

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ ENERJİ ENSTİTÜSÜ

**MAKİNE ÖĞRENMESİ KULLANIMIYLA EV TİPİ KLİMALARDA
PARAMETRE VE ENERJİ VERİMLİLİĞİ OPTİMİZASYONU**

YÜKSEK LİSANS TEZİ

Ezgi Sevgi ÖZDEMİR

Enerji Bilim ve Teknoloji Anabilim Dalı

Enerji Bilim ve Teknoloji Programı

ARALIK 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ ENERJİ ENSTİTÜSÜ

**MAKİNE ÖĞRENMESİ KULLANIMIYLA EV TİPİ KLİMALARDA
PARAMETRE VE ENERJİ VERİMLİLİĞİ OPTİMİZASYONU**

YÜKSEK LİSANS TEZİ

**Ezgi Sevgi ÖZDEMİR
301161036**

Enerji Bilim ve Teknoloji Anabilim Dalı

Enerji Bilim ve Teknoloji Programı

Tez Danışmanı: Prof. Dr. Gülgün KAYAKUTLU

ARALIK 2019

İTÜ, Enerji Enstitüsü'nün 301161036 numaralı Yüksek Lisans Öğrencisi Ezgi Sevgi ÖZDEMİR, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “MAKİNE ÖĞRENMESİ KULLANIMIYLA EV TİPİ KLİMALARDA PARAMETRE VE ENERJİ VERİMLİLİĞİ OPTİMİZASYONU” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Gülgün KAYAKUTLU**

İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Doç. Dr. Umut Asan**

İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Ali PINARBAŞI**

Yıldız Teknik Üniversitesi

Teslim Tarihi : **15 Kasım 2019**
Savunma Tarihi : **9 Aralık 2019**



ÖNSÖZ

Yüksek Lisans tezim sırasında bana her an destek ve yardımcı olan, bilgisi ile beni yönlendiren ve yol gösteren danışmanım Sayın Prof. Dr. Gülgün KAYAKUTLU'ya teşekkürlerimi sunarım.

Hayatım boyunca beni destekleyen aileme ve en büyük destekçim olan abime en içten sevgi ve teşekkürlerimi sunarım.

Son olarak tez çalışmalarım sırasında her zaman desteklerini esirgemeyen, ihtiyacım olan kaynak ve olanakların yaratılmasında yardımcı olmuş olan Arçelik LG Klima ailesine teşekkürü bir borç bilirim.

Aralık 2019

Ezgi Sevgi Özdemir



İÇİNDEKİLER

Sayfa

ÖNSÖZ	v
İÇİNDEKİLER	vii
KISALTMALAR	ix
SEMBOLLER	xi
ÇİZELGE LİSTESİ	xiii
ŞEKİL LİSTESİ	xv
ÖZET	xvii
SUMMARY	xvii
1. GİRİŞ	1
1.1 Tezin Amacı	1
1.2 Literatür Araştırması	3
1.3 Hipotez	3
1.4 Amaç Fonksiyonu	3
2. SOĞUTMA ÇEVİRİMİ	5
2.1 Teknoloji Tanımı.....	5
2.2 Klima Çalışma Prensibi ve Komponentleri.....	6
2.3 Mevsimsel Enerji Verimliliği Kavramı	8
2.4 Deney Parametreleri ve Deney Düzenegi	12
2.4.1 Deney Parametreleri.....	12
2.4.2 Soğutma Kapasite ve Verim Hesabı	12
2.4.3 Kalorimetre Odaları	13
2.5 Deney Sonuçları ve Verilerin Normalleştirilmesi	15
3. SOĞUTMA ÇEVİRİMİNDE MAKİNE ÖĞRENMESİ	19
3.1 Makine Öğrenmesi	19
3.2 Soğutma Çevriminde Makine Öğrenmesi	19
3.3 Delphi Tekniği.....	21
3.4 Doğrusal ve Çok Terimli Regresyon.....	21
3.5 Rassal Orman Yöntemi	22
3.6 Yapay Sinir Ağları	22
4. YAZILIM VE UYGULAMA	25
4.1 Veriler ve Algoritma	25
4.2 Bulgular	27
5. SONUÇ VE ÖNERİLER	31
KAYNAKLAR	33
EKLER	35
ÖZGEÇMİŞ	61



KISALTMALAR

HVAC	: Isıtma, Havalandırma, İklimlendirme
KT	: Kuru termometre
PID	: Oransal, integral, türev
Rms	: Kareler kök ortalaması
Rpm	: Rotation per minute
SEER	: Seasonal efficiency ratio
TSE	: Türk Standartları Enstitüsü
YSA	: Yapay Sinir Ağları
YT	: Yaş termometre



SEMBOLLER

EER_{rated}	: Nominal koşullarda enerji verimliliği oranı [kW/kW]
EER_{bin(T_j)}	: Her T _j koşulu için hesaplanmış enerji verimliliği [kW/kW]
H_{SB}	: Bekleme konumu süresi [h]
H_{TO}	: Termostat kapalı konumu süresi [h]
H_{CE}	: Soğutma konumu süresi [h]
H_{CK}	: Karter ısıtıcısı çalışma süresi [h]
H_{OFF}	: Kapalı konum süresi [h]
h_j	: bin saatleri [h]
max_A	: A grubu içerisindeki en büyük sayı
min_A	: A grubu içerisindeki en küçük sayı
N	: Deneysel sayı
New_max_A	: A grubu içerisinde yeni en büyük sayı
New_min_A	: A grubu içerisinde yeni en küçük sayı
SEER	: Mevsimsel verimlilik oranı [kW/kW]
SEER_{ON}	: Aktif moddaki mevsimsel verimlilik oranı [kW/kW]
pl(T_j)	: Kısmi yük oranı [kW/kW]
P_{designc}	: Soğutma modunda tasarım yükü [kW]
P_c(T_j)	: T _j dış ortam sıcaklığı için belirtilen kısmi yük [kW]
T_{designc}	: Tasarım sıcaklığı [°C]
T_{in}	: İç ortam sıcaklığı [°C]
T_j	: Dış ortam sıcaklığı [°C]
Q_c	: Referans yıllık soğutma talebi [kW]
Q_{CE}	: Senelik elektrik tüketimi [kW]
Q_H	: Yüksek sıcaklıktaki ortama doğru ısı geçişi [kW]
Q_L	: Düşük sıcaklıktaki ortamdaki dışarıya ısı geçişi [kW]
W_{net}	: Sisteme etkiyen net iş [kW]



ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1: Her bir fonksiyonel durumda geçirilen operasyon saatleri.....	10
Çizelge 2.2: Mevsimsel verimlilik hesaplamasında kullanılan kısmi yük ve sıcaklıklar.....	11
Çizelge 2.3: SEER hesaplamasında kullanılan bin saatleri sıcaklıklar ve geçirilen süreler.....	11
Çizelge 2.4: Mevsimsel verimlilik hesaplamasında kullanılan kısmi yük ve sıcaklıklar.....	13
Çizelge 4.1: Yöntemlere göre test ve öğrenme verilerinin sapma durumları.	27
Çizelge 4.2: Optimizasyon sonuçları.	28
Çizelge 4.3: Kapasite ve verim için gerçek durum ile program çıktısı karşılaştırması.	29
Çizelge A.1: Normalize edilmiş deney sonuçları.	37



ŞEKİL LİSTESİ

Sayfa

Şekil 1.1: Çalışmanın akış diyagramı.	2
Şekil 2.1: Bir iç ve bir dış üniteden oluşan ev tipi klima örneği.	5
Şekil 2.2: Soğutma makinesi çevrimi.	6
Şekil 2.3: Gerçek buhar sıkıştırırmalı çevrim şeması ve sıcaklık entropi diyagramı.....	6
Şekil 2.4: Klima komponentleri ve soğutucu akışkanın izlediği yol.	7
Şekil 2.5: Klima donanımsal parçaları.	8
Şekil 2.6: Klima sistemi optimizasyon akış şeması.	8
Şekil 2.7: Hazırlanan numune görselleri.	16
Şekil 2.8: Psikrometrik tip kalorimetre odası ve odanın elemanları.....	17
Şekil 3.1: Yapay sinir hücresinin temel elemanları.	23
Şekil 3.2: Dört katmanlı yapay sinir ağı mimarisi.	24
Şekil 4.1: Algoritma akış diyagramı.	25



MAKİNE ÖĞRENMESİ KULLANIMIYLA EV TİPİ KLİMALARDA PARAMETRE VE ENERJİ VERİMLİLİĞİ OPTİMİZASYONU

ÖZET

Enerji ihtiyacı günümüzün ve geleceğin vazgeçilmezi olacaktır. Kaynaklar yönünden kısıtlı olan enerjinin tasarrufu ise büyük önem taşır. Enerji verimliliği, enerji tasarrufunun en pratik yöntemi olduğundan bu tezde enerji verimliliği konusu ev tipi klimalar için makine öğrenmesi yöntemleri kullanarak nasıl artırılabilir konusu üzerinde durulmuştur.

Avrupa Birliğinin 2020 hedefleri doğrultusunda yürürlüğe giren ErP regülasyonu ile mevsimsel enerji verimliliği kavramı hayatımıza girmiştir. Bu kavram bir klimanın bütün yıl çektiği çeşitli yükler ve yıl boyunca çeşitli sıcaklıklarda çalışması ile oluşacak farklı kapasiteler sonucu farklı yıl içinde sürekli değişen farklı enerji verimlilik değerlerinde çalışıyor olmasının bir sonucudur.

Ev tipi klimaların kullanımını gün geçtikte artmaktadır ve enerji tüketimlerini azaltmak ve verimliliklerini arttırmak için sürekli çalışmalar yapılmaktadır. Bir klimanın enerji verimliliği termodinamik çevriminin verimini arttırarak ya da sistemin çalışma parametrelerini yazılımsal olarak en uygun şekilde seçerek artırılabilir.

Verimliliği arttırmak üzere çalışma kısmına geçmeden önce soğutma çevriminden bahsetmek gerekir. Tezin ikinci bölümünde teknoloji tanımı yapılmış ve soğutma çevrimi anlatılmıştır. HVAC, ısıtma, havalandırma ve iklimlendirme sözcüklerinin kısaltmasıdır ve en sondaki iklimlendirme kısmında klimalar yer almaktadır. Klimalar farklı kullanım alanlarına göre isimlendirilebilirler. Ev tipi klima olarak adlandırılan klimalar genellikle ev ve ofis gibi alanları iklimlendirmek üzere kullanılırlar. Genel olarak bir iç ve bir dış üniteden meydana gelirler ve termodinamik olarak prensipte bir soğutma makinesi gibi çalışmaktadırlar.

Isı geçişi azalan sıcaklık yönünde olduğundan, bunun tam ters yönde gerçekleşmesini sağlamak yani sıcak bir ortamdan sıcaklığı daha düşük olan bir ortama doğru ısı atmak için böyle bir sistem dışarıdan bir güce ihtiyaç duyacaktır. Bu işlemi gerçekleştirebilmek için soğutma makinesi kullanılabilir. Soğutma makinesi temelde 4 bileşenden oluşur ve kapalı bir çevrim izler. Sistemin içinde dolaşan soğutucu akışkan kompresörün sıkıştırmasıyla enerji kazanarak sistemde yolculuğuna başlar. Sıkıştırılıp basıncı ve sıcaklığı artan soğutucu akışkan, yoğunlaştırıcıya gider ve yüksek sıcaklıkta olduğundan, daha düşük sıcaklıktaki ortama doğru bir ısı geçişi gerçekleşir. Daha sonra kısılma vanasına gider, burada basınç ve sıcaklığı oldukça düşer. Buharlaştırıcıya geldiğinde sıcaklığı oldukça düşüktür ve dış ortamdan ısı geçişi bu kez soğutucu akışkana doğru gerçekleşir. Daha sonra tekrar kompresöre giderek çevrimi sürekli tekrarlar. Böylelikle buharlaştırıcının bulunduğu ortamın sıcaklığı düşürülmüş olur.

Klimalar da soğutma makinesi prensibiyle çalışırlar ve bu sistemlerin verimlilikleri ya termal sistemin çalışma noktalarının belirlenmesiyle ya da sistem komponentlerinin

değiştirilmesiyle çeşitlendirilir. Bu da klimanın veriminin ya yazılım ya da donanımdan geçtiğini anlatır.

Donanımsal parçalar sabit kalmak üzere yazılımsal olarak bir klimanın verimliliğini arttırabilmek için öncelikle klimada verimlilik hesabının yani mevsimsel verimliliğin nasıl hesaplandığının irdelenmesi gerekir.

Mevsimsel verimlilik, referans yıllık soğutma değerinin yıllık elektrik tüketimi değerine oranıyla bulunur. Referans yıllık soğutma değeri, klima tasarımı yapılırken, klima için ön görülen tasarım yükü yani klimanın gerekli koşullarda vermesi beklenen kapasite değeri ile bu değer senelik soğutma saatleriyle çarpılması ile bulunur. Yıllık enerji tüketimi ise klimalar için yalnızca soğutma yaptıkları zaman çektikleri elektrik yükünü değil, hazırda bekleme modundayken, termostat kapalı moddayken ya da karter ısıtıcısı modundayken çektikleri elektrik yükünü de hesaba katar. Bu durumlar için elektrik tüketimi süreleri TS EN 14825 standardınca belirtilmiş olup, standartta geçen veriler hesaplamalarda kullanılmıştır.

Mevsimsel verimlilik, esasen tüm yıl boyunca yapılan elektrik tüketimi ve gözlenen verimi içerdiğinden, sene boyunca klima farklı sıcaklıklarda çalışacağından standartta belirtildiği üzere 4 farklı dış ortam sıcaklığında kapasite ve güç tüketimi ölçümü almak gereklidir.

4 farklı dış ortam sıcaklığı için yazılım optimizasyonu gerçekleştirilmiştir. Bunun için kullanılan parametreler, klimanın iç ve dış ünite fan hızları, elektronik kısılma vanasının açıklığı ve kompresör dönme hızıdır.

Parametre değişikliklerinin klima kapasitesine ve gücüne verdiği cevabı görmek için deney düzeneği gereklidir. Klima kapasitesi ölçümü için gerekli formülizasyonlar verilmiştir. Deneylede psikrometrik tip deney odası kullanılmıştır. Bu odalar klimanın çalışmasını simüle edebilmek için bir iç ve dış ortam şeklinde iki kısımdan oluşur. Klimanın iç ünitesi iç ortama dış ünitesi dış ortama kurulur. Her iki ortam da şartlandırılabilir yani istenen kuru ve yaş termometre sıcaklıklarına ayarlanabilir. Bu sıcaklıklar sağlandıktan sonra kapasite ölçümü için iç ünitenin yarattığı hava debisi ve ortam sıcaklığında yarattığı değişim bulunmalıdır. Bu nedenle iç ünite bir hava debisi ölçüm sistemine bağlanır ve kapasite böylelikle hesaplanır.

Parametreler birbirinden farklı birimlerde olduğundan normalizasyon yapılmıştır. Minimum maksimum normalizasyon yöntemi kullanılmış ve verim değeri birimsiz bir büyüklük olduğundan normalizasyona dahil edilmemiştir.

3. bölümde makine öğrenmesinin genel tanımı ve soğutma çevrimine uyarlanması incelenmiştir, konuyla ilgili literatür araştırmaları yapılmıştır. Makine öğrenmesi yöntemleri incelenmiştir. Makine öğrenmesi, mevcut verileri kullanarak yani bunları öğrenerek, daha önce hiç görmemiş olduğu veriler hakkında çıkarım yapabilmeyi saylayan bir yöntemdir. Makine öğrenmesi geniş uygulama alanlarına sahiptir ve optimizasyon problemlerine de katkıda bulunur.

Makine öğrenmesi soğutma çevriminde genel olarak klimaların çalışma algoritmalarını düzenlemek için kullanılmıştır. Kullanıcı seçimlerinden yola çıkarak sonraki seçimleri kestirme, enerji verimliliği artırma ve kullanıcı konforu artırma gibi hedefleri vardır.

Makine öğrenmesi yöntemleri içerisinde regresyon, girdiler ve çıktılar arasındaki ilişkiyi inceler. Çıktının girdilerle nasıl değiştiğine dair regresyon denklemleri üretir. Böylelikle daha önce verilmiş verilerden yola çıkarak önceden görmediği bir veri için

çıktıyı kestirir. Rassal orman yöntemi de aslında bir regresyon türüdür ve karar ağaçları oluşturarak bir orman yaratır ve bu ormandan en iyi çıktıyı tahmin eder. Yapay sinir ağları, insan beynini ve nöronları taklit eden bir yapıya sahiptir. Kendine verilen örnekleri kullanarak öğrenme işlevini bir insan beyni gibi gerçekleştirir. Yapay sinir ağlarında her bir ağın kendine özgü bir ağırlık değeri vardır ve sakladığı bilgi bu sinir ağının ağırlığı ile orantılıdır. Eksik ve belirsiz bilgilerle dahi rahatlıkla çalışabilen bir yöntemdir.

Makine yöntemleri tanıtıldıktan sonra soğutma çevriminde bu yöntemlerin nasıl kullanılacağı anlatılmıştır. Öncelikle 158 adet 4 farklı ortam koşulunda deney gerçekleştirilmiş, bu deneylerin girdileri ve çıktı olarak kapasite değeri normalize edilmiş daha sonra bu verilerin %80'i makine öğrenmesi ile öğrenilmiştir. Verilerin geri kalan %20'si ise öğrenilen kısımdan yola çıkılarak makine tarafından tahmin edilmeye çalışılır ve buna test denir.

Sonuç olarak doğrusal regresyon, çok terimli regresyon, rassal orman ve yapay sinir ağları olmak üzere 4 farklı model için kodlar oluşturulmuş, veriler öğrenilmiş, test edilmiş öğrenme ve test sonuçları incelenmiştir. Rassal orman yöntemi bu yöntemler içinde test verilerinin ortalamadan sapmasının en düşük olduğu yöntem olarak belirlenmiştir.

Daha sonra bu yöntemlerin her biri için optimizasyon problemi tanımlanmış ve kısıtlar girilmiştir. Verimin maksimum olacağı ve kapasitenin belirlenen aralıklarda kalacağı durumlar için bu kez hem bu noktaları sağlayacak parametreler hem de bu parametrelerin çıktısı olacak sonuçlar 4 farklı yöntem için elde edilmiştir.

Elde edilen parametre değerleri normalize değerler olduğundan öncelikle bu değerler tekrar gerçek değerlerine çevrilmiştir. Daha sonra ayarlanabilecek en uygun tam sayıya yuvarlanmışlardır. Parametrelerin verilen değerlerinde deneyler tekrar edilmiştir. Deneylerin gerçekleşen sonuçları ile modellerin tahmin ettikleri sonuçlar kıyaslanmıştır.

Modeller arasından en iyi mevsimsel enerji verimliliği değerini rassal orman yöntemi vermiştir. Makine öğrenmesi yöntemleri kullanılmadan önce yapılan deneylerin en iyi sonuçlarının kombinasyonu ile hesaplanan SEER değeri 6,689 iken makine öğrenmesi sonrası bu değer 6,767'ye yükselmiştir. Yıllık enerji tüketimi ise ürün başına 183,14 kW'tan 181,02 kW değerine kadar düşmüştür.

Enerji verimliliği optimizasyonu konusunda makine öğrenmesi yöntemlerini kullanan bu çalışmayı geliştirmek için elde edilen sonuçlar tekrar öğrenme verilerinin içerisine yerleştirilerek sonuçlar git gide iyileştirilecektir.



OPTIMIZATION OF PARAMETER AND ENERGY EFFICIENCY IN RESIDENTIAL AIR CONDITIONERS USING MACHINE LEARNING

SUMMARY

The need for energy will be indispensable for present and future. Conservation of energy, which is limited in terms of resources, is of great importance. Since energy efficiency is the most practical method of energy saving, this thesis focuses on how energy efficiency can be increased by using machine learning methods for home air conditioners.

With the ErP regulation, which came into force in line with the 2020 targets of the European Union, the concept of seasonal energy efficiency came into our lives. This concept is the result of the fact that an air conditioner operates at different energy efficiency values that change continuously in different years as a result of the different loads that it has suffered all year and the different capacities that will occur at various temperatures throughout the year. In hot summer season, the air conditioner will consume more electricity in comparison with early or last days of summer. Considering this diversity of temperature throughout the year, the efficiency measurement for an air conditioner must vary depending on working temperatures.

