Bu durumda termal konforu lokal olarak değiştirerek bir şeyler yapılabilir. Örneğin eşdeğer sıcaklık konfor sıcaklığından daha düşükse ortama radyan panel yerleştirerek arttırılabilir.

Neyse ki, bireyler genellikle kendi termal konforlarını, giysilerini koşullara uyacak şekilde ayarlayarak, örneğin, bir yeleği çıkartarak, gömlek manşetlerini kıvrarak veya alternatif olarak bir ceket giyerek optimize edebilirler.

2.11 PMV ve PPD Ölçekleri

Bir iş yerindeki termal konfor mükemmel değilse, mükemmel olmaya ne kadar uzaktır? Ya da makul termal rahatlığı sağlamak için sıcaklık ve nemi için hangi sınırlar dahilinde koruyabiliriz? Bu soruların cevapları PMV indeksinden (Predicted Mean Vote - Öngörülen Ortalama Oylama) alınabilir. PMV indeksi, belirli bir çevrede bir grup insanın öznel değerlerinin ortalama değerini tahmin etmektedir.

PMV indeksi, -3 (soğuk) ila $+3$ (sıcak) arasında değişen yedi noktalı termal duyu ölçeğidir, burada 0 termal olarak nötr hissi temsil eder.

PMV indeksi 0 olsa bile, hepsi aynı şekilde giyinmiş ve aynı seviyede aktivite sahibi olmalarına bakılmaksızın sıcaklık seviyesinden memnun olmayan bazı bireyler olacaktır. Konfor değerlendirmesi kişiden kişiye farklılık gösterir.

Belirli bir termal ortamda kaç kişinin memnuniyetsiz olduğunu tahmin etmek için PPD indeksi (Predicted Percentage of Dissatisfied - Öngörülmemiş Memnuniyetsizlik Yüzdesi) kullanılır. PPD indeksinde PMV ölçeğinde $-3, -2, +2, +3$ oyu kullanan kişiler, termal olarak tatminsiz olarak kabul

edilmektedir. ISO7730 ve ASHRAE-55 standartlarına göre termal olarak konforlu olunan bölge

$$-0.5 < PMV < 0.5$$

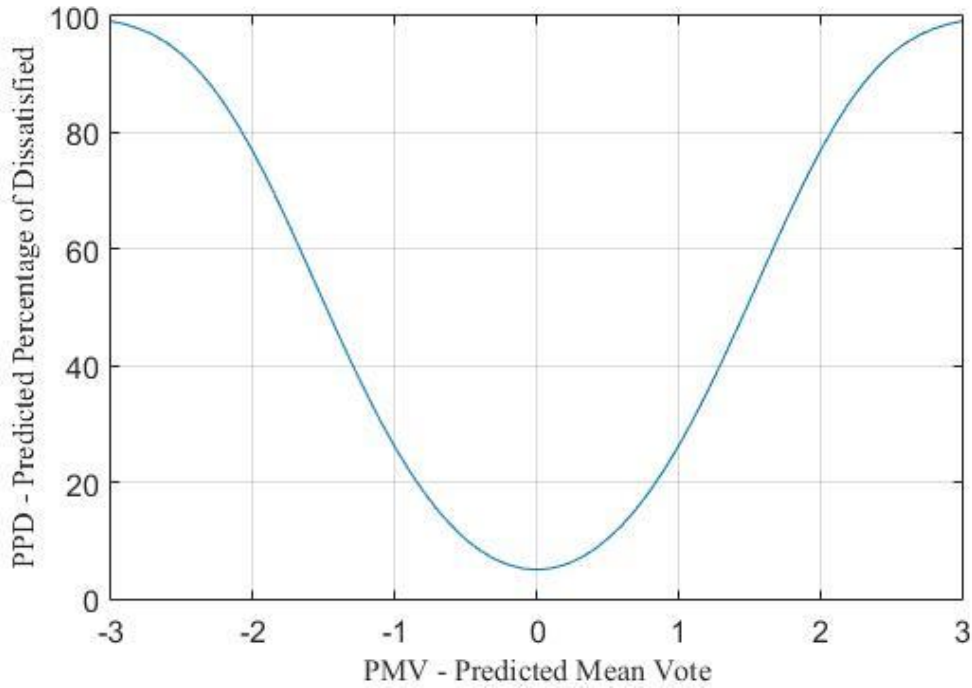
Şeklinde tanımlanır. PMV bu aralıkta olduğunda ortamdaki kişilerin en az %80'inin termal olarak konforlu düzeyde olduğu varsayılmaktadır.

PMV ve PPD denklemleri şu şekildedir:

$$PMV = (0.303 \cdot e^{-0.036 \cdot M} + 0.028) \cdot [(M - W) - H - E_c - C_{res} - E_{res}]$$

$$PPD = 100 - 95 \cdot e^{-(0.03353 \cdot PMV^4 + 0.2179 \cdot PMV^2)}$$

PMV-PPD grafiği şu şekildedir:



Şekil 2.1: PMV-PPD grafiği

PMV ve PPD arasındaki ilişkiyi gösteren eğrinin hiçbir zaman %5'in altına düşmediğine dikkat edin. Bu da birçok kişinin aynı anda termal olarak konforunda hissedebileceği bir ortam olmadığı manasına gelmektedir. Standartlarda %80 oranında termal konfor hissediliyorsa ortamın termal konforlu olduğundan bahsedilebilir.

2.12 Bir İşyerinde Termal Ölçüm Nasıl Yapılır?

İşyerinde ölçüm yaparken sensörler nereye yerleştirilmelidir? Profesyonel ölçümlerde oturma ve ayakta duran kişiler için baş hizası, ayak hizası ve gövde hizası olmak üzere 3 sensör yerleştirilmelidir.

Ancak genel olarak, sensörler kişinin ağırlık merkezine yerleştirilmelidir. Bu kuralın istisnaları, Dikey Hava Sıcaklığı Farklılıkları ve hava hızının ölçülmesidir. Bu ölçümler hem ayak bileği hem de boyun seviyelerinde yapılmalıdır.

Kuru Isı Kaybı H 'yi ölçmek için seçilen yönteme bağlı olarak bir, iki veya 3 adet sensöre ihtiyaç vardır. Seçenekler şunlardır:

- Kuru Isı Kaybı transdüseri
- Çalışma Sıcaklığı ve Hava Hızı transdüseri.
- Radyan Sıcaklık, Hava Sıcaklığı ve Hava Hızı transdüseri.

Bir işyerindeki oturarak ya da sakin yapılan hareketler söz konusu olduğunda termal konforun değerlendirilmesi için, ISO 7730 aşağıdaki şartları önerir:

- $-0.5 < PMV < 0.5$
- Boyun ve ayak bileğinde hava hızı $< \% 15$.
- Ayak bileğinden başa dikey Hava Sıcaklığı Farkı $3^{\circ}C$ 'den az olmalıdır.
- Soğuk pencerelerden kaynaklanan Radyan Sıcaklık asimetrisi $10^{\circ}C$ 'den az olmalıdır.
- Ilık tavanlardan kaynaklanan Radyan Sıcaklık Asimetrisi, $5^{\circ}C$ 'den az olmalıdır.
- Zeminlerin Yüzey Sıcaklığı $19^{\circ}C$ ile $29^{\circ}C$ arasında olmalıdır.
- Bağıl Nem $\%30$ ile $\%70$ arasında olmalıdır.

3. DAHA ÖNCEKİ YAKLAŞIMLAR

Daha önce de bahsedildiği gibi termal konfor problemi insanlığın var olduğu sürece var olmuş olan bir problem. Bu sebeple kontrol teorisi, makine öğrenmesi, önsüzli kontrol, adaptif kontrol, yapay sinir ağları, destek vektör makineleri gibi mühendislik konuları ilerledikçe bu konuların termal konfor problemine çözüm getirebilmeleri üzerine çalışmalar da başlanmıştır. Bu bölümde bu konudaki son dönemde yapılmış olan çalışmalardan bahsedilecektir.

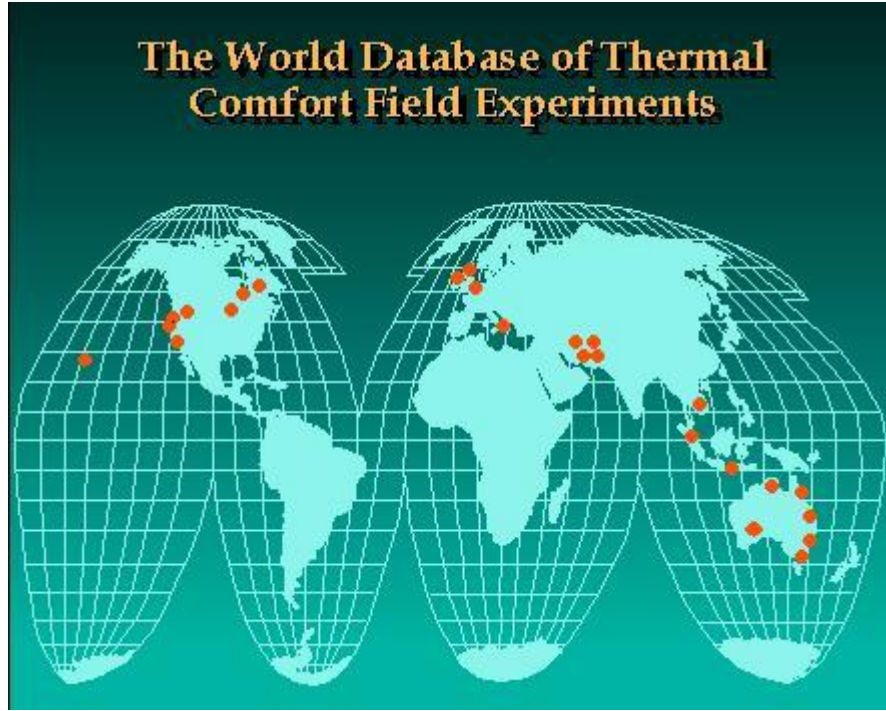
3.1 PMV Üzerine Yapılmış Çalışmalar

PMV modeli P.O. Fanger tarafından ilk yayınlanan zamanlardan beri çok ilgi görmüş ISO7730 ve ASHRAE 55 standartlarının temelini oluşturmuş bir çalışmadır. Bu çalışmalar bir grup üniversite çağındaki öğrenciler üzerine, değişmeyen termal koşulların olduğu ortamlarda, şartlandırılmış hava ile kontrol edilen laboratuvarlarda yapılmıştır. Ayrıca deney ortamı ortalama iklimin yaşandığı ve bu iklime alışkın olarak yaşayan kişilerle yapılmıştır. PMV hesabı ortalama insan ve çevre şartlarının olduğu birçok ortam için hala daha başarılı bir çalışma olarak görülmektedir. Ancak özellikle son yıllarda etkinliği üzerine oldukça fazla tartışılmaya başlanılmıştır. Bu durum öncelikle PMV hesabında kişisel ve çevresel birçok faktörün olamamasından kaynaklanır (Van Hoof 2008). Bu faktörler arasında demografik özellikler (cinsiyet, yaş, ekonomik durum), şartlar (bina tasarımı, yapı işlevi, mevsim, iklim, semantik, sosyal koşullandırma) ve kavram (tutum, tercih ve beklentiler) yer alabilir (de Dear ve diğ. 1997).

Bu durum karşısında çok çeşitli termal ortamlarda, farklı ırklar üzerine, farklı yaş gruplarında, farklı mekanlarda (ofis, stadyum, uçak, yurt vb.) çeşitli çalışmalar yapılmış farklı denek gruplarından farklı veriler oluşturulmuştur. Bu verilerin ortak özelliği neredeyse tamamının Fanger'in PMV metodu ile karşılaştırılarak çalışılmış olmasıdır.

3.2 ASHRAE RP-884 Adaptif Model Projesi

PMV ölçeğini yeni bir denek grubuyla farklı ortamlarda tekrar değerlendiren birçok çalışma vardır. Bu çalışmalardan belki de en büyüğü ASHRAE RP-884 Adaptif Model Projesi'dir (de Dear 1997). Bu proje çerçevesinde dünya genelinde 24 farklı şehirde, 160 farklı binada 46 farklı çalışma yapılmış ve toplam 20693 örnek toplanmıştır. Örnek alınan bölgelerin haritası şu şekildedir:



Şekil 3.1: ASHRAE RP-884 çalışmasının dünya üzerinde yapıldığı bölgeler.

Bu araştırma projesi, dünya çapında birçok noktada gerçekleştirilmiş olan daha önceki termal konfor çalışmalarından derlenen, kontrollü bir veri tabanının geliştirilmesi ve bu veri tabanının analizinin yapılabilmesi üzerine kurulmuştur. Bu projenin amacı adaptif sistemlere dair kavramsal öngörülerini iyileştirmek, adaptif sürecin deneysel bir modelini geliştirmek ve mevcut ASHRAE-55 Standardını desteklemek için değişken bir sıcaklık standardı önermek ve ileriye dönük çalışmaların yapılabilmesi için bu çalışmada oluşturulan veri tabanını kullanmaktır.

RP-884 Projesi'nin veri tabanı halka açıktır. Birçok öğrenen, adaptif algoritmaları termal konfor sistemi için karşılaştırma ya da deneysel çalışma yapmak amacıyla kullanılmıştır.

3.3 Yapay Sinir Ağları ile Yapılmış Çalışmalar

Termal konfor modelinin bölgesel ve kişisel olarak çözülmesine birçok yapay sinir ağı yaklaşımı mevcuttur. Bu çalışmalar hangi yapay sinir ağı algoritmasının kullanıldığından, hangi veri kümesi ile çalışma yapıldığına kadar çok farklı yöntemle geliştirmişlerdir. Bu çalışmalar arasından bu tezin konusuna en yakın olan birtakım örnekler belirlenmiş ve özet şeklinde bahsedilmiştir.

Li N. ve ekibi (2012) tarafından oluşturulan bir makalede GA-BP yapay sinir ağı modeli kullanılarak ve gerçek termal duyu örneklerine dayanarak, doğal olarak havalandırılmış ortamlarda, bina sakinlerinin termal hissini tahmin etmek için bir yöntem sunulmaktadır. Bu yöntem, GA-BP yapay sinir ağı, BP yapay sinir ağının olumsuz yönlerini ortadan kaldırmak için bir genetik algoritmayı dahil ederek geliştirilmiştir. Bu adaptif termal konfor sistemi Çin'deki Yangtze Nehri Bölgesi'ndeki bir konut binasında yapılan saha çalışmalarında, sonuçların gerçek termal duyu oyları ile karşılaştırılarak test edilmiştir. Sonuçlar maksimum sapmanın% 3,5 olduğunu göstermektedir.

Bu çalışmanın olumsuz yanı sonuçların çok kısıtlı bir bölgede ve çok kısıtlı bir veri sayısı ile test edilmiş olmasıdır. Sistemi eğitmek için 100 veri toplanmış bir diğer 41 veri de sistemin validasyonu için kullanılmıştır. Maksimum hata 0.21 (PMV ölçeği) ve maksimum öngörü deviasyonu %3,5 olarak verilmiştir.

Castilla, M. ve ekibi (2013) tarafından yapılmış olan bu çalışmada yapay sinir ağı ve 7. Dereceden polinom yaklaşımı metotları ile adaptif termal konfor modeli oluşturulmuştur. Veri kümesi PMV ile hesaplanmıştır. Ancak met değeri sabit 1,0 olarak alınmış, clo için ise kış analizleri için 1,0 yaz analizleri için 0,5 olmak üzere iki değer üzerinden veri kümesi oluşturulmuştur. Diğer 4 ölçülebilir parametre için oluşturulan veri kümesi aralığı şu şekildedir:

Hava sıcaklığı : 10 °C - 30 °C arasında 1.5 °C adımlarla

Globe Sıcaklığı (Rad. Sıc.) : 10 °C - 37.25 °C arasında 2 °C adımlarla

Hava hızı : 0.06m/s - 0.5m/s arasında 0.05m/s adımlarla

Bağıl Nem : %30 - %70 arasında %4 adımlarla

Sonuç olarak her iki sistem de bu verilere göre başarılı bir şekilde eğitilebilmiş ve oldukça düşük hatalar elde edilebilmiştir. Polinom yaklaşımı metodunun yapay sinir ağına göre çok daha fazla sistemsel yük getirdiği belirtilmiştir.

Ferreira, P.M. ve ekibi (2012) tarafından yazılan bu makalede yine 6 farklı sanal ortam koşulu yaratılarak farklı PMV değerleri ile bir eğitim kümesi oluşturulmuş ve bu eğitim kümesi ile yapay sinir ağı kullanılarak adaptif bir sistem geliştirilmiştir.

Burada oluşturulmuş olan veri kümesinin 6 parametresi şu şekilde oluşturulmuştur:

Hava sıcaklığı : 16 °C - 32 °C arasında rastgele seçimle

Bağıl Nem : %20 - %70 arasında rastgele seçimle

Globe Sıcaklığı (Rad. Sıc.) : her bir hava sıcaklığı için -3...+3 arasında rastgele bir sayı ile toplayarak

Hava hızı : 0.08 m/s sabit

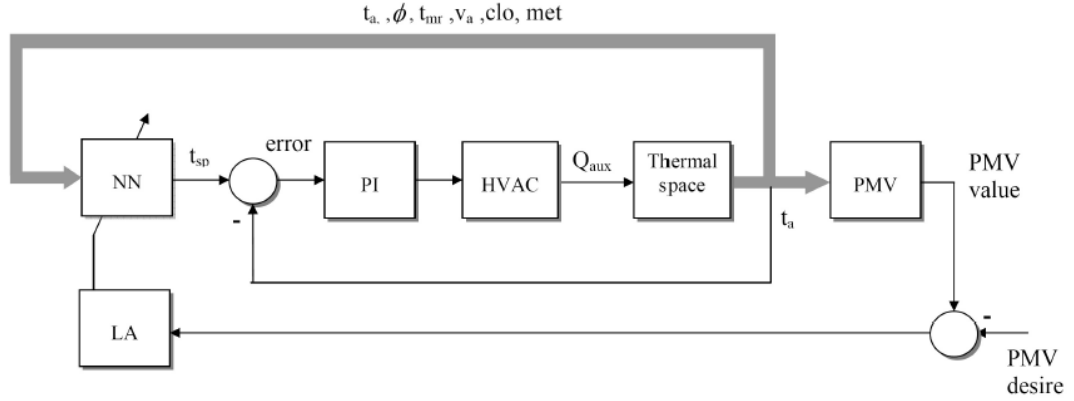
Metabolik Hız : 1.2 met sabit

Giysi Yalıtımı : 0.85 clo sabit

Bu parametrelere göre 1600 veri ve 80 nöron ile eğitilen bir yapay sinir ağı yaklaşık olarak 0.015 hata ile PMV tahmini yapabilmıştır. Bu sinir ağı ile kontrol edilmiş bir HVAC havalandırma sistemi %50 enerji tasarrufu sağlanmıştır.

Torres, J. L. ve ekibi (2008) tarafından yazılmış olan bu makalede diğer birçok makaleden farklı olarak termal konfor koşullarının kendisi değil bu koşulları

sağlayacak bir HVAC sistemi üzerine çalışılmıştır. Sistem kullanıcıdan bir PMV referans değeri alır ve ölçülen parametrelerden hesaplanan gerçek PMV değeri ile farkını alır. Bu fark hatasını 0'da tutmak için yapay sinir ağını online olarak eğiterek referans sıcaklığı kontrol eder. Bu referans sıcaklık da PI kontrollü HVAC sistemini çalıştırır. Sistem döngüsü şu şekildedir:



Şekil 3.2: Yapay sinir ağı ile sıcaklık kontrolü yapılarak PMV hatasını minimum yapmaya yönelik sistem.

Böyle bir sistemin avantajı PMV değerinin doğru olup olmamasıyla temelde ilgilenmiyor olmasıdır. PMV yanlış ölçülse bile burada insan faktörü PMV değerini kendi isteğine göre mevcut değer altında ya da üstünde girerek sistemi ayarlamayı başarabilir. Tabii böyle bir sistem tek bir parametreyi kontrol edebilen sistemler için kullanışlı olsa da eğer bir sistem T_a , T_r , V_a , bağıl nem değerlerinin birden fazlasını bağımsız kontrol edebiliyorsa ya da bunları ayrı ayrı kontrol edebilecek birden fazla alt sistemi mevcutsa hangi parametreyi düzeltmesi gerektiğini bilemez. Ayrıca kişilerin hava hızına ya da radyan sıcaklığa diğer parametrelerden daha fazla duyarlı olması durumunda yetersiz kalabilir. Bu aşamada PMV değerinin kişiye özel olarak tekrar düzenlenmesini gerekebilir. Bu da bizi PMV'nin öğrenilmesi problemine geri götürmektedir.

3.4 SVM ile Yapılmış Çalışmalar

Yapay sinir ağı ile probleme farklı yönlerden yaklaşan, farklı veri kümeleri ya da farklı yapay sinir ağı yapıları kullanan, farklı ortamlarda, farklı deney grupları ile yapılmış birçok çalışma bulunmaktadır ve bu çalışmaların bazılarında yukarıda

bahsedilmiştir. Aynı durum SVM için de geçerlidir. Bu çalışmaların bu tezin konusuna en yakın olanlarından ve en çok ilgi gören birkaç tanesi şu şekilde sıralanabilir.

Bin, S. ve Ke H. (2010) tarafından yapılmış olan bu çalışmada LS-SVM yöntemiyle regresyon yapılarak PMV tahmini yapılıyor. Bu çalışmanın en çarpıcı taraflarından biri hem C parametresinin hem de σ parametresinin “Particle Swarm Optimization” algoritması ile optimize edilmesi. Bu çalışmada rastgele 1000 veri seçilerek 800 veri ile öğrenme, 200 veri ile de test yapılıyor. PMV parametreleri şu aralıkta seçilmiş:

Hava sıcaklığı	: 22 °C - 28 °C arasında
Bağıl Nem	: %20 - %70 arasında
Radyan Sıcaklık	: 22 °C - 28 °C arasında
Hava hızı	: 0,1 m/s - 0,3 m/s arasında
Metabolik Hız	: 70-120 W/m ² (1.2 - 2.1 met)
Giysi Yalıtımı	: 0.07 - 0.16 m ² K/W (0.45 - 1.03 clo)

C ve σ parametresinin optimizasyonu yapılırken kullanılmış olan PSO algoritması aynı optimizasyon iterasyonu sayısında 0.0282 mutlak hata verirken “grid search” algoritması 0.040 mutlak hata vermiştir. Burada ortaya çıkan başka konu ise C parametresinin σ parametresi kadar sisteme etkisi olmamasının görülmesidir. Dolayısı ile belirli bir C değerinde σ parametresinin minimize edilmesi için tarama yapılması yeter olabilir.

Megri, A. C. Ve ekibi (2005) tarafından yapılmış olan bu makale, bu tezde bahsedilen konunun regresyonla yapılmış olanı gibidir. PMV ile 793 verilik bir eğitim kümesi oluşturulmuştur. Bu küme oluşturulurken PMV girişleri şu aralıkta tutulmuştur.

