



# ULUBORLU MESLEKİ BİLİMLER DERGİSİ (UMBD)

Uluborlu Journal of Vocational Sciences

<http://dergipark.gov.tr/umbd>

## MAKİNE ÖĞRENİMİ VE HİPER PARAMETRE OPTİMİZASYONU İLE BİNALARDAKİ ENERJİ VERİMLİLİĞİ ANALİZİ

Hüseyin Furkan ZENGİN<sup>1\*</sup>, Sinem BOZKURT KESER<sup>2</sup>

<sup>1</sup>Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, Eskişehir, Türkiye.

<sup>2</sup>Eskişehir Osmangazi Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Eskişehir, Türkiye.

\*Sorumlu Yazar: 503020210015@ogrenci.ogu.edu.tr

(Geliş/Received: 24.10.2023; Kabul/Accepted:19.11.2023)

**ÖZET:** Akıllı binaların sayısındaki artışın ve teknolojiye ilerlemenin bir sonucu olarak binalardaki enerji tüketimi giderek daha önemli bir hale gelmeye başlamıştır. Binalardaki enerji tüketimi tahmini, enerji verimliliği açısından kritik bir öneme sahiptir. Bu çalışmada, binalardaki enerji verimliliğini artırmak amacıyla Python programlama dilinde bulunan scikit-learn kütüphanesindeki çeşitli regresyon modelleri kullanıldı. Çalışmada, eğitim için hazır hale getirilmiş olan 768 satır ve 8 özellik içeren bir veri kümesi kullanıldı. Veriler önce normalizasyon işlemine tabi tutularak eğitime uygun hale getirildi. Çalışma, kütüphanede bulunan tüm modellerin varsayılan parametre değerleri kullanılarak eğitilmesiyle başladı. Daha sonra, bu modellerin performansları arasında MSE değeri 1,5 altında olan modeller belirlendi. Bu modeller, daha fazla iyileştirme için hiper-parametre optimizasyonuna tabi tutuldu. Bunun sonucunda, CatBoost regresyonu ve Bayes-Rastgele Orman hiper-parametre optimizasyonunun bir arada kullanılması ile 0,997 R2 ve 0,23 MSE skorları bu çalışmadaki en yüksek başarı sonucu olarak elde edildi. Bu çalışma, akıllı binaların yaygınlaşmasıyla birlikte enerji verimliliğini artırmak için Makine Öğrenmesi tekniklerini ve hiper-parametre optimizasyonunu kullanmanın potansiyelini göstermektedir. Teknolojik ilerlemeye paralel olarak, bu tür çalışmalar inşaat ve mimari alanında daha yeşil, sürdürülebilir ve çevre dostu yapıların geleceği için umut vaat etmektedir.

**Anahtar Kelimeler:** Enerji Verimliliği Tahmini, Hiper-Parametre, Makine Öğrenmesi, Optimizasyon, Regresyon.

## ENERGY EFFICIENCY ANALYSIS IN BUILDINGS WITH MACHINE LEARNING AND HYPER PARAMETER OPTIMIZATION

**ABSTRACT:** As a result of the increase in the number of smart buildings and the development in technology, energy consumption in buildings has become increasingly important. Estimation of energy consumption in buildings is of critical importance for energy efficiency. In this study, various regression models in the scikit-learn library in the Python programming language were used to increase energy efficiency in buildings. A dataset containing 768 lines and 8 features, which was prepared for training, was used in the study. The data was first subjected to normalization and made suitable for training. The work started by training all models in the library using default parameter values. Then, among the performances of these models, models with MSE values below 1.5 were identified and subjected to hyper-parameter optimization for further improvement. As a result, by using CatBoost regression and BA-RF hyper-parameter optimization together, 0.997 R2 and 0.23 MSE scores were obtained as the highest success result in this study. This study shows the potential of using Machine Learning techniques and hyper-parameter optimization to improve energy efficiency as smart buildings become more widespread.

**Keywords:** Energy Efficiency Prediction, Hyper-Parameter, Machine Learning, Optimization, Regression.

## 1. GİRİŞ

Tüm dünyada enerji kullanımı her sektörde yaygınlaşmış durumdadır. Enerji kullanımının artmasıyla beraber enerji alanında yapılan çalışmalar da artmaktadır. Bu çalışmalardan biri de konutlardaki enerji verimliliğini üst seviyeye çıkarmaktır. Mümkün olan en az enerji kullanımı ile binaların ısıtma ve soğutma yüklerinin en yüksek verimlilikte olması hem enerji tasarrufunda hem de sera gazı emisyonu üretiminin azaltılmasında önemli bir rol oynamaktadır [1].

Ev, işyeri vb. günlük yaşam alanlarında akıllı sistemlerin kullanımının artmasıyla beraber binalarda enerji tüketimi de artmaktadır. Dünyada yapay zekânın kullanımının gelişmesiyle beraber artık insan yaşamının birçok noktasında yapay zekâ uygulamaları çıkmaya başlamıştır. Birçok alanda olduğu gibi enerji verimliliğinde de yapay zekâ kilit bir rol oynamaktadır. Enerji tüketimi tahmininde özellikle Yapay Zekâ ve Makine Öğrenmesi teknikleri yaygın olarak kullanılmaktadır. Binalardaki ısıtma ve soğutma yükleri kullanımının bina özelliklerine göre değişiklik göstermesi bu tarz tekniklerin kullanımını önemli hale getirmektedir.

Enerji verimliliğini maksimuma çıkarmak adına, binalardaki ısıtma ve soğutma yüklerinin tahmininde Makine Öğrenmesi tekniklerinin başarılı oldukları görülmektedir. Bir binada bulunan dairelerin yapısı ve düzeni enerji tüketiminde etkili olmaktadır [2]. Bunların yanı sıra, binanın bulunduğu yüzey alanından, binada kullanılan duvarların ve camların alanına kadar birçok faktör bulunmaktadır. Isıtma yükü, oda sıcaklığını normal seviyede tutmak için gerekli olan ısı miktarı olarak tanımlanırken, soğutma yükü ise, dışarıya atılması gereken enerjiyi temsil etmektedir [3].