Through the seasonal energy efficiency calculation consumers could get more accurate information about the efficiency of an air conditioner.

The use of home air conditioners is increasing day by day and continuous efforts are being made to reduce energy consumption and increase their efficiency. The energy efficiency of an air conditioner can be increased by increasing the efficiency of the thermodynamic cycle or by selecting the operating parameters of the system in the most appropriate way.

In order to increase efficiency, it is necessary to talk about the cooling cycle before proceeding to the working principle. In the second part of the thesis, technology definition and cooling cycle are explained. HVAC stands for heating, ventilation and air conditioning, and air conditioning is located in the final part of the term. Air conditioners can be named according to different usage areas. Residential air conditioners generally used to air-condition areas such as home and office. They generally consist of an indoor unit and an outdoor unit, and thermodynamically operate in principle as a cooling machine.

Since the heat transfer is in the direction of the decreasing temperature, such a system will need external power to ensure that, this takes place in the opposite direction, which is, to extract heat from a hot environment to a lower temperature. A cooling machine can be used to perform this operation. The cooling machine consists essentially of 4 components and follows a closed cycle. The refrigerant circulating through the system gains energy by compressing the compressor and begins its journey in the system. The refrigerant, which is compressed and increased in pressure and temperature, goes to the condenser and, since it is at high temperature, a heat transfer occurs to the lower temperature medium. It then goes to the expansion valve, where the pressure and temperature drop considerably. When it comes to the evaporator, its temperature is

very low and the heat transfer from the outside takes place towards the refrigerant. It then goes back to the compressor and repeats the cycle continuously. Thus, the temperature of the environment where the evaporator is located is reduced.

Air conditioners operate on the principle of cooling machines, and their efficiency is varied either by determining the operating points of the thermal system or by replacing the system components. This means that the efficiency of the air conditioner passes either through software or hardware.

In order to increase the efficiency of an air conditioner in software with the hardware parts remaining constant, firstly, it is necessary to examine how the air conditioner efficiency calculation is calculated.

In order to explain software change, first parameters should be explain in detail. To get to the parameters, first air conditioner structure must be covered. An air conditioner consists of an outdoor unit, an indoor unit and a connection pipe for connecting two units together. Indoor unit consists main parts as an evaporator, a centrifugal fan and a control box for communication with outdoor unit. Outdoor unit also has a control box which consists of all of working algorithm that air conditioner has. Besides the control box, the outdoor unit also has a compressor, a condenser, an electronic expansion valve and a fan.

The parameters which effects the efficiency of the air conditioner are actually all of the parts that mentioned above. But since the hardware cannot be changed or modified easily only software is changed for this study. Software consists of indoor fan rotation speed, outdoor fan rotation speed, opening of electronic expansion valve and the compressor rotation speed. So all of these parameters can easilily be changed and for different combination of these parameters tests were done.

Seasonal efficiency is determined by the ratio of the reference annual cooling demand to the annual electricity consumption value. The reference annual cooling demand is calculated by multiplying the design load foreseen for the air conditioner, ie the capacity value expected to be provided by the air conditioner, and the annual cooling hours. The annual energy consumption does not only take into account the electrical charge they draw when air conditioners are cooling, but also the electrical load they take when in standby mode, thermostat off mode or crankcase heater mode. The electricity consumption times for these cases are specified in the TS EN 14825 standard and the data passed in the standard are used in the calculations.

Since seasonal efficiency mainly includes electricity consumption and observed efficiency throughout the year, it is necessary to take capacity and power consumption measurements at 4 different outdoor temperatures as specified in the standard since the air conditioner will operate at different temperatures throughout the year.

Software optimization was performed for 4 different outdoor temperatures. The parameters used for this are the indoor and outdoor unit fan speeds of the air conditioner, the opening of the electronic expansion valve and the compressor rotation speed.

A test setup is required to see the response of the parameter changes to the air conditioning capacity and power. The necessary formulations for air conditioning capacity measurement are given. Psychrometric type test room was used in the experiments. These rooms consist of two parts, one internal and one external, for simulating the operation of the air conditioner. The indoor unit of the air conditioner is installed indoors and the outdoor unit is installed outdoors. Both environments can be conditioned, ie adjusted to the desired dry and wet thermometer temperatures. Once these temperatures have been achieved, the air flow generated by the indoor unit and the change in ambient temperature must be found for capacity measurement. The

indoor unit is therefore connected to an air flow measuring system and the capacity is thus calculated.

Since the parameters were in different units, normalization was performed. The minimum maximum normalization method was used and efficiency was not included in the normalization since the yield value was a unitless magnitude.

In the third chapter, the general definition of machine learning and its adaptation to the refrigeration cycle are examined. Machine learning methods were examined. Machine learning is a method that counts the ability to make inferences about data that it has never seen before by using existing data. Machine learning has a wide range of applications and contributes to optimization problems.

Machine learning is generally used to regulate the operating algorithms of air conditioners in the cooling cycle. Based on user choices, it has the goals of predicting the next choices, increasing energy efficiency and increasing user comfort.

Within machine learning methods, regression examines the relationship between inputs and outputs. Generates regression equations for how output changes with inputs. Thus, based on the previously given data, the output is estimated for a data that it has not seen before. The random forest method is actually a type of regression and creates a forest by creating decision trees and predicts the best output from this forest. Artificial neural networks have a structure that mimics the human brain and neurons. Using the examples given to ann, ann performs the learning function like a human brain. In artificial neural networks, each network has its own weight value and the information it stores is proportional to the weight of the neural network. It is a method that can work easily even with incomplete and uncertain information.

After introducing the machine methods, how to use these methods in the cooling cycle is explained. Firstly, 158 experiments were carried out in 4 different environment conditions, the input value and output capacity of the experiments were normalized and then 80% of these data were learned by machine learning. The remaining 20% of the data is estimated by the machine based on the learned part and is called the test.

As a result, codes were formed for 4 different models: linear regression, polynomial regression, random forest and artificial neural networks, data were learned, tested learning and test results were examined. Random forest method has been determined as the lowest deviation of the test data from the average among these methods.

Then, optimization problem was defined for each of these methods and constraints were entered. In cases where the efficiency will be maximum and the capacity will remain within the specified ranges, both the parameters that will provide these points and the results that will be the output of these parameters are obtained for 4 different methods.

Since the obtained parameter values are normalized values, these values are first converted back to their real values. They are then rounded to the most appropriate integer that can be set. Experiments were repeated in the given values of the parameters. The actual results of the experiments were compared with the predicted results of the models.

Among the models, random forest method gave the best seasonal energy efficiency value. While the SEER value calculated by the combination of the best results of the experiments performed before using the machine learning methods was 6,689, this value increased to 6,767 after the machine learning. Annual energy consumption decreased from 183.14 kW to 181.02 kW per product.

In order to improve this study, which uses machine learning methods in energy efficiency optimization, the results obtained will be placed in the learning data again and the results will be improved gradually.



1. GİRİŞ

Enerji günümüzde ve gelecekte her zaman gerekli olacak bir ihtiyaçtır. Bununla birlikte sera gazlarının salımının en büyük kaynaklarından birini de enerji tüketimi oluşturur. Enerji tüketimini azaltmanın en pratik yoluysa enerji verimliliğidir. Enerji verimliliği daha az girdi ile aynı çıktıyı elde etmek demektir.

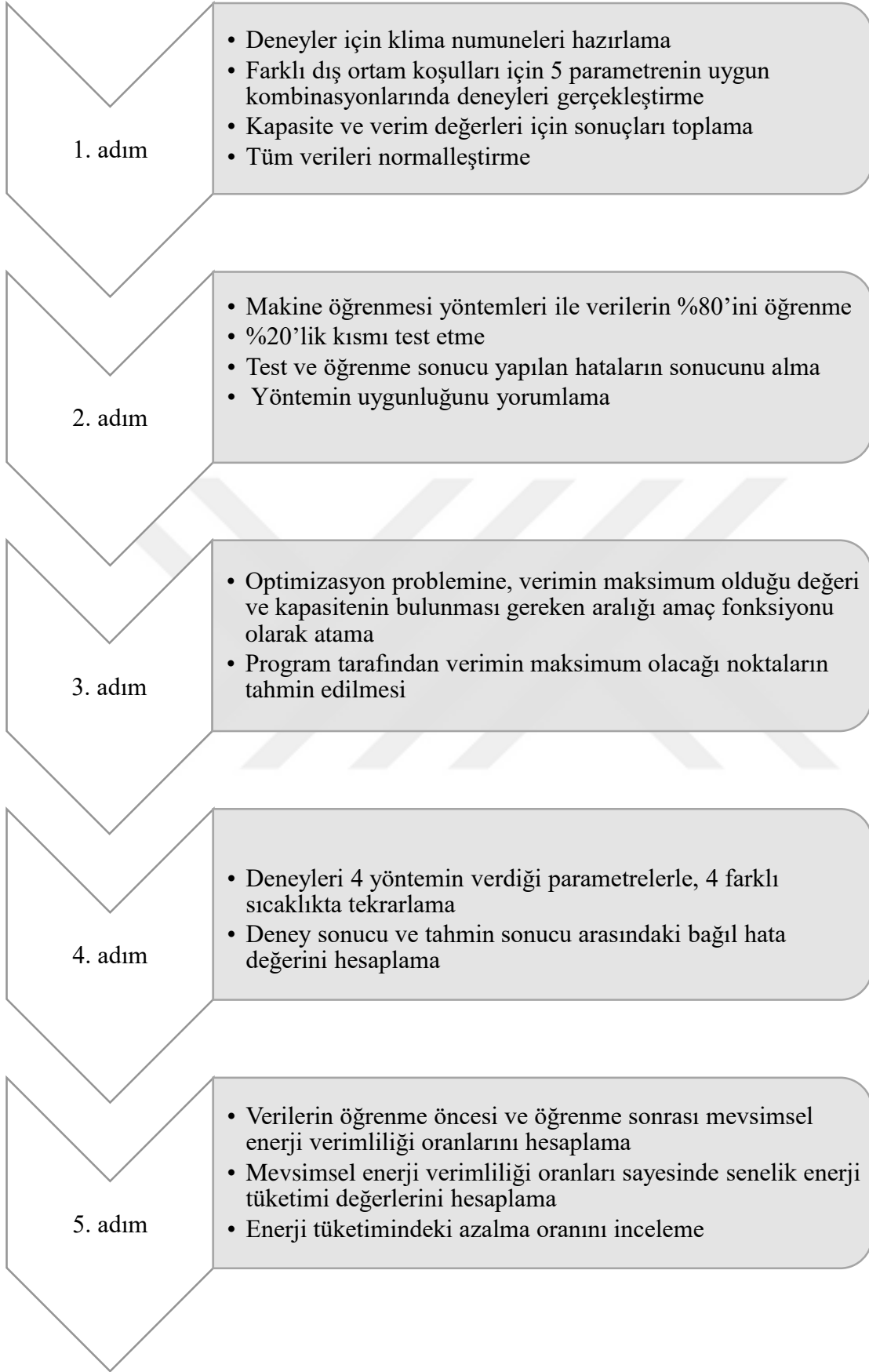
Avrupa Birliği iklim ve enerji yönünde 20-20-20 hedefleri olarak, yenilenebilir enerji kaynakları kullanımının %20 artırılması, sera gazı salımının %20 azaltılması ve enerji verimliliğinin %20 artırılması hedeflerini koymuştur. Bununla ilgili olarak da ErP (Enerji ile ilgili ürünler) yönetmeliğini 1 Ocak 2013'te yürürlüğe almıştır. Yönetmelik enerji verimliliğinin artırılması ve sera gazı salımının azaltılmasını amaçlar. ErP regülasyonu ile mevsimsel verimlilik kavramı hayatımıza girmiş bulunmaktadır. Mevsimsel verimlilik kavramı ısıtma ve soğutma yapan cihazların verimliliğini tüm yıl boyunca çalışma çeşitliliği durumlarını göz önünde bulundurarak hesaplamaları içerir. SEER, mevsimsel enerji verimliliği, klimalar için soğutmada mevsimsel enerji verimliliği ölçüsüdür.

Mevsimsel enerji tüketimi, klimalar için deklare edilen senelik güç tüketimi değerini doğrudan etkiler. Klimalar için verimlilik artışı, mevsimsel enerji verimliliği artışı demektir. Mevsimsel enerji tüketimi değerinde yapılacak olan iyileştirme ile senelik güç tüketimi azaltılmış olacaktır.

Şekil 1.1'de bu çalışmada yürütülecek olan akış diyagramı verilmiştir.

1.1 Tezin Amacı

Gün geçtikçe kullanıcı sayısı artan ev tipi klimaların enerji verimliliği, elektrik tüketimi ile doğrudan bağlantılıdır. Klimanın enerji verimliliğini etkileyen iki temel faktör bulunmaktadır; donanım ve yazılım. Bu tezde, elektrik ve dolayısıyla enerji kaynaklarının tüketimini azaltmak amacıyla, sabit bir donanım için yazılımsal algoritma değişikliği ile klima verimliliği, yapay zekâ yöntemleri ile arttırılmaya çalışılacaktır.



Şekil 1.1: Çalışmanın akış diyagramı.

1.2 Literatür Araştırması

Literatürde bulunan soğutma çevrimi parametre optimizasyonu ve yapay zekâ ile optimizasyon yöntemleri incelenmiştir.

1.3 Hipotez

Bu tezin amacı yapay zekâ yöntemlerini kullanarak sabit bir donanımda klima tasarımı için yazılım algoritmasını optimize etmeye çalışmaktır. Bu optimizasyon sonucu elde edilen yeni sonuçların, önceki duruma göre farkı irdelenecektir.

H_0 : Yapay zekâ yöntemleri kullanılarak yapılan klima algoritma optimizasyonu ile, enerji verimliliğinde anlamlı bir değişiklik gözlenmez.

H_a : Yapay zekâ yöntemleri kullanılarak yapılan klima algoritma optimizasyonu ile, enerji verimliliğinde anlamlı bir değişiklik gözlenir.

1.4 Amaç Fonksiyonu

Amaç fonksiyonu denklem 1.1’de verilmiştir. Kapasite $\in \{Min, Max\}$ ve max verim olacak şekilde, kısıtlar denklem 1.2, denklem 1.3 ve denklem 1.4’te yer almaktadır.

$$Min z = 0,25 d_1^- + 0,25 d_2^+ + 0,5 d_3^- \quad (1.1)$$

Öyle ki,

$$G_1: f(\text{iç ünite fan hızı, dış ünite fan hızı, frekans, eev açıklığı, dış ortam sıcaklığı}) + d_1^- - d_1^+ = max \quad (1.2)$$

$$G_2: f(\text{iç ünite fan hızı, dış ünite fan hızı, frekans, eev açıklığı, dış ortam sıcaklığı}) + d_2^- - d_2^+ = min \quad (1.3)$$

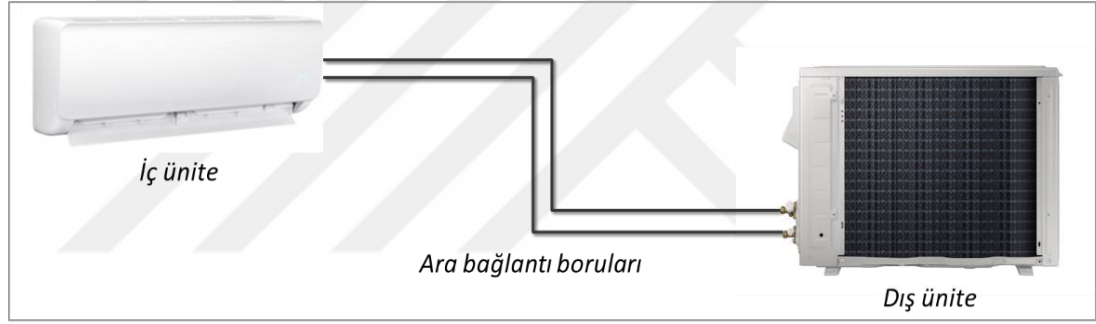
$$G_3: g(\text{iç ünite fan hızı, dış ünite fan hızı, frekans, eev açıklığı, dış ortam sıcaklığı}) + d_3^- - d_3^+ = max \quad (1.4)$$



2. SOĞUTMA ÇEVİRİMİ

2.1 Teknoloji Tanımı

Klima, ısıtma, soğutma, nemlendirme ve hava sirkülasyonu yapma gibi işlevlerin yapılmasına olanak veren bir iklimlendirme cihazıdır. Farklı kullanım amaçları için farklı tipleri mevcuttur ve evlerde kullanım için sıklıkla ev tipi klimalar kullanılır. Bu tip klimalar genellikle bir iç, bir dış ve bunların birbirleriyle bağlantısını sağlayan ara bağlantı borularından oluşurlar. Şekil 2.1’de ev tipi klima sistemi için bir örnek gösterilmiştir.



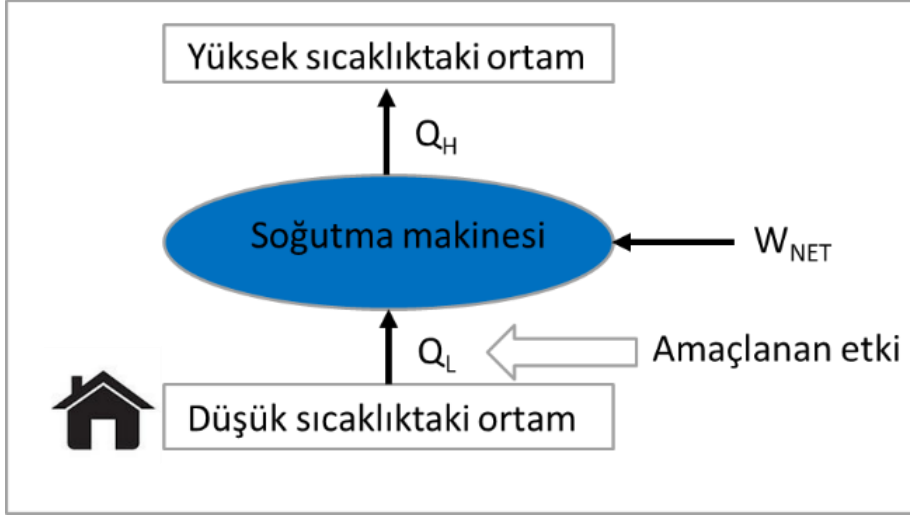
Şekil 2.1: Bir iç ve bir dış üniteden oluşan ev tipi klima örneği.

Klimalar, soğutma işlemi yaparken soğutma makinesi prensibiyle çalışırlar. Bir soğutma makinesi için çalışma prensibi Şekil 2.2’de gösterilmiştir.

Isı geçişi, doğal olarak, azalan sıcaklık yönündedir. Isı geçişinin yüksek sıcaklıktaki bir ortama doğru gerçekleşebilmesi için soğutma makinesi kullanılabilir. Bu yönlü ısı geçişi gerçekleşmesi için sisteme dışarıdan bir iş girdisi gerekir; klimalar için bu iş kompresör elemanı ile sağlanır.

Isı geçişi, doğal olarak, azalan sıcaklık yönündedir. Isı geçişinin yüksek sıcaklıktaki bir ortama doğru gerçekleşebilmesi için soğutma makinesi ya da ısı pompası kullanılabilir.

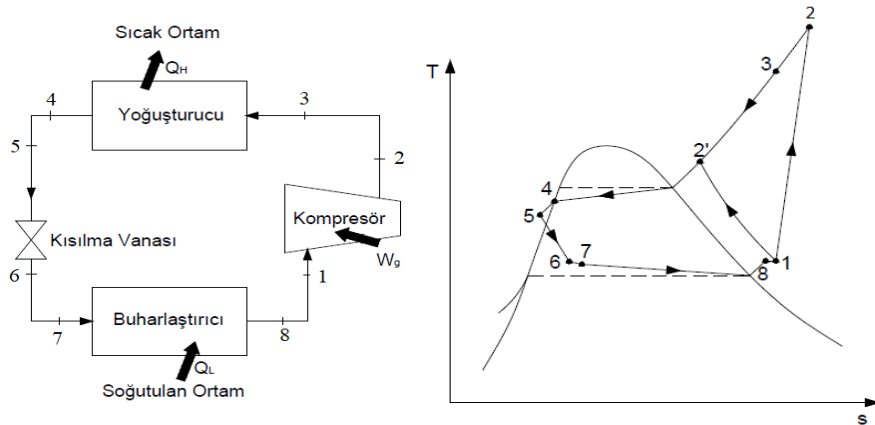
Soğutma makineleri ve ısı pompaları aynı çevrime göre çalışırlar fakat kullanım amaçları farklıdır.