Hava sıcaklığı	: 10 °C - 35 °C arasında
----------------	--------------------------

Bağıl Nem	: %0 - %100 arasında
Radyan Sıcaklık	: 10 °C - 35 °C arasında
Hava hızı	: 0.1 m/s - 3 m/s arasında
Metabolik Hız	: 0.7 - 6.3 met
Giysi Yalıtımı	: 0.1 - 1.17 clo

Görüldüğü gibi çalışılan aralık çok geniştir. Ancak büyük ihtimalle doğru seçilmiş bir veri kümesi ile bu geniş aralıkta bile sistem düzgün bir şekilde eğitilebilmiştir. 2014 yılında yazdığı benzer makalede de 793 verilik bir eğitim kümesinden bahsetmiş olması bu ihtimali arttırmaktadır. Her iki makalede de sistemin 18 verilik bir test kümesi ile test edildiğinden bahsetmektedir. Daha önceki çalışmasında lineer çekirdek fonksiyonu ile 2. ve 3. dereceden polinomial çekirdek fonksiyonları arasında karşılaştırma yapmış ve SVM'nin karmaşıklığı çok fazla arttırmadan en iyi sonucun 2. Dereceden polinom çekirdek fonksiyonu ile olduğunu belirtmiştir. Buna rağmen 2016 yılında yazdığı bu makalede doğrusal çekirdek fonksiyonu kullandığını yazmıştır. Sonuçların karşılaştırılması konusunda matematiksel çıkarım çok fazla yapılmamıştır. Aynı ekip benzer bir çalışmayı 2016 yılında da yapmıştır (Megri 2016).

Jiang, L. ve Yao, R. (2016) tarafından yapılmış olan bu çalışmada eğitim veri kümesi gerçek deneklerden alınmıştır. Aynı clo değerine sahip giysiler giyen ve aynı düzeyde aktivite yapan 20 kişi maksimum 10 defa olmak üzere 90 dakikalık deneylere katılmışlar 1199 adet veri toplamışlardır. Bu veriler -3 ile +3 arasında bir termal konfor oylaması ile toplanmıştır. Bu verilerin ortam şartları şu aralıkta ayarlanmıştır:

Hava sıcaklığı	: 24.94 °C - 29.58 °C arasında
Bağıl Nem	: %41.5 - %80.1 arasında
Radyan Sıcaklık	: 26.07 °C - 30.04 °C arasında
Hava hızı	: 0.11 m/s - 0.17 m/s arasında

Kernel olarak “Gaussian Radial Basis Function” kullanılmıştır. Sınıflandırma -3 ile +3 arasında 7 tane veriden oluştuğu için bu toplanan verilerle her bir SVM için “Grid Search” yapılarak optimum C ve σ bulunmuş ve çok sınıflı SVM eğitimi yapılmıştır.

Oluşturulmuş olan SVM modeli bu sefer yine aynı deneklerle teste tabi tutulmuştur. Yeni verilerle yapılan bu testte ortalama hata %89.82 bulunmuştur. 20 denekten sadece 3 tanesi %80'nin altında hatalı sonuç çıkarımı yapmıştır.

Aynı veriler Fagner tarafından önerilmiş olan PMV denklemleri ile karşılaştırıldığında Fagner'in PMV modelinin %49.71 doğru bilme oranına sahip olduğu belirtilmiştir.

Ahmad F., A. ve ekibi (2015) tarafından yapılmıştır olan bu çalışma bir sınıflandırma çalışmasıdır ve “konforsuz soğuk”, “konforlu” ve ”konforsuz sıcak” olarak 3 sınıfla yapılmıştır.

Bu çalışmanın en önemli özelliği veri kümesinin seçimidir. Daha önce bahsedilmiş olan RP-884 içinden seçilerek alınan veri kümesinde kuru hava sıcaklığı, radyan sıcaklık, kıyafet izolasyonu, metabolik hız, hava türbülansı, dış ortam sıcaklığı ve yaş, ektim uzayının parametreleri olarak belirlenmiştir. Yani -standart PMV modeline kıyasla bağıl nem kullanılmamış ama yaş ve dış hava sıcaklığı parametreleri eklenmiştir. Ancak daha sonrasında yapılan birçok simülasyon sonucunda ortalama radyan sıcaklık parametresinin de bir etkisi olmadığı görülmüş. Bu sebeple bu parametreyi veri setinden çıkararak tekrar 6 parametreye düşürmüşlerdir. Bu durumun niye gözlemlendiği konusunda bu tezde de bir çıkarım yapılacaktır.

Bu çalışmada daha sonra Fanger'in klasik PMV metodu, SVM metodu, “Random Forest” ve “Adaboost.M2” metodu ile yapılmış sistem karşılaştırılmıştır. Sonuçta SVM metodu diğer metotlara göre daha iyi sonuç vermiş, Fanger'in PMV metodu ise SVM'ye kıyasla yarı yarıya daha yanlış öngörülerde bulunmuştur.

Daha önce bu yapılan çalışmalar SVM'nin bu konuda oldukça iyi sonuçlar verdiğini gösteriyor. Sınıflandırma da regresyon olan çalışmalarda da bu etki gözlemlenmekte. Ayrıca bu çalışmalarda Fanger'in metodunun da bazı durumlarda

düşük doğrulukta sonuçlar verdiği belirtilmektedir. Bu da farklı bir adaptif sistem ihtiyacını daha fazla ortaya koyuyor. Bu çalışmada iki farklı konfor aralığında doğruluk oranları şu şekilde verilmiştir:

Tablo 3.1: Çalışma için doğru tahmin oranları

Konforlu aralık	$-0.5 < PMV < 0.5$	$-1 < PMV < 1$
SVM	56.7%	76.7%
Random Forest	52.1%	74.1%
Adaboost.M2	51.1%	61.4%
Fanger's Model	33.2%	35.4%

Bunun yanı sıra termal konfor modellemesi için Exhaustive General CHAID (Chi-square Automatic Interaction Detector), Boosting Tree, Random Forest ve MLP (multi-layer perceptron) algortimalarının kullanıldığı ve SVM ile karşılaştığı çalışmalar da mevcuttur (Kusiak, A. ve diğ. 2011).

4. DESTEK VEKTÖR MAKİNALARI

Bir Makine Öğrenmesi (Machine Learning) yaklaşımı olan Destek Vektör Makinesi (Support Vector Machines - SVM), etiketli bir eğitim verisi kümesinden girdi-çıkı haritalama fonksiyonları üreten öğreticili (supervised) bir öğrenme yöntemidir. Bu haritalama fonksiyonu, sınıflandırma (classification) fonksiyonu da regresyon fonksiyonu da olabilir. Sınıflandırma için girdi verilerini orijinal girdi uzayına göre daha ayrılabilir hale getiren yüksek boyutlu bir uzaya dönüştürmek için nonlinear kernel (çekirdek) fonksiyonları kullanılmaktadır. Bunun ardından veri sınıflandırmasını en iyi şekilde yapabilmek adına maksimum marjli hiper-düzlemler kullanılmaktadır. Bu şekilde üretilen model sadece sınıf sınırlarının yakınındaki eğitim verilerinin bir alt kümesine bağlıdır. İşte bu alt kümeye destek vektörleri adı verilmektedir.

İstatistiksel öğrenme teorisindeki sağlam matematiksel temeline ek olarak, SVM'ler, biyoinformatik, metin madenciliği, yüz tanıma ve görüntü işleme gibi çok gerçek uygulamalarda yüksek performansı sergilemiştir. Bu özellik SVM'leri makine öğrenimi ve veri madenciliği için gerekli diğer araçlara kıyasla (örneğin, yapay sinir ağları ve bulanık sistemler) daha gelişmiş bir araç haline getirmiştir (Kecman 2005).

4.1 SVM Tarihçesi

Günümüzde kullanılan Destek Vektör Makineleri çerçevesinde geliştirilmekte olan fikirlerin birçoğu ilk olarak "V. N. Vapnik" ve "A. Ya. Chervonenkis" (Rusya Bilimler Akademisi, Moskova, Rusya) tarafından, bilgisayar öğrenimi ve örüntü tanıma için "Genelleştirilmiş Portre Yöntemi" amacıyla ortaya atılmıştır. Bu fikirlerin gelişimi 1962'de başlamış ve ilk olarak 1964'te yayınlanmıştır (Schölkopf 2013). 1992'de Bernhard E. Boser, Isabelle M. Guyon ve Vladimir N. Vapnik, en yüksek marjli hiper-düzlemlere kernel fonksiyonu uygulayarak doğrusal olmayan sınıflandırıcılar yaratmanın bir yolunu önermişlerdir. Mevcut standart (yumuşak marjin) 1993 yılında Corinna Cortes ve Vapnik tarafından önerilmiş ve 1995 yılında yayınlanmıştır (Cortes 1995).

Günümüzde SVM'ler hem sınıflandırma hem de regresyon amacıyla birçok uygulamada kendine yer bulmaktadır.

4.2 SVM Uygulamaları

SVM, birçok gerçek problemde başarıyla kullanılmıştır (Schölkopf 1999). Bunlardan bazıları şu şekilde yazılabilir:

- Metin (ve hipermetin) kategorizasyonu
- Görüntü sınıflandırması
- Biyoinformatik (Protein sınıflaması, Kanser sınıflandırması)
- El yazısı karakter tanıma (Burges 1998)
- Yüz tanıma sistemleri
- Model öngörülü kontrol uygulamaları

SVM'ler hem sınıflandırma hem de regresyon olarak kullanılabilirler. Bu sebeple birçok çok-boyutlu problem için ideal çözüm yöntemleri sunarlar. SVM'lerin probleme göre parametrelerinin, çekirdek fonksiyonlarının değiştirilebilmesi bu noktada büyük avantajlardan biridir. Bunun dışında farklı problemlere farklı yaklaşımlar sağlayacak çeşitli SVM türleri de vardır. Bu konu oldukça geniştir ancak bu tezin konusu ve amacı olmadığı için burada anlatılması uygun görülmemiştir.

4.3 SVM'nin Diğer Sistemler ile Karşılaştırılması

SVM'ler özellikle çok sınıflı sınıflandırıcı uygulamalarında kendini kanıtlamış bir yöntemdir. Protein yapısı sınıflandırması için yapılmış bir çalışmada yapılan karşılaştırmada şu sonuç çıkarılmıştır: "polinom çok-sınıf SVM ve MLP-NN (MLP yapay sinir ağı), diğer sınıflandırıcılardan daha iyi performans gösterirken, SVM toplam doğruluk açısından en iyi performans gösteren algoritma olmuştur. Seçilen SVM, 27 sınıf problemde %42.8 doğruluk elde etti. Bu doğruluk oranı

karmaşık sınıflandırma problem için kabul edilebilir bir performanstır.” (Valavanis 2010)

Yine yapay sinir ağları ile karşılaştırılmış olan başka bir çalışmanın sonuç bölümünde şu şekilde bir ifade kullanılmıştır: “Bu çalışmada kullanılan SVM sisteminin, belirli bir ANN mimarisinden daha yüksek genel tahmin doğruluğu üretme kapasitesine sahip olduğu gösterilmiştir. Bu gözlemden yola çıkarak, SVM'nin, özellikle çok sayıda girdi değişken kullanıldığında, QSAR modelleme ve sanal taramada sınıflandırma görevleri için yararlı bir yöntemi temsil ettiği sonucuna vardık.” (Byvatov 2013)

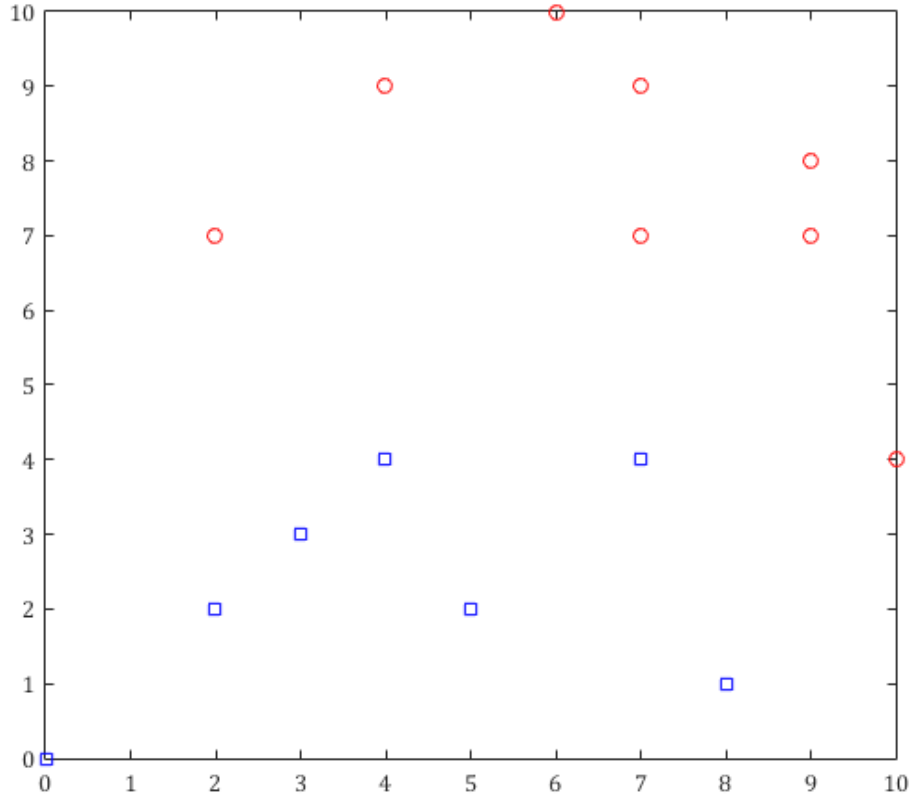
Bunun dışında bu tezin 4. Bölümünde de daha önce yapılmış çalışmaları incelerken SVM'nin “Random Forest”, “Adaboost.M2” gibi farklı algoritmalarla da karşılaştırılmış bir çalışmada ve farklı alanlarda yapılmış başka çalışmalarda da yine SVM'nin sınıflandırıcı konusundaki performansı dikkat çekmektedir (Jodas 2013).

4.4 Doğrusal Sınıflandırma

Varsayalım ki aşağıdaki formatta bir eğitim kümemiz olsun:

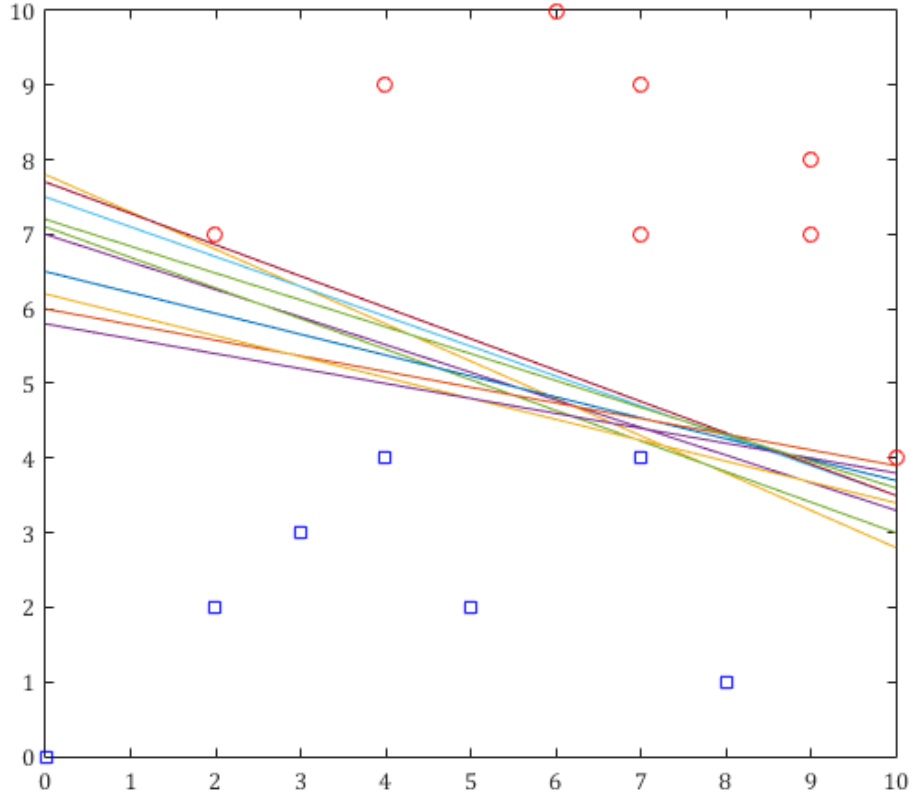
$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$$

Burada \mathbf{x}_i verisinin vektörel koordinatlarını verirken, y_i her bir \mathbf{x}_i verisinin hangi kümeye ait olduğu gösterir. y_i , -1 ya da $+1$ değerini alır. Her bir \mathbf{x}_i p-Boyutlu bir uzayda bulunan vektördür. Örnek olarak Şekil 3.1'de 2-boyutlu bir uzaydaki veri kümesleri gösterilmiştir.



Şekil 4.1: Örnek veri kümesi. Mavi kareler ' $y_i = -1$ ', kırmızı yuvarlaklar ' $y_i = +1$ ' olarak işaretlenmiş verileri gösterir.

Destek vektör makinelerinde veri sınıflandırması $y_i = -1$ ve $y_i = +1$ olarak işaretlenmiş veri gruplarının bir hiper-düzlem ile ayrılması ile sağlanmaktadır. Bu şekilde gruplandırılmış verileri ayıran sonsuz adet hiper-düzlem üretilebilir. Destek Vektör Makinaları lineer sınıflandırma problemlerinde hiper-düzlemler doğrusal olarak tanımlanmaktadır. Bu durumda örnekteki gibi iki boyutlu bir uzaydaki iki grubu ayıracak olan düzlemler çizgi ile gösterilebilir. Bu iki grubu ayıracak sonsuz adet çizgi tanımlanabilir. Şekil 3.2'de Bu iki grubu ayıracak olası çizgilerden bazıları gösterilmektedir.



Şekil 4.2: İki veri kümesinin ayırabilecek sonsuz adet hiper-düzlemlerden bazıları.

Bu hiper-düzlemler matematiksel olarak şu şekilde tanımlanabilirler:

$$\mathbf{w}^T \mathbf{x} + b = 0$$

Burada \mathbf{w} hiper-düzleme dik vektördür ve normalize edilmesi şart değildir. Bu hiper-düzlemlerden sınıflandırma işini en iyi yapabilecek olanı, marjini maksimize eden yani her iki veri kümesine de uzaklığı eşitlenmiş olan hiper-düzlemdir ve bu şekilde sadece bir adet hiper-düzlem vardır. Bu hiper-düzleme “maksimum-marjimli sınıflandırıcı” adı verilir.

Maksimum marjimli sınıflandırıcı hem en yakın $y_i = +1$ verisine hem de en yakın $y_i = -1$ verisine aynı uzaklıktadır. Bu verilere karşılık gelen vektörleri sırasıyla \mathbf{t}^+ ve \mathbf{t}^- olarak adlandıırırsak, maksimum marjimli sınıflandırıcının bu noktalara olan uzaklıkları şu şekilde hesaplanır.

\mathbf{t}^+ noktasının $\mathbf{w}^T \mathbf{x} + b = 0$ düzlemine olan uzaklığı:

$$\frac{|\mathbf{w}^T \mathbf{t}^+ + b|}{\|\mathbf{w}\|}$$

\mathbf{t}^- noktasının $\mathbf{w}^T \mathbf{x} + b = 0$ düzlemine olan uzaklığı:

$$\frac{|\mathbf{w}^T \mathbf{t}^- + b|}{\|\mathbf{w}\|}$$

Marjin bu iki uzaklığın toplamı olarak ifade edilirse:

$$m = \frac{|\mathbf{w}^T \mathbf{t}^+ + b|}{\|\mathbf{w}\|} + \frac{|\mathbf{w}^T \mathbf{t}^- + b|}{\|\mathbf{w}\|}$$

Ayrıca SVM sınıflandırmasında maksimum marjinli sınıflandırıcıya paralel ve bu düzleme en yakın noktaların üzerinden geçen düzlemler şu formülle ifade edilir:

\mathbf{t}^+ noktasından geçen düzlem:

$$\mathbf{w}^T \mathbf{t}^+ + b = 1$$

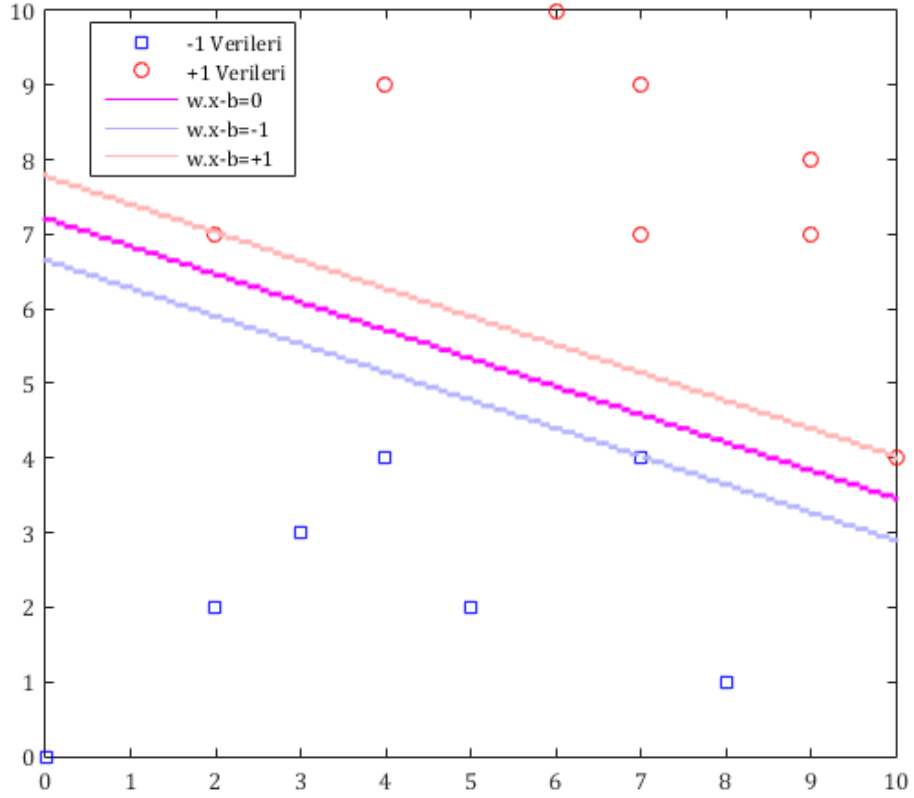
\mathbf{t}^- noktasından geçen düzlem:

$$\mathbf{w}^T \mathbf{t}^- + b = -1$$

Bu iki paralel düzlemin arasındaki uzaklık yani marjin şu şekilde hesaplanır:

$$m = \frac{|+1|}{\|\mathbf{w}\|} - \frac{|-1|}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$

Buna ait grafiği Şekil 3.3'de görebilirsiniz.



Şekil 4.3: Maksimum marjınlı lineer sınıflandırıcı

$m = \frac{2}{\|\mathbf{w}\|}$ formülü ile hesaplanan marjinin maksimum olduğu, yani $\|\mathbf{w}\|$ değerinin minimum olduğu düzlem bize maksimum marjınlı sınıflandırıcı düzleminin formülünü verir. $\mathbf{w}^T \mathbf{x} + b = 1$ ve $\mathbf{w}^T \mathbf{x} + b = -1$ düzlemleri üzerindeki noktalar da “Destek Vektörleri” olarak adlandırılır. Şekil 3.3’de bir adet $y_i = -1$ değerli ve iki adet $y_i = +1$ değerli destek vektörü görülmektedir.

4.4.1 Sert Marjınlı Sınıflandırıcı

Bir Destek Vektör Makinesinin eğitilmesini sağlayan algoritmaların kullandığı verinin bütünü “eğitim verisi” olarak adlandırılır. Bu eğitim verisinin lineer olarak ayrılabilirdiği üstteki gibi bir örnekte doğrusal sınıflandırıcı rahatlıkla kullanılabilir. Bu sınıflandırıcıda $y_i = +1$ değerli tüm veriler sınıflandırıcı hiperdüzlemin bir tarafında, $y_i = -1$ değerli veriler ise diğer tarafında yer almaktadır. Bu

durumda marjinin $\frac{2}{\|\mathbf{w}\|}$ olduğundan bahsetmiştik. Bu marjini maksimize etmek için $\|\mathbf{w}\|$ değerini minimize etmemiz gerektiği açıktır. Ancak burada iki tane kısıt vardır.