Akıllı binalarda enerji verimliliği için binaların şekli ve cephesinin yanı sıra, daire içerisindeki nem miktarı, hava kalitesi ve ortam sıcaklığı gibi değerler de dikkate alınarak kontrol parametrelerinde iyileştirme gibi çalışmalar yürütülmektedir. Makine Öğrenimi teknikleri, istatistiksel yöntemlerin temelinden oluşturulduğu için özellikle bu tür tahmin gerektiren çalışmalar için oldukça uygundur.

Bu çalışmada kullanılan ‘UCI Energy Efficiency Dataset’ veri kümesi, bu alanda yapılan birçok çalışmada kullanılmıştır [4]. Her çalışma, birbirlerinden farklı modeller ve farklı ön işleme yöntemleri kullanmaktadır. Bu çalışmada, diğer çalışmalardan farklı olarak hiper-parametre optimizasyonları yardımıyla kullanılan modellerin performanslarını artırmak ve farklı optimizasyon yöntemlerini karşılaştırmak amaçlanmıştır. Literatürde, bu alanda ve bu veri kümesinde hiper-parametre optimizasyon yöntemlerinin kullanımı ve performans karşılaştırmalarının yapıldığı bir çalışma bulunmamaktadır. Literatürde, enerji verimliliği tahmininde makine öğrenimi yöntemlerinin kullanımı ile ilgili çalışmalar bulunmaktadır.

Moradzadeh ve diğerleri [5], çalışmalarında Çok Katmanlı Algılayıcılar (MLP, Multilayer Perceptron) ve Destek Vektör Regresyonu (SVR, Support Vector Regression) modellerini kullanarak ısıtma ve soğutma yüklerinin tahmin performanslarını hesaplamışlardır. Abdurrahim Akgündoğdu [6], konut binalarının ısıtma ve soğutma yüklerinin tahmini için 6 farklı regresyon algoritmasının sonuçlarını karşılaştırmıştır ve en iyi tahmin sonuçlarını Uyarlamalı Sinirsel Bulanık Çıkarım Sisteminin (ANFIS) regresörü vermiştir. Reddy ve Kumar [7], yaptıkları çalışmada, UCI Enerji Verimliliği (Energy Efficiency) veri kümesini 19 farklı regresyon algoritması üzerinde herhangi bir ön işleme ve optimizasyon yöntemi kullanmadan test etmişlerdir.

Pachauri ve Ahn, çalışmalarında En Küçük Kareler Artırma (LS-Boosting, Least-squares boosting) algoritmasına dayanan ve Regresyon Ağacı Topluluğu (RTE, Regression Tree Ensemble) olarak adlandırılan algoritmayı, Kurbağa Sıçrama Algoritması (SFLA, Shuffled Frog Leaping) optimizasyon yöntemi ile birleştirerek bir yöntem oluşturmuşlardır [8]. Wei Gao ve arkadaşlarının [9], Weka ortamında 16 regresyon modeli üzerinde eğitim sonuçlarını karşılaştırmışlardır. Rastgele Orman (Random Forest) algoritması en iyi sonuca sahip olmuştur.

Moayedi ve diğerleri, 2019 yılında yaptıkları çalışmada toplam 6 regresyon modeli ile performans karşılaştırması yapmışlardır [10]. Yapılan çalışma sonucunda ısıtma yükündeki en başarılı sonucu Random Forest algoritması vermiştir. Goyal ve diğerleri, 2020 yılında hem Makine Öğrenmesi algoritmalarını hem de Ensemble algoritmalarını kullanarak binaların ısıtma ve soğutma yüklerinin tahmini için karşılaştırmalı bir çalışma yürütmüşlerdir [1]. Elde edilen sonuçlarda Gradyan Artırma algoritması en başarılı algoritma olmuştur. Al-Rakhami ve arkadaşları [2], 2019 yılındaki çalışmalarında enerji yüklerinin tahmini için XGBoost algoritmasını, çalışmada kullanılan UCI Energy Efficiency veri kümesi ile kullanarak yüksek bir tahmin sonucu elde etmişlerdir.

Bu çalışmanın amacı, binalardaki enerji verimliliğini artırmak için ısıtma ve soğutma yüklerini en doğru şekilde tahmin eden regresyon modelini ve hiper-parametre optimizasyonu yöntemini bulmaktır.

Hiper-parametreler, makine öğrenmesi algoritmalarının davranışlarını doğrudan olarak etkilemektedir. Uygun parametre değerlerini bulmak uzmanlık gerektiren bir iş olduğundan çoğu durumda kaba kuvvet (manuel olarak) olarak adlandırılan yöntem ile optimum performansı sağlayacak parametre değerleri bulunmaya çalışılmaktadır. Bu işlem ise zaman alan ve uzman bilgisi gerektiren bir iştir [11]. Bu nedenle, bu çalışmada çeşitli araçlar ve kütüphaneler kullanarak hiper-parametre optimizasyonunun bir sistem doğrultusunda gerçekleştirilmesi hedeflenmektedir. Bu doğrultuda, optimizasyon yöntemlerinin karşılaştırmalı analizi ile en uygun yöntemin belirlenmesi üzerine deneyler gerçekleştirilmiştir. Tablo 1’de benzer çalışmaları içeren literatür araştırmasının tablo gösterimi yer almaktadır.