Şekil 2.2: Soğutma makinesi çevrimi.

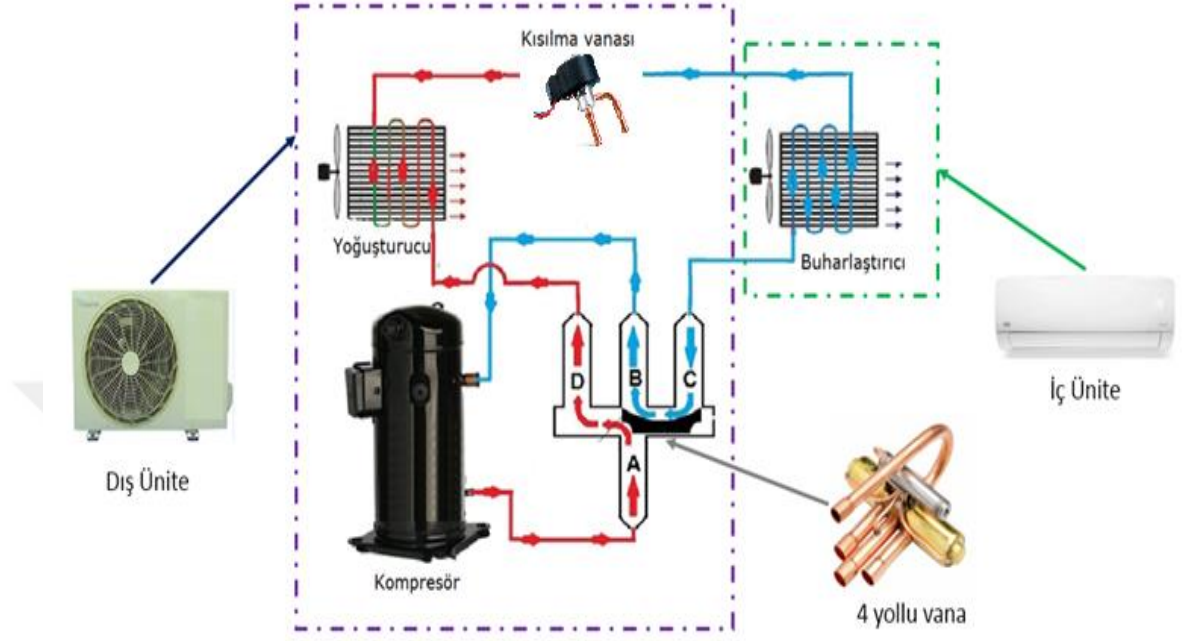
2.2 Klima Çalışma Prensibi ve Komponentleri

Klima temelde buhar sıkıştırımlı soğutma çevrimi ile çalışır. Bu çevrim termodinamik olarak en ideal şekilde ters Carnot çevrimine dayanır, fakat ters Carnot çevriminin gerçekleşmesi 2-3 aşamasında kompresörde sıvı-buhar karışımının sıkıştırılmasını ve 4-1 aşaması, sıvı oranı yüksek bir karışımın genişlemesini gerektirir. Bu güçlükler, çevrimin doyma eğrisi dışında gerçekleşmesiyle aşılabılır fakat bu kez de sabit sıcaklıkta ısı geçişi mümkün olmayacaktır. Bu nedenle Ters Carnot çevrimi uygulamada gerçekleşemeyecek fakat soğutma çevrimleri için bir standart oluşturacaktır. (Boles ve Çengel, 2015). Şekil 2.3'te gerçek buhar sıkıştırımlı bir çevrim için ilerlenen yollar ve sıcaklık entropi grafiği verilmiştir.



Şekil 2.3: Gerçek buhar sıkıştırımlı çevrim şeması ve sıcaklık entropi diyagramı.

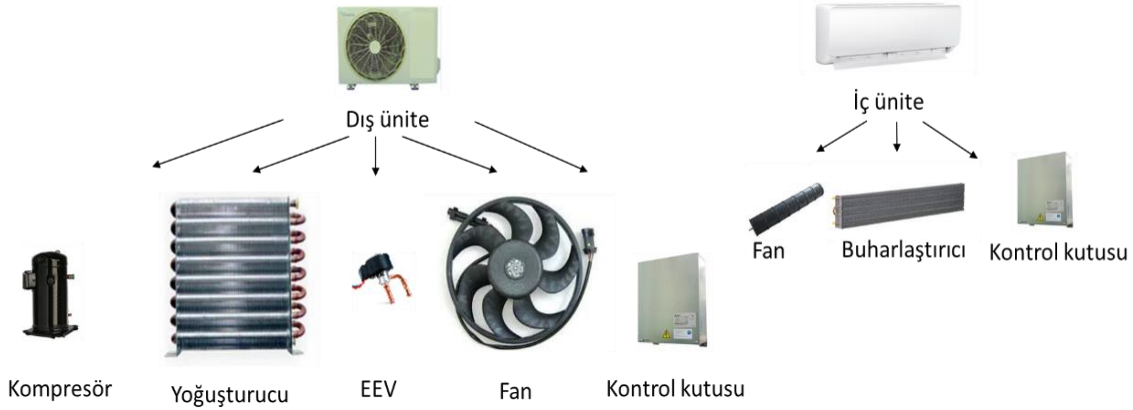
Gerçek buhar sıkıştırımlı çevrimde temel elemanlar kompresör, yoğuşturma ve buharlaştırma işlemlerini yapacak ısı değiştiriciler ve kısılma vanasından oluşur. İdealden farklı olarak, akış sürtünmelerini ve ısı kayıplarını içerir.



Şekil 2.4: Klima komponentleri ve soğutucu akışkanın izlediği yol.

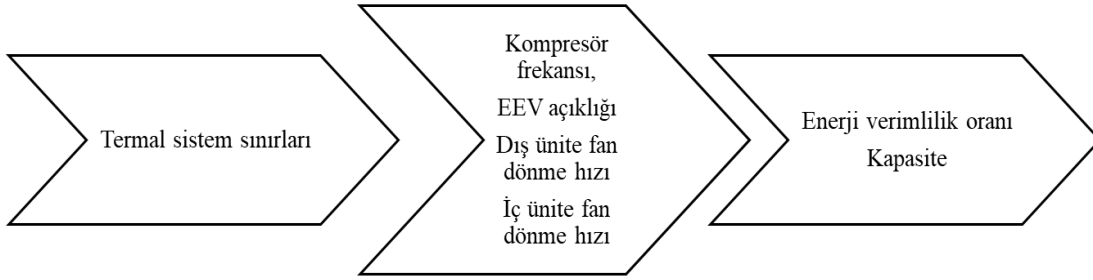
Soğutma çevrimi yapan bir klimada, soğutucu akışkanın izlediği yol Şekil 2.4'te verilmiştir. Kompresörde sıcaklığı ve basıncı artırılan soğutucu akışkan dört yollu vanadan geçerek yoğuşturucuya geçer. Yüksek sıcaklıktaki ortama, dış ünite fanının da yardımı ile ısı verir daha sonra kısılma vanasında basınç ve sıcaklığı düşürülen soğutucu akışkan buharlaştırıcıya gelerek düşük sıcaklıktaki ortamdan, iç ünite fanını da yardımıyla ısı çeker. Böylelikle buharlaştırıcının bulunduğu iç ortamda hava sıcaklığı düşürülmüş olur. Kompresör sıkıştırma oranı, elektronik kısılma vanasının açıklığı yani soğutucu akışkanın debisi, iç ve dış ünitelerde bulunan fan rpm değerleri klimanın yazılımı tarafından belirlenir. Yüksek ortam sıcaklıklarında daha fazla soğutma yükü oluşacağından daha yüksek sıkıştırma oranları yani daha yüksek kompresör frekansları gerekecektir. Bununla birlikte ısıyı yoğuşturucu ve buharlaştırıcıdan atmak zorlaşacağından daha yüksek fan dönme hızları gerekli olacaktır. Sistemin dengelenmesi için ise kısılma miktarının artırılması yani soğutucu akışkanın debisinin düşürülmesi gerekecektir. Klima performansını, klimanın donanımı ve yazılımı belirleyecektir. Donanımsal parçalar Şekil 2.5'te belirtilmiştir.

Klima yazılımı iç ve dış ünite de bulunan kontrol kutuları içerisindeki mikro çiplerde saklıdır.



Şekil 2.5: Klima donanımsal parçaları.

Klimanın yazılımıyla, kompresör frekansı, EEV (elektronik genişleme valfi) açıklığı, dış ünite fan dönme hızı ve iç ünite fan dönme hızı parametreleri belirlenebilir. Sabit donanımdaki bir klima için verimlilik ve kapasite değerleri bu parametrelerin kombinasyonlarıyla belirlenmektedir. Şekil 2.6’da gösterildiği üzere, termal sistem sınırları belirlendikten sonra parametreler belirlenir ve bu parametrelerdeki kapasite ve enerji verimlilik oranı değerleri deneysel olarak hesaplanır ve ölçülür.



Şekil 2.6: Klima sistemi optimizasyon akış şeması.

2.3 Mevsimsel Enerji Verimliliği Kavramı

SEER (Seasonal energy efficiency ratio), klimanın soğutma sezonundaki genel enerji verimliliğidir, klimanın farklı yüklerde çalışmasıyla, bu hava sıcaklıklarının mevsim içindeki dağılımının da hesaba katılmasıyla ortaya çıkar. SEER hesaplaması, TS EN 14825 (2019) Standardında belirtildiği üzere yapılır, standartta hesaplama için gerekli ortam koşulları ve formülizasyonlar belirtilir.

SEER değeri, klima için referans yıllık soğutma değerinin yıllık elektrik tüketimi değerine bölünmesiyle hesaplanır, denklem 2.1’de belirtilir.

$$SEER = Q_C / Q_{CE} \quad (2.1)$$

Referans yıllık soğutma değeri, Q_C ile gösterilir denklem 2.2’de görüleceği üzere $P_{desingC}$, soğutma tasarım yükünün, soğutmada geçirilen eşdeğer saat sayısı olan H_{CE} ile çarpılmasıyla elde edilir.

$$Q_C = P_{desingC} \cdot H_{CE} \quad (2.2)$$

Q_{CE} , yıllık elektrik tüketimi değeri, klimanın aktif mod, cihazın termostatının kapalı olduğu mod, hazır bekleme modu ve kapalı mod çalışma modundaki elektrik tüketimlerinden oluşmaktadır, denklem 2.3 ile belirtilmiştir. P_{TO} , termostatın kapalı olduğu durumdaki güç tüketimidir, bu da soğutma ya da ısıtma konumunun açık olduğu fakat operasyonel olarak hiçbir soğutma ya da ısıtma yükünün bulunmadığı durum demektir. Termostat kapalı modu iç ve dış ortam sıcaklıklarına göre oluşmaktadır. Termostat kapalı konumunun gerçekleşiyor olabilmesi için, soğutma modunda dış ortamın 16 °C’nin altında olması ya da sistemin çalışmıyor olması gerekmektedir. H_{TO} bu durumdaki saatlerin sayısıdır. Bu tezde standart ölçümler sırasında dış ortam sıcaklıkları 16 °C’ni altına düşmemektedir. P_{CK} , karter ısıtıcısının çalıştığı durumdaki güç tüketimidir, H_{CK} bu durumda geçirilen saatlerin sayısıdır. Kartier ısıtıcısı, genellikle, kompresör etrafına sarılarak, kışın klimanın dış ünitesinde bulunan kompresörün zarar görmemesi için içinde bulunan yağın donmasını engelleyerek bir elektrikli ısıtıcı gibi çalışmaktadır. Çoğunlukla ev tipi klimalarda kullanılmaz. Bu tezde kullanılacak klimada karter ısıtıcısı bulunmadığı için P_{CK} yok sayılacaktır. P_{SB} hazır bekleme konumunda klimanın çektiği güç değeridir. Klima zamanlayıcı ayarı ya da kumanda ile açılabilir durumdayken çekilen güç miktarını belirtir. H_{SB} ise bu durumda geçirilen saat sayısıdır. P_{OFF} klimanın kapalı moddaki güç tüketimidir. Kapalı mod için güç tüketimi yok kabul edilir, H_{OFF} klimanın bu modda geçirdiği saat sayısıdır.

$$Q_{CE} = (Q_C / SEER_{on}) + P_{TO} \cdot H_{TO} + P_{SB} \cdot H_{SB} + P_{CK} \cdot H_{CK} + P_{OFF} \cdot H_{OFF} \quad (2.3)$$

Her fonksiyonel durumda geçirilen operasyon saatleri Çizelge 2.1’de belirtilmiştir.

Çizelge 2.1: Her bir fonksiyonel durumda geçirilen operasyon saatleri.

Aktif Mod	Termostat kapalı mod	Bekleme modu	Karter ısıtıcısı modu	Kapalı mod
H _{CE} (h)	H _{TO} (h)	H _{SB} (h)	H _{CK} (h)	H _{OFF} (h)
350	221	2142	2672	0

SEER_{ON} değeri, denklem 2.4’te gösterildiği gibi hesaplanır.

$$SEER_{ON} = \sum_{j=0}^n \frac{h_j P_c(T_j)}{\frac{P_c(T_j)}{EER_{bin}(T_j)}} \quad (2.4)$$

Kısmi yük oranı, pl olarak gösterilir ve T_j, dış ortam sıcaklığındaki, pl kısmi yük oranı, pl(T_j) olarak gösterilir. Kısmi yük oranı denklem 2.5’te gösterildiği üzere hesaplanır. SEER hesabı için 4 farklı sıcaklıkta, 4 farklı kısmi yük oranı hesaplanacaktır. Çizelge 2.2’de klima için iç ve dış ortam sıcaklıklarına göre hangi kısmi yüklerde çalışılacağı, T_{design}, tasarım sıcaklığı 35 °C için belirtilmiştir. %100 kısmi yükteki deney SEER A şeklinde belirtilir. SEER hesaplaması için gereken 4 deney yapıldıktan sonra geri kalan sıcaklıklardaki yüklerin hesabı için iterasyon yapılır. Bu sıcaklık değerleri için geçirilen süreler Çizelge 2.3’te belirtilmiştir. Tasarım yükü, P_{designc}, soğutma konumunda tasarım sıcaklığında, üretici tarafından deklare edilen değerdir. Denklem 2.6’da gösterilen P_c kısmi yükü tasarım yükü ile kısmi yükün çarpımı ile bulunur.

$$pl(T_j) = (T_j - 16)/(T_{design} - 16) \quad (2.5)$$

$$P_c(T_j) = \frac{T_j - 16}{T_{design} - 16} * P_{designc} \quad (2.6)$$

Çizelge 2.2’de verilen 4 noktada deneyler yapıldıktan sonra bu noktalar için kapasite değerleri ölçülmüş, verim değerleri hesaplanmış olacaktır. 4 noktada deneyler yapıldıktan sonra Çizelge 2.3’teki tüm sıcaklıklar için tek tek kısmi yük hesaplanacaktır.

Çizelge 2.2: Mevsimsel verimlilik hesaplamasında kullanılan kısmi yük ve sıcaklıklar.

Test Kodu	Kısmi Yük Oranı (%)	Dış ortam KT(YT) sıcaklığı (°C)	İç ortam KT(YT) sıcaklığı (°C)
SEER A	100	35 (24)	27(19)
SEER B	74	30 (20,1)	27(19)
SEER C	47	25 (16,2)	27(19)
SEER D	21	20 (12,3)	27(19)

Çizelge 2.3: SEER hesaplamasında kullanılan bin saatleri sıcaklıklar ve geçirilen süreler.

j	T _j (°C)	H _j (h)
1	17	205
2	18	227
3	19	225
4	20	225
5	21	216
6	22	215
7	23	218
8	24	197
9	25	178
10	26	158
11	27	137
12	28	109
13	29	88
14	30	61
15	31	39
16	32	31
17	33	24
18	34	17
19	35	13
20	36	9
21	37	4
22	38	3
23	39	1
24	40	0

20 °C'ye kadar olan tüm sıcaklıklarda verim değeri 20 °C'deki ile aynı olarak kabul edilir. 20 °C üzerindeki sıcaklıklar için 25 °C olan ikinci ölçüm noktasına kadar olan tüm sıcaklıklar için bu iki ölçüm noktası arasında iterasyon yapılır.

Tüm bin sıcaklıkları için çekilen güç değeri $P_c (T_j)$ hesaplandıktan sonra Çizelge 2.3'te bulunan tüm bin sıcaklık süreleri birbirleriyle çarpılır ve daha sonra o

sıcaklıktaki EER(T_j) değerine bölünür. Böylelikle SEER_{ON} değeri için gerekli tüm değerler bulunmuş olacaktır. Daha sonra da SEER değeri hesaplanacaktır.

2.4 Deney Parametreleri ve Deney Düzenegi

2.4.1 Deney Parametreleri

SEER hesabı yapılabilmesi için 4 farklı dış oda sıcaklığında aşağıdaki parametrelerin farklı seviyeleri denenmiştir.

Kompresör frekansı, klima dış ünitesinde bulunan kompresörün saniyedeki dönme hızıdır ve birimi Hz'dir.

Kompresör frekansı, klima dış ünitesinde bulunan kompresörün saniyedeki dönme hızıdır ve birimi Hz'dir.

EEV (elektronik genleşme valfi dış ünite fan dönme hızı, dış ünite yoğuşturucusunun sıcaklığının atılmasını kolaylaştırır, dakikadaki dönme sayısını gösterir. Birimi rpm'dir.

İç ünite fan dönme hızı, iç ünite yoğuşturucusunun sıcaklığının atılmasını kolaylaştırır, dakikadaki dönme sayısını gösterir. Birimi rpm'dir.

$$\text{Senelik Enerji Tüketimi} = H_{CE} * P_d / SEER \quad (2.7)$$

Senelik enerji tüketimi değeri hesabı denklem 2.7 ile verilmiştir.

2.4.2 Soğutma Kapasite ve Verim Hesabı

Klimanın toplam soğutma kapasitesi denklem 2.8 ile hesaplanır. Φ , iç odada ölçülen toplam soğutma kapasitesidir, qml iç ünitenin m³/sn cinsinden hava debisidir. h_{a1} giriş havasına ait entalpi değeridir ve birimi J/kg'dır, h_{a2} çıkış havasına ait hava debisidir ve birimi J/kg'dır. V_n' klimanın nem alma kapasitesinin bir ölçüsü olan nem miktarıdır, birimi m³/kg'dır. W_n havanın nem miktarıdır, birimi kg/kg'dır.

$$\Phi_c = \frac{q_{ml}(h_{a1} - h_{a2})}{V_n'(1 + W_n)} + q_0 \quad (2.8)$$

Ürünün çektiği güç, hesaplama yapılmaksızın wattmetre cihazı ile ölçülerek, kapasite güç oranından performans katsayısı hesaplanır. Soğutma tasarım yükü, tasarımcı

tarafından belirlenir ve Çizelge 2.4'te tasarım yükü belirlenen klimanın kısmi yük koşullarında vermesi gerekli olan kapasite değerleri belirtilmiştir. Minimum sınır kısmi yük oranının getirdiği eşitlikle bulunurken, maksimum sınır bunun 1,07 katı olarak alınır. (TSE, 2019)

Çizelge 2.4: Mevsimsel verimlilik hesaplamasında kullanılan kısmi yük ve sıcaklıklar.