- $y_i = +1$ ile işaretlenmiş eğitim verileri $\mathbf{w}^T \mathbf{x} + b \geq 1$ kısıtını sağlamalıdır.
- $y_i = -1$ ile işaretlenmiş eğitim verileri $\mathbf{w}^T \mathbf{x} + b \leq -1$ kısıtını sağlamalıdır.

Bu durumda bütün i 'ler için problem şu şekilde yazılabilir:

$$\min_{w,b} \|\mathbf{w}\|$$

$$\text{kısıtlar: } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq +1 \quad i = 1, \dots, N$$

Bu optimizasyon probleminin çözümü bize sınıflandırmayı sağlayacak olan hiper-düzlemin fonksiyonunu yani " \mathbf{w} " ve " b " değerlerini verecektir. Destek Vektör Makinesini bu şekilde eğittikten sonra yeni bir \mathbf{x}_c verisinin hangi sınıfa ait olduğunu belirlemek için ise şu fonksiyon kullanılır:

$$\mathbf{x}_c \rightarrow \text{sgn}(\mathbf{w}^T \mathbf{x}_c + b)$$

Eğer $\mathbf{w}^T \mathbf{x}_c + b$ 'nin sonucu pozitifse veri +1 grubuna, negatifse -1 grubuna aittir.

4.4.2 Esnek Marjinli Sınıflandırıcı

Bazı durumlarda eğitim verisi doğrusal olarak ayrılabilir durumda olmaz ya da eğitim verisinin genel olarak lineer ayrılabilir durumda olsa da bazı veriler bu kurala uymuyor gibi görünebilir. Bu durumda esnek marjinli destek vektör makinesi kullanılır. Bu sayede yanlış tarafta bulunan veriler cezalandırılarak optimizasyon yapılır. Bunun için Hinge kayıp fonksiyonu kullanılabilir.

Hinge kayıp fonksiyonu şu şekildedir:

$$\max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$$

Burada y_i olması gereken çıkış değeri $(\mathbf{w}^T \mathbf{x}_i + b)$ ise destek vektör makinesi tarafından hesaplanan çıkış değeridir. Eğer daha önceki optimizasyon probleminde bahsetmiş olduğumuz $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq +1$ kısıtı sağlanıyorsa Hinge kayıp fonksiyonunun çıkışı '0' olur. Başka bir deyişle eğer veri doğru bölgede ve marjinin dışında ise fonksiyonun bir etkisi olmaz eğer veri yanlış taraftaysa $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq +1$ eşitsizliği sağlanamayacağından dolayı fonksiyonun çıkışı verinin yanlış tarafta olduğu oranda artar. Veri doğru tarafta ama marjinin içinde ise çıkış 1'den küçük olur. Hiper-düzlemin diğer tarafına geçtiği oranda 1'den büyük değerler alır.

Hinge kayıp fonksiyonu aslında kısıtları esnetmek için kullanılır ve aşağıdaki şekilde de ifade edilebilir:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

Burada ξ_i i değeri o verinin yaptığı hatayı ifade eder ve gevşek değişken olarak adlandırılmaktadır.

Sert marjinli sınıflandırmada $\|\mathbf{w}\|$ parametresi minimize edilirken yumuşak marjin de minimize edilecek şekilde minimizasyon formülüne ceza katsayısını yani gevşek değişkeni de ekleriz ve minimize etmek istediğimiz fonksiyon şu hali alır:

$$\min_{\mathbf{w}, b} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$$

Burada C değeri marjin genişliği ile \mathbf{x}_i değerinin doğru tarafta yer alması arasında bir tercih yapmak için ayarlanır. C sonsuz değer aldığında fonksiyon sert marjinli bir sınıflandırıcı haline dönüşür. Küçüldükçe yumuşak marjinli hale gelir.

4.5 Karesel Programlama

Daha önce görmüş olduğumuz Optimizasyon problemi aslında “karesel programlama problemi” olarak adlandırılan standart bir yapıdadır. Standart karesel programlama problemi şu şekilde tanımlanmaktadır:

$$\min \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{g}^T \mathbf{x},$$

$$\text{Kısıtlar: } \mathbf{A}_{eq} \mathbf{x} = \mathbf{b}_{eq}$$

$$\mathbf{A}_{ie} \mathbf{x} \geq \mathbf{b}_{ie}$$

Buradaki $\mathbf{x} \in \mathbb{R}^n$ tasarım değişkenleri vektörü olup diğer matrislerin boyutları şu şekildedir:

- n : Tasarım değişkenleri sayısı
- N_{eq} : Eşitlik kısıtları sayısı
- N_{ie} : Eşitsizlik kısıtları sayısı
- \mathbf{x} : $n \times 1$ boyutlu tasarım değişkenleri vektörü
- \mathbf{H} : $n \times n$ boyutlu simetrik kare matris, Hessian matrisi
- \mathbf{g} : $n \times 1$ boyutlu vektör
- \mathbf{A}_{eq} : $N_{eq} \times n$ boyutlu eşitlik kısıtları matrisi
- \mathbf{b}_{eq} : $N_{eq} \times 1$ boyutlu eşitlik kısıtları vektörü
- \mathbf{A}_{ie} : $N_{ie} \times n$ boyutlu eşitsizlik kısıtları matrisi
- \mathbf{b}_{ie} : $N_{ie} \times 1$ boyutlu eşitsizlik kısıtları vektörü

Buradaki amaç, yukarıdaki fonksiyonu minimize edecek \mathbf{x} -vektörünü bulmaktır. Karesel bir fonksiyon her zaman $\min \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{g}^T \mathbf{x}$ formuna dönüştürülebilir ve bu şekilde farklı standart algoritmalarla kolayca çözülebilir. Eğer \mathbf{H} matrisi pozitif yarı-tanımlı olduğu zaman problem konveks optimizasyon problemine dönüşerek çözümü kolaylaştıran özel bir hal alır.

Bizim mevcut optimizasyon problemimizi şu şekilde oluşturmuştuk:

$$\min_{w,b} \|\mathbf{w}\|$$

$$\text{kısıtlar: } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq +1 \quad i = 1, \dots, N$$

Bu problemi karesel programlama problemine şu şekilde dönüştürebilir:

$$\min_{\mathbf{w}, b} \frac{1}{2} \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}^T \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix}$$

$$\text{kısıtlar: } \begin{bmatrix} y_1 \mathbf{x}_1 & y_1 \\ \vdots & \vdots \\ y_i \mathbf{x}_i & y_i \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \geq 1 \quad i = 1, \dots, N$$

Sonuç olarak destek vektör makinelerinin optimizasyonu için problemi karesel programlama problemi haline dönüştürerek çözen özellikle bilgisayar ortamında yapılacak olan optimizasyon için bilgisayarın daha az kaynak kullanmasını sağlar. Bu tezde karesel programlama problemi MATLAB, “Optimization Toolbox” içerisinde yer alan “quadprog” fonksiyonu kullanılarak çözülmüştür. “interior point convex” algoritması tercih edilmiştir.

4.6 Doğrusal Olmayan Sınıflandırma

Birçok sınıflandırma problemi için doğrusal sınıflandırma yetersiz kalacaktır. Verilerin özellikle çok boyutlu problemlerde dağılımı doğrusal olmadığından veri sınıflandırması için kullanılacak olan hiper-düzlem de doğrusal olmayan yapıya sahip olmalıdır.

Destek vektör makinaları “Kernel Based” yani çekirdek fonksiyonu tabanlı sınıflandırıcılar grubuna girer. Özellikle doğrusal olmayan sınıflandırıcılar için çekirdek fonksiyonu kullanılarak doğrusal olmayan hiper-düzlem oluşturmayı kolaylaştırılır.

Vapnik tarafından 1963 yılında destek vektör makinaları ilk önerildiğinde sadece doğrusal sınıflandırma için oluşturulmuştu. Ancak 1992 yılında Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik çekirdek fonksiyonu kullanarak destek vektör makinaları ile doğrusal olmayan sınıflandırıcı yapılabileceğini gösterdiler.

Çekirdek fonksiyonu kullanımı için öncelikle problem dual uzay olarak adlandırılan Lagrange uzayına aktarılması gerekir. Lagrange uzayının özelliği problemi daha yüksek boyutlu bir uzaya taşıyarak daha rahat çözülmesini sağlamasıdır. Dual uzay ile primal uzayın çözümü aynıdır. Bu da primal uzayda bulunması zor bir çözümün dual uzaya dönüştürülerek bulunabileceği anlamına gelmektedir.

Dual uzaya geçişin ardından dual uzayda “Çekirdek fonksiyonu” olarak adlandırılan Kernel Fonksiyonu yöntemi kullanılarak problem nonlinear bir hale dönüştürülür.

Şu ana kadar bahsedilmiş olan fonksiyonların hepsi problemin gerçekten var olduğu boyuta yani primal uzaya aittir. Primal uzaydan dual uzaya geçiş Lagrange fonksiyonları kullanılarak dönüştürülür. Lagrange dönüşümü şu şekildedir:

4.6.1 Lagrange Dönüşümü

Öncelikle daha önce bahsetmiş olduğumuz karesel programlama probleminin yapısına bir daha bakalım:

$$\min \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{g}^T \mathbf{x},$$

$$\text{Kısıtlar:} \quad \mathbf{A}_{eq} \mathbf{x} = \mathbf{b}_{eq}$$

$$\mathbf{A}_{ie} \mathbf{x} \geq \mathbf{b}_{ie}$$

Karesel programlama yapısı destek vektör makinaları için özelleştirildiğinde \mathbf{g} vektörü ve eşitlik kısıtları kullanılmamaktadır. Bu yüzden bu dönüşümü yaparken problemi şu hale indirgeyebiliriz:

$$\min \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x}$$

$$\text{Kısıtlar:} \quad \mathbf{A}_{ie} \mathbf{x} \geq \mathbf{b}_{ie}$$

Bu bizim primal problemimizdir. Lagrange dönüşümü yapılarak ikincil probleme dönüştürülme işlemi şu şekilde yapılır:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}) &= \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} - \boldsymbol{\alpha}^T [\mathbf{A}_{ie} \mathbf{x} - \mathbf{b}_{ie}] \\ &= \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} - \sum_{i=1}^{N_{ie}} \alpha_i [\mathbf{A}_{ie}^i \mathbf{x} - b_{ie}^i]\end{aligned}$$

Kısıtlar: $\alpha_i \geq 0 \quad i = 1, \dots, N$

Burada $\boldsymbol{\alpha}$ vektörü dual değişkenleri içermektedir ve Lagrange Değişkenleri adı verilmektedir. Daha önce de bahsettiğimiz gibi bu problem konveks bir yapıdadır ve bu Lagrange ifadesi \mathbf{x} değişkenlerine göre türevi alındığında yani ekstremum noktalarında 0'a eşitlenmektedir.

$$\frac{\partial \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha})}{\partial \mathbf{x}} = \mathbf{H} \mathbf{x} - \mathbf{A}_{ie}^T \boldsymbol{\alpha} = \mathbf{0} \Rightarrow \mathbf{x} = \mathbf{H}^{-1} \mathbf{A}_{ie}^T \boldsymbol{\alpha}$$

Bu nokta ekstremum noktası olduğuna göre asıl denklemde yerine konduğunda Lagrange problemi şu hale dönüşmüş olur. Tek kısıt ise Lagrange değişkenlerinin 0 ya da sıfırdan büyük olmasıdır.

$$\mathcal{L}(\boldsymbol{\alpha}) = -\frac{1}{2} \boldsymbol{\alpha}^T \mathbf{A}_{ie} \mathbf{H}^{-1} \mathbf{A}_{ie}^T \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{b}_{ie}$$

Kısıtlar: $\alpha_i \geq 0 \quad i = 1, \dots, N$

4.6.2 Lagrange Dönüşümü SVM Uygulaması

Daha önce karesel programlama modeline dönüştürdüğümüz SVM fonksiyonlarımızı bir daha ele alalım. Ancak burada belirtmek gerekir ki daha önceki ifadede limit alımında \mathbf{H} matrisinin $\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ formundan dolayı b terimi etkisiz eleman olarak kaldığı için limit hesabında bu terimi yazmadan şu şekilde basitleştireceğiz. \mathbf{x} yerine de ilerideki karmaşıklıkları önlemek adına \mathbf{t} kullanacağız:

$$\min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w}$$

$$\text{kısıtlar: } y_i(\mathbf{w}^T \mathbf{t}_i + b) \geq +1 \quad i = 1, \dots, N$$

Lagrange dönüşümünü gerçekleştirsek:

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \alpha_i [y_i(\mathbf{w}^T \mathbf{t}_i + b) - 1] \\ &= \frac{1}{2} \mathbf{w}^T \mathbf{w} - \mathbf{w}^T \sum_{i=1}^N \alpha_i y_i \mathbf{t}_i - b \sum_{i=1}^N \alpha_i y_i + \sum_{i=1}^N \alpha_i \end{aligned}$$

Bu ifade \mathbf{w} ve b terimlerine göre türevinin $\mathbf{0}$ olması gerekir çünkü bu ifadenin maksimum noktaları buralardadır. Bu durumda:

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{0} \Rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{t}_i$$

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = 0 \Rightarrow \sum_{i=1}^N \alpha_i y_i = 0$$

Lagrange fonksiyonunu bu minimum noktalar için tekrar yazarsak:

$$\mathcal{L}(\boldsymbol{\alpha}) = -\frac{1}{2} \mathbf{w}^T \mathbf{w} + \sum_{i=1}^N \alpha_i$$

$$\mathcal{L}(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{t}_i \mathbf{t}_j + \sum_{i=1}^N \alpha_i$$

Bu problemi minimizasyon problemine dönüştürürsek:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{t}_i \mathbf{t}_j - \sum_{i=1}^N \alpha_i$$

Kısıtlar:

$$\sum_{i=1}^N \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \quad i = 1, \dots, N$$

Dual problem bu haliyle de karesel programlama problemidir. Burada dikkat edilmesi gereken nokta şudur: primal problemdeki uzay, verilerin boyutu kadardı, dual problemdeki uzayın boyutu ise veri sayısı kadardır. Bu problem çözüldüğünde primal problemdeki her bir veriye karşılık gelen negatif olmayan Lagrange çarpanı (α) elde edilir. Bu Lagrange çarpanlarının 0'dan büyük olanlarına karşılık gelen \mathbf{t} vektörleri destek vektörleridir. Destek vektörleri daha önce de bahsedildiği gibi marjinin sınırında yer aldıkları için $y_i(\mathbf{w}^T \mathbf{t}_i + b) = 1$ şartının sağlarlar.

Primal problemde yeni bir verinin hangi sınıfa ait olduğunun belirleyen fonksiyon yani sınıflandırıcı modeli şu şekildedir:

$$y(\mathbf{t}) = \mathbf{w}^T \mathbf{t} + b$$

Dual modelde ise bu ifade şu şekilde dönüşür:

$$y(\mathbf{t}) = \sum_{i=1}^N \alpha_i y_i \mathbf{t}_i^T \mathbf{t} + b$$

Buradaki b terimi destek vektörlerinin $y_i(\mathbf{w}^T \mathbf{t}_i + b) = 1$ eşitliğini sağlaması gerekliliğinden yola çıkılarak bulunurlar. Eğer fonksiyonun çıkışı negatif ise, verinin -1 sınıfında, pozitif ise de verinin $+1$ sınıfında yer aldığını göstermektedir.

Lagrange çarpanlarının sadece destek vektörlerine denk gelenlerinin 0'dan büyük, diğer veriler için ise 0 olduğunda daha önce bahsedilmişti. 0 değerinde olan Lagrange çarpanlarının yukarıdaki modele bir etkisi olmayacaktır. Bu sebeple destek vektör makinesi modeli sadece destek vektörleri kullanılarak çalıştırılabilir. Diğer verilerin modelin çalışmasına bir etkisi yoktur.

4.6.3 Dual uzay, Esnek Marjinli Sınıflandırıcı

Esnek marjlinli sınıflandırıcıdan daha önce primal uzay için bahsedilmişti. Tekrar şu şekilde yazılabilir:

$$\min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i$$

Kısıtlar:

$$y_i(\mathbf{w} \mathbf{t}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

Primal uzaydaki Bu sınıflandırıcı problemi $C \sum_{i=1}^N \xi_i$ teriminden ötürü artık karesel programlama problemi değildir. Bu problem Lagrange uzayına şu şekilde dönüştürülebilir:

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [y_i(\mathbf{w}^T \mathbf{t}_i + b) - 1 + \xi_i] - \sum_{i=1}^N \beta_i \xi_i \\ &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i - \mathbf{w}^T \sum_{i=1}^N \alpha_i y_i \mathbf{t}_i - b \sum_{i=1}^N \alpha_i y_i + \sum_{i=1}^N \alpha_i - \sum_{i=1}^N (\alpha_i + \beta_i) \xi_i \end{aligned}$$

Burada $\boldsymbol{\alpha}$ ve $\boldsymbol{\beta}$ olarak iki tane Lagrange çarpanı kullanılmıştır. Bu fonksiyon da \mathbf{w} , b ve ξ değişkenlerine göre minimize edildiğinde minimum noktasındaki bu değişkenlere göre olan türevi 0 olacaktır.

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \mathbf{w}} = \mathbf{0} \Rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{t}_i$$

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = 0 \Rightarrow \sum_{i=1}^N \alpha_i y_i = 0$$

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_i} = 0 \Rightarrow C - (\alpha_i + \beta_i) = 0 \quad i = 1, \dots, N$$

Bu eşitlikler asıl Lagrange ifadesinde yerine konulduğunda ve eşitlik kısıtı haline getirildiğinde şu sonuç elde edilir:

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{t}_i \mathbf{t}_j - \sum_{i=1}^N \alpha_i$$

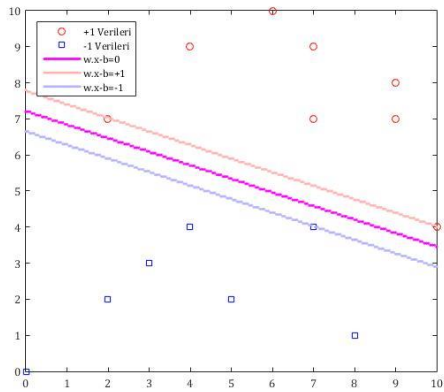
Kısıtlar:

$$\sum_{i=1}^N \alpha_i y_i = 0$$

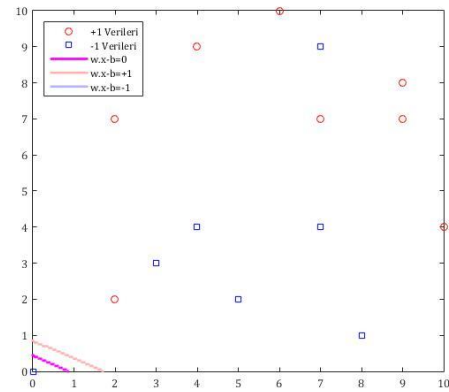
$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, N$$

Bu sonuçta iki durum gözlemlenebilir: Birincisi β Lagrange çarpanı ortadan kalkmış ve sert marjin ifadesinde olduğu gibi sadece α çarpanı kalmıştır. İkincisi ise α çarpanlarına C üst limiti getirilmesi dışında sert marjin problemi ile aynı kalmıştır. 0 olmayan α çarpanları yine destek vektörü olarak adlandırılır ve makinenin çalıştırılmasında sadece bu vektörler kullanılır.

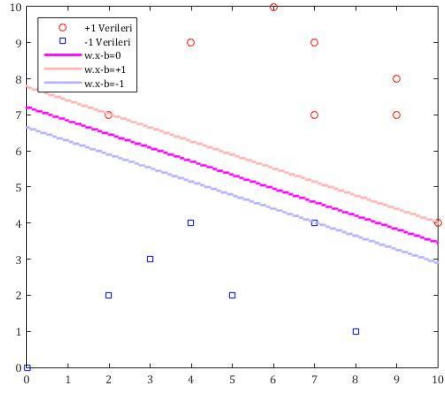
Önceki örnekten yola çıkarak oluşturulmuş, biri lineer ayrılabilir, diğeri lineer ayrılamaz olan iki ayrı veri kümesinin, farklı C değerlerindeki grafikleri aşağıda görülebilir:



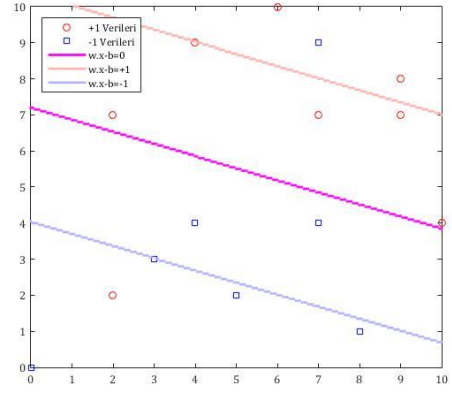
(a) $C = \infty$



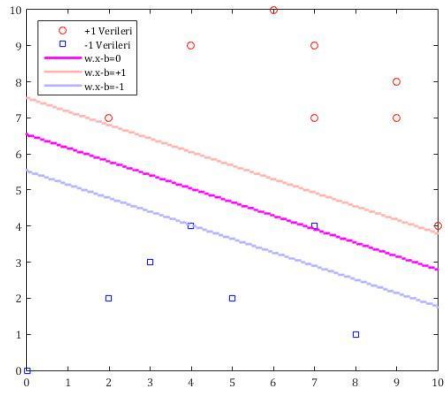
(b) $C = \infty$



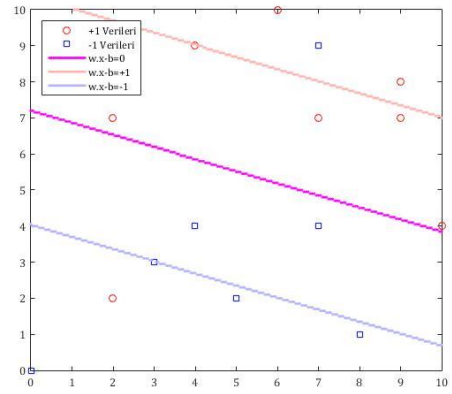
(c) $C = 10$



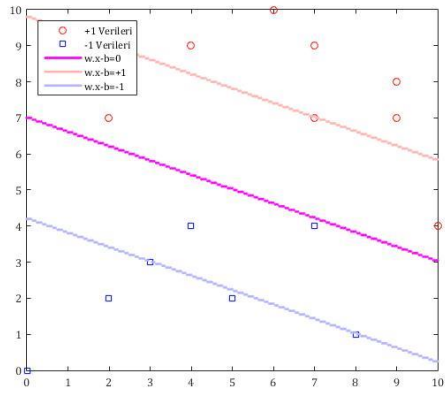
(d) $C = 10$



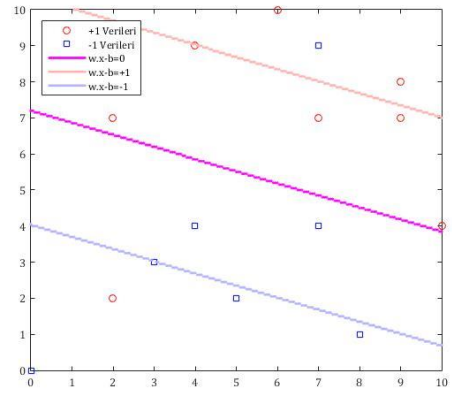
(e) $C = 1$



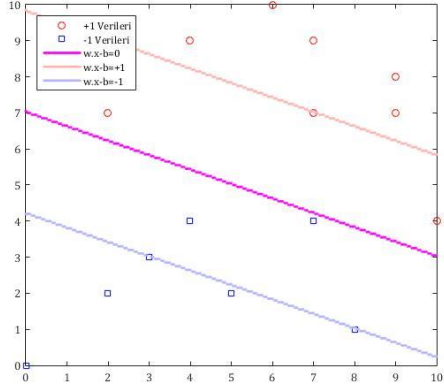
(f) $C = 1$



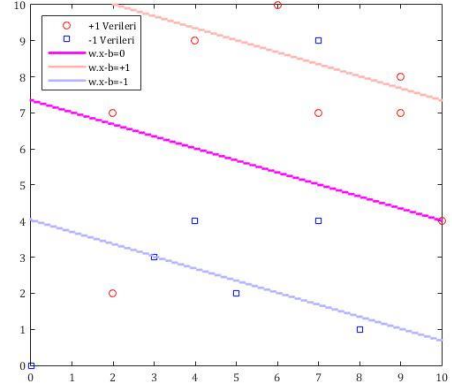
(g) $C = 0.1$



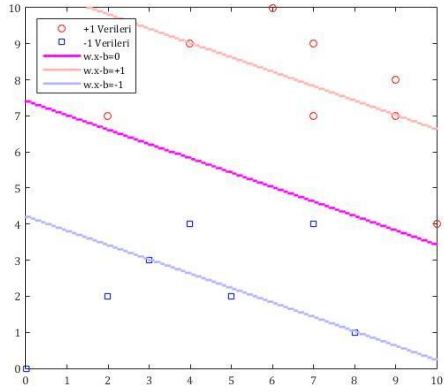
(h) $C = 0.1$



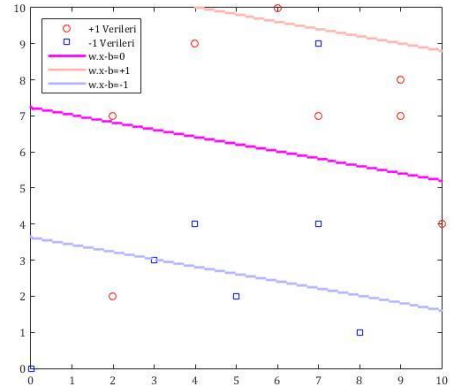
(i) $C = 0.05$



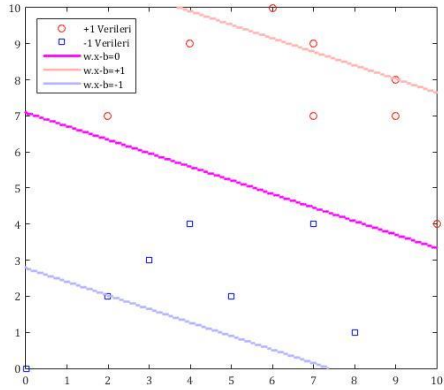
(j) $C = 0.05$



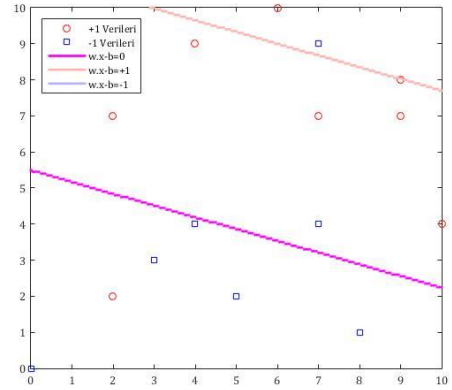
(k) $C = 0.03$



(l) $C = 0.03$



(m) $C = 0,01$



(n) $C = 0,01$

Şekil 4.4: Lineer olarak ayrılabilir ve ayrılmaz olan veri kümelerinin esnek marjlinli sınıflandırıcı kullanılarak farklı C değerlerinde sınıflandırılması, (a,b) $C = \infty$, (c,d) $C = 10$, (e,f) $C = 1$, (g,h) $C = 0.1$, (i,j) $C = 0.05$, (k,l) $C = 0.03$, (m,n) $C = 0.01$

Bu grafiklerden şu sonuçlar çıkarılabilir:

- C azaldıkça marjin artmaktadır. SVM hatalara karşı daha toleranslı davranmaktadır.

- Lineer olarak ayrılabilir verilerde C 'nin deđiřmesi karar çizgisini çok fazla etkilememekte, sadece marjini deđiřtirmektedir.
- $C = \infty$ için lineer olarak ayrılamayan veride SVM düzgün çalışmamaktadır. $C = \infty$ olduğunda SVM sert marjinli bir sınıflandırıcı olarak çalışmakta, bu da veri kümesine uygun olmadığı için algoritma doğru sonuç elde edememektedir.

4.6.4 Kernel (Çekirdek) Fonksiyonu

Birçok veri kümesi doğrusal olarak (2 boyutta çizgi, 3 boyutta düzlem, n boyutta, $n - 1$ boyutlu düzlem) ayrılamaz durumda olabilir. Bazı durumlarda ise doğrusal olarak ayrılabilir olsalar dahi doğrusal olmayan destek vektör makinesi marjin genişliğini arttırılabilmekte, daha rahat ayrılabilir bir sınıflandırıcı oluşturabilmektedir.

Primal uzayda doğrusal olmayan sınıflandırıcı elde etmek oldukça zordur. Ancak Lagrange uzayının boyutu veri sayısı ile orantılı olduğu için çok daha yüksektir. Bu uzaya geçildiğinde primal uzaydaki doğrusal olmayan verileri doğrusal ayrılabilir hale getirebilecek bir geçiş fonksiyon bulunabilirse, doğrusal ayırım yapan karesel optimizasyon ile çözüm elde edilebilir. Primal uzaydaki doğrusal ayrılamaz olan verileri doğrusal olamayan dönüşüm yaparak dual uzayda doğrusal ayrılabilir hale getiren fonksiyon "Kernel Fonksiyonu" olarak adlandırılır.

Her bir \mathbf{t}_i verisinin doğrusal olmayan bir dönüşümle $\Phi(\mathbf{t}_i)$ 'ye dönüřtürüldüğü varsayılrsa, bu durumda dual uzayda destek vektör makinesi eğitimi řu şekilde tanımlanır (Gunn 1998):

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \Phi^T(\mathbf{t}_i) \Phi(\mathbf{t}_j) - \sum_{i=1}^N \alpha_i$$

Kısıtlar:

$$\sum_{i=1}^N \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, N$$

Bu SVM sınıflandırıcısı modeli ise şu şekilde elde edilecektir:

$$y(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i \Phi^T(\mathbf{t}_i) \Phi(\mathbf{x}) + b$$

Görüldüğü gibi $\mathbf{t}_i \Rightarrow \Phi(\mathbf{t}_i)$ dönüşümü tek başına kullanılmamakta her zaman $\mathbf{t}_i^T \mathbf{t}_j \Rightarrow \Phi^T(\mathbf{t}_i) \Phi(\mathbf{t}_j)$ çarpımı şeklinde tanımlanmaktadır. Böylece $\mathbf{t}_i^T \mathbf{t}_j \Rightarrow \Phi^T(\mathbf{t}_i) \Phi(\mathbf{t}_j) = K_{ij}(\mathbf{t}_i, \mathbf{t}_j)$ şeklinde bir doğrusal olmayan dönüşüm fonksiyonu yazılabilmektedir. Bu durumda Φ fonksiyonu da bulunmak zorunda kalmadan $K_{ij}(\mathbf{t}_i, \mathbf{t}_j)$ Kernel (çekirdek) fonksiyonu kullanılarak SVM oluşturulmuş olur.

Bir fonksiyonun Kernel fonksiyonu olabilmesi için birtakım özellikler taşınması gerekir. Bu özellikleri 1909 yılında Mercer belirlemiştir (Mercer 1909). Özetle bir fonksiyonun Kernel fonksiyonu olarak kullanılabilmesi için pozitif tanımlı olması gerek ve yeter koşuldur.

Literatürde Kernel fonksiyonu olarak kullanılabilen birçok fonksiyon bulunmaktadır. Bu fonksiyonlar ve karşılaştırmaları bu tezin konusu olmadığı için hepsinden bahsedilmeyecektir. Bu tezin çalışmalarında “Gaussian Radial Basis Function” kullanılmıştır. Bu Kernel fonksiyonu şu formdadır:

$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}$$

Bu fonksiyon her bir veri noktasına ortalanmış Gauss eğrisi fonksiyonu kullanarak karar eğrisini oluşturur. Bu Gauss fonksiyonun genişliğinin σ parametresi ile ayarlanabilmesi de “Gaussian Radial Basis” kernelinin en büyük özelliğidir.

Kernel kullanmanın başka bir avantajı da bias terimi olan b ’nin kernel içerisinde yer almasının sağlanmasıyla bu terimin hesaplanmasının kolaylaştırılmasıdır. Bunun uygulanabileceği birçok Kernel vardır “Gaussian Radial Basis Kernel” da bunlardan biridir (Girosi 1997). Özetle b teriminin kernel içine alınması için “ $\sum_{i=1}^N \alpha_i y_i = 0$ ” kısıtlık eşitinin kaldırılması yeterlidir. Bu durumda SVM eğitimi şu şekilde olacaktır:

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{t}_i, \mathbf{t}_j) - \sum_{i=1}^N \alpha_i$$

Kısıtlar:

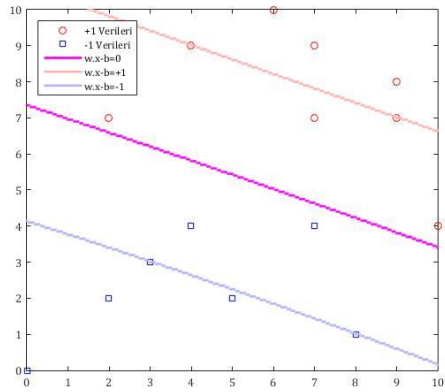
$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, N$$

α 'nın sıfır'dan büyük değerlerine karşılık gelen veriler destek vektörleridir.

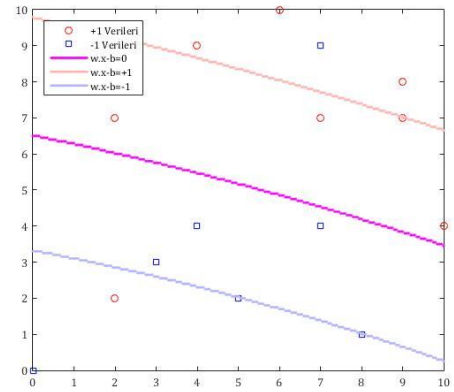
Sadece bu vektörler kullanılarak oluşturulacak model ise şu şekildedir:

$$y(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i K(\mathbf{t}_i, \mathbf{t}_j)$$

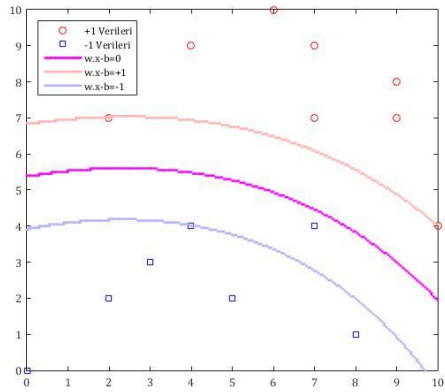
Lineer olarak ayrılabilir ve ayrılmaz olan örnek veri kümelerimizde farklı σ değerleri için karar eğrisi örnekleri şu şekildedir:



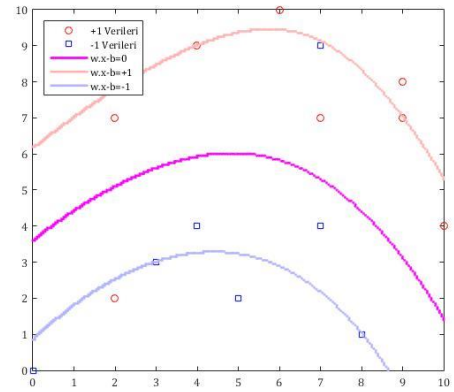
(a) $\sigma = 30$



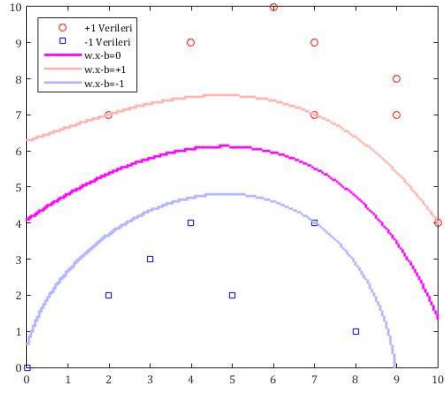
(b) $\sigma = 30$



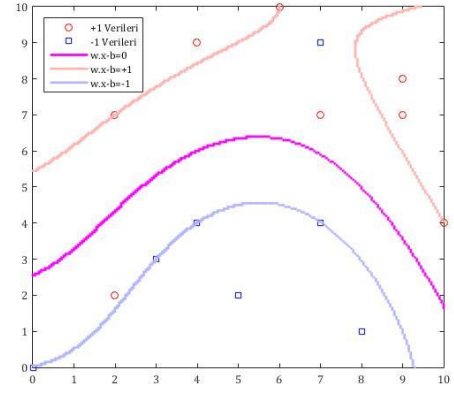
(c) $\sigma = 10$



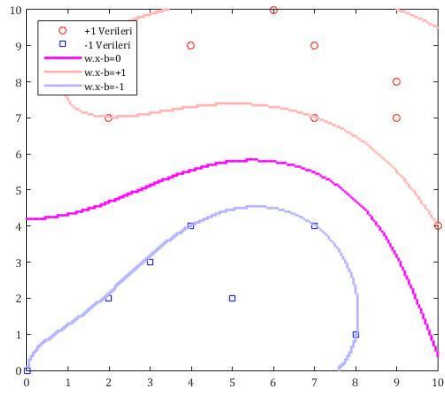
(d) $\sigma = 10$



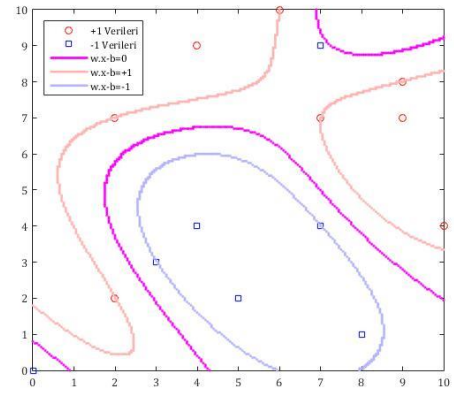
(e) $\sigma = 5$



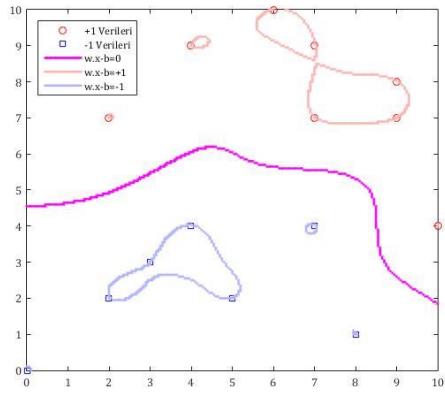
(f) $\sigma = 5$



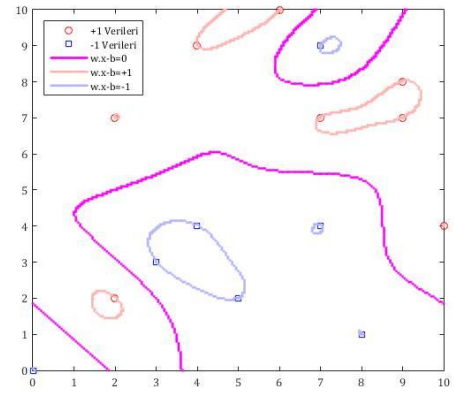
(g) $\sigma = 3$



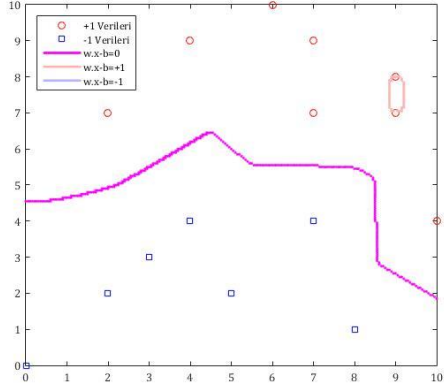
(h) $\sigma = 3$



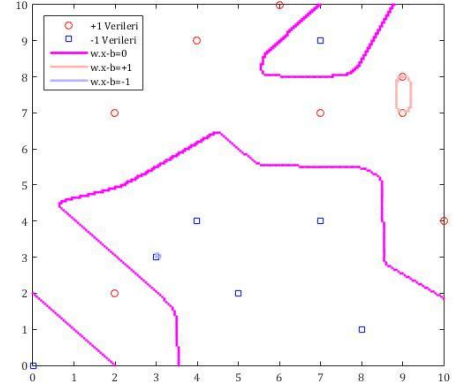
(i) $\sigma = 1$



(j) $\sigma = 1$



(k) $\sigma = 0.5$



(l) $\sigma = 0.5$

Şekil 4.5: $C = 30$ iken farklı σ değerlerinde iki farklı veri kümesinde SVM sınıflandırması (a,b) $\sigma = 30$, (c,d) $\sigma = 10$, (e,f) $\sigma = 5$, (g,h) $\sigma = 3$, (i,j) $\sigma = 1$, (k,l) $\sigma = 0.5$.

Dikkat edilirse bu grafiklerden şu sonuçlar çıkarılabilir:

- σ değeri küçüldükçe karar çizgisi verilere yakınlaşmakta verilerin etrafında dolaşmaktadır. Buna sistemin “ezberlemesi” denir ve genellikle istenmeyen bir durumdur çünkü bu durumda hataların da ezberlenmesi söz konusu olduğu için yeni gelen bir verinin sorgulanması durumunda daha önceki bir hatayı baz alması ve yanlış sonuç vermesi gözlemlenebilir.
- σ değeri arttıkça karar çizgisi daha genel davranmaktadır. Hatta çok büyük değerlerde lineer sınıflandırıcı gibi çalışmaktadır.
- Lineer olarak ayrılabilen örnek veri kümesine bakıldığında, $\sigma = 5$ ya da $\sigma = 3$ durumları lineer sınıflandırıcı ile karşılaştırıldığında, karar çizgisinin bükülerek marjini arttırdığı gözlemlenebilir.

4.7 Çok-Sınıflı SVM

Şu zamana kadar verilerin ya “-1” ya da “+1” olarak iki sınıfa ayrıldığı durumu ele alarak sonuç çıkardık. Ancak gerçek bir veri kümesinde çok daha fazla sınıf olabilir. Örneğin el yazısı ile yazılmış harfleri tanıyacak bir sistem yapmak istediğimizde sistemde harf sayısı kadar sınıflandırıcı bulunmaktadır. Bu tür sınıflandırıcılara çok sınıflı (Multiclass) sınıflandırıcı denir.

Çok sınıflı verilerin SVM ile nasıl ayrılacağına dair farklı çalışmalar olsa da en çok kabul göreni çok sınıflı verilerin 2 sınıflı kombinasyonları üretilerek 2 sınıflı SVM'ler ile çözülmesi sonra da her bir SVM'den çıkan kararların ele alınmasıyla yeni verinin hangi sınıfa ait olduğunun belirlenmesidir.

Örneğin 4 sınıflı bir veri ele alalım. Bu sınıflar şimdilik 1, 2, 3 ve 4 olarak adlandırılmış olsun. İki adet 2 sınıflı sınıflandırıcı kullanarak bu problemi çözebiliriz. Birinci sınıflandırıcımız 1 ve 2 ile işaretlenmiş verileri “-1” olarak işaretler, 3 ve 4 olarak işaretlenmiş verileri ise “+1” olarak işaretler. İkinci sınıflandırıcı ise 1 ve 3 ile işaretlenmiş verileri “-1” olarak işaretler, 2 ve 4 olarak işaretlenmiş verileri ise “+1” olarak işaretler. SVM eğitimleri buna göre yapılır. Bir tablo ile özetlemek gerekirse:

Tablo 4.1: Çok sınıflı SVM: sınıfların SVM'lere yarılmaları

	-1 değerlikli veriler	+1 değerlikli veriler
SVM ₁	1, 2	3, 4
SVM ₂	1, 3	2, 4

Yeni bir verinin hangi sınıfa ait olduğunu belirlemek için ise bu veri her iki sınıflandırıcıda da ayrı ayrı sorgulanarak gelen sonuçlara göre karar verilir. Yani bu durumda aşağıdaki gibi bir tablo kullanılabilir:

Tablo 4.2: Çok sınıflı SVM: SVM sonuçlarına göre sınıfların belirlenmesi

	SVM'lerden çıkan sonuçlar			
SVM ₁	-1	-1	+1	+1
SVM ₂	-1	+1	-1	+1
Sınıf	1	2	3	4

Tabi ki farklı sayıdaki sınıfları ayırmanın birçok farklı yöntemi vardır. Eğer verinin dağılım şekli biliniyorsa ayırım buna uygun olacak şekilde düzenlenebilir. Eğer bilinmiyorsa ikili sayı sistemi, ya da “grey code” kullanılarak veriler sistematik bir şekilde ayrılabilir. Kaç adet SVM kullanılacağı ise şu formülle hesaplanabilir:

$$\#SVM = \lceil \log_2(\#CLASS) \rceil$$

$\#CLASS$: Sınıf sayısını belirler.

$\lceil x \rceil$: Yukarı yuvarlama fonksiyonu. x 'e eşit ya da x 'den büyük olan tam sayıya yuvarlar.

$\#SVM$: Destek vektörü sayısı

Burada dikkat edilmesi gereken noktalardan biri de şudur: özellikle sınıf sayısının $2^{\#SVM}$ 'den az olduğu durumlarda bazı SVM çıktılarının bir karşılığı olmayabilir. Örneğin 5 farklı sınıflı bir veri kümesi 3 tane SVM ile ayrıştırılabilir. Ancak 3 SVM 8 farklı çıkış verebilir yani 3 tane çıkış kombinasyonunun bir karşılığı yoktur. Şöyle ki:

Tablo 4.3: Çok sınıflı SVM: 3 SVM'li sınıflandırma örneği

	SVM'lerden çıkan sonuçlar							
SVM ₁	-1	-1	-1	-1	+1	+1	+1	+1
SVM ₂	-1	-1	+1	+1	-1	-1	+1	+1
SVM ₃	-1	+1	-1	+1	-1	+1	-1	+1
Sınıf	1	2	3	4	5	?	?	?

Bu durumu çözmek için şunlar yapılabilir:

- Bazı SVM'lere öncelik vererek buna göre karar vermek. Örneğin yukarıdaki örnek için SVM₁ "+1" çıkışı verdiği diğer SVM'lere bakmadan veriyi "5" olarak sınıflandırmak.
- İki sınıf arasından sadece bir adet karar çizgisi geçecek şekilde ayırım yapmak. Bu sayede farklı SVM'lerin aynı karara etkisi olmasını engellemek. Bu çözüm her veri kümesi için uygulanabilir değildir.
- Sınıflandırıcılar ayarlanırken boş kombinasyon kalmayacak şekilde bütün durumlara bir sınıf ataması yapmak. Örneğin yukarıdaki veri kümesinde "?" olan yerlere 5 sınıftan birini atamak. Veri kümelenmesi biliniyorsa buna göre yapılması daha uygun olacaktır.
- Verinin belirli bir delta aralığında yerini değiştirerek olası en yakın sınıfa atama yapmak.