**Tablo 1.** Benzer çalışmaları içeren literatür araştırması

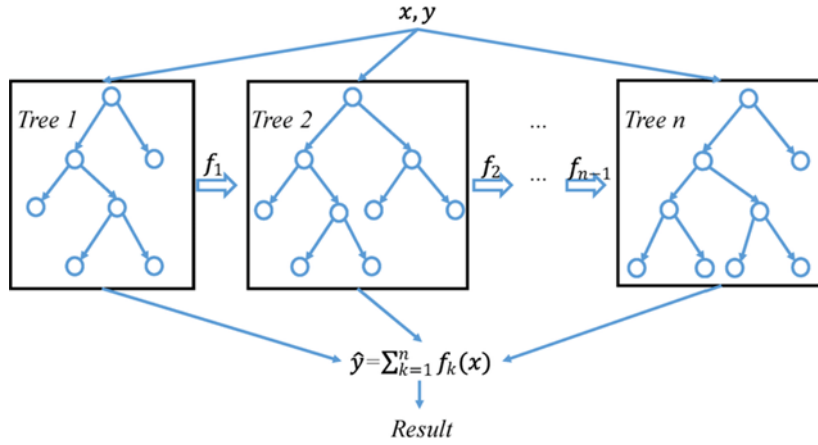
Referans	Yöntem	Veri Kümesi	En İyi Skor
[5]	MLP, SVR	UCI Energy Efficiency	0.78 MSE Isıtma, 3.02 MSE Soğutma - SVR
[15]	LR, MLP, RBF, SVR, GP, ANFIS	UCI Energy Efficiency	0.68 RMSE Isıtma, 1.60 RMSE Soğutma - ANFIS
[7]	LR, SVR, Fine Tree, Medium Tree, Coarse Tree, Boosted Trees, Bagged Trees, GP	UCI Energy Efficiency	0.46 RMSE Isıtma, 1.24 RMSE Soğutma - Matern Kovaryansı ile kullanılmış GP
[8]	LS-Boosting, RTE, SFLA	UCI Energy Efficiency, Simulation Data	0.20 MSE Isıtma, 0.47 MSE Soğutma
[9]	RF, RDT, Lazy K-Star, AMT, MLP, M5Rules, DPC, MPR, XNV, RBFR, MLR,GPR,SMOR,LMSR,LWL,EN	UCI Energy Efficiency	0.23 RMSE Isıtma, 0.62 RMSE Soğutma - RF
[10]	MLP, RBF, ElasticNet, RF, Tree Model, LWL	UCI Energy Efficiency	0.46 RMSE - RF
[1]	MLP, KNN, SVR, RF, GBM, XGBoost	UCI Energy Efficiency	0.23 MSE Isıtma, 3.20 MSE Soğutma - GBM
[2]	XGBoost	UCI Energy Efficiency	0.26 RMSE Isıtma, 0.46 MSE Soğutma

Makalenin 2. Bölümünde, çalışmada kullanılmış olan materyal ve metotlar detaylandırılmıştır. Çalışmada kullanılan regresyon modelleri ve hiper-parametre optimizasyon yöntemleri bu bölümde açıklanmıştır. 3. Bölümde, çalışmada kullanılan veri kümesi ve deneysel sonuçlar ile ilgili açıklamalar yer almaktadır. 4. Bölümde çalışma sonucu ve gelecekte yapılabilecek çalışmalarla ilgili açıklamalar yer almaktadır.

## 2. MATERYAL VE METOT

### 2.1. Ekstrem Gradyan Artırma (XGBoost, eXtreme Gradient Boosting)

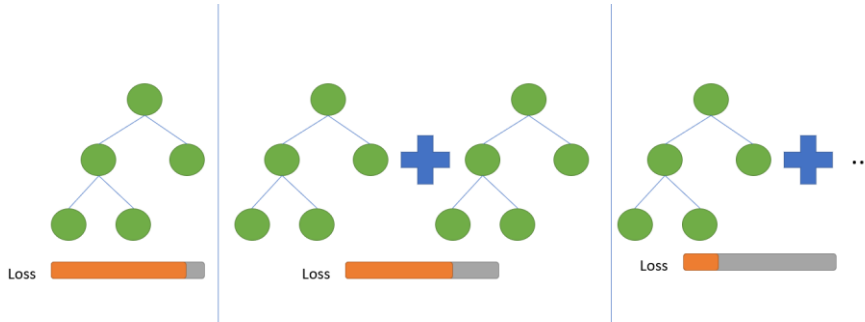
XGBoost veya diğer adıyla eXtreme Gradient Boosting, algoritması Tianqi, Chen ve Carlos Guestrin' in 2016 yılında çıkardıkları "XGBoost: A Scalable Tree Boosting System" adlı makale ile ortaya çıkmıştır [12]. XGBoost, Karar Ağacı temelli çalışan bir gradyan artırma tekniği kullanılmaktadır. Tianqi'ye göre diğer Makine Öğrenmesi algoritmalarına oranla 10 kat daha hızlıdır. Bu algoritma oluşan "hata"ları Kara Ağaçlarına göndererek hatayı minimuma indirmeyi amaçlamaktadır. Birçok çalışmada gösterdiği başarılı tahminler ile akademik olarak tercih sebebi olmaktadır. Algoritmanın genel mimarisi Şekil 1'de gösterilmiştir.