Test Kodu	Kısmi Yük Oranı (%)	Minimum kapasite değeri	Maksimum kapasite değeri
SEER A	100	0,95	1
SEER B	74	0,6	0,7
SEER C	47	0,29	0,34
SEER D	21	0	Sınır yok

2.4.3 Kalorimetre Odaları

Klimalarda kapasite ve güç ölçümleri için kalorimetre odaları kullanılmaktadır. Kalorimetre odaları balans tipi kalorimetre odası ya da psikrometrik tip kalorimetre odası olabilir. Balans tipi odalarda soğutma modunda, klimanın iç ünitesinin soğutma ve nem alma kapasitesi; ısıtma modunda ise klimanın iç ünitesinin ısıtma kapasitesi, ölçüm cihazının (balans tipi kalorimetre odasının) ısı kapasitesi ile dengelenir ve termal girdi ölçülür. Soğutma ve ısıtma modları için aynı durum dış ünite için de geçerlidir ve cihazın dış ünitesinin kapasitesi de balans tipi odanın ısı kapasitesi ile dengelenir. Psikrometrik tip odalarda ise farklı sıcaklık ve nem değerlerine ayarlanabilen iki ayrı odadan oluşan bir yapıda psikrometrik yöntemler kullanılarak klimanın soğutma ve ısıtma kapasiteleri hesaplanarak bulunur.

Bu tezde yapılan ölçümler psikrometrik tip deney odasında yapılmıştır. Psikrometrik tip deney odalarında, klima kapasitesinin hesaplanabilmesi için sıcaklık ölçüm sistemi ve hava debisi ölçüm sistemleri bulunmaktadır. Sıcaklık ölçüm sistemi, havanın kuru ve yaş termometre sıcaklıklarını ölçmek için kullanılır. Hava debisi ölçüm sistemi ise klimanın hava üfleme ağzına bağlanarak, klimanın hava debisinin ölçülmesini sağlar. Şekil 2.8'de psikrometrik bir oda ve odanın elemanları gösterilmiştir. Bu şekilde göre;

1 numara ile belirtilen, klima dış ünitesidir, dış oda olarak belirtilen bölümde bulunur.

2 numara ile belirtilen, klima iç ünitesidir, iç oda olarak belirtilen bölümde bulunur.

3 numara ile belirtilen, klima iç ve dış ünitelerinin hava emiş bölgelerine konan sıcaklık ölçüm sistemi kollarıdır.

4 numara ile belirtilen, sıcaklık ölçüm sistemidir.

5 numara ile belirtilen, klima iç ünitesinin hava debisinin ölçümü için gerekli kanal bağlantısıdır.

6 numara ile belirtilen hava debisi ölçüm sistemine ait giriş basıncı sıfırlama bölümüdür.

7 numara ile belirtilen hava debisi ölçüm sistemine ait hava örnekleyicidir, klima üfleme havası sıcaklıklarının ölçümünün yapıldığı bölümdür.

8 numara ile belirtilen hava debisi ölçüm sistemi basınçlandırma bölümüdür.

9 numara ile belirtilen hava debisi ölçüm sisteminde birinci basınç odasının basıncının sıfırlanabilmesi için gerekli emiş fanıdır.

10 numara ile belirtilen 9 numaralı fana ait motordur.

11 numara ile belirtilen iç ve dış odaların şartlandırılması için gerekli havayı odaya dağıtmak için kullanılan hava sirkülasyon fanıdır.

12 numara ile belirtilen hava işleme ünitesi toz tutucu ızgara filtresidir.

13 numara ile belirtilen hava işleme ünitesi ısıtıcısıdır.

14 numara ile belirtilen hava işleme ünitesi soğutma ünitesidir.

15 numara ile belirtilen hava akışının odaya homojen dağılması için gerekli hava delikleridir.

16 numara ile belirtilen şartlandırılmış havanın akış yönüdür.

17 numara ile belirtilen, sıcaklık ölçüm sistemi bağlantı borusudur.

İç ünite, iç odaya askı aparatlarıyla kurulduktan sonra kanal bağlantısı ile hava debisi ölçüm sistemine izolasyonlu bantlarla bağlanarak montajlanır. Dış ünite dış odaya alınır ve iç ve dış ünite ile iç ünite arasındaki ara bağlantı borularının sıvı ve gaz hattı için ayrı ayrı bağlantıları rakor ve somun bağlantıları ile yapılır. İç ve dış ünite birbiriyle kapalı bir sistem oluşturduğunda içerisinde hava kalmamasına emin olmak için 5.10^{-2} Torr değerine kadar sistem basıncı düşürülür. Vakum yapıldıktan sonra ürünün dış ünitesine soğutucu akışkan şarj edilir. Ürün kurulumu sağlandıktan sonra iç ve dış odalar yapılacak deneyin sıcaklıklarına göre şartlandırılır. Odaların sıcaklıkları stabilize olduktan sonra ürün çalıştırılır ve deney başlatılır. İlk 90 dakika boyunca ürünün stabil hale gelmesi beklenir ve stabilizasyon periyodunun ardından 7 dakikada bir olmak üzere 5 kez sonuç alındıktan sonra bu değerlerin ortalaması alınarak kapasite ve güç değerleri elde edilmiş olunur. Kapasitenin güce bölünmesi ile de verim değeri hesaplanmaktadır.

Ölçüm belirsizliği ise iç ve dış oda kuru termometre sıcaklığı için $0,01$ °C, yağ termometre sıcaklığı için $0,02$ °C, basınç ölçümü için ise oransal olarak $0,01$ mertebesindedir. Bununla birlikte test numunesinin kapasite ve verim hesabının ölçüm belirsizlikleri toplamda % $0,6$ mertebesindedir.

2.5 Deney Sonuçları ve Verilerin Normalleştirilmesi

Veri normalleştirme, her bir özelliğe eşit katkı sağlamak için verilerin ölçeklendiği veya dönüştürüldüğü ön işleme yaklaşımlarından biridir. Makine öğrenmesi algoritmalarının başarısı, sınıflandırma probleminin genelleştirilmiş bir tahmin modelini elde etmek için verilerin kalitesine bağlıdır. Veri kalitesinin iyileştirilmesinde ve ardından makine öğrenmesi algoritmalarının performansında veri normalizasyonunun önemi birçok çalışmada sunulmuştur. (Singh ve Singh, 2019).

Deneyler iki farklı iç ve dış ünitelerden oluşan klima setleri için yapılmıştır. İlk numune için deney düzeneği kurulmuş ve Şekil 2.7’de görüleceği gibi numuneler hazırlanmıştır. SEER değeri hesaplanması için gereken A, B, C ve D testleri toplamda 158 ayrı deney yapılarak, Çizelge A.1’deki şekilde kapasite ve verim değerleri elde edilmiştir.

Deneyler için istikrar periyodu 90 dakika ve veri toplama periyodu 35 dakika olacak şekilde veri toplanmıştır. İç ve dış olarların rejime girme süreleri yaklaşık 30 dakika ile

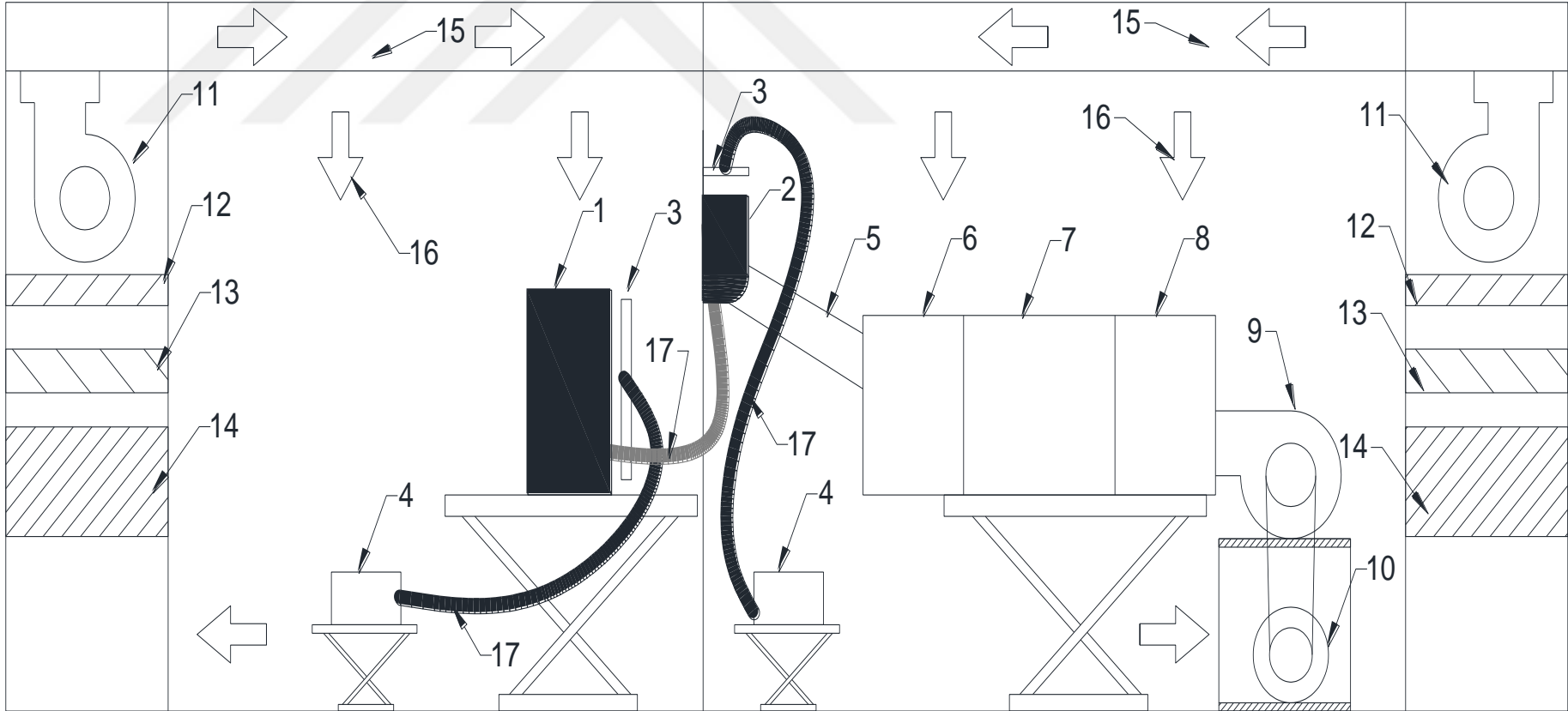
Min-max normalizasyonu, orijinal veriler üzerinde doğrusal bir değişiklik yapar. Değerler verilen aralık dahilinde normalleştirilir. minA ve maxA aralığında yer alan bir A değerinin, new_minA ve new_maxA aralığında eşlenmesi için hesaplama denklem 2.9'daki gibi yapılır. Min max normalizasyonunun yararı, verilerin tamamının belli bir aralık içinde yer almasıdır (Saranya ve Manikandan, 2013).

60 dakika arasında değişmektedir. Uzman görüşleri çerçevesinde, deney sürelerinin uzunluğu göz önünde bulundurularak deney parametreleri en uygun olacak şekilde seçilmiştir. En yüksek performansı gerektiren ve en zorlu koşul olan SEER A deneyinde kompresör frekansları en yüksek olacak konumlarda olacak şekilde seçilmiştir. Bununla birlikte yoğuşturuculardan atılan sıcaklıkların da en fazla olması gerekmektedir. Böylelikle yoğuşturuculardan ısı atılmasını sağlayan fan dönme hızlarının da yüksek kademelerde olması beklenmektedir. Kısılma vanası açıklığı kompresör frekansını dengeleyecek şekilde seçilmiştir.



Şekil 2.7: Hazırlanan numune görselleri.

Bu deneylerde oda sıcaklıkları iç oda için sabit kalacağından yalnızca dış oda sıcaklığı değişkenliği 4 ayrı kademeye belirtilmiştir. Deneydeki değişkenler olan iç ünite fan rpm değeri, dış ünite fan rpm değeri, kompresör frekansı ve dış ortam sıcaklığı değerleri ürünün vereceği kapasiteyi ve gücü etkileyecektir. Kapasite ve gücün oranından çıkan verim ve güç değerlerini iki ayrı çıktı olarak belirtmek gereksiz olduğundan çıktılar arasında güç değeri belirtilmemiştir.



Şekil 2.8: Psikrometrik tip kalorimetre odası ve odanın elemanları.

Z skoru normalizasyonu, sıfır ortalama normalizasyonu olarak da adlandırılır. Burada veriler ortalama ve standart sapmalara göre normalleştirilmiştir. Ondalık ölçek normalizasyonunda ise veriler arasında en büyük verinin basamak değerini, 10'un kuvveti olarak alıp, tüm verileri bu sayıya bölerek normalize eder. Saranya ve Manikandan'ın veri setlerinin gizliliğini korumak için yaptıkları çalışmada görüldüğü üzere min-max normalizasyonu diğer iki normalizasyonla karşılaştırıldığında çok daha az sayıda sınıflandırma hatası gösterir.

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A \quad (2.9)$$

Bu çalışmada parametrelerin birimleri birbirinden farklı olduğundan normalizasyon yapılacaktır ve normalizasyon aralığı 0 ile 1 arasında olacak şekilde seçilmiştir. new_maxA değeri 1 ve new_minA değeri 0 olacak şekilde tüm giriş parametreleri ve çıktıda kapasite değeri için normalizasyon formülü uygulanır. Verim çıktısı hali hazırda birimsiz olduğundan normalizasyon gerektirmemektedir.

3. SOĞUTMA ÇEVİRİMİNDE MAKİNE ÖĞRENMESİ

3.1 Makine Öğrenmesi

Bilgisayar aracılığıyla bir problemi çözebilmek için algoritmalara ihtiyaç duyulur. Algoritma, bir girdiyi, çıktıya dönüştürebilmek için gerekli bir dizi yönergeyi takip eder. Makine öğrenmesinde, söz konusu makine olan bilgisayarın öğrenebilmesi için öğrenme girdilerine ihtiyacı vardır. Bir problemi çözmek için çeşitli algoritmalar kullanılabilir ve en etkili yöntemi ararken en az sayıda adım ve en hızlı çözüme götüren aranabilir. Makine öğrenmesinde, süreci tamamen öğreniyor olamayız fakat uygun ve yararlı denebilecek bir yaklaşım yapabiliyor oluruz (Alpaydin, 2009).

Makine öğrenmesi kullanılmaya başlandığından bu yana optimizasyon algoritmalarını kullanmış bunun yanı sıra optimizasyon problemlerine de katkıda bulunmuştur. Optimizasyon yaklaşımları, geniş uygulanabilirlikleri ve teorik özellikleri sayesinde, makine öğrenmesi alanında önemli bir yer almıştır. (Sra, Nowozin ve Wright 2012)

Makine öğrenmesi, eldeki verilerden bilgi edinme hakkında, istatistik, yapay zekâ, bilgisayar, tahmin analizi ve istatistiksel öğrenme gibi alanların kesişiminde olan bir alandır. Makine öğrenmesi ile verilen girdi ve çıktı setlerinden, daha önce öğrenilmemiş olan girdilerin çıktıları tahmin edilebilir. Bu noktada verilerin seçimi ve ana kütleli temsili önem arz eder. Python, birçok veri bilimi uygulaması için ortak dil olmuştur. Python veri işleme, istatistik, görüntü işleme ve daha birçok alanda kütüphaneler içerir. (Müller ve Guido 2016).

3.2 Soğutma Çevriminde Makine Öğrenmesi

Makine öğrenmesinin soğutma çevrimine ve daha geniş kapsamlı olarak iklimlendirme cihazlarına olan etkilerini araştıran makaleler incelenmiştir.

Domanski ve Yashar, kanatlı borulu kondanserin devre tasarımı ile kapasiteyi maksimize edecek bir modül geliştirmek için çalışmışlardır. Bu çalışmada yöntem

olarak genetik algoritmalar kullanılmış bu sayede model ile manuel olarak tasarlanan devrelere eş ya da daha yüksek kapasiteli devreler elde edebilmişlerdir (2007).

Huang ve Lam, çalışmalarında HVAC sisteminin optimum performansta çalışabilmesi için PID denetleyicinin otomatik olarak çalışmasını sağlayacak bir sistem için genetik algoritmalar yöntemi kullanmışlardır. Darwin'in doğal seçiliminin mekaniğine dayanan arama prosedürleri olan genetik algoritmalar, karmaşık problemler alanlarda en uygun çözümlere ulaşmada sağlam ve etkili oldukları kanıtlanmış olduklarından kullanılmaktadır. HVAC sistemleri için modül kullanılarak PID kontrol cihazlarının otomatik olarak ayarlanması için yararlı olduğunu göstermişlerdir. Bu çalışma tasarım optimizasyonundan ziyade bir çalışma optimizasyonu niteliğindedir (1997).

Ghahramani, Karvigh ve Becerik-Gerber yaptıkları çalışmada HVAC sistemlerinin enerji kullanımını optimize etmek için ya sistemin matematiksel modeline ya da çok fazla kullanma verisine ihtiyaç duyulacağını belirtmişlerdir. Geçmiş veriler olmadan en iyi ayarları öğrenmeye başlayan ve HVAC işlemlerini optimize eden, modelsiz bir kontrol yöntemi amaçlamışlardır. Bu çalışmada adaptif hibrit metahüristik yöntemler, makine öğrenmesi yöntemleri kullanılmıştır. Bu çalışma da tasarım optimizasyonundan ziyade sistemin çalışmasını optimize eder.

Zhou ve Wu, çalışmalarında HVAC sistem kontrolü için yerel bir kontrol yöntemi geliştirmeye çalışmış bunun için de K-ortalama kümeleme yöntemini klimalı odanın farklı bölgelere ayrılması için kullanmışlardır. Popüler bir makine öğrenmesi türü olan aşırı derin öğrenme yöntemi kullanılmıştır. (2015)

Zhang et al, gerçek yaşam koşullarının gereksinimlerini karşılayabilen bir talep yanıtı ve ev enerji yönetimi sistemi oluşturmak için makine öğrenmesi, optimizasyon ve veri yapısı tasarımını birleştiren disiplinlerarası bir mekanizma geliştirmeye çalışır, HVAC'de enerji tüketim modelini öğrenmek için bir öğrenme sisteminin nasıl tasarlanması gerektiğini, en uygun talep yanıt politikalarını oluşturmak için öğrenme mekanizmasının optimizasyon teknikleriyle nasıl bütünleştirileceğini araştırır. (2016)

Kusiak, Tang ve Xu'nun yaptığı çalışmada HVAC sisteminin öngörücü bir modeli veri madenciliği algoritmaları ile türetilmiştir. İç ortam hava kalitesi korunurken enerjiyi en aza indirmek için çok amaçlı bir optimizasyon modeli geliştirilmiştir. Bu modelin çözümleri evrimsel bir hesaplama algoritmasıyla türetilmiş kontrol sisteminin noktalarıdır. Kontrol edilebilir giriş değişkenleri enerji kullanımını azaltmak için

üretir. Evrimsel hesaplama algoritması tarafından üretilen sonuçlar, kontrol stratejisinin bir HVAC sisteminin operasyonlarını optimize ederek enerji tasarrufu sağladığını gösterir (2011).

Mirinejad, Welch ve Spicer, HVAC sisteminde termal konforu artırıp enerji tüketimini azaltmayı temel hedef olarak almış ve bunun için kullanılan yöntemleri incelemiştir. Yapay sinir ağları ve genetik algoritmalar incelenmiştir (2012).

3.3 Delphi Tekniği

Delphi tekniği Soğuk Savaş Döneminde, Delphi, Sovyetler Birliği'nin olası saldırı riskini belirlemek için US Rand Corporation tarafından geliştirilmiştir. Bu yöntem neredeyse yarım asırdan beri var olan uzmanların görüşlerine dayanarak karar verme esasına dayanan popüler bir tekniktir. Uzmanların kontrollü görüş geri bildirimlerini içeren bir dizi yoğun anket aracılığıyla en güvenilir görüş birliğinin elde edilmesi hedeflenir (Dalkey ve Helmer, 1962).

Delphi tekniği, kendi uzmanlık alanındaki katılımcılardan veri toplamak için yaygın olarak kullanılan ve kabul gören bir yöntemdir. Bu teknik, gerçek dünyayla ilgili belirli bir konuda fikir birliğine varmayı hedefleyen bir, grup iletişim süreci olarak tasarlanmıştır. Delphi süreci, program, planlama, ihtiyaç değerlendirmesi, politika belirleme ve çeşitli alternatifler geliştirmek için kaynak kullanımı, temel varsayımları araştırmak veya ortaya koymak ve geniş kapsamlı bir konudaki yargıları ilişkilendirmek gibi çeşitli çalışma alanlarında kullanılmıştır (Hsu & Sandford, 2007).