5. SVM İLE TERMAL KONFOR MODELLEMESİ

Bir iç mekanın termal olarak konfora getirilmesi termal konfor koşullarının belirlenmesi ile başlar. Bu koşullar birçok parametreye göre değişiklik göstermektedir. Bu parametrelerden bazıları şu şekilde özetlenebilir:

- Cinsiyet: Bu konuda yapılan araştırmalar ufak farklar olsa da cinsiyetin termal konfor hissine etkisi olduğunu kanıtlamıştır.
- Soğuğa duyarlılık farkları: Hamileler, Engelliler, Hastalar soğuğa daha fazla duyarlılık gösterebilirler.
- Yaş: Yapılan bir çalışma 14 yaş altı ve 60 yaş üstü kişilerin sıcaklık değişimlerine daha fazla duyarlı olduklarını göstermiştir.
- Psikolojik ve sosyolojik etkiler: İnsan psikolojisi çevresel hissiyatta önemli rol oynar. Ayrıca sosyolojik yapı ve alışkanlıklar da termal konfor hissiyatına etkilidir.
- Bölgesel etkiler: İnsanların yaşadıkları bölgelerdeki hava şartları insan vücudunun bu duruma adapte olarak çalışmasını sağlar.

Termal konfor modellemesinden daha önce bahsederken PMV - PPD modelinden söz edilmişti. Bu model özellikleri birbirleri ile aynı kabul edilen birçok denek kullanılarak anket yöntemi ile oluşturulmuş bir modeldir. Uzun yıllardır kullanılıyor olmasına karşın yukarıda belirtilen sebeplerden ötürü her koşula ve her kişiye uygulanabilir değildir. Ayrıca met ve clo değerleri ölçülebilir değerler olmadığı için bu veriler PMV modeline girilirken yanlışlık yapılması olasılığı yüksektir. Bunun için kişiye özel termal konfor modellemesine ihtiyaç vardır. Kişiye özel termal konforun sağlanabilmesi için öğrenilebilir, değişken koşullara göre hareket edebilen adaptif bir yapı gerekmektedir. Bunun için Destek Vektör Makinaları ile termal konfor modellemesi yapılacaktır.

Bu amaçla sırayla şöyle bir yöntem izlenmiştir.

1) Belirlenen aralıklarda rastgele olarak bir veri set üretilmektedir. Bu veri setinin her bir elemanının PMV değeri hesaplanarak aşağıdaki şekilde sınıflandırması yapılmıştır:

$$PMV < -0.5 \Rightarrow \text{Sınıf: 1 (konforsuz, soğuk)}$$

$$-0.5 \leq PMV < 0 \Rightarrow \text{Sınıf: 2 (konforlu, serin)}$$

$$0 \leq PMV \leq 0.5 \Rightarrow \text{Sınıf: 3 (konforlu, ılık)}$$

$$0.5 < PMV \Rightarrow \text{Sınıf: 4 (konforsuz, sıcak)}$$

2) Bu veri setine ve bu sınıflandırmaya göre “Gaussian Radial Basis Function” kullanılarak iki tane SVM eğitilecektir.

SVM1: 1. ve 2. Sınıflar ile 3. Ve 4. sınıfı birbirinden ayırır. Yani PMV'nin pozitif mi negatif mi olduğunu öğrenir.

SVM2: 2. ve 3. Sınıfları ile 1. ve 4. sınıfları birbirinden ayıracaktır. Yani PMV'nin konfor sınırları içinde olup olmadığını öğrenir.

Her iki SVM de optimum “ σ ” değerini belirlemek için bir dizi tarama yapar. Bundan ileriki bölümlerde bahsedilecektir.

3) Oluşturulmuş olan SVM modelleri yeni bir teste tabi tutulurlar. Daha önce belirlenmiş olan veri aralığında yine rastgele veri noktaları üretilir ve PMV değerlerine göre sınıfları belirlenir. Aynı veriler SVM modelinden de geçirilerek PMV ile aynı sınıfı tahmin edip etmediklerine bakılır. Bu testte en az %80 oranında başarı beklenmektedir.

Bu SVM modelleri ve tüm program kodları MATLAB programında yazılmış olup programın kodları EK.A bölümünde verilmiştir.

5.1 Veri Setinin Seçilmesi

PMV modelinde daha önce de bahsedildiği gibi kuru hava sıcaklığı, radyan sıcaklık, hava hızı, nem, metabolik hız ve kıyafet izolasyonu olmak üzere 6 tane parametre vardır. Bu 6 parametrenin PMV fonksiyonuna girilmesi ile ortamın termal konfor seviyesi hesaplanabilir.

Yapılmış olan SVM termal konfor modelinde de bu 6 parametre kullanılacaktır. Sistemin çalışma testleri ise yine bu 6 parametre kullanılarak yapılacaktır. Burada sonuçların denekler üzerinde test edilmesi imkanı olmadığı için bu tezde PMV modeli referans alınarak hareket edilecektir. Eğer sistem PMV modelini öğrenebilir hale gelebilirse bu modele uymayan kişilerin konfor parametresini de öğrenebilir çünkü her ne olursa olsun koşullar belirli bir standart sapma içinde olacaktır.

SVM öğrenmesi için öncelikle bir veri setinin oluşturulması gereklidir. Bahsedilen 6 tane parametrenin hangi aralıkta verileceği ISO 7720 ve ASHRAE55 standartları baz alınarak iç mekan koşullarının olabileceği aralıklar varsayılarak verilmiştir. Burada bahsedilmelidir ki her ne kadar SVM’yi iç mekan koşullarına göre optimize edecek olsak da dış mekanda sadece eğitim kümesinin veri aralığı değişeceğinden SVM, aynı parametrelerle ya da ufak değişikliklerle tekrar optimize edilebilecektir.

SVM için kullanılacak veri aralığı şu şekilde belirlenmiştir:

T_a - Kuru hava sıcaklığı	: 22 - 28 °C
T_r - Radyan sıcaklık	: 22 - 28 °C
V_a - Hava hızı	: 0.1 - 0.3 m/s
HR - Bağıl nem oranı	: %40 - %60
met - Metabolik hız	: 1 - 2 (58 - 116 W/m ²)
clo - Giysi yalıtımı	: 0.5 - 1 (0.08 - 0.16 m ² K/W)

Veri kümesi belirlenirken oluşturulan ikinci kısıt ise PMV kısıtıdır. PMV değerleri $-1.5 \leq PMV \leq 1.5$ aralığında olacak şekilde veri kümesi oluşturulmuştur. Bunun sebebi öncelikle iç mekanın çok soğuk ya da çok sıcak olamayacağının öngörülmesidir. Böyle bir durum yaşansa dahi bu bölgedeki veriler o karar çizimine uzak kalacağından dolayı SVM’nin çalışmasına etkisi olmayacaktır. Veri kümesinin küçültülmesi teknik olarak SVM’nin eğitime algoritmasını hızlandıracağı için bu yola başvurulmuştur. Tabii ki bunu yapmamızın temel sebebi veri kümesini oluştururken PMV hesabını referans olarak almamızdır. Gerçek bir veri kümesi hesaplanırken böyle bir optimizasyona gerek olmayacaktır.

SVM eğitimi için bu yukarıda belirtilen aralıkta ve PMV limitine göre rastgele veri seti üretilecektir.

5.2 PMV ile Hesaplanması

Veri kümesi PMV değerleri ile karşılaştırılacağı için öncelikle PMV hesaplaması yapan bir algoritma oluşturulması gereklidir. Örnek bir algoritma ISO 7730'da ve ASHRAE 55'de yer almaktadır. Bu programı "EK A.1 PMV Hesaplama Programı" bölümünde bulunabilir. Bu program 6 adet termal konfor verisini skaler olarak alır ve dışarıya PMV değerini verir.

Bu fonksiyon şu yapıdadır:

```
function PMV = PMVCalculator(TA, TR, VEL, HR, MET, CLO)
```

Fonksiyonun girişleri ve çıkışları şu şekildedir:

TA - Kuru hava sıcaklığı	: Skaler	Birimi: °C
TR - Radyan sıcaklık	: Skaler	Birimi: °C
VEL - Hava hızı	: Skaler	Birimi: m/s
HR - Bağıl nem oranı	: Skaler	Birimi: %
MET - Metabolik hız	: Skaler	Birimsiz
CLO - Giysi yalıtımı	: Skaler	Birimsiz
PMV - Tahmini ortalama oy	: Skaler	Birimsiz

"EK A.2 Rastgele çok boyutlu PMV verisi Üreteci" bölümünde ise PMV hesaplama programında rastgele veri üreten program yer almaktadır. Bu program ise şu yapıdadır:

```
function [INPUTs, LABELs, PMVs]  
= RandomMultiDimPMVCalculator(DTA, DTR, DVEL, DHR, Dmet, Dclo, n)
```

"PMVCalculator" fonksiyonu bu fonksiyonun içinde çağırılmaktadır. Farklı olan parametreleri ise şu şekildedir:

n - Rastgele üretilecek veri sayısı	: Skaler	Birimsiz
INPUTs - her bir verinin koordinatlarını verir.	: 6xn	Birimsiz
LABELs - her bir verinin sınıfını verir.		
1, 2, 3 ya da 4 değerlerinden birini alır.	: 1xn	Birimsiz
PMVs- her bir verinin PMV değerini verir.	: 1xn	Birimsiz

Bu fonksiyonlar sayesinde oluşturulan veri kümesi ile SVM'nin eğitimi yapılabilir.

5.3 SVM Modeli Oluşturulması

SVM modeli oluşturan MATLAB kodu "EK A.4 Ana Program" bölümünde bulunabilir.

Ana program öncelikle daha önce bahsedilen "RandomMultiDimPMVCalculator" fonksiyonunu kullanarak veri aralıkları belirlenmiş rastgele PMV verisini oluşturur ve bu verinin sınıflandırmasını yapar.

"MIMOSMC" fonksiyonu Bu veriyi kullanarak çok boyutlu ve çok sınıflı SVM'leri oluşturur. Bu program SVM için ana program niteliğindedir ve detaylarından daha sonra bahsedilecektir. Çıkışında destek vektörlerini, bu destek vektörlerine karşılık gelen Lagrange çarpanlarını, optimal sigma değerini, eğitim verilerini ve bu verilere karşılık gelen sınıfları verir. Bu sayede eğitilmiş olan çok boyutlu ve çok sınıflı SVM modelleri çalıştırılabilir.

"DrawPMVvsSVMGraph" programı sabit hava hızı, nem, met ve clo değerlerinde değişken Ta ve Tr verisinde rastgele PMV veri sınıfı üretir ve bu veriyi aynı düzlemdeki SVM ile grafik üzerinde karşılaştırır.

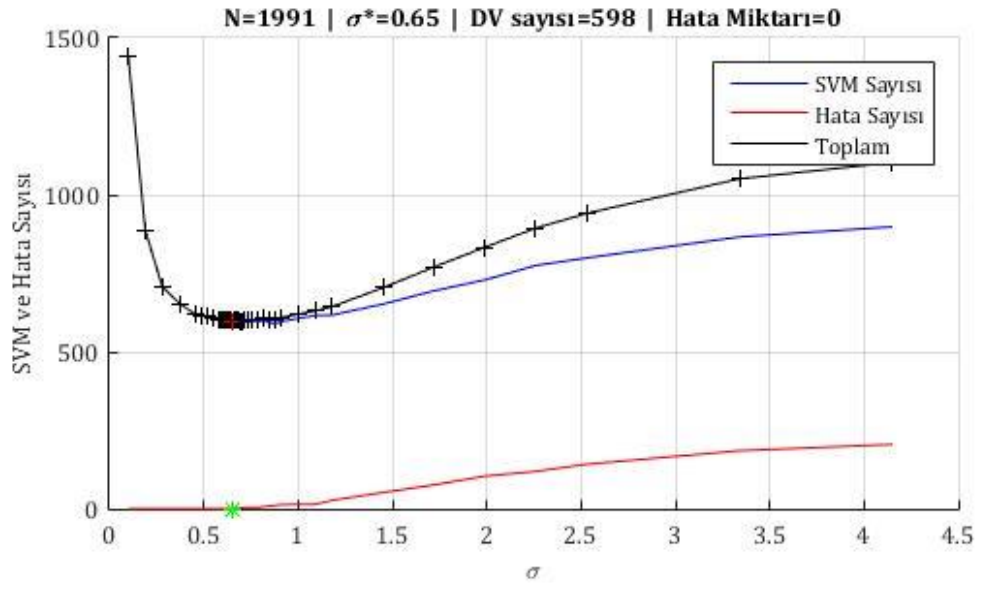
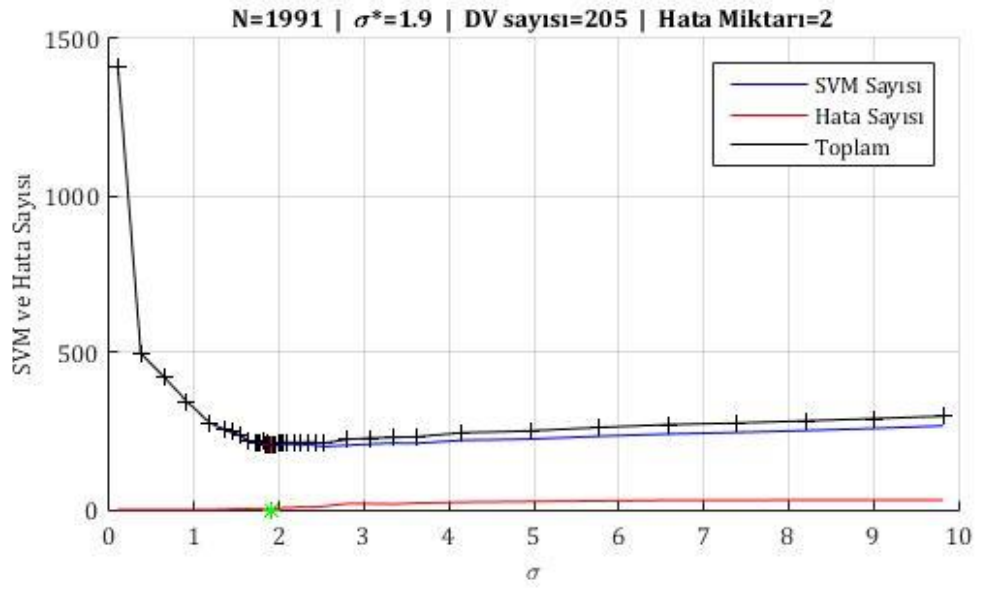
"MIMOSMC" Fonksiyonu sırasıyla şu işlemleri yapmaktadır:

1) Verilerin ve bunların sınıflarının 2 boyutlu grafikte gösterilmesi: SVM veri kümesi 6 boyutlu olsa da görsel bir ifade elde edebilmek açısından bu grafik oluşturulmuştur. Bu sayede SVM çalışması üzerine daha fazla bilgi edinilebilir hale gelir. 6 boyutlu bir uzayı 2 boyutlu bir düzlemde göstermenin sıkıntısı verilerin diğer

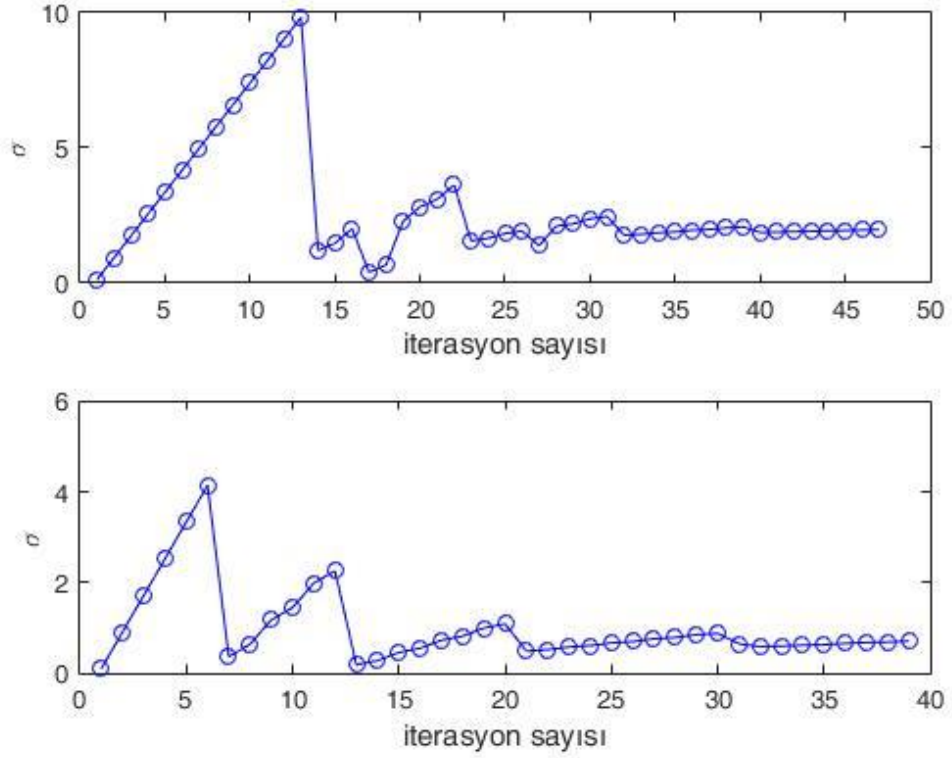
boyutlardaki yerlerinin tam olarak belirlenemiyor olmasıdır. Örneği 3 boyutlu bir veri kümesini ele aldığımızda bu uzaya sadece 2 boyut üzerinden ve perspektif olmadan bakarsak derinlik algısını elde edemeyeceğimiz için tüm veriler 2 boyutlu bir düzlem üzerindeymiş gibi görünür. Ancak yine de bu çizim karar çizgisi üzerine bir fikir verir.

2) 4 sınıflı bir veri kümesi kullanacağımız için 2 tane SVM'ye ihtiyacımız vardır. Daha önce de belirtildiği gibi SVM₁'de 1 ve 2 sınıfları "+1", 3 ve 4 sınıfları ise "-1" olarak belirlenir. SVM₂'de ise 2 ve 3 sınıfları "+1", 1 ve 4 sınıfları ise "-1" olarak belirlenir. "MIMOSMC" fonksiyonundaki bu adım gelen verideki sınıfları o an hangi SVM programlanıyorsa ona göre "-1" ya da "+1" yapmaktır.

3) Daha önce de bahsedildiği gibi optimum " σ " değerinin belirlenmesi için bir tarama algoritması yapılmaktadır. Bu tarama 0.1 ile belirli bir maksimum değer arasında farklı " σ " değerlerinde ilgili veri kümesini eğitir ve yine eğitilmiş SVM'nin sonuçlarına bakarak kaç tane destek vektör makinesi kullanıldığını ve kaç tane veriyi yanlış tahmin ettiğini belirler. Bu hata sayısı ile destek vektörü sayısını toplar ve TOTAL değeri bulur. Buradaki amaç hangi " σ " değerinin TOTAL değeri minimum yaptığını bulmaktır. Normalde 0.1'den başlayarak 0.01 aralıklarla tarama yapılarak sonuç bulunması en güvenilir yoldur ancak çok fazla zaman alır. Bunun için farklı bir algoritma da burada kullanılmıştır. Öncelikler 0.81 aralıklarla tarama yapılır. TOTAL'in minimum olduğu değer civarında bu sefer 0.27 aralıkla tarama yapılır ve yine TOTAL'in minimum olduğu veri bulunur. Bu şekilde 0.09, 0.03 ve 0.01 aralıkla tarama yapıldığında optimum " σ " değeri bulunmuş olur. SVM bu " σ " ile eğitilmiş olarak model parametrelerini oluşturulur. Bu tarama işlemi her iki SVM için de ayrı ayrı yapılmaktadır. Optimum " σ " tarama algoritmasının grafikleri aşağıda verilmiştir.



Şekil 5.1: Optimum " σ " değeri tarama algoritmasının çalışması



Şekil 5.2: İterasyonlara göre σ değerinin taranması

4) Yukarıdaki döngü içinde belirli bir “ σ ” değerinin önce SVM eğitimi yapılmalıdır. Bunun için “SVCtrain” fonksiyonu kullanılmaktadır. Bu fonksiyon eğitim veri kümesini, eğitim verisinin sınıflarını, “ σ ” ve C değerini kullanarak. Önce **H** matrisini oluşturur. Ardından gerekli “ α ” değerleri için olan alt ve üst limitle beraber MATLAB kütüphanelerinde bulunan quadratic programın optimizasyon algoritması “quadprog”u çağırır. Bu sayede “ α ” değerleri bulunmuş olur. Bu “ α ” değerlerinin 10^{-4} ’ten daha büyük olanları destek vektörü olarak kabul edilir.

5) SVM eğitiminin hemen ardından aynı veriler SVM modeline girilerek çıkışlar alınır. Bu fonksiyonun adı “SVCoutput”dur. Bu fonksiyonun çıkışında bulunmuş olan sınıfla gerçek sınıflar karşılaştırılır. Böylece destek vektörü ve hata sayısı belirlenerek TOTAL değeri bulunmuş olur. Amaç TOTAL değerini minimum yapacak “ σ ” değerini bulmaktır.

6) Her iki SVM için de optimal “ σ ” değeri bulunduktan sonra yine her iki SVM için de destek vektörleri, “ σ ” değerleri, eğitim veri kümesi, eğitim veri

kümesinin sınıfları ve optimal “ σ ” değerleri kaydedilerek SVM sınıflandırıcısında kullanılmak üzere fonksiyon dışına aktarılır.

5.4 Karşılaştırma

Yukarıdaki amaçlar doğrultusunda ve anlatılan yöntemlerle çeşitli denemeler yapılarak elde edilen bir sistemi şu şekilde tarifleyebiliriz.

$T_a=22$ °C’den başlayarak 1.5 °C aralıklarla 28 °C’ye kadar 5 veri

$T_r=22$ °C’den başlayarak 1.5 °C aralıklarla 28 °C’ye kadar 5 veri

$v_a=0.1$ m/s’den başlayarak 0.1 m/s aralıklarla 0.3 m/s’ye kadar 3 veri

HR=%40’dan başlayarak %10 aralıklarla %60’a kadar 3 veri

met=1’den başlayarak 0.2 aralıklarla 1.4’e kadar 3 veri

clo=0.5’den başlayarak 0.25 aralıklarla 1’e kadar 3 veri.

Toplam 2025 veri hesaplanmış ancak $-1.5 < PMV < 1.5$ limitinden dolayı 1991’e düşmüştür. Bu veri kümesinin tamamı eğitim için kullanılmıştır.