Şekil 1. Ekstrem Gradyan Artırma genel mimarisi [13]

## 2.2. Kategori Artırma (CatBoost, Category Boosting)

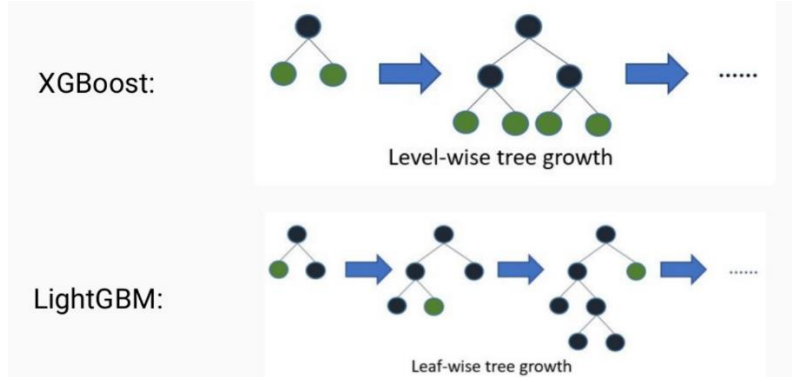
CatBoost algoritması, Yandex tarafından geliştirilen bir makine öğrenmesi algoritmasıdır. Gradient Boosting tabanlı olan algoritma, 2018 yılında XGBoost ve LightGBM algoritmalarına alternatif olarak Prokhorenkova ve arkadaşları tarafından yayınlanan bir makale ile ortaya çıkmıştır [14]. Diğer algoritmalarından farklı olarak sıralı artırma yöntemi uygulanmıştır. Ayrıca tahmin kayması probleminin de bu sayede çözülebileceği belirtilmiştir. Tahmin kayması problemi eğitilen veriler ile test verileri arasındaki dağılımın farklı olmasından meydana gelmektedir. Bu da “overfitting” adı verilen aşırı öğrenmeye yol açmaktadır. Şekil 2’de CatBoost algoritmasının çalışma mantığı gösterilmiştir.



Şekil 2. Kategori Artırma çalışma mantığı [15]

## 2.3. Hafif Gradyan Artırma Makineleri (LightGBM, Light Gradient Boosting Machines)

LightGBM, 2017 yılında geliştirilmiş olan bir gradient boosting algoritmasıdır. Guolin Ke ve arkadaşları tarafından “LightGBM: A Highly Efficient Gradient Boosting Decision Tree” adlı makale ile ortaya çıkmıştır [16]. Histogram tabanlı çalışan algoritma, sürekli değerlere sahip olan değişkenleri kesikli yaparak maliyeti azaltmaktadır. Karar Ağaçları, öğrenmede iki strateji kullanabilir. Bunlar level-wise (seviye odaklı) ve leaf-wise (yaprak odaklı) stratejilerdir. LightGBM algoritması, leaf-wise stratejiyi kullanarak daha az hata oranına sahip olmaktadır. Leaf-wise strateji, az sayıda veri bulunduğu zaman aşırı öğrenmeye yol açtığı için çok sayıda veri ile kullanılması daha uygundur. LightGBM ve XGBoost algoritmaları arasındaki çalışma mantığı farkı Şekil 3’te gösterildiği gibidir.



Şekil 3. Ekstrem Gradyan Artırma ve Hafif Gradyan Artırma Makineleri arasındaki çalışma mantığı farkı [17]

### 2.3. Izgara Arama (Grid Search) Optimizasyonu

Grid Search (Izgara Arama) optimizasyon algoritması, hiper-parametre optimizasyonunda en popüler yöntemlerden bir tanesidir. Belirlenen hiper-parametre uzayında her kombinasyonu modele uygulayarak en optimal sonucu elde etmeye çalışmaktadır. Hiper-parametre değerlerinin sınırları manuel olarak kullanıcı tarafından ayarlanmaktadır. Belirlenen sınırlar dahilindeki tüm kombinasyonlar teker teker denir. Oluşturulan hiper-parametre uzayındaki her kümenin birbiriyle Kartezyen çarpımları model üzerinde eğitilir. Daha sonra Grid Search, en yüksek başarıyı sağlayan parametreleri belirler.

### 2.4. Bayes Optimizasyonu

Bayesian Optimization (BO), daha önce elde ettiği sonuçlardan yola çıkarak yapacağı tahminleri belirleyen bir algoritmadır. Surrogate model (SM) ve acquisition function (AF) adı verilen iki ana bileşenden oluşmaktadır. SM, önceden test edilen parametrelerin sonuç dağılımlarını AF' a verir. AF ise bu dağılımlardan yola çıkarak daha iyi bir tahmin için sonuçları dengelemeye çalışır. BO için en yaygın 3 SM bu çalışmada değerlendirilmek üzere kullanılmıştır. Bunlar Gauss Process (GP), Random Forest (RF) ve Tree Parzen Estimator (TPE)' dir.

### 2.5. Gauss İşlemi (GP, Gauss Process)

GP, lineer olmayan bir modeldir. BO için en çok tercih edilen varsayılan modellerden bir tanesidir. Eğitim sırasındaki belirsizliğin ölçülmesini sağlar. GP' in en çok tercih edilen model olmasının sebebi ise maksimum entropi dağılımının ön bilgi oranını en aza indirgeyerek belirsizliği oldukça fazla azaltmasıdır. Ancak GP' in bir dezavantajı ise yüksek boyutlu verilerde düşünüldüğü kadar iyi sonuç vermemesidir. Bu yüzden genellikle düşük boyutlu verilerde tercih edilmiştir.

### 2.6. Rastgele Orman (RF, Random Forest)

RF algoritması, birden çok karar ağacının farklı örnekler üzerinde eğitilmesi ile oluşturulur. Bu sayede çeşitli modeller üretilerek en optimum sonuç bulunmaya çalışılır. Oluşan her bir karar ağacı en iyi tahmin sonucu için oylanır. Oylanın karar ağaçlarının sonuçlarından yola çıkarak en iyi tahmin sonucu bulunmuş olur. Buradaki en önemli özellik ise ağaç derinliğinin iyi seçilmesidir.