İlgili uzmanlara danışırken risk faktörlerinin olası sonuçlarını tahmin etmek için kullanılır. Delphi Tekniği çeşitli projelerin yönetimi alanında neredeyse yarım yüzyıldır teknik ve bilimsel araştırmalarda yaygın olarak kullanılmıştır (Markmann et al., 2013).

Bu çalışmada Delphi yöntemi ile deney ve çalışma sayısı yapılan tahmin sayısını minimuma indirmek için uzman görüşleri alınmıştır.

3.4 Doğrusal ve Çok Terimli Regresyon

Regresyon, değişkenlerin birbirine bağlılığı ile ilgilendirir. Bir çıktının, girdiye ne kadar bağlı olduğu sorusunu sorar. Regresyon analizi birçok araştırma projesinin merkezinde yer almıştır. Birçok istatistiksel analiz yönteminde olduğu gibi regresyon analizinde

de gözlenen verinin mümkün olduğunca basit ve kullanışlı olacak şekilde özetlenmesi amaçlanır (Weisberg, 2005).

Çok terimli regresyon esasen bir doğrusal regresyon türüdür. Doğrusal regresyonda bağımlı ve bağımsız değişken arası ilişkinin doğrusal olması beklenirken çok terimli regresyonda bu koşul aranmaz. Daha karmaşık ilişkiler için kullanılabilir. (Agarwal, 2018)

Çoklu regresyonun iki temel kullanım amacı bulunmaktadır, bunlar Bu tezde, regresyon analizi, girdi ve çıktı verilerinin birbirini etkilediği bilindiğinden bu etkileşimin modelini kurabilmek adına kullanılmıştır.

3.5 Rassal Orman Yöntemi

Rassal ormanlar, ilk olarak Breiman tarafından 2001 yılında ortaya konmuştur, sınıflandırma ve regresyon problemleri için oldukça popüler ve verimli bir algoritmadır. Rassal ormanların ilkesi, öğrenme örnekleminde gelen birkaç örneği kullanarak oluşturulan birçok açıklayıcı ağacı birleştirmek ve açıklayıcı değişkenlerden sonra her bir düğümde rastgele seçim yapmaktır. (Genuer, Poggi ve Tuleau-Malot, 2010)

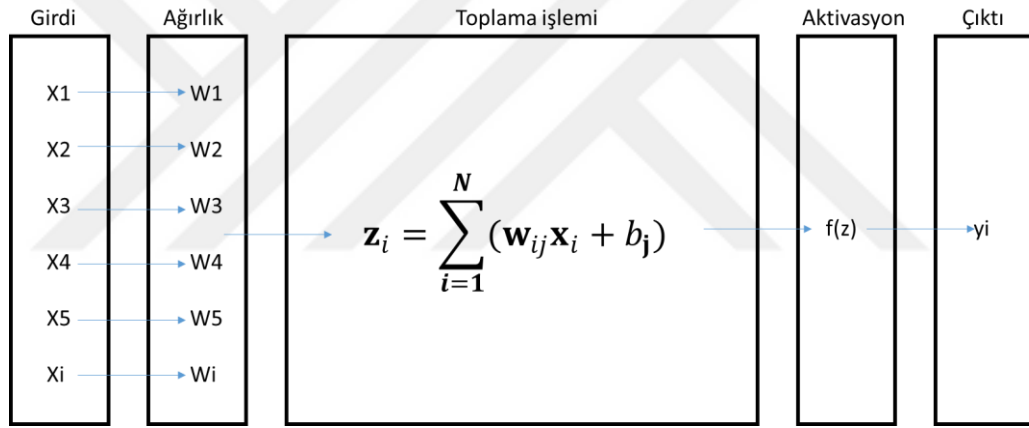
Rastgele ormanlar, ağaç kestiricilerin bir birleşimidir, her ağaç bağımsız bir şekilde ve ormandaki tüm ağaçlar için aynı dağılıma sahip olan bir rastgele vektör numunesinin değerlerine bağlıdır. Genelleme hatası ormandaki ağaç sayısı arttıkça limite doğru yakınsar. Genelleme hatası ormandaki ağaçların gücü ve onların arasındaki korelasyona bağlıdır (Breiman, 2001).

3.6 Yapay Sinir Ağları

Yapay sinir ağları, öğrenmiş olduğu veriler sayesinde girdi ve çıktılar arasında ilişkiler kurarak, önceden görmemiş olduğu verilerle ilgili de sonuca varabilen sistemlerdir. İnsan beynini taklit ederek öğrenme işlevini kendine verilen örnekler aracılığı ile yapar. Her bir yapay sinir ağının kendine özgü bir ağırlık değeri vardır, w_i ile gösterilir. Sinir ağında bilginin ne kadarının mevcut olduğu bu ağırlık fonksiyonunda bulunur. Yapay sinir ağları kendisine verilen bilgiler belirsiz ya da eksik olsa dahi, adaptasyon yetenekleri çok yüksek olduğundan rahatlıkla çalışabilir (Öztemel, 2003).

Yapay sinir ağıları, birden fazla methodu içinde bulundurduğu için çok fazla durum için uygulanabilirlikleri bulunur. Karmaşık bir problemin çözümü için yapay sinir ağlarının sunduğu olasılıklar ile birçok farklı adım izlenebilir (Zupan, 2003).

Yapay sinir ağıları, eğitim verilerini tanımlayan kuralları öğrenme ve daha önce öğrenilmiş olan bilgilerden yeni kalıplara cevap verme yeteneğine sahiptir. Sinir ağıları, birkaç nöronun veya bağlantının kaybedilmesi, bilgi işlemede çok sayıda nöron ve bağlantı içerdiğinden davranışlarını önemli ölçüde etkilememesi anlamında hataya dayanıklıdır. Yapay sinir hücresi, yapay sinir ağının temel bileşenidir. Şekil 3.1’de gösterildiği üzere bir yapay sinir hücresinin bileşenlerini, x ile belirtilen girdi sinyalleri, w ile belirtilen sinaptik bağlantıların ağırlığını gösteren ağırlık fonksiyonu, toplama fonksiyonu, f(z) ile belirtilmiş olan aktivasyon fonksiyonu ve y ile belirtilen çıkış değerleri oluşturur. (Omondi ve Rajapakse, 2006).



Şekil 3.1: Yapay sinir hücresinin temel elemanları.

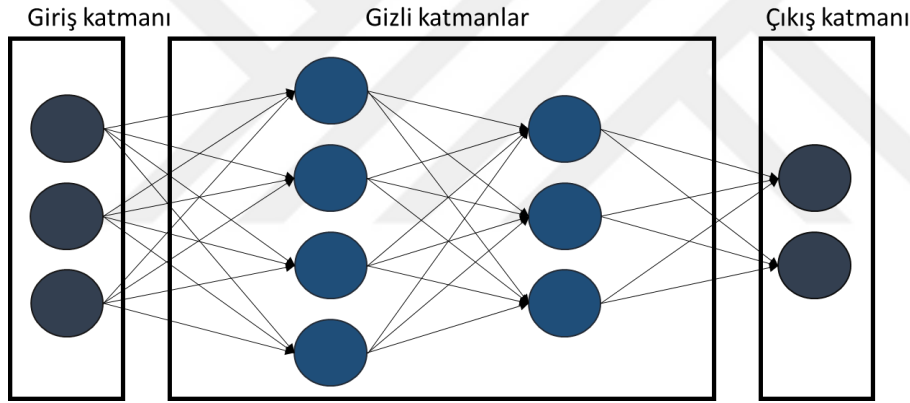
Şekil 3.2’de iki gizli katmanı bulunan 4 katmanlı bir sinir ağı modeli gösterilmiştir. Bu çalışmada yapay sinir ağları modeli kullanılarak 5 hücreli giriş ve 2 hücreli çıkış katmanı ve 10, 15, 10 ve 5 hücreden oluşan 4 gizli katmanlı bir sinir ağı modeli oluşturulmuştur. Giriş ve gizli katmanlarda relu fonksiyonu aktivasyon fonksiyonu olarak kullanılmış çıkış katmanında ise doğrusal aktivasyon fonksiyonu kullanılmıştır.

Optimizasyon algoritması için Adam veya adaptif momentum kullanılmıştır. Parametrelerin her birinin öğrenme oranlarını bunun yanında momentum değişikliklerini önbellekte tutar. Öğrenme hızını kendisi öğrenir.

Model öğrenirken, modele sokulan verilerin hepsi öğrenilmez. Bir kısmı öğrenilir ve modelin ne kadar başarılı olduğu test edilir, ne kadar başarılı olduğuna göre

geriyeyayılım ile ağırlıklar güncellenir. Güncellemenin ardından bir kısım daha öğrenilir ve tekrar ağırlıklar güncellenir. Her bir adımda model için en uygun ağırlıklar bulunmaya çalışılır ve bu adımlar epoch olarak adlandırılır. “Derin öğrenmede parametrelerin güncellenmesi geriyeyayılım işlemi ile yapılmaktadır. Bu işlemde geriye doğru türev olarak farkı bulunur ve bulunan fark değerinin öğrenme oranı parametresiyle çarpıp, çıkan sonucun ağırlık değerlerinden çıkarılarak yeni ağırlık değerinin hesaplanmasıyla yapılmaktadır.” (Çarkacı, 2018). Adam optimizasyonunun varsayılan öğrenme oranı 0,001’dir.

Yapay sinir ağlarının çalışma prensibi iki ana faktörden oluşur, bunlar öğrenme ve test aşamalarıdır. Öğrenme yapılırken, yapay sinir ağına verilerin bir kısmı öğretilir ve ağırlık fonksiyonları öğrenme algoritmasının gerektirdiği şekilde hesaplanır, çıktılar belirlenir. Test kısmında ise, yapay sinir ağının öğrenmemiş olduğu verilerle çıktılar hesaplanır (Yavuz ve Deveci, 2012).



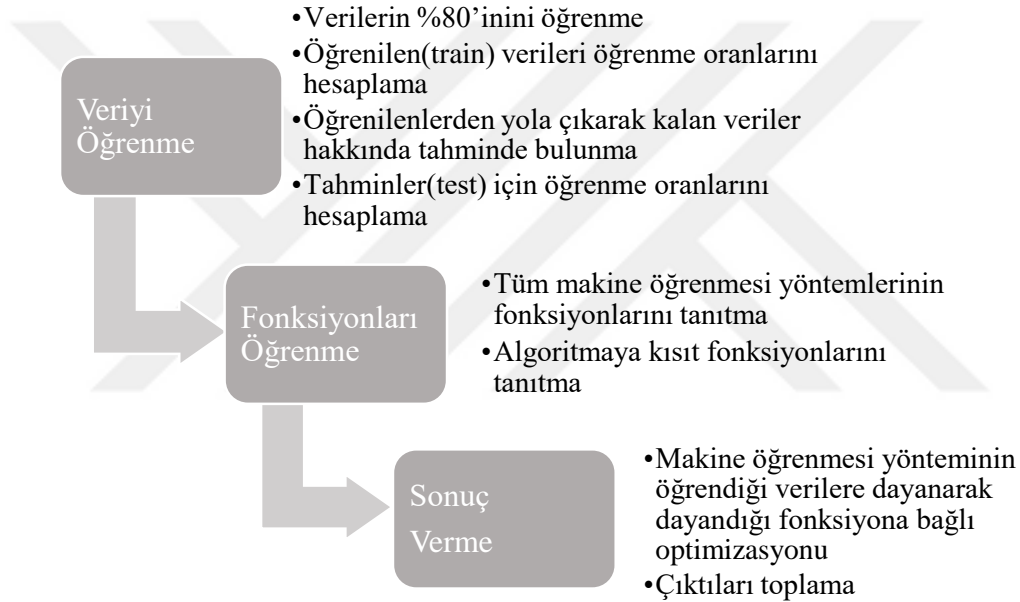
Şekil 3.2: Dört katmanlı yapay sinir ağı mimarisi.

4. YAZILIM VE UYGULAMA

4.1 Veriler ve Algoritma

Verilerin toplanması ve normalleştirilmesi işlemleri 2. bölümde anlatılmıştır. Bu verilerin algoritmaya dönüştürülmesinde programlama dili olarak Python kullanılmıştır.

Algoritmayı özetleyen akış diyagramı Şekil 4.1’de verilmiştir.



Şekil 4.1: Algoritma akış diyagramı.

Ek B’de, 158 ayrı deney arasında öğrenme oranı %80 olacak şekilde, verilerin öğrenilmesi sağlanmıştır. Burada belirtilen train değeri, %80 içerisindeki tüm girdilerin, çıktı değerlerinin gerçek değeri ile farkının karelerinin karekökleri ortalamasıdır ve denklem 4.1’deki gibi elde edilir. N, deney sayısını belirtir.

$$kapasite_test_hata_{rms} = \sqrt{\frac{\sum_i^N kapasite_test_hata_i^2}{N}} \quad (4.1)$$

Kapasite için öğrenilen değerlerin hatalarının karekökler ortalaması ise denklem 4.2 ile elde edilir.

$$kapasite_train_hata_{rms} = \sqrt{\frac{\sum_i^N kapasite_train_hata_i^2}{N}} \quad (4.2)$$

Verim değeri için test edilen değerlerin hatalarının karekökler ortalaması denklem 4.3 ile elde edilir.

$$verim_test_hata_{rms} = \sqrt{\frac{\sum_i^N verim_test_hata_i^2}{N}} \quad (4.3)$$

Verim için öğrenilen değerlerin hatalarının karekökler ortalaması ise denklem 4.4 ile elde edilir.

$$verim_train_hata_{rms} = \sqrt{\frac{\sum_i^N verim_train_hata_i^2}{N}} \quad (4.4)$$

Algoritma, gerekli kütüphanelerin ve verilerin tanıtılması ile başlar. Yapılan deney sonuçlarını, Python programı Excel dosyasından okur. Bu dosyada ilk 5 satıra sırasıyla iç ünite fan rpm, dış ünite fan rpm, eev açıklığı, dış ortam sıcaklığı ve kompresör frekansı yazılır, bunlar deneylerin girdileridir. Daha sonraki iki satırda ise bu noktalara karşılık gelen kapasite ve verim değerleri yazılmıştır, bunlar ise deneylerin çıktılarıdır.

Daha sonra algoritmanın kullanacağı temel fonksiyonlar tanıtılır. Kullanılacak yöntemle göre fonksiyon türü seçilir.

Verilerin tamamını program rastgele olarak %80 ve %20 şeklinde ayırır. Öncelikle ayırdığı %80'lik kısmın içindeki tüm deney girdilerini ve çıktılarını öğrenir. Daha sonra öğrendiği verilerden yola çıkarak kullanılan yöntemin fonksiyonuna bağlı olarak %20'lik kısmın girdilerini okur ve çıktıları hesaplar.

Hesaplanan çıktılar ile gerçek verilerin farkı algoritmanın öğrenme ölçüsüdür. Test verisi hatasının kontrolü için karekökler ortalaması değerlendirilmiştir.

Sapma, algoritmanın test verisi hatasının bir ölçüsüdür. Sapma değerinin fazla olması algoritmanın test sonucunun gerçek değerden farklı olduğunu gösterir. Test sonucu gerçekten ne kadar uzaksa algoritmanın ileride tahmin sonuçları da gerçekten o kadar uzak olacaktır. Denklem 4.5 ile gösterilmiştir.

$$Sapma(\%) = \frac{Test\ verisi\ rms}{Ortamala\ deęer} * 100 \quad (4.5)$$

Öncelikle veriyi öğrenen algoritma, daha sonra kullanacağı fonksiyonları öğrenir ve bu fonksiyonların her biri için sapma değerleri ayrı ayrı hesaplanır.

4.2 Bulgular

Çizelge 4.1: Yöntemlere göre test ve öğrenme verilerinin sapma durumları.

Doęrusal regresyon		
Sonuçlar	Kapasite	Verim
Öğrenme verisi rms	0,02	0,52
Test verisi rms	0,02	0,49
Sapma (%)	2,73	8,40
Rassal Orman		
Sonuçlar	Kapasite	Verim
Öğrenme verisi rms	0,01	0,12
Test verisi rms	0,01	0,14
Sapma (%)	1,02	0,02
Çok terimli regresyon		
Sonuçlar	Kapasite	Verim
Öğrenme verisi rms	0,01	0,09
Test verisi rms	0,01	0,16
Sapma (%)	1,39	2,81
Yapay Sinir Ağları		
Sonuçlar	Kapasite	Verim
Öğrenme verisi rms	0,03	0,54
Test verisi rms	0,04	0,62
Sapma (%)	6,55	10,65

Ek B ve Ek C’de verilen doğrusal regresyon, çok terimli regresyon, rassal orman ve yapay sinir ağları öğrenme yöntemleri ile alınan sonuçların sapma durumları Çizelge 4.1’de verilmiştir. Buna göre test verilerinin ortalamadan sapmasının en düşük olduğu yöntem %0,02 ile rassal orman yöntemidir. En yüksek olduğu yöntem ise %10,65 değeri ile yapay sinir ağları olmuştur.

Çizelge 4.2’de verilen kısıtlar için beklenen optimizasyon yapılmıştır. Bu sonuçlara göre elde edilen iç ünite fan hızı, dış ünite fan hızı, EEV açıklığı ve frekans değerleri ile ürün tekrar test edilmiştir. Öncelikle normalize veriler tekrar gerçek veri haline getirilir ve ondalıklı değerler ayarlanamayacağından ayarlanabilen en yakın değere yuvarlanır.

Çizelge 4.2: Optimizasyon sonuçları.

NORMAL VERİLER						
	Verim	Kapasite	İç ünite	Dış ünite	EEV	Frekans
35 C için tahminler	Tahmin	Tahmini	fan	fan		
Doğrusal regresyon	4,15	1,00	0,74	0,54	0,25	1,00
Çok Terimli Regresyon	3,62	0,95	0,74	0,54	0,00	0,88
Rassal Orman	2,95	0,96	0,93	0,86	0,45	0,75
Yapay sinir ağları	4,06	1,00	0,74	0,54	0,05	1,00
30 C için tahminler	Verim	Kapasite	İç ünite	Dış ünite	EEV	Frekans
	Tahmin	Tahmini	fan	fan		
Doğrusal regresyon	5,41	0,60	0,76	0,44	0,00	0,44
Çok Terimli Regresyon	5,34	0,60	0,64	0,90	0,00	0,44
Rassal Orman	4,76	0,62	0,86	0,44	0,20	0,40
Yapay sinir ağları	5,08	0,62	0,85	0,44	0,00	0,44
25 C için tahminler	Verim	Kapasite	İç ünite	Dış ünite	EEV	Frekans
	Tahmin	Tahmini	fan	fan		
Doğrusal regresyon	9,87	0,34	0,54	0,00	0,45	0,31
Çok Terimli Regresyon	9,88	0,29	0,80	0,00	0,95	0,12
Rassal Orman	9,03	0,30	0,54	0,00	0,15	0,13
Yapay sinir ağları	10,22	0,34	0,54	0,00	0,00	0,31
20 C için tahminler	Verim	Kapasite	İç ünite	Dış ünite	EEV	Frekans
	Tahmin	Tahmini	fan	fan		
Doğrusal regresyon	14,28	0,10	0,00	0,00	0,00	0,09
Çok Terimli Regresyon	14,67	0,02	0,00	0,00	0,00	0,00
Rassal Orman	14,87	0,02	0,00	0,00	0,10	0,00
Yapay sinir ağları	14,31	0,12	0,00	0,00	0,00	0,09

Verilerin ilk durumu için yapılan SEER hesaplamaları ile SEER değeri 6,689 ve buna bağlı olarak senelik güç tüketimi 183,14 kW olmaktadır.

Çizelge 4.3: Kapasite ve verim için gerçek durum ile program çıktısı karşılaştırması.