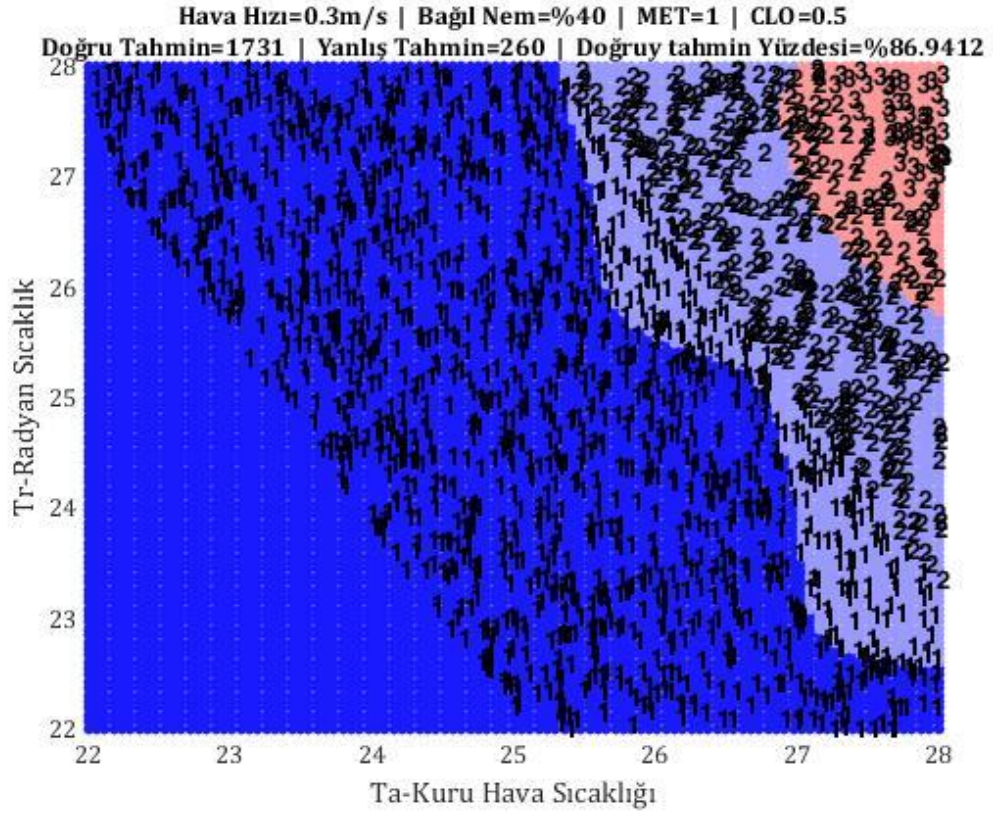
C değeri 100 olarak alınmıştır.

“ σ ” tarama sonuçlarına göre. 1. SVM için optimum “ σ ” 1.9 olarak bulunmuştur. Bu değerde 1991 olan toplan verinin sadece 205 tanesi destek vektörü olarak kullanılmış ve eğitim verisinin test edilmesi sonucu 2 tane verinin yanlış sınıflandırıldığı görülmüştür. 2. SVM için optimum “ σ ” 0.65 olarak bulunmuştur. Bu değerde 1991 verinin 598 tanesi destek vektörü olarak tanımlanmıştır. Eğitim verisinin tekrar test edilmesi sonucu hiçbir verinin yanlış sınıflandırılmadığı görülmüştür. Bu tarama grafikleri önceki bölümde gösterilmiştir.

Bu “ σ ” ile eğitilen SVM’ler başka veri kümeleriyle 5 farklı ortam testine tabi tutulmuştur. Bundan sonraki grafikler test grafikleridir. Test grafiklerinde arka plan 4 farklı sınıfı temsilen 4 bölgeye ayrılmıştır. Bu bölgeler eğitilmiş olan SVM tarafından

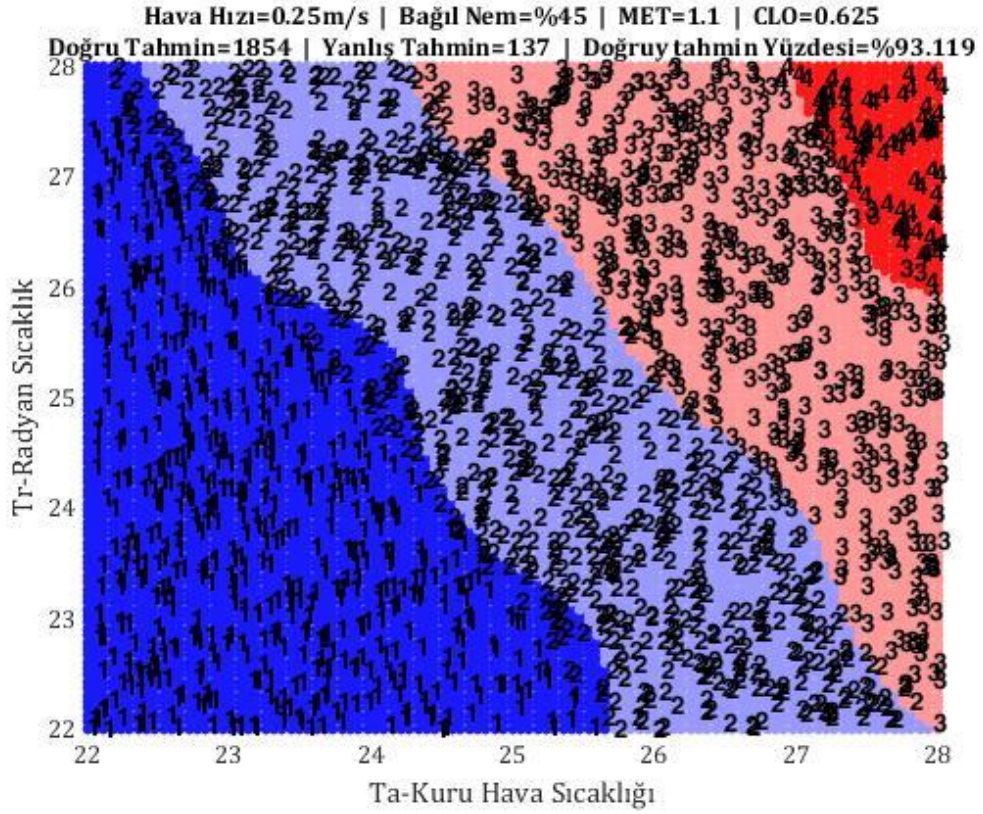
oluşturulan alanlardır. Konforsuz soğuk bölge koyu mavi renktedir ve $PVM < -0.5$ değerlerini yani sınıf 1'i gösterir. Konforlu serin bölge açık mavi renktedir ve $-0.5 < PVM < 0.5$ aralığındaki değerleri yani sınıf 2'yi gösterir. Konforlu ılık bölge açık kırmızı renktedir ve $0 < PVM < 0.5$ aralığındaki değerleri yani sınıf 3'ü gösterir. Konforsuz sıcak bölge koyu kırmızı renktedir ve $0.5 < PVM$ değerlerini yani sınıf 4'ü gösterir. Her bir test grafiği için hava hızı, nem, met ve clo değerleri sabit olacak şekilde kuru hava sıcaklığı ve ortalama radyan sıcaklık için rasgele değerler oluşturulmuştur. Bu vektörlerin PMV hesabına göre bulunmuş olması gereken sınıf değerleri vektörün gösterdiği nokta üzerine siyah punto ile yazılmıştır. Bu vektörler ayrıca SVM algoritmasından geçirilmiş ve oradan da alınan sınıf kararı ile PMV'den alınan sınıf kararı birbiri ile karşılaştırılmıştır. Bu karşılaştırmanın sonucuna göre SVM'nin belirli hava hızı, nem, met ve clo koşullarında doğru tahmin yüzdesi bulunmuştur.

1. test soğuk ortam testidir. Değerleri şu şekildedir, $v_a = 0.3$, HR = %40, met = 1, clo = 0.5. Test sonucunda %86.94 oranında doğru karar verilmiştir. Grafiği şu şekildedir:



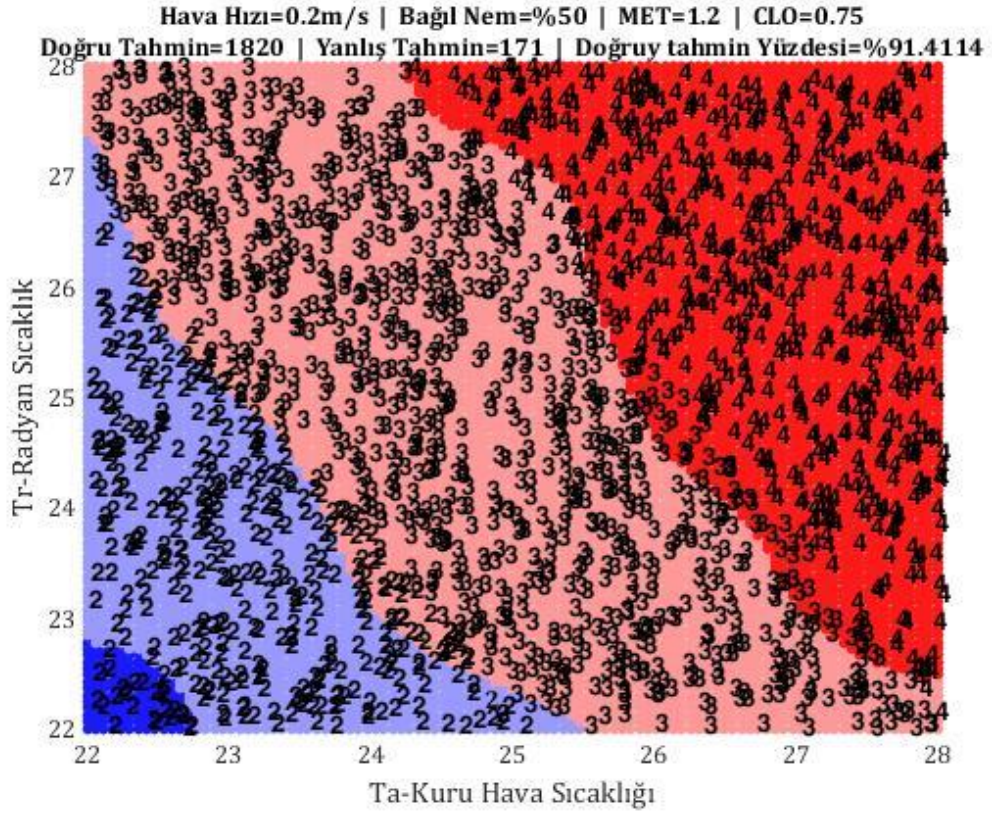
Şekil 5.3: $v_a = 0.3$, HR = %40, met = 1, clo = 0.5. için test grafiği

2. test serin ortam testidir. Değerleri şu şekildedir, $v_a = 0.25$, $HR = \%45$, $met = 1.1$, $clo = 0.625$. Test sonucunda $\%93.12$ oranında doğru karar verilmiştir. Grafiği şu şekildedir:



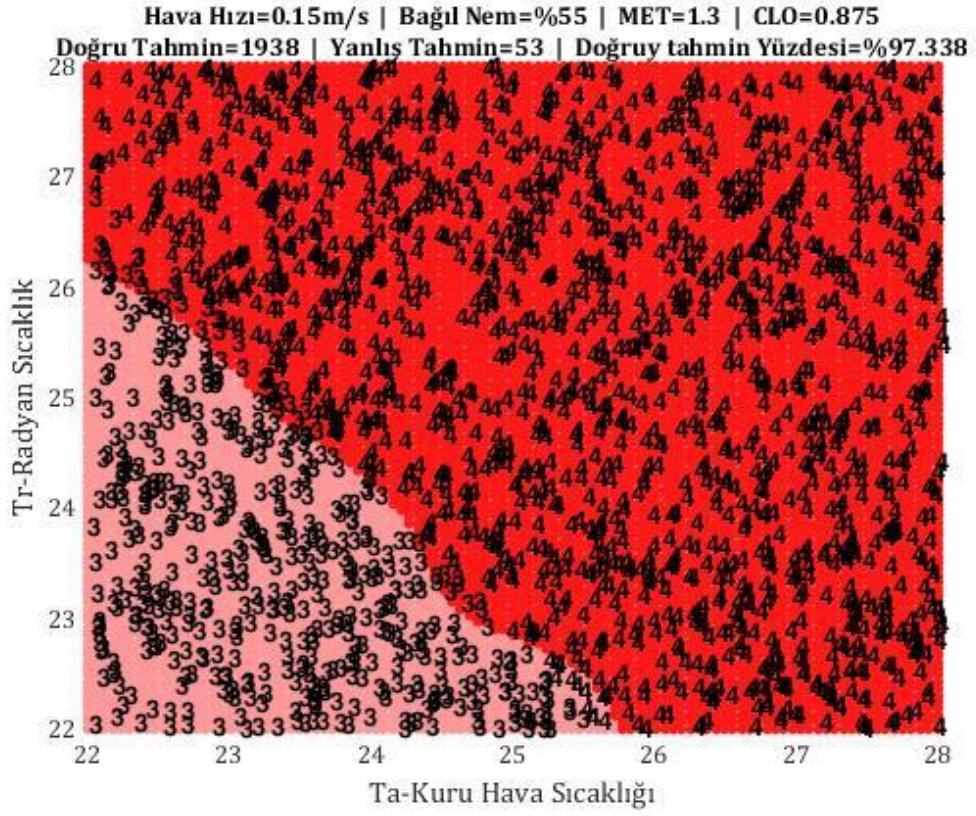
Şekil 5.4: $v_a = 0.25$, $HR = \%45$, $met = 1.1$, $clo = 0.625$ için test grafiği

3. test ortalama ortam testidir. Değerleri şu şekildedir, $v_a = 0.2$, HR = %50, met = 1.2, clo = 0.75. Test sonucunda %91.41 oranında doğru karar verilmiştir. Grafiği şu şekildedir:



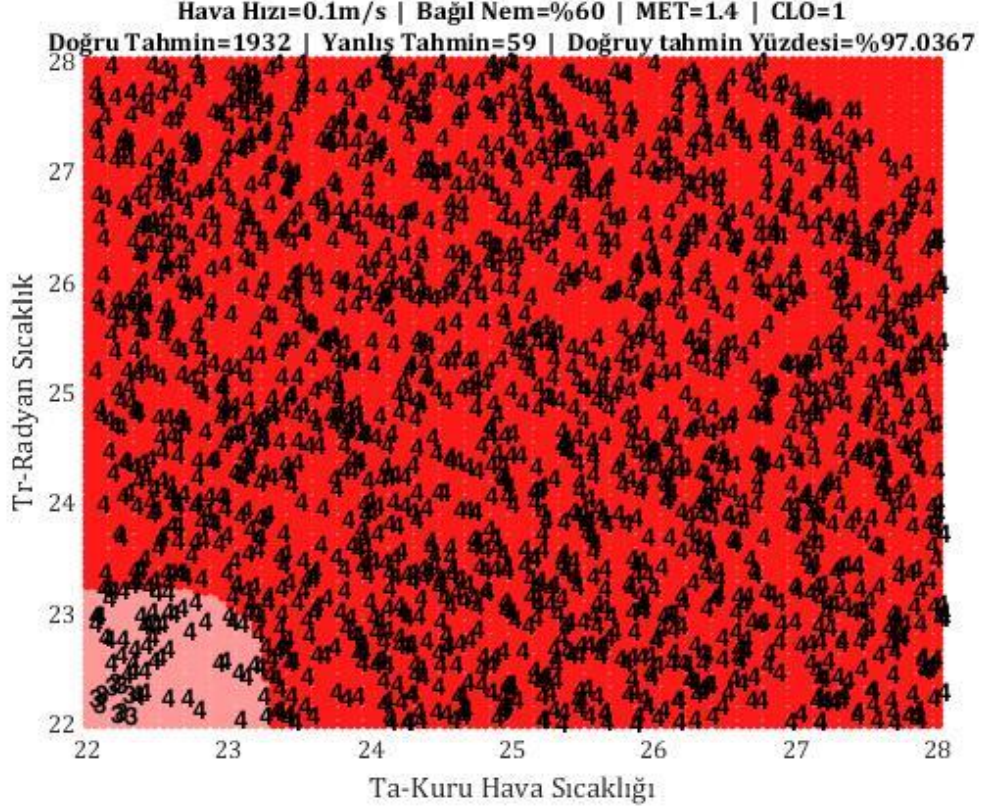
Şekil 5.5: $v_a = 0.2$, HR = %50, met = 1.2, clo = 0.75 için test grafiği

4. test ılık ortam testidir. Değerleri şu şekildedir, $v_a = 0.15$, HR = %55, met = 1.3, clo = 0.875. Test sonucunda %97.34 oranında doğru karar verilmiştir. Grafiği şu şekildedir:



Şekil 5.6: $v_a = 0.15$, HR = %55, met = 1.3, clo = 0.875 için test grafiği

5. test sıcak ortam testidir. Değerleri şu şekildedir, $v_a = 0.1$, HR = %60, met = 1.4, clo = 1. Test sonucunda %97.04 oranında doğru karar verilmiştir. Grafiği şu şekildedir:



Şekil 5.7: $v_a = 0.1$, HR = %60, met = 1.4, clo = 1 için test grafiği

Test sonuçlarının en kötüsünde %86.94 oranında doğru sınıflandırma yapılmıştır. Ortalama doğru tahmin oranı %93.24'tür. Bu sonuçlar sistemin oldukça iyi eğitildiğini düzgün karar verdiğini göstermektedir.

5.5 Gerçek Ortam Karşılaştırması

Daha önce bahsedilmiş olan RP-884 projesi dünyanın birçok yerinde birçok veri toplamış ve bu verileri halka açık şekilde paylaşmıştır. Bu veri kümesi kullanılarak oluşturulacak bir model ile sınıflandırıcının gerçek ortamda nasıl davranacağı konusunda bir çalışma niteliğinde olacağı açıktır.

RP-884 veri kümesi çok detaylı ve güzel analizler sunmaktadır. Toplanan her bir veri için çok farklı parametreler de toplanmış ve hesaplanmıştır. Örneğin çalışmadaki deneklerden termal ortamı -3 ile $+3$ arasında değerlendirmeleri istenmiştir. İkinci bir soru olarak da ortamın termal olarak kabul edilebilir olup olmadığı sorusudur. Bu soruda normal şartlar altında termal ortam değerlendirmesindeki “ -1.5 ” ile “ 1.5 ” arasındaki oylamalar için konforlu yanıtı beklenmektedir. Bunun ardından gelen soru ise “Nasıl olsun isterdiniz?” sorusudur. Cevapları ise “daha soğuk”, “daha sıcak”, “değişim istemiyorum” şeklinde verilebilmektedir. Normal şartlar altında bu 3 sorunun birbiri ile bağlantılı olarak cevaplanması beklenirken bazı durumlarda bunun tersi ile karşılaşılmaktadır. Bu ise ayrı bir çalışmanın konusu olabilir.

Bu tezde bahsi geçen çalışmadan 7 tane parametre kullanılmıştır. Bunlardan oylama yöntemi ile alınan 3 parametre şu şekildedir:

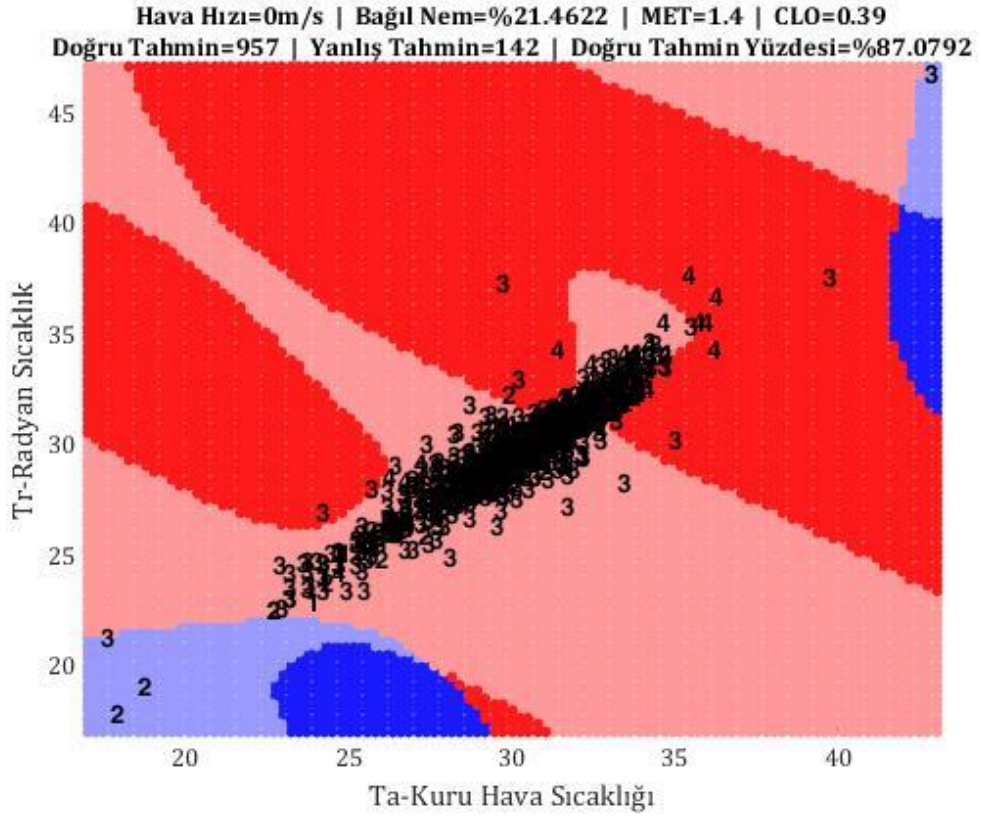
ASH: “ASHRAE Termal his skalası [$-3, +3$]” bireylerin kendi oyları ile oluşturulmuş konfor verisidir.

met: “Oylama yoluyla alınan metabolik hız verisi”

INSUL: Kıyafet izolasyonu ile sandalye, koltuk, yatak vb. eşyalardan kaynaklanan izolasyonun toplamı.

Diğer 4 parametre ortalama kuru hava sıcaklığı, ortalama radyan sıcaklık, hava hızı ve nem ölçülebilir verilerdir ve yukarıda bahsi geçen 3 parametre ile beraber Excel dosyasından okunarak sistem eğitimi için gerekli matrislere dönüştürülmüştür.

Bu veri kümesi birçok noktadan veri alınarak elde edilmiş oldukça geniş kapsamlı bir çalışmadır. Şekil 4.1’den de görüleceği gibi ülkemize en yakın nokta Atina’da toplanmış olan verilerdir (Baker 1995). Ancak Atina bölgesindeki çalışmalar incelendiğinde verilerin sadece yaz aylarında ve doğal havalandırılmalı ortamlardan (soğutma ya da iklimlendirme sistemi kullanılmayan) alındığı gözlemlenmiştir. Bu sebeple toplanan verilerin çok çeşitli olmadığı gözlemlenmiştir. Toplam 1099 verini %70’i eğitim için kullanılmıştır. Bu koşullar altında test edilmiş olan SVM modelimizin çıkışı şu şekildedir:



Şekil 5.8: RP-884 Atına çalışmalarının modellenmesi,

Aynı zamanda oluşturulmuş olan doğruluk tablosu da şu şekildedir:

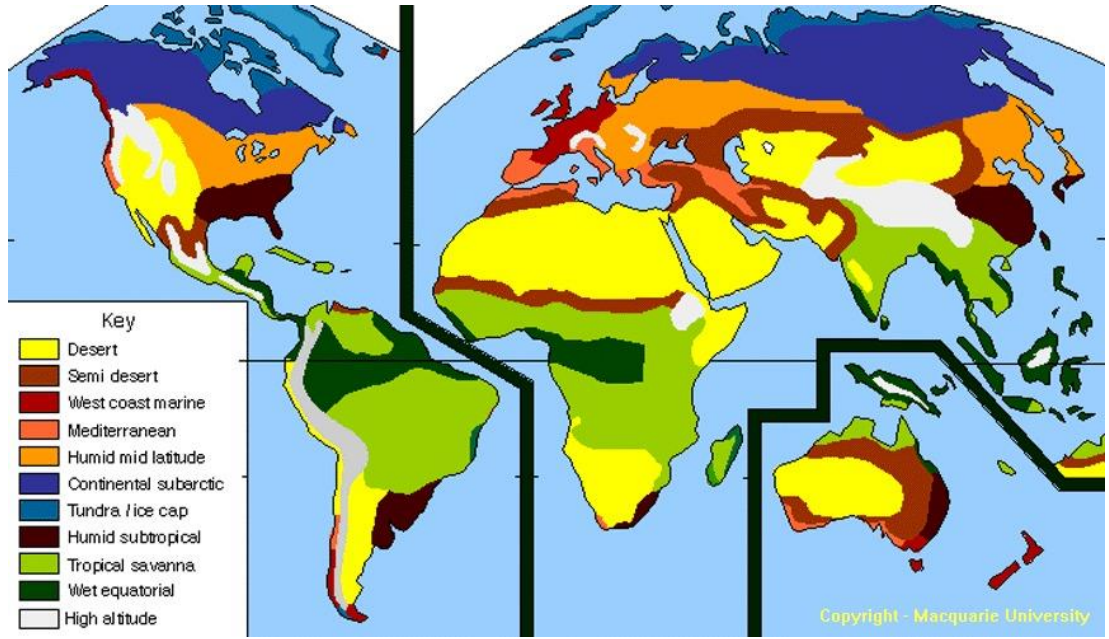
Tablo 5.1: Veri sınıflarına göre SVM tahminleri

	Veri sınıfı: 1	Veri sınıfı: 2	Veri sınıfı: 3	Veri sınıfı: 4
SVM Tahmini:1	0	0	0	0
SVM Tahmini:2	0	6	0	0
SVM Tahmini:3	7	46	842	36
SVM Tahmini:4	14	2	37	109

Yukarıdaki tabloya göre veri sınıfı ile SVM tahminin aynı olduğu sayılar doğru tahmin edilen sayılardır. Sistemin doğru tahmin etme oranı %87dir. Bu tabloda da veri sınıflarının 3 ve 4 te ağırlıklı olarak toplandığı görülebilir. Bunun sebebi bu verilerin yazın da doğal havalandırılmalı bir ortam ortamdaki ortamdan toplanmış olmasıdır.