## 2.7. Ağaç Yapılı Parzen Tahmincisi (TPE, Tree Parzen Estimator)

TPE, tahmin dağılımına dayalı bir algoritma değildir. Belirtilen koşullu bağımlılığı koruyarak her süreç için iyileştirmeyle sonuçlanan bir ağaç yapısı sunmaktadır. 2 adet hiyerarşik model oluşturur. Bunlar  $l(x)$  ve  $g(x)$ ' tir. Eğer amaç fonksiyon  $y^*$  değişkeninin altında veya üstünde kalırsa model buna göre şekillenir. TPE algoritmasının formülü Eşitlik 1'de verilmiştir.

$$p(x | y, D) = l(x) \text{ if } y < y^* \text{ if, else } g(x) \quad (1)$$

## 3. BULGULAR VE TARTIŞMA

### 3.1 Veri Kümesi

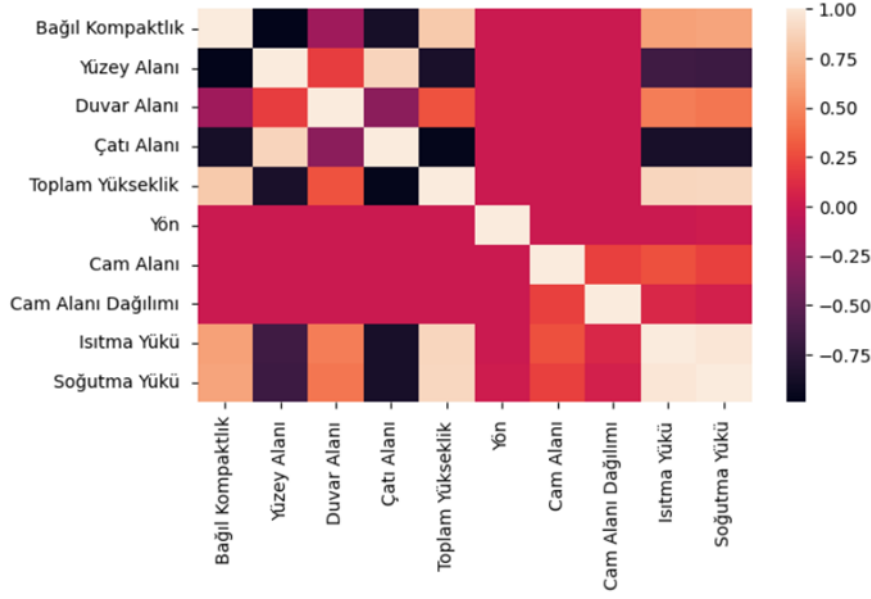
Çalışmada, UCI Energy Efficiency veri kümesi kullanılmıştır [4]. Veri kümesi Ecotect kullanılarak 12 farklı bina yapısının simüle edilmesi ile oluşturulmuştur. 768 örnek ve 8 farklı özellikten oluşmaktadır. Veri kümesinde yer alan özellikler Tablo 2'de, veri kümesinin dağılımı ise Tablo 3'te gösterilmiştir. Çalışmada kullanılan veri kümesine ait ısı haritası ise Şekil 4'te gösterilmiştir.

**Tablo 2.** Veri kümesi özellikleri

Sütun Adı	Özellik
X1	Bağıl Kompaktlık (Relative Compactness)
X2	Yüzey Alanı (Surface Area)
X3	Duvar Alanı (Wall Area)
X4	Çatı Alanı (Roof Area)
X5	Toplam Yükseklik (Overall Height)
X6	Yön (Orientation)
X7	Cam Alanı (Glazing Area)
X8	Cam Alanı Dağılımı (Glazing Area Distribution)
y1	Isıtma Yüğü (Heating Load)
y2	Soğutma Yüğü (Cooling Load)

**Tablo 3.** Veri kümesinin dağılım özeti

	X1	X2	X3	X4	X5	X6	X7	X8	y1	y2
count	768.00	768.00	768.00	768.00	768.00	768.00	768.00	768.00	768.00	768.00
mean	0.764167	671.71	318.50	176.0	5.25	3.50	0.234375	2.81	22.31	24.59
std	0.105777	88.09	43.63	45.17	1.75	1.12	0.133221	1.55	10.09	9.51
min	0.620000	514.50	245.00	110.25	3.50	2.00	0.000000	0.000000	6.01	10.90
25%	0.682500	606.38	294.00	140.88	3.50	2.75	0.100000	1.75	12.99	15.62
50%	0.750000	673.75	318.50	183.75	5.25	3.50	0.250000	3.00	18.95	22.08
75%	0.830000	741.13	343.00	220.50	7.00	4.25	0.400000	4.00	31.67	33.13
max	0.980000	808.50	416.50	220.50	7.00	5.00	0.400000	5.00	43.10	48.03



Şekil 4. Çalışmada kullanılan UCI Energy Efficiency veri kümesine ait ısı haritası

Isı haritası, bir veri kümesindeki birden fazla değişken arasındaki ilişkiyi renk kodlu matris olarak görüntüleyen bir araçtır. Şekil 4'te yer alan ve bu çalışmada kullanılan veri kümesine ait olan ısı haritasında, açık renklerle temsil edilen kutucuklar, iki değişken arasında pozitif yönde güçlü bir ilişki olduğunu gösterirken, koyu renklerle temsil edilen kutucuklar, negatif yönde güçlü bir ilişki olduğunu göstermektedir. Yüzey ve çatı alanı değişkenlerinin, hedef değişkenler olan ısıtma ve soğutma yükleri ile arasında negatif yönde, toplam yükseklik değişkeninin ise pozitif yönde güçlü bir ilişkisinin olduğu söylenebilmektedir. Böylece, bu çalışmada kullanılan UCI Energy Efficiency veri kümesinin özellikleri arasındaki ilişkiler görselleştirilerek anlaşılabilirliği artırılmıştır.

### 3.2. Deneysel Sonuçlar ve Tartışma

Çalışmada, 12 farklı bina yapısından elde edilen 768 farklı örnek kullanılmıştır. Örnekler eğitilmeden önce veri kümesine Normalizasyon işlemi uygulanmıştır. Regresyon modelleri Python programlama dili kütüphanelerinden birisi olan scikit-learn kütüphanesine dahil olan modellerden seçilmiştir. Tahmin için 10 regresyon modeli seçilmiştir. Bu çalışmada aynı zamanda 4 farklı Hiper-parametre Optimizasyon yöntemi kullanılmıştır (Grid Search, Bayesian-Gauss Process, Bayesian-Random Forest, Bayesian-Tree Parzen Estimator).