35 C için tahminler	Verim Tahmini	Kapasite Tahmini	Verim Sonucu	Kapasite Sonucu	Verim Bağıl Hata (%)	Kapasite Bağıl Hata (%)
Doğrusal regresyon	4,15	1,00	2,21	1,02	87,57	-2,00
Çok Terimli regresyon	3,62	0,95	2,58	0,97	40,18	-1,95
Rassal Orman	2,95	0,96	2,97	0,93	-0,80	2,29
YSA	4,06	1,00	2,26	1,04	79,86	-2,64
30 C için tahminler	Verim Tahmini	Kapasite Tahmini	Verim Sonucu	Kapasite Sonucu	Verim Bağıl Hata (%)	Kapasite Bağıl Hata (%)
Doğrusal regresyon	5,41	0,60	4,93	0,66	9,77	-5,89
Çok Terimli regresyon	5,34	0,60	4,65	0,67	14,75	-7,29
Rassal Orman	4,76	0,62	5,01	0,59	-5,05	3,73
YSA	5,08	0,62	4,89	0,67	3,81	-4,70
25 C için tahminler	Verim Tahmini	Kapasite Tahmini	Verim Sonucu	Kapasite Sonucu	Verim Bağıl Hata (%)	Kapasite Bağıl Hata (%)
Doğrusal regresyon	9,87	0,34	5,25	0,39	88,09	-7,88
Çok Terimli regresyon	9,88	0,29	6,65	0,23	48,50	12,08
Rassal Orman	9,03	0,30	7,75	0,29	16,55	2,34
YSA	10,22	0,34	4,44	0,38	130,28	-5,77
20 C için tahminler	Verim Tahmini	Kapasite Tahmini	Verim Sonucu	Kapasite Sonucu	Verim Bağıl Hata (%)	Kapasite Bağıl Hata (%)
Doğrusal regresyon	14,28	0,10	10,09	0,14	41,55	-11,02
Çok Terimli regresyon	14,67	0,02	13,66	0,04	7,42	-4,39
Rassal Orman	14,87	0,02	14,47	0,03	2,79	-4,89
YSA	14,31	0,12	10,09	0,17	41,79	-11,55

Optimizasyon sonucu elde edilen parametrelerin çıktıları ve bu çıktılar ile programların tahminleri arasındaki bağıl hata Çizelge 4.3'te verilmiştir. Bağıl hata değerleri verim için normalizasyon yapılmadığından doğrudan tahmin ve gerçekleşen sonuç değerleri arasında alınmıştır fakat kapasite için normalize değerler üzerinden

bağıl hata hesabı anlamlı olmadığından gerçek değer ve tahmin değerinin normalizasyondan çıkarılmış karşılığı arasında hesaplama yapılmıştır.

Optimizasyon sonrası en iyi SEER sonucu rassal orman yöntemiyle alınmıştır. Buna göre SEER değeri 6,767'dir ve senelik güç tüketimi asd 181,02 kW olmaktadır.



5. SONUÇ VE ÖNERİLER

Bu çalışmada makine öğrenmesi kullanımıyla ev tipi klimalarda parametre ve enerji verimliliği optimizasyonu 4 farklı makine öğrenmesi yöntemi ile yapılmıştır. Bu öğrenme yöntemleri, doğrusal regresyon, çok terimli regresyon, rassal orman yöntemi ve yapay sinir ağlarıdır. Bu yöntemler arasından en başarılı sonuç veren yöntem rassal orman yöntemi olmuştur.

Yapılan çalışmayla görülmektedir ki makine öğrenmesi yöntemlerinin klima parametrelerini optimize etmek için kullanılmasıyla enerji verimliliği artırılabilen ve senelik enerji tüketimi azaltılabilmektedir. Daha iyi sonuçlar alınabilmesi için çalışmanın ileriki aşamalarında yeni sonuçların da öğrenme verilerinin içine konulduktan sonra süreci devam ettirmek gereklidir. Yapılacak ileri çalışmalar ile verim artışı ve enerji tüketimi miktarı daha da azaltılabilir, çalışma ev tipi klimaları için genele yaygınlaştırılabilir. Veri sayısının artırılması öğrenmeyi arttıracığından model daha fazla geliştirilebilir.

Sonuç olarak enerji kaynaklarının akıllıca kullanımı, sınırsız ihtiyaçlarımızı sınırlı kaynaklarla yönetmeye çalıştığımız çağımızın en önemli gerekliliklerindedir. Enerjiden tasarruf etmenin en pratik yolu enerji verimliliği olduğundan, enerji verimliliğini arttırmaya yönelik çalışmalar gittikçe önem kazanmaya devam edecektir.



KAYNAKLAR

- Alpaydin, E.** (2009). *Introduction to machine learning*. MIT press.
- Breiman, L.** (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Cengel, Y. A., & Boles, M. A.** (2002). *Thermodynamics: an engineering approach*. Sea, 1000, 8862.
- Çarkacı, N.** (2018). Derin Öğrenme Uygulamalarında En Sık kullanılan Hiperparametreler
- Dalkey, N., & Helmer, O.** (1963). An Experimental Application of the Delphi Method to the Use of Experts. *Management science*, 9(3), 458-467.
- Domanski, P. A., & Yashar, D.** (2007). Optimization of finned-tube condensers using an intelligent system. *International Journal of Refrigeration*, 30(3), 482-488.
- Ethington, C. A., Thomas, S. L., & Pike, G. R.** (2002). Back to the basics: Regression as it should be. In *Higher education: Handbook of theory and research* (pp. 263-293). Springer, Dordrecht.
- Genuer, R., Poggi, J. M., & Tuleau-Malot, C.** (2010). Variable selection using random forests. *Pattern Recognition Letters*, 31(14), 2225-2236.
- Ghahramani, A., Karvigh, S. A., & Becerik-Gerber, B.** (2017). HVAC system energy optimization using an adaptive hybrid metaheuristic. *Energy and Buildings*, 152, 149-161.
- Huang, W., & Lam, H. N.** (1997). Using genetic algorithms to optimize controller parameters for HVAC systems. *Energy and Buildings*, 26(3), 277-282.
- Kahane, L. H.** (2007). *Regression basics*. Sage publications.
- Klimalarin Enerji Etiketlemesine Dair Tebliğ. (2013). T. C. Resmi Gazete, 28861,**
- Kusiak, A., Tang, F., & Xu, G.** (2011). Multi-objective optimization of HVAC system with an evolutionary computation algorithm. *Energy*, 36(5), 2440-2449.
- Markmann, C., Darkow, I., & Gracht, H. V.** (2013). A Delphi-based risk analysis - Identifying and assessing future challenges for supply chain security in a multi-stakeholder environment. *Technological Forecasting and Social Change*, 80(9) 1815-1833.
- Mirinejad, H., Welch, K. C., & Spicer, L.** (2012, May). A review of intelligent control techniques in HVAC systems. In *2012 IEEE Energytech* (pp. 1-5). IEEE.
- Müller, A. C., & Guido, S.** (2016). *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc."

- Omondi, A. R., & Rajapakse, J. C. (Eds.).** (2006). *FPGA implementations of neural networks* (Vol. 365, p. 6). Dordrecht, The Netherlands: Springer.
- Öztemel, E.** (2003), *Yapay Sinir Ağları. Papatya Yayıncılık, İstanbul.*
- Saranya, C., & Manikandan, G.** (2013). A study on normalization techniques for privacy preserving data mining. *International Journal of Engineering and Technology (IJET)*, 5(3), 2701-2704.
- Sra, S., Nowozin, S., & Wright, S. J. (Eds.).** (2012). *Optimization for machine learning*. Mit Press.
- TSE** (2019). *Air conditioners, liquid chilling packages and heat pumps, with electrically driven compressors, for space heating and cooling. Testing and rating at part load conditions and calculation of seasonal performance* (TS EN 14825)
- Url-1**< <https://journals.plos.org/plosone/article/file?type=supplementary&id=info:doi/10.1371/journal.pone.0005943.s009>>, erişim tarihi 04.09.2019.
- Url-2**< <https://towardsdatascience.com/polynomial-regression-bbe8b9d97491>>, erişim tarihi 04.09.2019.
- Yavuz, S., & Deveci, M.** (2012). İstatiksel Normalizasyon Tekniklerinin Yapay Sinir Ağın Performansına Etkisi. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, (40), 167-187.
- Zhang, D., Li, S., Sun, M., & O'Neill, Z.** (2016). An optimal and learning-based demand response and home energy management system. *IEEE Transactions on Smart Grid*, 7(4), 1790-1801.
- Zhou, H., Soh, Y. C., & Wu, X.** (2015). Integrated analysis of CFD data with K-means clustering algorithm and extreme learning machine for localized HVAC control. *Applied Thermal Engineering*, 76, 98-104.
- Zupan, J.** (2003). Basics of artificial neural network. *Nature-inspired Methods in Chemometrics: Genetic Algorithms and Artificial Neural Networks* (Leardi, R., ed.), 199-229.
- Weisberg, S.** (2005). *Applied linear regression* (Vol. 528). John Wiley & Sons.

EKLER

EK A: Normalleştirilmiş deney sonuçları.

EK B: Regresyon modelleri ve rassal orman için kullanılan algoritmalar.

EK C: Yapay sinir ağı algoritması.





EK A**Çizelge A.1:** Normalleştirilmiş edilmiş deney sonuçları.

Deney No	İç Ünite Fan RPM	Dış Ünite Fan Rpm	EEV Açıklığı	Dış Ortam KT Sıcaklığı	Kompresör Frekansı	Soğutma Kapasitesi	Verim
1	1,000	0,714	0,462	1,000	0,932	0,947	2,788
2	1,000	0,714	0,462	1,000	0,949	0,967	2,781
3	1,000	0,714	0,462	1,000	0,966	0,982	2,760
4	1,000	0,714	0,492	1,000	0,966	0,970	2,749
5	1,000	0,714	0,539	1,000	0,949	0,955	2,782
6	1,000	0,714	0,539	1,000	0,966	0,968	2,751
7	1,000	0,714	0,539	1,000	0,983	0,982	2,719
8	1,000	0,714	0,569	1,000	0,983	0,981	2,726
9	1,000	0,714	0,615	1,000	0,983	0,978	2,726
10	1,000	0,714	0,646	1,000	0,983	0,978	2,729
11	1,000	0,714	0,692	1,000	0,983	0,971	2,720
12	1,000	0,714	0,723	1,000	0,983	0,972	2,722
13	1,000	0,714	0,769	1,000	0,983	0,969	2,717
14	1,000	0,714	0,800	1,000	0,983	0,949	2,657
15	1,000	0,714	0,846	1,000	0,983	0,948	2,657
16	1,000	0,714	0,877	1,000	0,983	0,953	2,678
17	1,000	0,714	0,923	1,000	0,983	0,954	2,682
18	1,000	0,714	0,954	1,000	0,983	0,941	2,656
19	1,000	0,714	1,000	1,000	0,983	0,936	2,648
20	1,000	0,714	0,800	1,000	1,000	0,961	2,647
21	1,000	0,714	0,846	1,000	1,000	0,959	2,641
22	1,000	0,714	0,877	1,000	1,000	0,958	2,643
23	1,000	0,714	0,923	1,000	1,000	0,957	2,640
24	1,000	0,714	0,877	1,000	1,000	0,954	2,636
25	1,000	0,714	1,000	1,000	1,000	0,955	2,639
26	1,000	1,000	0,308	1,000	0,898	0,961	2,960
27	1,000	1,000	0,308	1,000	0,898	0,948	2,895
28	1,000	1,000	0,308	1,000	0,915	0,962	2,856
29	1,000	1,000	0,339	1,000	0,915	0,959	2,877
30	1,000	1,000	1,000	0,385	1,000	0,915	0,945
31	1,000	0,714	0,714	0,462	1,000	0,966	0,943
32	1,000	0,714	0,714	0,462	1,000	0,966	0,964
33	1,000	0,462	0,462	1,000	0,966	0,977	2,802
34	1,000	0,462	0,462	1,000	0,983	0,976	2,696
35	1,000	0,462	0,462	1,000	0,966	0,971	2,743
36	1,000	0,154	0,154	0,667	0,492	0,660	4,641

Çizelge A.1 (devam): Normalleştirilmiş edilmiş deney sonuçları.

Deney No	İç Ünite Fan RPM	Dış Ünite Fan Rpm	EEV Açıklığı	Dış Ortam KT Sıcaklığı	Kompresör Frekansı	Soğutma Kapasitesi	Verim
37	1,000	0,714	0,185	0,667	0,492	0,658	4,630
38	1,000	0,714	0,154	0,667	0,509	0,678	4,585
39	1,000	0,714	0,154	0,667	0,525	0,691	4,519
40	1,000	0,714	0,185	0,667	0,509	0,674	4,567
41	1,000	0,714	0,231	0,667	0,509	0,672	4,563
42	1,000	0,714	0,262	0,667	0,509	0,673	4,566
43	1,000	0,714	0,308	0,667	0,509	0,671	4,566
44	1,000	0,714	0,339	0,667	0,509	0,666	4,544
45	1,000	0,714	0,385	0,667	0,509	0,664	4,540
46	1,000	0,714	0,415	0,667	0,509	0,665	4,548
47	1,000	0,714	0,462	0,667	0,509	0,663	4,546
48	1,000	0,714	0,000	0,667	0,458	0,645	4,770
49	1,000	0,714	0,031	0,667	0,458	0,635	4,762
50	1,000	0,714	0,077	0,667	0,458	0,621	4,749
51	1,000	0,714	0,108	0,667	0,458	0,624	4,762
52	1,000	0,714	0,154	0,667	0,458	0,619	4,745
53	1,000	0,714	0,185	0,667	0,458	0,621	4,762
54	1,000	0,714	0,231	0,667	0,458	0,616	4,735
55	1,000	0,714	0,262	0,667	0,458	0,616	4,743
56	1,000	0,714	0,308	0,667	0,458	0,614	4,738
57	1,000	0,714	0,000	0,667	0,475	0,663	4,672
58	1,000	0,714	0,031	0,667	0,475	0,656	4,681
59	1,000	0,714	0,077	0,667	0,475	0,640	4,682
60	1,000	0,714	0,108	0,667	0,475	0,639	4,692
61	1,000	0,714	0,154	0,667	0,475	0,633	4,672
62	1,000	0,714	0,185	0,667	0,475	0,631	4,657
63	1,000	0,714	0,231	0,667	0,475	0,628	4,649
64	1,000	0,714	0,262	0,667	0,475	0,628	4,655
65	1,000	0,714	0,308	0,667	0,475	0,608	4,549
66	1,000	0,714	0,000	0,667	0,458	0,625	4,657
67	1,000	0,714	0,077	0,667	0,475	0,614	4,561
68	1,000	0,714	0,077	0,667	0,475	0,625	4,549
69	0,702	0,714	0,077	0,333	0,186	0,327	7,677
70	0,702	0,714	0,077	0,333	0,203	0,350	7,521
71	0,702	0,714	0,077	0,333	0,220	0,373	7,384
72	0,702	0,714	0,108	0,333	0,186	0,325	7,662

Çizelge A.1 (devam): Normalleştirilmiş edilmiş deney sonuçları.

Deney No	İç Ünite Fan RPM	Dış Ünite Fan Rpm	EEV Açıklığı	Dış Ortam KT Sıcaklığı	Kompresör Frekansı	Soğutma Kapasitesi	Verim
73	0,702	0,714	0,154	0,333	0,186	0,325	7,683
74	0,702	0,714	0,185	0,333	0,186	0,322	7,664
75	0,702	0,714	0,231	0,333	0,186	0,321	7,678
76	0,702	0,714	0,262	0,333	0,186	0,322	7,706
77	0,702	0,714	0,308	0,333	0,186	0,317	7,652
78	0,702	0,714	0,339	0,333	0,186	0,314	7,633
79	0,702	0,714	0,385	0,333	0,186	0,312	7,644
80	0,702	0,524	0,231	0,333	0,186	0,289	7,337
81	0,702	0,524	0,262	0,333	0,186	0,293	7,413
82	0,702	0,524	0,308	0,333	0,186	0,293	7,416
83	0,702	0,595	0,262	0,333	0,186	0,299	7,439
84	0,702	0,714	0,262	0,333	0,186	0,281	7,140
85	0,702	0,714	0,262	0,333	0,186	0,283	7,246
86	0,000	0,000	0,077	0,000	0,000	0,025	14,666
87	0,000	0,000	0,077	0,000	0,017	0,045	13,773
88	0,000	0,000	0,077	0,000	0,034	0,072	13,156
89	0,000	0,000	0,077	0,000	0,051	0,098	12,615
90	0,000	0,000	0,077	0,000	0,068	0,121	11,949
91	0,000	0,000	0,000	0,000	0,000	0,023	14,386
92	0,000	0,000	0,031	0,000	0,000	0,022	14,526
93	0,000	0,000	0,108	0,000	0,000	0,020	14,668
94	0,000	0,000	0,154	0,000	0,000	0,020	14,822
95	0,000	0,000	0,154	0,000	0,000	0,015	14,480
96	0,000	0,000	0,154	0,000	0,000	0,000	13,322
97	0,000	0,000	0,154	0,000	0,000	0,004	14,251
98	0,000	0,000	0,077	0,000	0,000	0,018	14,660
99	0,000	0,000	0,077	0,000	0,017	0,038	13,733
100	0,000	0,000	0,077	0,000	0,034	0,064	13,163
101	0,000	0,000	0,077	0,000	0,051	0,092	12,711
102	0,000	0,000	0,077	0,000	0,068	0,115	12,138
103	0,000	0,000	0,000	0,000	0,000	0,021	14,790
104	0,000	0,000	0,031	0,000	0,000	0,018	14,822
105	0,000	0,000	0,108	0,000	0,000	0,016	15,126
106	0,000	0,000	0,154	0,000	0,000	0,012	14,787
107	0,000	0,000	0,154	0,000	0,000	0,012	13,697
108	0,000	0,000	0,154	0,000	0,000	0,009	14,227
109	0,000	0,000	0,154	0,000	0,000	0,015	14,506

Çizelge A.1 (devam): Normalleştirilmiş edilmiş deney sonuçları.

Deney No	İç Ünite Fan RPM	Dış Ünite Fan Rpm	EEV Açıklığı	Dış Ortam KT Sıcaklığı	Kompresör Frekansı	Soğutma Kapasitesi	Verim
110	0,702	0,524	0,231	0,333	0,186	0,284	7,320
111	0,702	0,524	0,262	0,333	0,186	0,296	7,518
112	0,702	0,524	0,308	0,333	0,186	0,297	7,575
113	0,702	0,714	0,262	0,333	0,186	0,308	7,234
114	0,702	0,714	0,262	0,333	0,186	0,294	7,396
115	0,702	0,714	0,262	0,333	0,186	0,302	7,520
116	0,702	0,714	0,000	0,667	0,458	0,596	4,488
117	0,702	0,714	0,031	0,667	0,458	0,606	4,601
118	0,702	0,714	0,077	0,667	0,458	0,604	4,677
119	0,702	0,714	0,108	0,667	0,458	0,592	4,654
120	0,702	0,714	0,154	0,667	0,458	0,583	4,648
121	0,702	0,714	0,185	0,667	0,458	0,579	4,633
122	0,702	0,714	0,231	0,667	0,458	0,579	4,644
123	0,702	0,714	0,262	0,667	0,458	0,577	4,634
124	0,702	0,714	0,308	0,667	0,458	0,575	4,626
125	0,702	0,714	0,000	0,667	0,475	0,609	4,380
126	0,702	0,714	0,031	0,667	0,475	0,628	4,547
127	0,702	0,714	0,077	0,667	0,475	0,630	4,626
128	0,702	0,714	0,108	0,667	0,475	0,618	4,618
129	0,702	0,714	0,154	0,667	0,475	0,601	4,575
130	0,702	0,714	0,185	0,667	0,475	0,598	4,579
131	0,702	0,714	0,231	0,667	0,475	0,596	4,586
132	0,702	0,714	0,262	0,667	0,475	0,594	4,584
133	0,702	0,714	0,308	0,667	0,475	0,581	4,503
134	0,702	0,714	0,000	0,667	0,458	0,607	4,585
135	0,702	0,714	0,077	0,667	0,475	0,622	4,489
136	1,000	0,714	0,077	0,667	0,475	0,638	4,645
137	1,000	0,714	0,077	0,667	0,475	0,659	4,698
138	1,000	0,714	0,800	1,000	0,983	0,930	2,604
139	1,000	0,714	0,877	1,000	0,983	0,927	2,614
140	1,000	0,714	0,923	1,000	0,983	0,927	2,622
141	1,000	0,714	0,954	1,000	0,983	0,933	2,628
142	1,000	0,714	1,000	1,000	0,983	0,926	2,628
143	1,000	0,714	0,800	1,000	1,000	0,944	2,607
144	1,000	0,714	0,846	1,000	1,000	0,948	2,615
145	1,000	0,714	0,877	1,000	1,000	0,944	2,609
146	1,000	0,714	0,923	1,000	1,000	0,942	2,606

Çizelge A.1 (devam): Normalleştirilmiş edilmiş deney sonuçları.