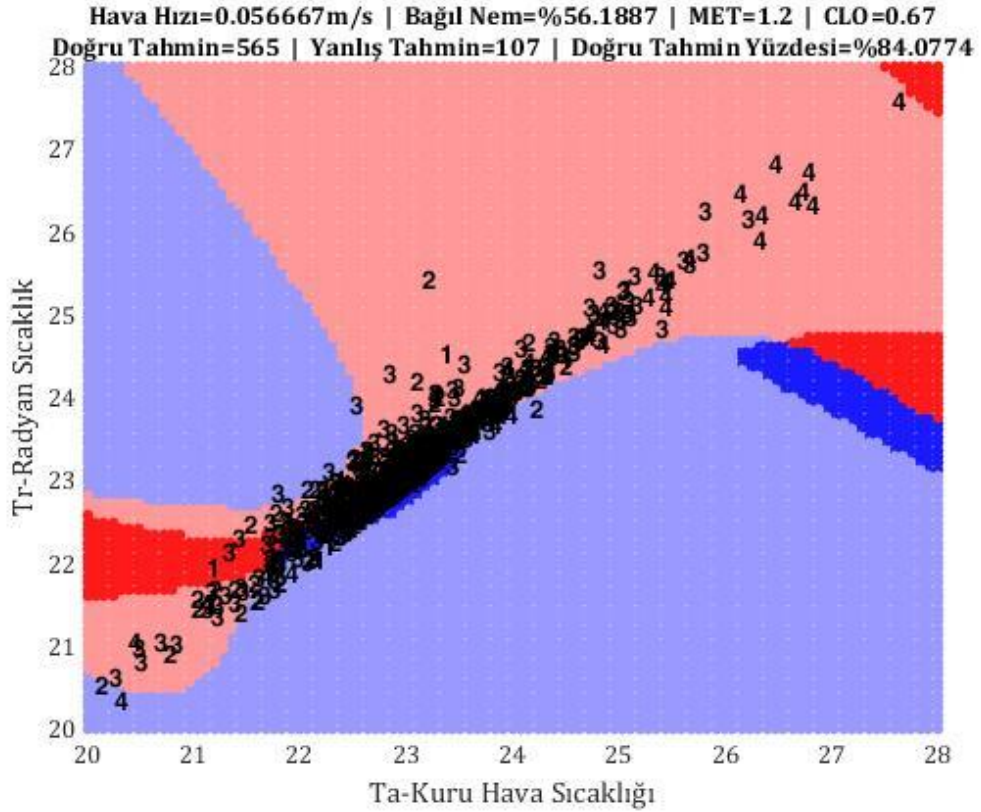
Yine Akdeniz benzeri iklimine sahip ve daha çeşitli veri kümesi içeren bir çalışmadan veri almak için araştırma yapıldığında RP-884'ün burada da kolaylık

oluşturduğu benzer iklimdeki bölgeleri işaretlediği anlaşıldı. Buna göre kendilerinin oluşturdukları harita şu şekildedir:



Şekil 5.9: RP-884 projesini iklim bölgeleri.

Buna göre bir sonraki çalışma yine “Mediterranean” olan ama daha geniş aralıkta verilerin bulunduğu “San Francisco Bay Area” bölgesinden yazın alınmış HVAC soğutmalı bir binanın verileri kullanılmıştır (Schiller 1988).



Şekil 5.10: San Francisco Bay Area bölgesi HVAC sistemli yaz verileri

471 veri ile eğitilen ve 672 veri ile test edilen modelin %84'ün üzerinde doğru tahmin yaptığı gözlemlenmiştir. Doğruluk tablosu şu şekilde verilmiştir:

Tablo 5.2: Veri sınıflarına göre SVM tahminleri

	Veri sınıfı: 1	Veri sınıfı: 2	Veri sınıfı: 3	Veri sınıfı: 4
SVM Tahmini:1	21	4	7	1
SVM Tahmini:2	3	96	27	3
SVM Tahmini:3	3	24	389	10
SVM Tahmini:4	2	4	19	59

Bu verilerle yapılmış olan diğer testlerin de sonuçları birbirine yakın çıkmakta %80 doğruluğun altına düşmemektedir. Bu verilerdeki özellik kuru hava ve radyan sıcaklık değerlerinin birbiriyle neredeyse lineer olarak çalışmasıdır. Tabii ki de bu durum sınıflandırma açısından zorluklar çıkarsa da yine de yüksek oranlarda tahminler elde edilebilmektedir. Bu durum Farhan (2015) makalesindeki “Radyan

sıcaklığın bir etkisi olmadığı görülmüştür” cümlesinin de neye dayanarak kurulmuş olabileceğini gösteriyor.

Bunun dışında Farhan (2015), çalışmasında olduğu gibi termal konforlu olan bölgeyi ayırmadan yani bu tezdeki 2 ve 3 bölgelerini birleştireşsek yukarıdaki tablo şu hali alır:

Tablo 5.3: Veri sınıflarına göre SVM tahminleri

	Veri sınıfı: 1	Veri sınıfı: 2 ve 3	Veri sınıfı: 4
SVM Tahmini:1	21	11	1
SVM Tahmini:2 ve 3	6	536	13
SVM Tahmini:4	2	23	59

Sınıflandırmanın doğruluk oranı bu haliyle %91.67'ye çıkmaktadır.

6. SONUÇ VE ÖNERİLER

Bu tezde termal konforun SVM ile sağlanabileceğine dair çalışma yapılmış ve yüksek oranda doğruluklar elde edilmiştir. İlk çalışmalarda PMV kullanılarak iç mekan koşullarını simule edecek belirli bir aralıkta limitleri olan düzenli veri üretilmiş ve bu veri ile SVM eğitilmiştir. Bunun sonucunda minimum %86,94, ortalama olarak da %93.24 doğru tahmin yapılmıştır. Bunun ardından gerçek ortam testlerine geçilmiş ve ASHRAE RP-884 verileri kullanılarak yapılan testler sonucunda %80'nin üzerinde sonuçlar elde edilmiştir. Ayrıca termal konforu “soğuk”, “konforlu” ve “sıcak” olarak 3 kategoriye ayırdığımızda doğru tahmin oranı %90'ın üstüne çıkmaktadır.

Uygun parametrelerin bulunması halinde SVM'lerin çok sınıflı sınıflandırma konusunda yüksek başarımlar elde ettiği gözlemlenmiştir. Uygun parametrelerin bulunabilmesi için bu makalede olduğu gibi grid search uygun olabileceği gibi farklı optimizasyon teknikleri de kullanılabilir.

Elde edilen bu sonuçlar termal konfor parametrelerinin çok sınıflı SVM sınıflandırıcıları kullanılarak yüksek doğrulukta bulunabileceğini göstermiştir.

Bu tez ve konusu çalışmakta olduğum MİR ARAŞTIRMA ve GELİŞTİRME A.Ş. şirketinde hala yürümekte olan, ulusal ve uluslararası koordinatörlüğünü şahsımın yaptığı, TÜBİTAK 1509 destekli “ÖĞRENEN ALGORİTMA İLE DESTEKLENECEK OPERATİF SICAKLIK ESASINA DAYANAN AKILLI TERMOSTAT GELİŞTİRİLMESİ” başlıklı, İspanya ortaklı EUREKA projesinin önemli bir ayağını oluşturmaktadır. Bu proje kişiye özel termal konfor koşullarının belirlenmesi ve iç mekan sıcaklık kontrol cihazlarının buna göre yönetilmesini esas almaktadır.

Bundan sonra, başta sistemin daha az kaynak kullanarak daha hızlı çalışmasını sağlayacak ve eğitim kümesi değişimlerine hızlı tepki verebilecek online-SVM gibi farklı SVM türleri denenmesi planlanmaktadır. İkincil olarak sisteme sadece 6 verinin değil, yaş, cinsiyet gibi kişiye özel ve hava durumu, iklim koşulları,

binanın termal davranışı vb. dış etkenlerin de eğitim kümesine eklenmesi düşünülmektedir. Ayrıca kişini alışkanlıkları, eve giriş çıkış saatleri gibi parametreler de öğrenen algoritmaya dahil edilerek sadece termal konfor değil, sistem kullanım kolaylığı da sağlanması düşünülmektedir.

7. TEŞEKKÜR

Bu tez çalışması esnasında bana her türlü desteğini veren, bilgi birikimini, çalışmalarını, zamanın benimle paylaşmaktan imtina etmeyen, tez danışmanım sayın Prof. Dr Serdar İplikçi'ye,

İşe başladığım günden beri beni destekleyen, değer veren, gerek bu tez için gerekse başka gerekçelerle ihtiyacım olduğunda kaynaklarını benimle paylaşmaktan kaçınmayan, TÜBİTAK 1509 destekli “ÖĞRENEN ALGORİTMA İLE DESTEKLENECEK OPERATİF SICAKLIK ESASINA DAYANAN AKILLI TERMOSTAT GELİŞTİRİLMESİ” başlıklı projenin koordinatörlüğünü bana teslim eden ve bu sayede hem bu tezin konusunu almamı sağlayan, tez çalışmalarım olduğunda anlayışla yaklaşarak kişisel gelişimime ve eğitimime destek veren, MİR ARAŞTIRMA ve GELİŞTİRME A.Ş.'ye, yöneticilerime ve mesai arkadaşlarıma,

Bahsi geçen projeyi destekleyerek şirketimin benim ve projede çalışan arkadaşlarımla gelişimini, eğitimini, yetkililer kazanmasını sağlayan Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na,

Bu Tez'in çalışması başta olmak üzere her konuda beni destekleyen, elimden tutan, arka çıkan, yönlendiren, yardımcı olan, emek veren, güzel eşim Dr. Yasemin Kaya Uncuoğlu'na,

Hem iyi günde hem kötü günde hep benim yanımda olan, Bu günlere gelmemi sağlayan, gerektiğinde canlarından can veren, benim için her şeyi yapmaya hazır, haklarını asla ödeyemeyeceğim canım annem Nurgün Uncuoğlu ve babam Mehmet Ali Uncuoğlu'na

Bana bir kardeş kadar yakın olan, Gürcan Güray'a

Ve tüm dostlarıma,

Sonsuz teşekkürlerimi iletirim.

8. KAYNAKLAR

Ahmad F., A. & Pattipati, K., Wang, B., Luh, P., “Predicting individual thermal comfort using machine learning algorithms”, *2015 IEEE International Conference on Automation Science and Engineering (CASE)* 708-713, 10.1109/CoASE.2015.7294164, (2015).

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., “ASHRAE Standard 55-2010 Thermal Environmental Conditions for Human Occupancy”, (2010).

Baker, N., Standeven, M., "A Behavioural Approach to Thermal Comfort Assessment in Naturally Ventilated Buildings". *Proceedings from CIBSE National Conference*, 76-84, (1995)

Bin, S., Ke, H., “Indoor Thermal Comfort PMV Index Prediction Based on Particle Swarm Algorithm and Least Square Support Vector Machine”, *Proceeding of International Conference on Intelligent System Design and Engineering Application (ISDEA)*, 857 - 860, 10.1109/ISDEA.2010.322, (2010).

Burges, C.J., “A tutorial on support vector machines for pattern recognition”, *Knowledge Discov Data Mining*, 2, 2, 1–43, (1998).

Byvatov, E., Fechner, U., Sadowski, J., Schneider, G., “Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification”, *Journal of chemical information and computer sciences*, 43, 1882-9. 10.1021/ci0341161, (2003).

Castilla, M., Álvarez, J.D., Ortega, M.G., Arahall, M.R., “Neural network and polynomial approximated thermal comfort models for HVAC systems”, *Building and Environment*, 59, 107-115, (2013).

Cortes, C., Vapnik, V.N., “Support Vector Networks”, *Machine Learning*. 20. 273-297. 10.1023/A:1022627411411, (1995).

de Dear, R., Brager, G., Cooper, D., “Developing an Adaptive Model of Thermal Comfort and Preference - Final Report on RP-884”, *ASHRAE Transactions*, 104, (1997).

European Committee for Standardization (CEN), “Ergonomics of the thermal environment - Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria (ISO 7730:2005)”, (2005).

European Committee for Standardization (CEN), “Ergonomics of the thermal environment - Instruments for measuring physical quantities (ISO 7726:1998)”, (2001).

Enerji Ve Tabii Kaynaklar Bakanlığı (ETKB), "Mavi Kitap 2016", (2017)

Fanger, P.O., “Thermal Comfort: analysis and applications in environmental engineering”, *Danish Technical Press*, Copenhagen, Denmark, Danish Kr. 76, 50, 244 , (1970).

Ferreira, P.M., Ruano, A.E., Silva, S., Conceição, E.Z.E., “Neural networks based predictive control for thermal comfort and energy savings in public buildings”, *Energy and Buildings*, 55, 238-251, (2012).

Frontczak, M., Wargocki, P., “Literature survey on how different factors influence human comfort in indoor environments”, *Building and Environment*, 46, 922-937, (2011), 10.1016/j.buildenv.2010.10.021.

Girosi, F., “An Equivalence Between Sparse Approximation and Support Vector Machines”, *Technical Report AIM-1606*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, (1997).

Gunn, S., “Support Vector Machines for Classification and Regression”, *ISIS Technical Report*, (1998).

Hall, J., Guyton, A., “Textbook of Medical Physiology 11th Edition”, Saunders, 2007.

Innova AirTech Instruments, " INNOVA Thermal Comfort Booklet", (2002)

Jiang, L., Yao, R. “Modelling personal thermal sensations using C-Support Vector Classification (C-SVC) algorithm”, *Building and Environment*, 99. pp. 98-106. ISSN 0360-1323, (2016).

Jodas, D., Marranghello, N., Pereira, A. S. Guido, R., “Comparing Support Vector Machines and Artificial Neural Networks in the Recognition Of Steering Angle for Driving of Mobile Robots Through Paths in Plantations”, *Procedia Computer Science*, 18, 240–249. 10.1016/j.procs.2013.05.187, (2013).

Kecman, V., “Support Vector Machines – An Introduction”, *Support Vector Machines: Theory and Applications*, 177, 605-605, 10.1007/10984697_1, (2005).

Kusiak, A., Tang, F., Xu, G., “Multi-objective optimization of HVAC system with an evolutionary computation algorithm”, *Energy*, 36, 5, 2440-2449 , (2011).

Ku, K. L., Liaw, J. S., Tsai, M. Y., Liu, T. S., “Automatic Control System for Thermal Comfort Based on Predicted Mean Vote and Energy Saving”, *IEEE Transactions on Automation Science and Engineering*, vol. 12(1), 378-383, doi: 10.1109/TASE.2014.2366206, (2015).

Li, N., Yu, W., Li, B., “Assessing Adaptive Thermal Comfort Using Artificial Neural Networks in Naturally-Ventilated Buildings”, *International Journal of Ventilation*, 11, 2, 205-218, (2012).

Megri, A. C., El Naqa, I., Haghghat, F., “A Learning Machine Approach for Predicting Thermal Comfort Indices”, *International journal of ventilation*, 3, 363-376, 10.1080/14733315.2005.11683930, (2005).

Megri, A. C., El Naqa, I., “Prediction of the thermal comfort indices using improved support vector machine classifiers and nonlinear kernel functions”, *Indoor and Built Environment*, 25, 1, 6-16, (2014)

Mercer, j., “XVI. Functions of positive and negative type, and their connection the theory of integral equations”, 209, 415-446, DOI: 10.1098/rsta.1909.0016., (1909).

Pamuk, O., “Thermal Manikins: an Overview”, *e-Journal of New World Sciences Academy*, 3,1,A0057, (2008).

Prek, M., “Thermodynamic analysis of human heat and mass transfer and their impact on thermal comfort”, *Int. J. Heat Mass Transfer*, 48, 731–739, (2005).

Schiller, G. E., Arens, E., Bauman, F., Benton, C., Fountain, M., Doherty, T., “A Field Study of Thermal Environments and Comfort in Office Buildings: Final Report–ASHRAE 462”, *CEDR:UC Berkeley*, (1988).

Schölkopf, B., Burges, C., Smola, A., *Advances in kernel methods: support vector: learning*, Cambridge, MA: MIT Press, (1999).

Schölkopf, B., Luo, Z., Vovk, V., *Empirical inference. Festschrift in honor of Vladimir N. Vapnik*. Springer-Verlag Berlin Heidelberg, 10.1007/978-3-642-41136-6. (2013).

Swiss Federal Office of Energy (SFOE), “Energie-Forschung 2012”. *berblicksberichte der Programmleiter (Program yöneticilerinin genel raporları)*, Swiss Federal Office of Energy SFOE, (2013).

Valavanis, I. K., Spyrou, G. M., Nikita K. S., “A comparative study of multi-classification methods for protein fold recognition”, *International Journal of Computational Intelligence in Bioinformatics and Systems Biology (IJCIBSB)*, 1, 3, (2010).

Van Hoof, J., “Forty years of Fanger’s model of thermal comfort: comfort for all?”, *Indoor Air*, 18, 182-201, (2008).

Torres, J. L., Marcelo L. M., “Adaptive Control of Thermal Comfort Using Neural Networks.” (2008).

EKLER

9. EKLER

EK A.1 PMV Hesaplama Programı

Bu program ISO7730 ve ASHRAE-55 standartlarında yer alan haliyle yazılmıştır.

```
function PMV=PMVCalculator(TA,TR,VEL,HR,met,clo)

EPS=0.00015;
ICL=0.155*clo;
M=met*58.15;
if ICL<=0.078
    FCL=1+1.29*ICL;
else
    FCL=1.05+0.645*ICL;
end
HCF=12.1*sqrt(VEL);
TAA=TA+273;
TRA=TR+273;
TCLA=TAA+(35.5-TA)/(3.5*ICL+0.1);

P1=ICL*FCL;
P2=P1*3.96;
P3=P1*100;
P4=P1*TAA;
P5=308.7-0.028*M+P2*(TRA/100)^4;

XN=TCLA/100;
XF=XN;
for N=1:150
    XF=(XF+XN)/2;
    HCN=2.38*(abs(100*XF-TAA))^0.25;
    if HCF>HCN
        HC=HCF;
    else
        HC=HCN;
    end
    XN=(P5+P4*HC-P2*XF^4)/(100+P3*HC);
    if abs(XN-XF)<EPS
        break;
    end
end
TCL=100*XN-273;
PA=HR*10*exp(16.6536-4030.183/(TA+235));
HL1=3.05*0.001*(5733-6.99*M-PA);
if M>58.15
    HL2=0.42*(M-58.15);
```

```

else
    HL2=0;
end
HL3=1.7*0.00001*M*(5867-PA);
HL4=0.0014*M*(34-TA);
HL5=3.96*FCL*(XN^4-(TRA/100)^4);
HL6=FCL*HC*(TCL-TA);
TS=0.303*exp(-0.036*M)+0.028;
PMV=TS*(M-HL1-HL2-HL3-HL4-HL5-HL6);

```

EK A.2 Rastgele Çok Boyutlu PMV Verisi Üretici

PMV hesaplama programını kullanarak istenilen aralıkta ve sayıda 6 boyutlu veri kümesini sınıflarıyla birlikte oluşturur.

```

function
[INPUTs, LABELs, PMVs]=RandomMultiDimPMVCalculator(DTA, DTR, DVEL, DHR, Dmet, Dclo, n)

PMVs=zeros(1, n);
INPUTs=zeros(6, n);
LABELs=zeros(1, n, 'int8');
TAmin=DTA(1);
TAdiff=DTA(size(DTA, 2))-TAmin;

TRmin=DTR(1);
TRdiff=DTR(size(DTR, 2))-TRmin;

VELmin=DVEL(1);
VELdiff=DVEL(size(DVEL, 2))-VELmin;

HRmin=DHR(1);
HRdiff=DHR(size(DHR, 2))-HRmin;

metmin=Dmet(1);
metdiff=Dmet(size(Dmet, 2))-metmin;

clomin=Dclo(1);
clodiff=Dclo(size(Dclo, 2))-clomin;

for k=1:n
    TA=TAmin+rand()*TAdiff;
    TR=TRmin+rand()*TRdiff;
    VEL=VELmin+rand()*VELdiff;
    HR=HRmin+rand()*HRdiff;
    met=metmin+rand()*metdiff;
    clo=clomin+rand()*clodiff;
    PMV=PMVCalculator(TA, TR, VEL, HR, met, clo);

    if PMV<-0.5
        LABEL=1;
    end
end

```



```

else if PMV<0
    LABEL=2;
else if PMV<=0.5
    LABEL=3;
else
    LABEL=4;
end
end
end
INPUTs (:, k)=[TA;TR;VEL;HR;met;clo];
PMVs (k)=PMV;
LABELs (k)=LABEL;
end

```

EK A.3 Düzenli Çok Boyutlu PMV Verisi Üretici

Girişinde verilmiş olan vektörleri kullanarak 6 boyutlu uzayda veri kümesi oluşturur. Veri kümesi sayısı vektörlerin elemanlarının çarpımı kadar olur. PMV hesaplama programına göre performansı optimize edildiği için farklı yazılmıştır.

```

function
[INPUTs, LABELs, PMVs]=MultiDimPMVCalculator (DTA, DTR, DVEL, DRH, Dmet, Dclo)
o)

t=0;
EPS=0.00015;
for r=1:size(Dclo,2)
    clo=Dclo(r);
    ICL=0.155*clo;
    if ICL<=0.078
        FCL=1+1.29*ICL;
    else
        FCL=1.05+0.645*ICL;
    end
    P1=ICL*FCL;
    P2=P1*3.96;
    P3=P1*100;
    for o=1:size(DVEL,2)
        VEL=DVEL(o);
        HCF=12.1*sqrt(VEL);
        for n=1:size(DTR,2)
            TR=DTR(n);
            TRA=TR+273;
            for q=1:size(Dmet,2)
                met=Dmet(q);
                M=met*58.15;
                P5=308.7-0.028*M+P2*(TRA/100)^4;
                for m=1:size(DTA,2)
                    TA=DTA(m);
                    TAA=TA+273;
                    P4=P1*TAA;