Veriler başlangıçta, parametre değerleri değiştirilmeden eğitilmiştir. Veriler, yüzde 70 eğitim ve yüzde 30 test olacak şekilde ayrılmıştır. MSE (Mean Squared Error) skor değeri 1,5 ve altında olan modeller hiper-parametre optimizasyon yöntemleri ile beraber kullanılarak test edilmiştir. Aşağıda Şekil 5'te scikit-learn kütüphanesinde bulunan regresyon modellerin varsayılan parametre değerleri ile test sonuçları verilmiştir.



	Default
<b>XGBoost</b>	R2-Score = 0.9953871968525841 MSE = 0.39942730890325634
<b>CatBoost</b>	R2-Score = 0.9956385217861196 MSE = 0.3779968562488117
<b>LightGBM</b>	R2-Score = 0.9863305444900909 MSE = 1.1835744276034177
<b>Extra Trees</b>	R2-Score = 0.9808084471941609 MSE = 1.6515453252606205
<b>Random Forest</b>	R2-Score = 0.9822285122110108 MSE = 1.5308173098917806
<b>Lineer Reg</b>	R2-Score = 0.9001628811412268 MSE = 8.940205919775199
<b>Ridge</b>	R2-Score = 0.900984720777771 MSE = 8.869277548628338
<b>Lasso</b>	R2-Score = 0.7611055470784795 MSE = 21.606869302738183
<b>KNN</b>	R2-Score = 0.9037313250008457 MSE = 8.624893125541124
<b>SVR</b>	R2-Score = 0.9059030114928801 MSE = 8.411125582960683

Şekil 5. Varsayılan parametreler ile tüm modellerin test sonuçları

Şekil 5 incelendiğinde 10 model arasından MSE değeri 1,5 ve altında olan XGBoost, CatBoost ve LightGBM modelleri hiper-parametre optimizasyon yöntemleri ile kullanılmak üzere seçilmiştir. Modeller, ısıtma ve soğutma olarak ayrı ayrı eğitilmektedir. Elde edilen test sonuçları birleştirilerek modellerin genel başarısını gösteren tek bir sonuç elde edilmektedir. Şekil 6' da seçilen modeller ve hiper-parametre optimizasyon yöntemleri ile yapılmış olan test sonuçları yer almaktadır.

	XGBoost	CatBoost	LightGBM
<b>Default</b>	R2-Score = 0.9953871968525841 MSE = 0.39942730890325634	R2-Score = 0.9956385217861196 MSE = 0.3779968562488117	R2-Score = 0.9863305444900909 MSE = 1.1835744276034177
<b>Grid Search</b>	R2-Score = 0.9937790804039337 MSE = 0.5386089194925365	R2-Score = 0.05307574502681783 MSE = 90.68334489471461	R2-Score = 0.993220766194616 MSE = 0.6199639638185508
<b>Bayesian - GP</b>	R2-Score = 0.9954828384042511 MSE = 0.39240479913018733	R2-Score = 0.9972158192925058 MSE = 0.2570559616283685	R2-Score = 0.9953339629654173 MSE = 0.42960907640013934
<b>Bayesian - RF</b>	R2-Score = 0.9952447685155331 MSE = 0.41201243408386323	R2-Score = 0.9975227651802987 MSE = 0.23002354459448468	R2-Score = 0.9953048215927915 MSE = 0.4316252164668367
<b>TPE</b>	R2-Score = 0.996100738868018 MSE = 0.3397977323868261	R2-Score = 0.9968962166613382 MSE = 0.2868937298622186	R2-Score = 0.9946535663012934 MSE = 0.49021524293033697

Şekil 6. Hiper-Parametre Optimizasyonu ile birlikte elde edilen sonuçlar

Şekil 7’ de CatBoost algoritmasının hiper-parametre optimizasyon yöntemi sonrası elde edilen parametre değerleri gösterilmektedir. Isıtma ve soğutma için parametre değerleri ayrı ayrı gösterilmiştir.

	CATBOOST	
	ISITMA	SOĞUTMA
Grid Search	{'iterations': 10, 'l2_leaf_reg': 70, 'learning_rate': 0.01, 'loss_function': 'RMSE', 'random_strength': 0.01}	{iterations': 10, 'l2_leaf_reg': 70, 'learning_rate': 0.01, 'loss_function': 'RMSE', 'random_strength': 0.001}
Bayesian-GP	[('iterations', 2000), ('l2_leaf_reg', 7), ('learning_rate', 0.06335998007074319), ('loss_function', 'Tweedie:variance_power=1.10'), ('random_strength', 6.244236126212707e-08)]	[('iterations', 1911), ('l2_leaf_reg', 71), ('learning_rate', 0.5537373996465697), ('loss_function', 'Tweedie:variance_power=1.25'), ('random_strength', 6.466177885950986e-06)]
Bayesian-RF	[('bagging_temperature', 0.14082945805593675), ('depth', 4), ('iterations', 721), ('l2_leaf_reg', 11), ('learning_rate', 0.47108081824409764), ('loss_function', 'Tweedie:variance_power=1.5'), ('random_strength', 0.16057692871276674)]	[('iterations', 1362), ('l2_leaf_reg', 34), ('learning_rate', 0.07942610543471301), ('loss_function', 'Tweedie:variance_power=1.10'), ('random_strength', 0.015493733272030732)]
TPE	{'iterations': 976, 'l2_leaf_reg': 26, 'learning_rate': 0.34204545216863824, 'loss_function': 'RMSE', 'random_strength': 0.18835603017348032}	{'iterations': 1890, 'l2_leaf_reg': 74, 'learning_rate': 0.2729407918213027, 'loss_function': 'Tweedie:variance_power=1.10', 'random_strength': 1.389529742220915}

Şekil 7. CatBoost algoritmasının optimizasyon sonrası elde edilen parametre değerleri

CatBoost algoritması, en başarılı sonucu, Bayesian-RF optimizasyon yönteminde elde edilen parametreler ile sağlamıştır. Şekil 8’ de XGBoost algoritmasının hiper-parametre optimizasyon yöntemi sonrası elde edilen parametre değerleri gösterilmektedir.