Deney No	İç Ünite Fan RPM	Dış Ünite Fan Rpm	EEV Açıklığı	Dış Ortam KT Sıcaklığı	Kompresör Frekansı	Soğutma Kapasitesi	Verim
147	1,000	0,714	0,877	1,000	1,000	0,943	2,615
148	1,000	0,714	1,000	1,000	1,000	0,942	2,614
149	0,851	0,714	0,385	1,000	0,932	0,944	2,747
150	0,851	1,000	0,308	1,000	0,898	0,954	2,922
151	0,851	1,000	0,308	1,000	0,915	0,961	2,864
152	0,851	1,000	0,339	1,000	0,915	0,965	2,899
153	0,851	1,000	0,385	1,000	0,915	0,957	2,906
154	0,851	0,714	0,462	1,000	0,966	0,953	2,634
155	1,000	0,714	0,462	1,000	0,966	0,961	2,697
156	1,000	1,000	0,462	1,000	0,966	0,999	2,840
157	1,000	0,714	0,462	1,000	0,983	1,000	2,726
158	1,000	0,714	0,462	1,000	0,966	0,989	2,777



EK B

Bu kısımda gerekli kütüphaneler tanıtılıp, yüklenmiştir.

```
import pandas as pd
import numpy as np
```

```
from math import sqrt
```

```
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn import linear_model
from sklearn.preprocessing import PolynomialFeatures
```

```
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn.metrics import mean_squared_error
```

```
import warnings
warnings.filterwarnings("ignore")
```

#Bu kısımda 158 adet veri makine tarafından okunmuştur.

```
data = pd.read_excel(io = ".\dataall.xlsx")
k = np.transpose(data)
```

#Yukarıda okunan verilerden ilk 5 satır girdi, daha sonraki iki satır çıktı olarak tanımlanmıştır.

```
x = k[[0,1,2,3,4]]
y = k[[5,6]]
```

#Girdi verileri ve çıktı verileri için öğrenme ve test etme verileri ayrılmış ayrıca öğrenme verisi oranı %80 olarak belirlenmiştir.

```
x_train, x_test, y_train, y_test = train_test_split(x,y, train_size = 0.80)
```

Doğrusal regresyon modeli için kapasite değerini anlatacak tanımlamalar yapılmıştır.

```
regr_btu = linear_model.LinearRegression()
```

Yukarıda verilen öğrenme verisi ve btu etiketi ile modelin kapasite eğitimi yapılmıştır.

```
regr_btu.fit(x_train, y_train[5])
```

Yukarıda verilen test verisi ile modelin kapasite sonuçları test hatasını öğrenmek için sonuçları alınmıştır.

```
pred_out_btu = regr_btu.predict(x_test)
```

Yukarıda verilen öğrenme verisi ile modelin kapasite değeri öğrenme hatasını öğrenmek için sonuçları alınmıştır.

```
pred_out_btu_train = regr_btu.predict(x_train)
```

Doğrusal regresyon modeli için verim değerini anlatacak tanımlamalar yapılmıştır.

```
regr_wv = linear_model.LinearRegression()
```

Yukarıda verilen öğrenme verisi ve ww etiketi ile modelin verim eğitimi yapılmıştır.
regr_ww.fit(x_train, y_train[6])

Yukarıda verilen test verisi ile modelin verim sonuçları test hatasını öğrenmek için sonuçları alınmıştır.

```
pred_out_ww = regr_ww.predict(x_test)
```

Yukarıda verilen öğrenme verisi ile modelin verim değeri öğrenme hatasını öğrenmek için sonuçları alınmıştır.

```
pred_out_ww_train = regr_ww.predict(x_train)
```

Öğrenme ve test değerleri için öğrenme verilerinde yapılan hata, yani olması gereken değer ile modelin çıktısı olan değer arasındaki farkların kareköklerinin toplamının ortalaması kapasite değeri için aranmaktadır. Ortalama hatayı bulabilmek için kapasite değeri ortalaması hesaplanır ve test hatası ile çarpılarak ortalama hata bulunur.

```
rmse_test = sqrt(mean_squared_error(pred_out_btu,y_test[5]))  
rmse_train = sqrt(mean_squared_error(pred_out_btu_train,y_train[5]))  
print('train: ' + str(rmse_train))  
print('test: ' + str(rmse_test))  
print('mean: ' + str(k[5].mean()))  
print('percentage mean: ' + str(rmse_test/(k[5].mean()*100) + '%')
```

Doğrusal regresyon ile elde edilen, kapasite değeri için sonuçlar aşağıdaki gibidir. Kapasite değeri için test verileri bu yöntemle ortalamadan yaklaşık %2,72 sapmaktadır.

```
train: 0.01622595316996263  
test: 0.01660048731443698  
mean: 0.608613610192131  
percentage mean: 2.7275905494779247%
```

Öğrenme ve test değerleri için öğrenme verilerinde yapılan hata, yani olması gereken değer ile modelin çıktısı olan değer arasındaki farkların kareköklerinin toplamının ortalaması kapasite değeri için aranmaktadır. Ortalama hatayı bulabilmek için verim değeri ortalaması hesaplanır ve test hatası ile çarpılarak ortalama hata bulunur.

```
rmse_test = sqrt(mean_squared_error(pred_out_btu,y_test[6]))  
rmse_train = sqrt(mean_squared_error(pred_out_btu_train,y_train[6]))  
print('train: ' + str(rmse_train))  
print('test: ' + str(rmse_test))  
print('mean: ' + str(k[6].mean()))  
print('percentage mean: ' + str(rmse_test/(k[6].mean()*100) + '%')
```

Doğrusal regresyon ile elde edilen, verim değeri için sonuçlar aşağıdaki gibidir. Verim değeri için test verileri bu yöntemle ortalamadan yaklaşık %12,72 sapmaktadır.

```
train: 0.5200217674931072  
test: 0.48605348682219374  
mean: 5.785254240036011  
percentage mean: 8.401592508390232%
```

Rassal orman modeli için kapasite değerini anlatacak tanımlamalar yapılmıştır.

```
random_btu = RandomForestRegressor()
random_btu.fit(x_train, y_train[5])
randomResultBtu1 = random_btu.predict (x_train)
randomResultBtu2 = random_btu.predict (x_test)
```

Yukarıdaki adımlar takip edilerek kapasite değeri hataları bulunacaktır.

```
rmse_test = sqrt(mean_squared_error(randomResultBtu1,y_train[5]))
rmse_train = sqrt(mean_squared_error(randomResultBtu2,y_test[5]))
print('train: ' + str(rmse_train))
print('test: ' + str(rmse_test))
print('mean: '+ str(k[5].mean()))
print('percentage mean: ' + str(rmse_test/(k[5].mean()*100) + '%')
```

Rassal orman modeli ile elde edilen, kapasite değeri için sonuçlar aşağıdaki gibidir. Kapasite değeri için test verileri bu yöntemle ortalamadan yaklaşık %1,01 sapmaktadır.

```
train: 0.014205932323538912
test: 0.006200494492990827
mean: 0.608613610192131
percentage mean: 1.018789982536443%
```

Rassal orman modeli için verim değerini anlatacak tanımlamalar yapılmıştır.

```
random_ww = RandomForestRegressor()
random_ww.fit (x_train, y_train[6])
randomResultWw1 = random_ww.predict (x_train)
randomResultWw2 = random_ww.predict (x_test)
```

Yukarıdaki adımlar takip edilerek verim değeri hataları bulunacaktır.

```
rmse_test = sqrt(mean_squared_error(randomResultWw1,y_train[6]))
rmse_train = sqrt(mean_squared_error(randomResultWw2,y_test[6]))
print('train: ' + str(rmse_train))
print('test: ' + str(rmse_test))
print('mean: '+ str(k[6].mean()))
print('percentage mean: ' + str(rmse_test/(k[6].mean()*100) + '%')
```

Rassal orman modeli ile elde edilen, verim değeri için sonuçlar aşağıdaki gibidir. Verim değeri için test verileri bu yöntemle ortalamadan yaklaşık %2,34 sapmaktadır.

```
train: 0.12077888314220009
test: 0.13530349048194296
mean: 5.785254240036011
percentage mean: 2.3387648125400404%
```

Çok terimli regresyon modeli doğrusal regresyon modeli gibi çalışmaktadır ancak bu modelin çalışması için çok terimli model dönüşümünü özellik dönüşümü ile yapılır.
#burada veriler feature transformation yapılır

```
poly = PolynomialFeatures(degree=2)
poly_features = PolynomialFeatures(degree=2)
```

```
x_train2 = poly.fit_transform(x_train)
x_test2 = poly.fit_transform(x_test)
```

```
X_train_poly = poly_features.fit_transform(x_train)
```

Çok terimli regresyon modeli için kapasite değerini anlatacak tanımlamalar yapılmıştır. Model eğitilir.

```
poly_model_btu = linear_model.LinearRegression()
poly_model_btu.fit(X_train_poly,y_train[5])
```

Daha sonra bu model ile yeni bir veri eğitilmek istenirse poly_features.fit_transform fonksiyonu ile dönüştürülerek verilir.

```
poly_model_btu1 = poly_model_btu.predict(X_train_poly)
poly_model_btu2 = poly_model_btu.predict(poly_features.fit_transform(x_test))
```

Çok terimli regresyon modeli için verim değerini anlatacak tanımlamalar yapılmıştır. Model eğitilir.

```
poly_model_ww = linear_model.LinearRegression()
poly_model_ww.fit(X_train_poly,y_train[6])
```

Aynı şekilde bu model ile yeni bir veri eğitilmek istenirse poly_features.fit_transform fonksiyonu ile dönüştürülerek verilir.

```
poly_model_ww1 = poly_model_ww.predict(X_train_poly)
poly_model_ww2 = poly_model_ww.predict(poly_features.fit_transform(x_test))
```

Yukarıdaki adımlar takip edilerek kapasite değeri hataları bulunacaktır.

```
rmse_test = sqrt(mean_squared_error(poly_model_btu1,y_train[5]))
rmse_train = sqrt(mean_squared_error(poly_model_btu2,y_test[5]))
print('train: ' + str(rmse_train))
print('test: ' + str(rmse_test))
print('mean: '+ str(k[5].mean()))
print('percentage mean: ' + str(rmse_test/(k[5].mean()*100) + '%')
```

Çok terimli regresyon modeli ile elde edilen, verim değeri için sonuçlar aşağıdaki gibidir. Kapasite değeri için test verileri bu yöntemle ortalamadan yaklaşık %1,38 sapmaktadır.

```
train: 0.011163503078583858
test: 0.008431591895864359
mean: 0.608613610192131
percentage mean: 1.3853768227763785%
```

Yukarıdaki adımlar takip edilerek verim değeri hataları bulunacaktır.

```
rmse_test = sqrt(mean_squared_error(poly_model_ww1,y_train[6]))
rmse_train = sqrt(mean_squared_error(poly_model_ww2,y_test[6]))
print('train: ' + str(rmse_train))
print('test: ' + str(rmse_test))
print('mean: '+ str(k[6].mean()))
print('percentage mean: ' + str(rmse_test/(k[6].mean()*100) + '%')
```


Çok terimli regresyon modeli ile elde edilen, verim değeri için sonuçlar aşağıdaki gibidir. Kapasite değeri için test verileri bu yöntemle ortalamadan yaklaşık %2,81 sapmaktadır.

train: 0.09411850148147562
test: 0.16277669620993349
mean: 5.785254240036011
percentage mean: 2.813648103543333%

Üç farklı yöntem için optimize edilecek verim değeri ve bu verime karşılık gelecek kapasite değerini bulmak için fonksiyonlar tanımlanmıştır.

```
def sonucGetir(newBtuValsLinear, newWwValsLinear, newBtuValsPolynomial,  
newWwValsPolynomial,  
newBtuValsRandomForest,newWwValsRandomForest,valX,valY):
```

```
    index2=0
```

Kapasite değeri, her sıcaklık için belirli aralıklarda yer aldığından limit fonksiyonu yazılmıştır.

```
    for x in range(index):
```

```
        if newBtuValsLinear[x]>valX and newBtuValsLinear[x]<valY :  
            index2=index2+1
```

```
    refinedBtu=np.zeros( (index2,1) )
```

```
    refinedWw=np.zeros( (index2,1) )
```

```
    refinedInput=np.zeros( (5,index2) )
```

Doğrusal regresyon fonksiyonu için tahmin edilen verim ve kapasite değerleri ve bu değerlere karşılık gelen parametrelerin değerleri alınacaktır.

```
    index2=0
```

```
    for x in range(index):
```

```
        if newBtuValsLinear[x]>valX and newBtuValsLinear[x]<valY :  
            refinedBtu[index2]=newBtuValsLinear[x]  
            refinedWw[index2]=newWwValsLinear[x]  
            refinedInput[:,index2]=allInput[:,x]  
            index2=index2+1
```

```
    ww=refinedWw[refinedWw.argmax()]
```

```
    btu=refinedBtu[refinedWw.argmax()]
```

```
    print(('Doğrusal Regresyon Verim Çıktısı: ',np.str(ww)))
```

```
    print(('Doğrusal Regresyon Kapasite Çıktısı: ',np.str(btu)))
```

```
    print(('Doğrusal Regresyon En İyi Değerler:  
,refinedInput[:,refinedWw.argmax()]'))
```

Çok terimli regresyon fonksiyonu için tahmin edilen verim ve kapasite değerleri ve bu değerlere karşılık gelen parametrelerin alınacaktır.

```
    index2=0
```

```
    for x in range(index):
```

```
        if newBtuValsPolynomial[x]>valX and newBtuValsPolynomial[x]<valY :  
            index2=index2+1
```

```

refinedBtu=np.zeros( (index2,1) )
refinedWw=np.zeros( (index2,1) )
refinedInput=np.zeros( (5,index2) )
index2=0
for x in range(index):
    if newBtuValsPolynomial[x]>valX and newBtuValsPolynomial[x]<valY :
        refinedBtu[index2]=newBtuValsPolynomial[x]
        refinedWw[index2]=newWwValsPolynomial[x]
        refinedInput[:,index2]=allInput[:,x]
        index2=index2+1
ww=refinedWw[refinedWw.argmax()]
btu=refinedBtu[refinedWw.argmax()]

print(('Çok terimli Regresyon Verim Çıktısı: ',np.str(ww)))
print(('Çok terimli Regresyon Kapasite Çıktısı: ',np.str(btu)))
print(('Çok terimli Regresyon En İyi Değerler:
',refinedInput[:,refinedWw.argmax()])))

# Rassal orman fonksiyonu için tahmin edilen verim ve kapasite değerleri ve bu
değerlere karşılık gelen parametrelerin değerleri alınması için gerekli döngü
girilmiştir.
index2=0
for x in range(index):
    if newBtuValsRandomForest[x]>valX and newBtuValsRandomForest[x]<valY
:
        index2=index2+1

refinedBtu=np.zeros( (index2,1) )
refinedWw=np.zeros( (index2,1) )
refinedInput=np.zeros( (5,index2) )

index2=0
for x in range(index):
    if newBtuValsRandomForest[x]>valX and newBtuValsRandomForest[x]<valY
:
        refinedBtu[index2]=newBtuValsRandomForest[x]
        refinedWw[index2]=newWwValsRandomForest[x]
        refinedInput[:,index2]=allInput[:,x]
        index2=index2+1
ww=refinedWw[refinedWw.argmax()]
btu=refinedBtu[refinedWw.argmax()]
print(('Rassal Orman Verim Çıktısı: ',np.str(ww)))
print(('Rassal Orman Kapasite Çıktısı: ',np.str(btu)))
print(('Rassal Orman En İyi Değerler: ',refinedInput[:,refinedWw.argmax()])))

return 'tamamlandı'

```

Normalize dış sıcaklık değerinin 1 olduğu nokta için, iç ünite fanı, dış ünite fanı, eev ve frekans değerleri geniş aralıklarda taranmıştır. Kodun çalışabilmesi için normalize değerler burada 100 ile çarpılmıştır.

```
index=0
for iduFan in range (74,101,2):
    for oduFan in range (54,101,2):
        for eev in range (0,71,5):
            for frekans in range (75,101,1):
                index=index+1
```

```
newBtuValsLinear=np.zeros((index,1))
newWwValsLinear=np.zeros((index,1))
```

```
newBtuValsPolynomial=np.zeros((index,1))
newWwValsPolynomial=np.zeros((index,1))
```

```
newBtuValsRandomForest=np.zeros((index,1))
newWwValsRandomForest=np.zeros((index,1))
```

```
allInput = np.zeros( (5,index) )
```

```
index=0
for iduFan in range (74,101,2):
    for oduFan in range (54,101,2):
        for eev in range (0,71,5):
            for frekans in range (75,101,1):
                disSic=1
```

Normalize değerlere geri dönebilmek için değerler burada 100'e bölünmüştür.

```
inputData = np.zeros( (5,1) )
inputData[0]=iduFan/100
inputData[1]=oduFan/100
inputData[2]=eev/100
inputData[3]=disSic
inputData[4]=frekans/100
inputData=np.transpose(inputData)
allInput[:,index]=inputData
newBtuValsLinear[index]=regr_btu.predict(inputData)
newWwValsLinear[index]=regr_ww.predict (inputData)
```

```
newBtuValsPolynomial[index]=poly_model_btu.predict(poly_features.fit_transform(inputData))
newWwValsPolynomial[index]=poly_model_ww.predict (poly_features.fit_transform(inputData))
```

```
newBtuValsRandomForest[index]=random_btu.predict(inputData)
newWwValsRandomForest[index]=random_ww.predict (inputData)
index=index+1
```

Kapasite değerinin 0,95 ile 1 normalize değerleri arasında 3 ayrı yöntem için bulunan verim, kapasite ve bunlara karşılık gelen parametre değerleri aşağıdaki gibi bulunmuştur.

```
valX=0.95
valY=1
sonucGetir(newBtuValsLinear,newWwValsLinear,newBtuValsPolynomial,newWw
ValsPolynomial,newBtuValsRandomForest,newWwValsRandomForest,valX,valY)
```

```
('Doğrusal Regresyon Verim Çıktısı: ', '[4.14528382]')
('Doğrusal Regresyon Kapasite Çıktısı: ', '[0.99750101]')
('Doğrusal Regresyon En İyi Değerler: ', array([0.74, 0.54, 0.25, 1. , 1. ]))
```

```
('Çok terimli Regresyon Verim Çıktısı: ', '[3.61669755]')
('Çok terimli Regresyon Kapasite Çıktısı: ', '[0.95004579]')
('Çok terimli Regresyon En İyi Değerler: ', array([0.74, 0.54, 0. , 1. , 0.88]))
```

```
('Rassal Orman Verim Çıktısı: ', '[2.94609644]')
('Rassal Orman Kapasite Çıktısı: ', '[0.9560999]')
('Rassal Orman En İyi Değerler: ', array([0.93, 0.86, 0.45, 1. , 0.75]))
```

'tamamlandı'

Normalize dış sıcaklık değerinin 0,66 olduğu nokta için, iç ünite fanı, dış ünite fanı, eev ve frekans değerleri geniş aralıklarda taranmıştır. Kodun çalışabilmesi için normalize değerler burada 100 ile çarpılmıştır.

```
index=0
for iduFan in range (64,91,1):
    for oduFan in range (44,91,1):
        for eev in range (0,71,5):
            for frekans in range (40,45,1):
                index=index+1

newBtuValsLinear=np.zeros((index,1))
newWwValsLinear=np.zeros((index,1))

newBtuValsPolynomial=np.zeros((index,1))
newWwValsPolynomial=np.zeros((index,1))

newBtuValsRandomForest=np.zeros((index,1))
newWwValsRandomForest=np.zeros((index,1))

allInput = np.zeros( (5,index) )
```

```
index=0
for iduFan in range (64,91,1):
    for oduFan in range (44,91,1):
        for eev in range (0,71,5):
            for frekans in range (40,45,1):
                disSic=1
```