```


EK A.4 Ana Program

```
clear
close all
clc

C=1000;

verbose = true;

% DTA=22:1.5:28;
% DTR=22:1.5:28;
% DVEL=0.1:0.1:0.3;
% DHR=40:10:60;
% Dmet=1:0.2:1.4;
% Dclo=0.5:0.25:1;
%
% % DVEL=0.1;
% % DHR=50;
% % % Dmet=1;
% % Dclo=0.75;
%
%
[INPUTs,LABELs,PMVs]=MultiDimPMVCalculator(DTA,DTR,DVEL,DHR,Dmet,Dclo);
% n=length(LABELs);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Random point Test %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DTA=[22,28];
DTR=[22,28];
DVEL=[0.1,0.3];
DHR=[40,60];
Dmet=[1,1.4];
Dclo=[0.5,1];

% DTA=[20,28];
% DTR=[20,28];
% DVEL=[0.1,0.1];
% DHR=[50,50];
% Dmet=[1,1.4];
% Dclo=[0.7,0.7];

% DVEL=0.1;
% DHR=50;
% Dmet=1;
% Dclo=0.75;

n=100;
[INPUTs,LABELs,~]=RandomMultiDimPMVCalculator(DTA,DTR,DVEL,DHR,Dmet,Dclo,n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Random point Test %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[AllALPHAs,AllSVs,AllSGMs,AllTrain,AllYtrain]=MIMOSMC(INPUTs,LABELs,
verbose,C);
drawnow

TVEL=max(DVEL);
```

```

THR=min(DHR);
Tmet=min(Dmet);
Tclo=min(Dclo);
DrawPMVvsSVMGraph(AllALPHAs,AllSVs,AllSGMs,AllTrain,AllYtrain,DTA,DT
R,TVEL,THR,Tmet,Tclo,n)
drawnow

```

```

TVEL=min(DVEL)+3*(max(DVEL)-min(DVEL))/4;
THR=min(DHR)+(max(DHR)-min(DHR))/4;
Tmet=min(Dmet)+(max(Dmet)-min(Dmet))/4;
Tclo=min(Dclo)+(max(Dclo)-min(Dclo))/4;
DrawPMVvsSVMGraph(AllALPHAs,AllSVs,AllSGMs,AllTrain,AllYtrain,DTA,DT
R,TVEL,THR,Tmet,Tclo,n)
drawnow

```

```

TVEL=(max(DVEL)+min(DVEL))/2;
THR=(max(DHR)+min(DHR))/2;
Tmet=(max(Dmet)+min(Dmet))/2;
Tclo=(max(Dclo)+min(Dclo))/2;
DrawPMVvsSVMGraph(AllALPHAs,AllSVs,AllSGMs,AllTrain,AllYtrain,DTA,DT
R,TVEL,THR,Tmet,Tclo,n)
drawnow

```

```

TVEL=min(DVEL)+(max(DVEL)-min(DVEL))/4;
THR=min(DHR)+3*(max(DHR)-min(DHR))/4;
Tmet=min(Dmet)+3*(max(Dmet)-min(Dmet))/4;
Tclo=min(Dclo)+3*(max(Dclo)-min(Dclo))/4;
DrawPMVvsSVMGraph(AllALPHAs,AllSVs,AllSGMs,AllTrain,AllYtrain,DTA,DT
R,TVEL,THR,Tmet,Tclo,n)
drawnow

```

```

TVEL=min(DVEL);
THR=max(DHR);
Tmet=max(Dmet);
Tclo=max(Dclo);
DrawPMVvsSVMGraph(AllALPHAs,AllSVs,AllSGMs,AllTrain,AllYtrain,DTA,DT
R,TVEL,THR,Tmet,Tclo,n)
drawnow

```

EK A.5 Destek Vektörünü Eğitim ve Test Etme Programı

```

function
[AllALPHAs,AllSVs,AllSGMs,AllTrain,AllYtrain]=MIMOSMC(INPUT,LABELS,v
erbose,C)

LineRes=100;
SGMMin=0.1;
SGMFinal=10;

```

```

%-----
if verbose==true
    figure(2)
    for k=1:length(LABELS)

h=text(INPUT(1,k),INPUT(2,k),num2str(LABELS(k)),'FontWeight','bold')
;
        set(h,'color',[0 0 1])
    end
    hold on
    T1min = floor(min(INPUT(1,:)));
    T1max = ceil(max(INPUT(1,:)));
    T2min = floor(min(INPUT(2,:)));
    T2max = ceil(max(INPUT(2,:)));
    axis([T1min,T1max,T2min,T2max])
    pTz=gca; set(pTz,'FontName','Cambria');set(pTz,'FontSize',10)
    set(gcf,'color',[1 1 1])
    set(gcf,'Position',[200 200 600 500])

end
%-----

%-----
NUMofCLASSIFIERS=2;
%-----

%-----
NegativeClassesMatrix=[3 4;1 4];
PositiveClassesMatrix=[1 2;2 3];
%-----

AllALPHAs = [];
AllSVs = [];
AllSGMs = [];
ClassificationStart=tic;
for ClassifierIndex=1:NUMofCLASSIFIERS
    %-----
    NegativeClasses = NegativeClassesMatrix(ClassifierIndex,:);
    PositiveClasses = PositiveClassesMatrix(ClassifierIndex,:);
    Ttrain=INPUT;
    Ytrain=(LABELS==PositiveClasses(1) |
LABELS==PositiveClasses(2))-(LABELS==NegativeClasses(1) |
LABELS==NegativeClasses(2));
    %-----

    %-----
    KernelParameter = [];
    NUMofSVMs = [];
    NUMofERRORs = [];
    %-----

    %-----
    SGMIterationSteps=[0.81 0.27,0.09,0.03,0.01];
    SGMScanningStart=SGMMin;
    SGMScanningFinal=SGMFinal;
    SGMStep=1;
    SGMIterationStep=SGMIterationSteps(SGMStep);
    BestSGMChanged=false;
    BestTOTAL = inf;
    MinimumNOTFound=true;

```

```

FirstIteration=true;
%-----
PeriodTime=0;
OutputTime=0;
PeriodStart=tic;
while MinimumNOTFound
    TOTALPrevious = inf;
    for SGM=SGMScanningStart:SGMIterationStep:SGMScanningFinal
        %-----
        if SGM~=SGMScanningStart
            PeriodTime = toc(PeriodStart);
            fprintf('training Time: %f seconds\n',TrainTime)
            fprintf('Output Time: %f seconds\n',OutputTime)
            fprintf('Period Time: %f seconds\n',PeriodTime)
            fprintf('Time Percentage: %f
seconds\n', (TrainTime+OutputTime)/PeriodTime)
        end
        PeriodStart=tic;

        %-----
        disp('SGM:')
        disp(SGM)
        fprintf('\n');

        %-----

        %-----
        IndexOfPreviouslyCalculatedSGM=find(abs(KernelParameter-
SGM)<0.0001);
        if isempty(IndexOfPreviouslyCalculatedSGM)
            TrainStart=tic;
            [ALPHAS,SV] = SVCtrain(Ttrain,Ytrain,SGM,C);
            TrainTime = toc(TrainStart);
            OutputStart=tic;
            [y] =
SVCOutput(Ttrain,ALPHAS(SV),Ttrain(:,SV),Ytrain(SV),SGM);
            OutputTime = toc(OutputStart);
            NUMofERR = sum(Ytrain~=sign(y));
            KernelParameter = [KernelParameter,SGM];
            NUMofSVMs = [NUMofSVMs,length(SV)];
            NUMofERRORs = [NUMofERRORs,NUMofERR];
            TOTAL = length(SV) + NUMofERR;
        else
            TOTAL =
NUMofSVMs(IndexOfPreviouslyCalculatedSGM)+NUMofERRORs(IndexOfPreviou
slyCalculatedSGM);
        end
        %-----

        %-----
        figure(1)
        subplot(NUMofCLASSIFIERS,1,ClassifierIndex)
        hold on
        plot(SGM,TOTAL,'k+')
        %-----

        %-----
        if TOTAL<=BestTOTAL
            if FirstIteration
                BestTOTAL = TOTAL;

```

```

        BestSGM = SGM;
        BestALPHAS = ALPHAS(SV);
        BestSV = SV;
        BestNUMofERRORs = NUMofERR;
        BestTrain = Ttrain(:,SV);
        BestYtrain = Ytrain(SV);
    else
        if (TOTAL<BestTOTAL && SGM ~= BestSGM) ||
(TOTAL==BestTOTAL && SGM < BestSGM)
            BestTOTAL = TOTAL;
            BestSGM = SGM;
            BestALPHAS = ALPHAS(SV);
            BestSV = SV;
            BestNUMofERRORs = NUMofERR;
            BestTrain = Ttrain(:,SV);
            BestYtrain = Ytrain(SV);

            %-----
-
            SGMStep=SGMStep-1;
            if SGMStep<1; SGMStep=1;end
            SGMIterationStep=SGMIterationSteps(SGMStep);
            if BestSGM-2*SGMIterationStep<SGMMin
                SGMScanningStart=SGMMin;
            else
                SGMScanningStart=BestSGM-
2*SGMIterationStep;
            end
            if BestSGM+2*SGMIterationStep>SGMFinal
                SGMScanningFinal=SGMFinal;
            else
                SGMScanningFinal=BestSGM+2*SGMIterationStep;
            end
            SGMStep=SGMStep+1;
            BestSGMChanged=true;
            SGMIterationStep=SGMIterationSteps(SGMStep);
            break;
            %-----
-
        end

    end

end
%-----

%-----
if (TOTALPrevious==inf && SGMStep==1)
    TOTALComparisonHeightDiff=TOTAL/5;
    TOTALComparisonHeight=TOTAL/2;
    ContinuousIncrementCount=0;
else
    if (TOTAL>TOTALPrevious)
        if (TOTAL-BestTOTAL>TOTALComparisonHeightDiff )
ContinuousIncrementCount=ContinuousIncrementCount+1;
            if (ContinuousIncrementCount>2 &&
TOTAL>TOTALComparisonHeight)
                break;
            end
        end
    end
end
end

```

```

        else if TOTAL<TOTALPrevious
            ContinuousIncrementCount=0;
        end
    end
end
TOTALPrevious=TOTAL;
%-----

end %the end of
"SGM=SGMScanningStart:SGMIterationStep:SGMScanningFinal"
%-----

%-----
FirstIteration=false;
plot (BestSGM,BestTOTAL,'r+')
disp (BestSGM)
if BestSGMChanged==false
    if BestSGM-2*SGMIterationStep<SGMMin
        SGMScanningStart=SGMMin;
    else
        SGMScanningStart=BestSGM-2*SGMIterationStep;
    end

    if BestSGM+2*SGMIterationStep>SGMFinal
        SGMScanningFinal=SGMFinal;
    else
        SGMScanningFinal=BestSGM+2*SGMIterationStep;
    end
    SGMStep=SGMStep+1;
    if SGMStep>size (SGMIterationSteps)
        MinimumNOTFound=false;
        SGMStep=SGMStep-1;
    end;
    SGMIterationStep=SGMIterationSteps (SGMStep);

end
BestSGMChanged=false;
%-----

end

%-----
AllALPHAs (1:length (BestALPHAS),ClassifierIndex) = BestALPHAS';
AllSVs (1:length (BestSV),ClassifierIndex) = BestSV';
AllSGMs (1,ClassifierIndex) = BestSGM';

AllTrain (1:size (BestTrain,1),1:size (BestTrain,2),ClassifierIndex) =
BestTrain;
AllYtrain (ClassifierIndex,1:length (BestYtrain)) = BestYtrain;
%-----

%-----
[KernelParameterSorted,SortingVector]=sort (KernelParameter);
NUMofSVMs=NUMofSVMs (SortingVector);
NUMofERRORs=NUMofERRORs (SortingVector);
%-----

%-----
figure (3)
subplot (NUMofCLASSIFIERS,1,ClassifierIndex)
plot (KernelParameter,'b-o')

```



```

xlabel('iterasyon sayısı');
ylabel('\sigma')
%-----

%-----
figure(1)
subplot(NUMofCLASSIFIERS,1,ClassifierIndex)
p1=plot(KernelParameterSorted,NUMofSVMs,'b');
hold on
p2=plot(KernelParameterSorted,NUMofERRORs,'r');
p3=plot(KernelParameterSorted,NUMofSVMs+NUMofERRORs,'k');
plot(BestSGM,0,'g*')
grid
pTz=xlabel('\sigma'); set(pTz,'FontName','Cambria');
set(pTz,'FontSize',10);
pTz=ylabel('SVM ve Hata Sayısı'); set(pTz,'FontName','Cambria');
set(pTz,'FontSize',10);
legend([p1 p2 p3], 'SVM Sayısı', 'Hata Sayısı', 'Toplam')
pTz=title(['N=',num2str(length(Ytrain)), ' | '
', '\sigma*=', num2str(BestSGM), ' | ', 'DV
sayısı=', num2str(length(BestSV)), ' | ', 'Hata
Miktarı=', num2str(BestNUMofERRORs)]); set(pTz,'FontName','Cambria');
set(pTz,'FontSize',10); %set(h,'FontAngle','Italic')
pTz=gca; set(pTz,'FontName','Cambria');set(pTz,'FontSize',10)
set(gcf,'color',[1 1 1])
set(gcf,'Position',[863.4000 65 565.6000 712])
%-----

%-----
if verbose==true
    Tzero=[];
    for t1=T1min:(T1max-T1min)/LineRes:T1max
        PREVy=0;
        for t2=T2min:(T2max-T2min)/LineRes:T2max
            xtest=[t1;t2];

[y]=SVCOutput(xtest,BestALPHAS,Ttrain(1:2,BestSV),Ytrain(BestSV),Bes
tSGM);

            if PREVy*y<0
                Tzero=[Tzero;[t1,t2]];
            end
            PREVy=y;

        end
    end

figure(2)
for t2=T2min:(T2max-T2min)/LineRes:T2max
    PREVy=0;
    for t1=T1min:(T1max-T1min)/LineRes:T1max
        xtest=[t1;t2];

[y]=SVCOutput(xtest,BestALPHAS,Ttrain(1:2,BestSV),Ytrain(BestSV),Bes
tSGM);

            if PREVy*y<0
                Tzero=[Tzero;[t1,t2]];
            end
            PREVy=y;

        end
    end
end

```

```

        end
        figure(2)
        if ClassifierIndex==1
            if ~isempty(Tzero)
                pTz1=plot(Tzero(:,1),Tzero(:,2),'b. ');
set(pTz1,'markersize',5); set(pTz1,'Color',[1 0 1]);
            end
            pTzm1=plot([0;0],[0;0],'b'); set(pTzm1,'Color',[1 0
1],'LineWidth',2);
            end
            if ClassifierIndex==2
                if ~isempty(Tzero)
                    pTz2=plot(Tzero(:,1),Tzero(:,2),'b. ');
set(pTz2,'markersize',5); set(pTz2,'Color',[1 0 0]);
                end
                pTzm2=plot([0;0],[0;0],'b'); set(pTzm2,'Color',[1 0
0],'LineWidth',2);
            end
            end
            end
            %-----
end
TotalTime = toc(ClassificationStart);
fprintf('Total Time: %f seconds\n',TotalTime)
legend([pTzm1 pTzm2], 'SVM1', 'SVM2', 'Location','northwest');
pText=xlabel('Ta-Kuru Hava
Sıcaklığı');set(pText,'FontName','Cambria');
set(pText,'FontSize',10);
pText=ylabel('Tr-Radyan Sıcaklık'); set(pText,'FontName','Cambria');
set(pText,'FontSize',10);
grid on

```

EK A.6 Destek Vektörü Sınıflandırıcısı Eğitim Programı

```

function [ALPHAS,SV]=SVCtrain(Ttrain,Ytrain,SGM,C)
N = length(Ytrain);
H=zeros(size(Ytrain));
for i=1:N;
    for j=1:N;
        H(i,j)=Ytrain(i)*Ytrain(j)*kernelfunction(Ttrain(:,i),Ttrain(:,j),SG
M);
    end
end
g = -ones(N,1);
Aie = [];
bie = [];
Aeq = [];
beq = [];
lb = zeros(N,1); % Set the bounds: alphas >= 0
ub = C*ones(N,1); % alphas <= C
x0 = zeros(N,1); % The starting point is [0 0 0 ... 0]
opts = optimset('Algorithm','interior-point-
convex','Display','off');
[ALPHAS] = quadprog(H, g, Aie, bie, Aeq, beq, lb, ub, x0, opts);
SV=(find(abs(ALPHAS)>1e-4));

```

EK A.7 Destek Vektörü Sınıflandırıcısı Programı

```
function [y]=SVCoutput (Tall,ALPHAS,Ttrain,Ytrain,SGM)

y=0;
for m=1:size(Tall,2)
    xtest=Tall(:,m);
    y(m)=0;
    for i=1:length(ALPHAS)
        y(m)= y(m) +
ALPHAS(i)*Ytrain(i)*kernelfunction(Ttrain(:,i),xtest,SGM);
    end
end
```

EK A.8 Kernel Fonksiyonu

```
function K=kernelfunction(x,y,ss)
K=exp(-((x-y)'*(x-y))/(2*ss*ss));
```

EK A.9 Öğrenilmiş bir SVM Kullanarak Veri Kümesi Çizme Programı

```
function
DrawPMVvsSVMGraph (AllALPHAs,AllSVs,AllSGMs,AllTrain,AllYtrain,DTA,DT
R,DVEL,DHR,Dmet,Dclo,n)

T1min = floor(min(DTA));
T1max = ceil(max(DTA));
T2min = floor(min(DTR));
T2max = ceil(max(DTR));

figure
hold on
axis ([T1min,T1max,T2min,T2max])

LineRes=100;

T4=[];
T3=[];
T2=[];
T1=[];
BestSV1=AllSVs (AllSVs(:,1)>0,1);
BestSV2=AllSVs (AllSVs(:,2)>0,2);
BestALPHAS1=AllALPHAs (1:length(BestSV1),1);
BestALPHAS2=AllALPHAs (1:length(BestSV2),2);
Ttrain1=AllTrain(:,1:length(BestSV1),1);
Ttrain2=AllTrain(:,1:length(BestSV2),2);
Ytrain1=AllYtrain(1,1:length(BestSV1));
Ytrain2=AllYtrain(2,1:length(BestSV2));
BestSGM1=AllSGMs(1);
```

```

BestSGM2=AllSGMs(2);

for t1=T1min:(T1max-T1min)/LineRes:T1max
    for t2=T2min:(T2max-T2min)/LineRes:T2max
        xtest=[t1;t2;DVEL(1);DHR(1);Dmet(1);Dclo(1)];

y1=SVCOutput(xtest,BestALPHAS1,Ttrain1(:, :),Ytrain1(:),BestSGM1);

y2=SVCOutput(xtest,BestALPHAS2,Ttrain2(:, :),Ytrain2(:),BestSGM2);
    if y1>0
        if y2>0
            T2=[T2;[t1,t2]];
        else
            T1=[T1;[t1,t2]];
        end
    else
        if y2>0
            T3=[T3;[t1,t2]];
        else
            T4=[T4;[t1,t2]];
        end
    end
end
end

hold on
if ~isempty(T1)
pT1=plot(T1(:,1),T1(:,2),'b. '); set(pT1,'markersize',15);
set(pT1,'Color',[0.1 0.1 1]);
end
if ~isempty(T2)
pT2=plot(T2(:,1),T2(:,2),'b. '); set(pT2,'markersize',15);
set(pT2,'Color',[0.6 0.6 1]);
end
if ~isempty(T3)
pT3=plot(T3(:,1),T3(:,2),'b. '); set(pT3,'markersize',15);
set(pT3,'Color',[1 0.6 0.6]);
end
if ~isempty(T4)
pT4=plot(T4(:,1),T4(:,2),'b. '); set(pT4,'markersize',15);
set(pT4,'Color',[1 0.1 0.1]);
end

[TestINPUTs,TestLABELs,~]=RandomMultiDimPMVCalculator(DTA,DTR,DVEL(1)
),DHR(1),Dmet(1),Dclo(1),n);

correct=0;
false=0;
for t=1:n
    xtest=TestINPUTs(:,t);

y1=SVCOutput(xtest,BestALPHAS1,Ttrain1(:, :),Ytrain1(:),BestSGM1);

y2=SVCOutput(xtest,BestALPHAS2,Ttrain2(:, :),Ytrain2(:),BestSGM2);
    if y1>0
        if y2>0 %LABEL 2
            if TestLABELs(t)==2
                correct=correct+1;
            else

```

```

        false=false+1;
    end
else %LABEL 1
    if TestLABELs(t)==1
        correct=correct+1;
    else
        false=false+1;
    end
end
else
    if y2>0 %LABEL 3
        if TestLABELs(t)==3
            correct=correct+1;
        else
            false=false+1;
        end
    else %LABEL 4
        if TestLABELs(t)==4
            correct=correct+1;
        else
            false=false+1;
        end
    end
end
end
end

percentage=100*correct/(correct+false);

pTitle=title({'Hava Hızı=', num2str(DVEL(1)), 'm/s', ' | ', 'Bağıl
Nem=', num2str(DHR(1)), ' | ', 'met=', num2str(Dmet(1)), ' |
', 'clo=', num2str(Dclo(1))}; ['Doğru Tahmin=', num2str(correct), ' |
', 'Yanlış Tahmin=', num2str(false), ' | ', 'Doğru Tahmin
Yüzdesi=', num2str(percentaje)]}); set(pTitle, 'FontName', 'Cambria');
set(pTitle, 'FontSize', 10);
pTitle=gca;
set(pTitle, 'FontName', 'Cambria'); set(pTitle, 'FontSize', 10)

for k=1:length(TestLABELs)

pText=text(TestINPUTs(1,k), TestINPUTs(2,k), num2str(TestLABELs(k)), 'F
ontWeight', 'bold');
    set(pText, 'color', [0 0 0])
end
xlabel('Ta-Kuru Hava Sıcaklığı'); set(pText, 'FontName', 'Cambria');
set(pText, 'FontSize', 10);
ylabel('Tr-Radyan Sıcaklık'); set(pText, 'FontName', 'Cambria');
set(pText, 'FontSize', 10);

```

10. ÖZGEÇMİŞ

Adı Soyadı : Emre Uncuoğlu

Doğum Yeri ve Tarihi : Denizli - 22.05.1984

Lisans Üniversite : İstanbul Teknik Üniversitesi

Elektronik posta : uncuoglu.emre@gmail.com

İletişim Adresi : Mareşal Çakmak mah. Bağcılar cad.
No:118/8 Güngören/ İstanbul

Yayın Listesi :

• Karadağ, S. , Görüşük, E. M., Çetinkaya, E. , Deveci, S. , Dönmez, K. B., Uncuoğlu, E. and Doğu, M., Development of an automated flow injection analysis system for determination of phosphate in nutrient solutions. Journal of the Science of Food and Agriculture, 98: 3926-3934. doi:10.1002/jsfa.8911, (2018)

Konferans listesi :

• Emre Uncuoğlu, Özcan Kalenderli, Hakan Kuntman, “Transistör Tetiklemeli Tesla Bobini Tasarımı ve Yapımı”, Elektrik-Elektronik Bilgisayar Mühendisliği Ulusal Kongresi, Ankara, s. 301-304, 23-26 Aralık 2009.