	XGBOOST	
	ISITMA	SOĞUTMA
Grid Search	{'colsample_bytree': 0.7, 'learning_rate': 0.07, 'max_depth': 7, 'min_child_weight': 4, 'n_estimators': 500, 'nthread': 4, 'objective': 'reg:linear', 'silent': 1, 'subsample': 0.7}	{'colsample_bytree': 0.7, 'learning_rate': 0.07, 'max_depth': 6, 'min_child_weight': 4, 'n_estimators': 500, 'nthread': 4, 'objective': 'reg:linear', 'silent': 1, 'subsample': 0.7}
Bayesian-GP	[('colsample_bytree', 0.5140258469066882), ('learning_rate', 0.01), ('max_depth', 12), ('n_estimators', 5000), ('reg_alpha', 1e-09), ('reg_lambda', 5.0582970828989655), ('subsample', 1.0)]	[('colsample_bytree', 0.8565795396295001), ('learning_rate', 0.9355196460762675), ('max_depth', 12), ('n_estimators', 5000), ('reg_alpha', 1e-09), ('reg_lambda', 29.19798799879909), ('subsample', 0.8965021484296037)]
Bayesian-RF	colsample_bytree=0.5808223876619614, learning_rate = 0.1976606173151928, reg_alpha = 0.022482236946905126, reg_lambda = 39.87193370390488, max_depth=9, n_estimators=4211, subsample =0.8812701807127893	('colsample_bytree', 0.6398890698940642), ('learning_rate', 0.7233786463928277), ('max_depth', 10), ('n_estimators', 2021), ('reg_alpha', 1e-09), ('reg_lambda', 95.46830020484617), ('subsample', 0.9586457304444828)
TPE	{'colsample_bytree': 0.6415972382520492, 'learning_rate': 0.1843600126042833, 'max_depth': 62, 'n_estimators': 2001, 'reg_alpha': 0.06278466604654144, 'reg_lambda': 85.63551971399562, 'subsample': 0.9562555457425306}	{'colsample_bytree': 0.6590415366764466, 'learning_rate': 0.616771665434688, 'max_depth': 54, 'n_estimators': 3095, 'reg_alpha': 0.09061695962130986, 'reg_lambda': 96.01559703962853, 'subsample': 0.8615401175211749}

Şekil 8. XGBoost algoritmasının optimizasyon sonrası elde edilen parametre değerleri

XGBoost algoritması, en başarılı sonucu, TPE optimizasyon yönteminde elde edilen parametreler ile sağlamıştır. Şekil 9’ da LightGBM algoritmasının hiper-parametre optimizasyon yöntemi sonrası elde edilen parametre değerleri gösterilmektedir.

		LIGHTGBM	
		ISITMA	SOĞUTMA
Grid Search	{'colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_samples': 1, 'n_estimators': 150, 'subsample': 0.7}		{'colsample_bytree': 0.9, 'grid = learning_rate': 0.1, 'max_depth': 6, 'min_child_samples': 10, 'n_estimators': 150, 'subsample': 0.7}
Bayesian-GP	[('colsample_bytree', 0.9306623318575136), ('learning_rate', 0.29706601518564324), ('max_depth', 53), ('n_estimators', 5000), ('num_leaves', 418), ('reg_alpha', 0.0010246346352370656), ('reg_lambda', 0.00014001003359904277), ('reg_sqrt', False), ('subsample', 1.0), ('subsample_freq', 7)]		[('colsample_bytree', 1.0), ('learning_rate', 0.48121430387525616), ('max_depth', -1), ('n_estimators', 5000), ('num_leaves', 512), ('reg_alpha', 0.00025775560388258025), ('reg_lambda', 5.2555625089210585e-05), ('reg_sqrt', True), ('subsample', 1.0), ('subsample_freq', 10)]
Bayesian-RF	[('colsample_bytree', 0.8255202892609332), ('learning_rate', 0.47734278480653514), ('max_depth', 210), ('n_estimators', 790), ('num_leaves', 141), ('reg_alpha', 2.5906532928815245e-09), ('reg_lambda', 3.147269506194319e-07), ('reg_sqrt', True), ('subsample', 0.987364651793906), ('subsample_freq', 7)]		[('colsample_bytree', 0.7588125694231656), ('learning_rate', 0.3247464789572122), ('max_depth', 45), ('n_estimators', 4911), ('num_leaves', 267), ('reg_alpha', 0.0003552800760497431), ('reg_lambda', 4.190846413206602e-09), ('reg_sqrt', True), ('subsample', 0.9315444862044924), ('subsample_freq', 7)]
TPE	{'colsample_bytree': 0.7581183172319083, 'learning_rate': 0.9160777144032961, 'max_depth': 116, 'n_estimators': 2904, 'num_leaves': 184, 'reg_alpha': 0.26459491141567815, 'reg_lambda': 63.11576848073227, 'subsample': 0.9425303820282646, 'subsample_freq': 4}		{'colsample_bytree': 0.73305275336064, 'learning_rate': 0.8541108665754867, 'max_depth': 61, 'n_estimators': 3868, 'num_leaves': 437, 'reg_alpha': 0.04172759102969725, 'reg_lambda': 38.66978057588459, 'subsample': 0.8519803607257674, 'subsample_freq': 7}

**Şekil 9.** LightGBM algoritmasının optimizasyon sonrası elde edilen parametre değerleri

LightGBM algoritması, en başarılı sonucu, Bayesian-GP optimizasyon yönteminde elde edilen parametreler ile sağlamıştır. Yapılan karşılaştırmalar sonucunda, en yüksek tahmin oranına, CatBoost Regressor ve Bayesian-Random Forest optimizasyon yönteminin birlikte kullanılması ile ulaşılmıştır. Hem ısıtma hem soğutma yükleri ile yapılan test sonuçları birleştirilmiştir. Elde edilen sonuçlarda 0.997 R2-Score ve 0.23 MSE (Mean Squared Error) değerleri elde edilmiştir.