Normalize değerlere geri dönebilmek için değerler burada 100'e bölünmüştür.

```
inputData = np.zeros( (5,1) )
inputData[0]=iduFan/100
inputData[1]=oduFan/100
inputData[2]=eev/100
inputData[3]=disSic
inputData[4]=frekans/100
inputData=np.transpose(inputData)
allInput[:,index]=inputData
newBtuValsLinear[index]=regr_btu.predict(inputData)
newWwValsLinear[index]=regr_ww.predict (inputData)

newBtuValsPolynomial[index]=poly_model_btu.predict(poly_features.fit_transform(inputData))
newWwValsPolynomial[index]=poly_model_ww.predict (poly_features.fit_transform(inputData))

newBtuValsRandomForest[index]=random_btu.predict(inputData)
newWwValsRandomForest[index]=random_ww.predict (inputData)
index=index+1
```

Kapasite değerinin 0,6 ile 0,7 normalize değerleri arasında 3 ayrı yöntem için bulunan verim, kapasite ve bunlara karşılık gelen parametre değerleri aşağıdaki gibi bulunmuştur.

```
valX=0.6
valY=0.7
sonucGetir(newBtuValsLinear,newWwValsLinear,newBtuValsPolynomial,newWwValsPolynomial,newBtuValsRandomForest,newWwValsRandomForest,valX,valY)
```

```
('Doğrusal Regresyon Verim Çıktısı: ', '[5.41163473]')
('Doğrusal Regresyon Kapasite Çıktısı: ', '[0.60087851]')
('Doğrusal Regresyon En İyi Değerler: ', array([0.76, 0.44, 0. , 0.66, 0.44]))
```

```
('Çok terimli Regresyon Verim Çıktısı: ', '[5.33597019]')
('Çok terimli Regresyon Kapasite Çıktısı: ', '[0.60051881]')
('Çok terimli Regresyon En İyi Değerler: ', array([0.64, 0.9 , 0. , 0.66, 0.43]))
```

```
('Rassal Orman Verim Çıktısı: ', '[4.75678854]')
('Rassal Orman Kapasite Çıktısı: ', '[0.62028027]')
('Rassal Orman En İyi Değerler: ', array([0.86, 0.44, 0.2 , 0.66, 0.4 ]))
```

'tamamlandı'

Normalize dış sıcaklık değerinin 0,33 olduğu nokta için, iç ünite fanı, dış ünite fanı, eev ve frekans değerleri geniş aralıklarda taranmıştır. Kodun çalışabilmesi için normalize değerler burada 100 ile çarpılmıştır.

```
index=0
for iduFan in range (54,81,2):
    for oduFan in range (0,71,2):
        for eev in range (0,101,5):
```

```

    for frekans in range (10,32,1):
        index=index+1

newBtuValsLinear=np.zeros((index,1))
newWwValsLinear=np.zeros((index,1))

newBtuValsPolynomial=np.zeros((index,1))
newWwValsPolynomial=np.zeros((index,1))

newBtuValsRandomForest=np.zeros((index,1))
newWwValsRandomForest=np.zeros((index,1))

allInput = np.zeros( (5,index) )

index=0
for iduFan in range (54,81,2):
    for oduFan in range (0,71,2):
        for eev in range (0,101,5):
            for frekans in range (10,32,1):
                disSic=1
                # Normalize değerlere geri dönebilmek için değerler burada 100'e bölünmüştür.
                inputData = np.zeros( (5,1) )
                inputData[0]=iduFan/100
                inputData[1]=oduFan/100
                inputData[2]=eev/100
                inputData[3]=disSic
                inputData[4]=frekans/100
                inputData=np.transpose(inputData)
                allInput[:,index]=inputData
                newBtuValsLinear[index]=regr_bt.predict(inputData)
                newWwValsLinear[index]=regr_ww.predict (inputData)

                newBtuValsPolynomial[index]=poly_model_bt.predict(poly_features.fit_transform(inputData))
                newWwValsPolynomial[index]=poly_model_ww.predict (poly_features.fit_transform(inputData))

                newBtuValsRandomForest[index]=random_bt.predict(inputData)
                newWwValsRandomForest[index]=random_ww.predict (inputData)
                index=index+1

# Kapasite değerinin 0,6 ile 0,7 normalize değerleri arasında 3 ayrı yöntem için bulunan verim, kapasite ve bunlara karşılık gelen parametre değerleri aşağıdaki gibi bulunmuştur.
valX=0.29
valY=0.34
sonucGetir(newBtuValsLinear,newWwValsLinear,newBtuValsPolynomial,newWwValsPolynomial,newBtuValsRandomForest,newWwValsRandomForest,valX,valY)

('Doğrusal Regresyon Verim Çıktısı: ', '[9.8746338]')

```

```
('Doğrusal Regresyon Kapasite Çıktısı: ', '[0.33718841]')
('Doğrusal Regresyon En İyi Değerler: ', array([0.54, 0. , 0.45, 0.33, 0.31]))
```

```
('Çok terimli Regresyon Verim Çıktısı: ', '[9.87552323]')
('Çok terimli Regresyon Kapasite Çıktısı: ', '[0.29104236]')
('Çok terimli Regresyon En İyi Değerler: ', array([0.8 , 0. , 0.95, 0.33, 0.12]))
```

```
('Rassal Orman Verim Çıktısı: ', '[9.03299665]')
('Rassal Orman Kapasite Çıktısı: ', '[0.30230483]')
('Rassal Orman En İyi Değerler: ', array([0.54, 0. , 0.15, 0.33, 0.13]))
```

'tamamlandı'

Normalize dış sıcaklık değerinin 0 olduğu nokta için, iç ünite fanı, dış ünite fanı, eev ve frekans değerleri geniş aralıklarda taranmıştır. Kodun çalışabilmesi için normalize değerler burada 100 ile çarpılmıştır.

```
index=0
for iduFan in range (0,71,2):
    for oduFan in range (0,61,2):
        for eev in range (0,101,5):
            for frekans in range (0,10,1):
                index=index+1

newBtuValsLinear=np.zeros((index,1))
newWwValsLinear=np.zeros((index,1))

newBtuValsPolynomial=np.zeros((index,1))
newWwValsPolynomial=np.zeros((index,1))

newBtuValsRandomForest=np.zeros((index,1))
newWwValsRandomForest=np.zeros((index,1))

allInput = np.zeros( (5,index) )
```

```
index=0
for iduFan in range (0,71,2):
    for oduFan in range (0,61,2):
        for eev in range (0,101,5):
            for frekans in range (0,10,1):
                disSic=1
                # Normalize değerlere geri dönebilmek için değerler burada 100'e bölünmüştür.
                inputData = np.zeros( (5,1) )
                inputData[0]=iduFan/100
                inputData[1]=oduFan/100
                inputData[2]=eev/100
                inputData[3]=disSic
                inputData[4]=frekans/100
                inputData=np.transpose(inputData)
                allInput[:,index]=inputData
                newBtuValsLinear[index]=regr_btu.predict(inputData)
```

```
newWwValsLinear[index]=regr_ww.predict (inputData)

newBtuValsPolynomial[index]=poly_model_btu.predict(poly_feature
s.fit_transform(inputData))
newWwValsPolynomial[index]=poly_model_ww.predict
(poly_features.fit_transform(inputData))

newBtuValsRandomForest[index]=random_btu.predict(inputData)
newWwValsRandomForest[index]=random_ww.predict (inputData)
index=index+1
```

Kapasite değerinin 0'ın üzerindeki normalize değerleri arasında 3 ayrı yöntem için bulunan verim, kapasite ve bunlara karşılık gelen parametre değerleri aşağıdaki gibi bulunmuştur.

```
valX=0
```

```
valY=1
```

```
sonucGetir(newBtuValsLinear,newWwValsLinear,newBtuValsPolynomial,newWw
ValsPolynomial,newBtuValsRandomForest,newWwValsRandomForest,valX,valY)
```

```
('Doğrusal Regresyon Verim Çıktısı: ', '[14.28217713]')
```

```
('Doğrusal Regresyon Kapasite Çıktısı: ', '[0.09517196]')
```

```
('Doğrusal Regresyon En İyi Değerler: ', array([0. , 0. , 0. , 0. , 0.09]))
```

```
('Çok terimli Regresyon Verim Çıktısı: ', '[14.67291414]')
```

```
('Çok terimli Regresyon Kapasite Çıktısı: ', '[0.02460581]')
```

```
('Çok terimli Regresyon En İyi Değerler: ', array([0., 0., 0., 0., 0.]))
```

```
('Rassal Orman Verim Çıktısı: ', '[14.87366624]')
```

```
('Rassal Orman Kapasite Çıktısı: ', '[0.01713674]')
```

```
('Rassal Orman En İyi Değerler: ', array([0. , 0. , 0.1, 0. , 0. ]))
```

```
'tamamlandı'
```


EK C

Bu bölümde ihtiyaç duyulan kütüphaneler eklenmiştir.

```
import pandas as pd
import numpy as np
```

```
from math import sqrt
```

```
import warnings
warnings.filterwarnings("ignore")
```

```
from keras.callbacks import ModelCheckpoint
from keras.layers import Dense, Activation, Flatten
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
```

```
import seaborn as sb
import matplotlib.pyplot as plt
import numpy as np
import warnings
```

```
import keras
from keras.models import Sequential, Model
from keras.optimizers import Adam
```

#Aşağıdaki kısımlar EK B'deki ile aynı şekilde tanımlanmıştır.

```
data = pd.read_excel(io = "./DataAll.xlsx")
k = np.transpose(data)
```

```
x = k[[0,1,2,3,4]]
y = k[[5,6]]
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y, train_size = 0.80)
```

```
train_data=x_train
test_data=x_test
train_label=y_train[[5,6]]
test_label=y_test[[5,6]]
```

```
NN_model = Sequential()
```

#Burada yapay sinir ağı modeli tanımlanmış ve 5 girdisi olan giriş katmanını belirlenmiştir.

```
NN_model = Sequential()
NN_model.add(Dense(5, activation='linear', input_shape=(5,)))
```

#Gizli katmanlar verilmiştir.

```
NN_model.add(Dense(10, activation='linear'))
```

```
NN_model.add(Dense(15, activation='linear'))
NN_model.add(Dense(10, activation='linear'))
NN_model.add(Dense(5, activation='linear'))
```

Verim ve kapasite olmak üzere 2 adet çıktı olduğu için burada çıktı katmanı ve 2 adet çıkış tanımlanmıştır.

```
NN_model.add(Dense(2, activation='linear', name="cikti"))
```

Neural network modelinin hesaplamasında kullanacağı hata metriği belirlenmiştir ve ayrıca optimizasyon için kullanılacak optimizasyon algoritması Adams Optimizer olarak seçilmiştir.

```
NN_model.compile(loss='mean_absolute_error', optimizer = Adam())
```

```
history = NN_model.fit(train_data, train_label, batch_size=1, epochs=1000,
verbose=1,
validation_data=(test_data, test_label))
```

Öğretilen model ile test ve öğrenme verilerinin hatalarının hesaplanması için veriler modele tekrar sokulmuştur.

```
pred_out_test = NN_model.predict(x_test)
pred_out_train = NN_model.predict(x_train)
```

Verilerin çıktıları ve hatalarının bu bölümde yazdırılır.

```
rmse_test_btu = sqrt(mean_squared_error(pred_out_test[:,0],y_test[5]))
rmse_test_verim = sqrt(mean_squared_error(pred_out_test[:,1],y_test[6]))
```

```
rmse_train_btu = sqrt(mean_squared_error(pred_out_train[:,0],y_train[5]))
rmse_train_verim = sqrt(mean_squared_error(pred_out_train[:,1],y_train[6]))
```

```
print('Test verisi btu hatası: ' + str(rmse_test_btu))
print('Test verisi verim hatası: ' + str(rmse_test_verim))
```

```
print('Test verisi btu ortalama hata (%) : ' + str(rmse_test_btu/(k[5].mean()*100) +
'%')
```

```
print('Test verisi verim ortalama hata (%) : ' +
str(rmse_test_verim/(k[6].mean()*100) + '%')
```

```
print('Öğrenme verisi btu hatası: ' + str(rmse_train_btu))
print('Öğrenme verisi verim hatası: ' + str(rmse_train_verim))
```

```
print('Öğrenme verisi btu ortalama hata (%) : ' +
str(rmse_train_btu/(k[5].mean()*100) + '%')
print('Öğrenme verisi verim ortalama hata (%) : ' +
str(rmse_train_verim/(k[6].mean()*100) + '%')
```

```
Test verisi btu hatası: 0.03983654051509484
Test verisi verim hatası: 0.6160504296989606
```

Test verisi btu ortalama hata (%) : 6.545456731162977%
Test verisi verim ortalama hata (%) : 10.648631920714447%
Öğrenme verisi btu hatası: 0.02947245783229067
Öğrenme verisi verim hatası: 0.5388932828247474
Öğrenme verisi btu ortalama hata (%) : 4.842556482262468%
Öğrenme verisi verim ortalama hata (%) : 9.314945557542048%

Normalize dış sıcaklık değerinin 1 olduğu nokta için, iç ünite fanı, dış ünite fanı, eev ve frekans değerleri diğer yöntemlerle aynı aralıklarda taranmıştır. Kodun çalışabilmesi için normalize değerler burada 100 ile çarpılmıştır.

```
index35=0
for iduFan in range (74,101,2):
    for oduFan in range (54,101,2):
        for eev in range (0,71,5):
            for frekans in range (75,101,1):
                index35=index35+1
```

```
newVals35=np.zeros((index35,2))
allInput35 = np.zeros( (5,index35) )
```

```
index35=0
for iduFan in range (74,101,2):
    for oduFan in range (54,101,2):
        for eev in range (0,71,5):
            for frekans in range (75,101,1):
                disSic=1
```

Normalize değerlere geri dönebilmek için değerler burada 100'e bölünmüştür.

```
inputData = np.zeros( (5,1) )
inputData[0]=iduFan/100
inputData[1]=oduFan/100
inputData[2]=eev/100
inputData[3]=disSic
inputData[4]=frekans/100
inputData=np.transpose(inputData)
allInput35[:,index35]=inputData
newVals35[index35,:]=NN_model.predict(inputData)
index35=index35+1
```

Alınan değerler içerisinde en iyi sonucun bulunması için tanımlanan fonksiyon yazılmıştır.

```
def getNNresult(newVals,allInput,lowPoint,highPoint,indexT):
    newBtuVals=newVals[:,0]
    newWwVals=newVals[:,1]
```

```
index2=0
for x in range(indexT):
    if newBtuVals[x]<highPoint and newBtuVals[x]>lowPoint :
```

```

index2=index2+1

refinedBtu=np.zeros( (index2,1) )
refinedWw=np.zeros( (index2,1) )
refinedInput=np.zeros( (5,index2) )

index2=0
for x in range(indexT):
    if newBtuVals[x]<highPoint and newBtuVals[x]>lowPoint :
        refinedBtu[index2]=newBtuVals[x]
        refinedWw[index2]=newWwVals[x]
        refinedInput[:,index2]=allInput[:,x]
        index2=index2+1

print(("NN Tarafından Bulunan Btu" ,refinedBtu[refinedBtu.argmax()]))
print(("NN Tarafından Bulunan Verim" ,refinedWw[refinedWw.argmax()]))
print(refinedInput[:,refinedWw.argmax()])

return "Tamamlandı"

# Tüm işlemler dış ortam sıcaklığının normalize değerinin 0,66 olduğu nokta için
tekrarlanmıştır.
index30=0
for iduFan in range (64,91,1):
    for oduFan in range (44,91,1):
        for eev in range (0,71,5):
            for frekans in range (40,45,1):
                index30=index30+1

newVals30=np.zeros((index30,2))
allInput30 = np.zeros( (5,index30) )
index30=0
for iduFan in range (64,91,1):
    for oduFan in range (44,91,1):
        for eev in range (0,71,5):
            for frekans in range (40,45,1):
                disSic=0.66
                inputData = np.zeros( (5,1) )
                inputData[0]=iduFan/100
                inputData[1]=oduFan/100
                inputData[2]=eev/100
                inputData[3]=disSic
                inputData[4]=frekans/100
                inputData=np.transpose(inputData)
                allInput30[:,index30]=inputData
                newVals30[index30]=NN_model.predict (inputData)
                index30=index30+1

```

Tüm işlemler dış ortam sıcaklığının normalize değerinin 0,33 olduğu nokta için tekrarlanmıştır.

```
index25=0
for iduFan in range (54,81,2):
    for oduFan in range (0,71,2):
        for eev in range (0,101,5):
            for frekans in range (10,32,1):
                index25=index25+1

newVals25=np.zeros((index25,2))
allInput25 = np.zeros( (5,index25) )
index25=0
for iduFan in range (54,81,2):
    for oduFan in range (0,71,2):
        for eev in range (0,101,5):
            for frekans in range (10,32,1):
                disSic=0.33
                inputData = np.zeros( (5,1) )
                inputData[0]=iduFan/100
                inputData[1]=oduFan/100
                inputData[2]=eev/100
                inputData[3]=disSic
                inputData[4]=frekans/100
                inputData=np.transpose(inputData)
                allInput25[:,index25]=inputData
                newVals25[index25]=NN_model.predict(inputData)
                index25=index25+1
```

Tüm işlemler dış ortam sıcaklığının normalize değerinin 0 olduğu nokta için tekrarlanmıştır.

```
index20=0
for iduFan in range (0,71,2):
    for oduFan in range (0,61,2):
        for eev in range (0,101,5):
            for frekans in range (0,10,1):
                index20=index20+1

newVals20=np.zeros((index20,2))
allInput20 = np.zeros( (5,index20) )

index20=0
for iduFan in range (0,71,2):
    for oduFan in range (0,61,2):
        for eev in range (0,101,5):
            for frekans in range (0,10,1):
                disSic=0
                inputData = np.zeros( (5,1) )
                inputData[0]=iduFan/100
                inputData[1]=oduFan/100
                inputData[2]=eev/100
```

```
inputData[3]=disSic
inputData[4]=frekans/100
inputData=np.transpose(inputData)
allInput20[:,index20]=inputData
newVals20[index20]=NN_model.predict(inputData)
index20=index20+1
```

Bu kısımda tüm dış ortam sıcaklıklarındaki maksimum verim ve bu verime karşılık gelen kapasite değeri ayrıca bunları oluşturan giriş parametreleri bulunmuştur.

```
print('35 °C için sonuçlar: ')
```

```
getNNresult(newVals35,allInput35,0.95,1,index35)
```

```
print('30 °C için sonuçlar: ')
```

```
getNNresult(newVals30,allInput30,0.6,0.7,index30)
```

```
print('25 °C için sonuçlar: ')
```

```
getNNresult(newVals25,allInput25,0.29,0.34,index25)
```

```
print('20 °C için sonuçlar: ')
```

```
getNNresult(newVals20,allInput20,0,1,index20)
```

Yapay sinir ağı modeli için sonuçlar aşağıdaki gibidir.

35 °C için sonuçlar:

```
('NN Tarafından Bulunan Btu', array([0.99999958]))
```

```
('NN Tarafından Bulunan Verim', array([4.06473875]))
```

```
[0.74 0.54 0.05 1. 1. ]
```

30 °C için sonuçlar:

```
('NN Tarafından Bulunan Btu', array([0.62022835]))
```

```
('NN Tarafından Bulunan Verim', array([5.07623816]))
```

```
[0.85 0.44 0. 0.66 0.44]
```

25 °C için sonuçlar:

```
('NN Tarafından Bulunan Btu', array([0.33999974]))
```

```
('NN Tarafından Bulunan Verim', array([10.22463608]))
```

```
[0.54 0. 0. 0.33 0.31]
```

20 °C için sonuçlar:

```
('NN Tarafından Bulunan Btu', array([0.12086993]))
```

```
('NN Tarafından Bulunan Verim', array([14.30663395]))
```

```
[0. 0. 0. 0. 0.09]
```

```
'Tamamlandı'
```

ÖZGEÇMİŞ



Ad Soyad: Ezgi Sevgi Özdemir
E-posta: ezgizdmr@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2017, İstanbul Teknik Üniversitesi, Makine Fakültesi, Makine Mühendisliği

MESLEKİ DENEYİM

- 2018-Halen Arçelik-LG Klima Sanayi ve Ticaret Lim. Şti.'nde AR-GE Sistem Tasarım Uzman Mühendis