#### 4. SONUÇ

Enerji kullanımının artması ile binalarda enerji verimliliğinin tahmin edilmesi, günümüzde oldukça popüler bir konudur. Binalardaki enerji verimliliğinin tahmini için yapay zekâ ve makine öğrenimi tekniklerinin kullanımı, son zamanlarda sıklıkla üzerinde çalışılan konular arasındadır. Bu çalışmada, akıllı binalardaki enerji tüketiminin tahmini için regresyon modellerinin hiper-parametre optimizasyon yöntemleri ile kullanımı önerilmektedir. Enerji tahmin modellerinin başarımlarını iyileştirmek için Hiper-parametre Optimizasyon yöntemleri uygulanmıştır. 10 regresyon modeli ve 4 Hiper-parametre Optimizasyon yöntemi ile 22 adet eğitim modeli oluşturulmuştur. Bunların 10 tanesi, parametre değerleri değiştirilmeden eğitilen modellerdir. 12 tanesi ise MSE (Mean Squared Error) değeri 1,5 altında olan modellerin, hiper-parametre optimizasyon yöntemleri ile birleştirilmesinden oluşan modellerdir. Deney sonuçlarında, CatBoost algoritmasının Bayesian-RF optimizasyon yöntemi ile entegre edilmesi elde edilen optimum hiper-parametreler en başarılı sonucu vermiştir. 0.997 R2 Skoru ve 0.23 MSE skoru ile diğer regresyon modeli ve hiper-parametre optimizasyonu kombinasyonlarına göre daha yüksek bir başarı sergilemiştir. Gelecek çalışmalar için farklı tipteki akıllı binalardan elde edilen daha büyük boyutlu verilerin ve bu verilerin ön işleme aşamasının çalışmaya uygun olarak yapılmasıyla veya önceki çalışmalara göre daha iyi hazırlanmış olan gelişmiş sinir ağlarının da eklenmesiyle enerji tüketimi tahmin modellerinin performansı iyileştirilebilir.

## 5. KAYNAKLAR

- [1] Goyal, M., Pandey, M., & Thakur, R. (2020, June). Exploratory analysis of machine learning techniques to predict energy efficiency in buildings. In 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) (pp. 1033-1037). IEEE.
- [2] Al-Rakhami, M., Gumaei, A., Alsanad, A., Alamri, A., & Hassan, M. M. (2019). An ensemble learning approach for accurate energy load prediction in residential buildings. *IEEE Access*, 7, 48328-48338.
- [3] Sajjad, M., Khan, S. U., Khan, N., Haq, I. U., Ullah, A., Lee, M. Y., & Baik, S. W. (2020). Towards efficient building designing: Heating and cooling load prediction via multi-output model. *Sensors*, 20(22), 6419.
- [4] Tsanas, A., & Xifara, A. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and buildings*, 49, 560-567.
- [5] Moradzadeh, A., Mansour-Saatloo, A., Mohammadi-Ivatloo, B., & Anvari-Moghaddam, A. (2020). Performance evaluation of two machine learning techniques in heating and cooling loads forecasting of residential buildings. *Applied Sciences*, 10(11), 3829.
- [6] Akgundogdu, A. (2020). Comparative analysis of regression learning methods for estimation of energy performance of residential structures. *Erzincan University Journal of Science and Technology*, 13(2), 600-608.
- [7] Venkat Ramana Reddy, A., & Sudheer Kumar, M. (2020). A Comparative Analysis of Regression Algorithms for Energy Estimation in Residential Buildings. In *ICICCT 2019–System Reliability, Quality Control, Safety, Maintenance and Management: Applications to Electrical, Electronics and Computer Science and Engineering* (pp. 300-311). Springer Singapore.
- [8] Pachauri, N., & Ahn, C. W. (2022, November). Regression tree ensemble learning-based prediction of the heating and cooling loads of residential buildings. In *Building Simulation* (Vol. 15, No. 11, pp. 2003-2017). Beijing: Tsinghua University Press.
- [9] Gao, W., Alsarraf, J., Moayedi, H., Shahsavari, A., & Nguyen, H. (2019). Comprehensive preference learning and feature validity for designing energy-efficient residential buildings using machine learning paradigms. *Applied Soft Computing*, 84, 105748.
- [10] Moayedi, H., Bui, D. T., Dounis, A., Lyu, Z., & Foong, L. K. (2019). Predicting heating load in energy efficient buildings through machine learning techniques. *Applied Sciences*, 9(20), 4338.
- [11] Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, 17(1), 26-40.
- [12] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).
- [13] Wang, Y., Pan, Z., Zheng, J., Qian, L., & Li, M. (2019). A hybrid ensemble method for pulsar candidate classification. *Astrophysics and Space Science*, 364, 1-13.
- [14] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.
- [15] Yenigün, O. (2022), Smart Aspects of CatBoost Algorithm, 25 Nisan 2023 tarihinde <https://python.plainenglish.io/smart-aspects-of-catboost-algorithm-2720a6de4da6> adresinden alındı.
- [16] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- [17] Khandelwal, E. (2020), Which algorithm takes the crown: Light GBM vs XGBOOST?, 25 Nisan 2023 tarihinde [www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/](http://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/) adresinden alındı.