symmetry

MDPI

Article

Machine Learning Models for the Prediction of Energy Consumption Based on Cooling and Heating Loads in Internet-of-Things-Based Smart Buildings

Bita Ghasemkhani [1,*], Reyat Yilmaz [2], Derya Birant [3] and Recep Alp Kut [3]

[1] Graduate School of Natural and Applied Sciences, Dokuz Eylul University, Izmir 35390, Turkey
[2] Department of Electrical and Electronics Engineering, Dokuz Eylul University, Izmir 35390, Turkey; reyad.yilmaz@deu.edu.tr
[3] Department of Computer Engineering, Dokuz Eylul University, Izmir 35390, Turkey; derya.birant@deu.edu.tr (D.B.); alp.kut@deu.edu.tr (R.A.K.)
* Correspondence: bita.ghasemkhani@ogr.deu.edu.tr

Abstract: In this article, the consumption of energy in Internet-of-things-based smart buildings is investigated. The main goal of this work is to predict cooling and heating loads as the parameters that impact the amount of energy consumption in smart buildings, some of which have the property of symmetry. For this purpose, it proposes novel machine learning models that were built by using the tri-layered neural network (TNN) and maximum relevance minimum redundancy (MRMR) algorithms. Each feature related to buildings was investigated in terms of skewness to determine whether their distributions are symmetric or asymmetric. The best features were determined as the essential parameters for energy consumption. The results of this study show that the properties of relative compactness and glazing area have the most impact on energy consumption in the buildings, while orientation and glazing area distribution are less correlated with the output variables. In addition, the best mean absolute error (MAE) was calculated as 0.28993 for heating load (kWh/m^2) prediction and 0.53527 for cooling load (kWh/m^2) prediction, respectively. The experimental results showed that our method outperformed the state-of-the-art methods on the same dataset.

Keywords: machine learning; Internet of things; energy consumption; smart buildings; tri-layered neural networks; cooling load; heating load

check for updates

Citation: Ghasemkhani, B.; Yilmaz, R.; Birant, D.; Kut, R.A. Machine Learning Models for the Prediction of Energy Consumption Based on Cooling and Heating Loads in Internet-Of-Things-Based Smart Buildings. *Symmetry* **2022**, *14*, 1553. https://doi.org/10.3390/sym14081553

Academic Editors: Tzu Chuen Lu, Wun-She Yap and Biswapati Jana

Received: 5 July 2022
Accepted: 25 July 2022
Published: 28 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In information and communication technology, the *Internet of things* (IoT), as state-of-the-art technology for intelligent interconnectivity, has recently been presented to communicate at any time, from anywhere, and through any object. The IoT is extensively utilized in different fields, e.g., manufacturing [1], home [2], agriculture [3], healthcare [4], environment [5], military [6], retail [7], and sports [8]. It can be used in a great variety of applications for different purposes in mentioned fields, e.g., temperature control, appliance control, communication, quality control, threat analysis, situational awareness, risk assessment, patient care, fitness trackers, crop management, fire detection, species tracking, weather prediction, traffic flow, smart parking, theft protection, inventory control, and focused marketing. Using IoT-based solutions provides many opportunities for different processes, intelligent devices, real-time applications, and operating platforms to facilitate the accessibility of specific information and services, to enhance people's lifestyle as an enabler in various environments, especially in industry [9,10].

An essential subfield of computer science that IoT can use is *machine learning* which aids computer software in making a prediction from former data. Based on it, the learning process can be grouped into four different types: supervised learning, unsupervised learning, semisupervised learning, and reinforcement learning. In this study, we focused

on supervised learning that builds a model from historical data to be able to predict an output value associated with a particular input vector.

Rapid developments in various information technologies have simplified the advent of Internet-based devices that deliver observation and measurement from the real physical world. Thus, the total number of such devices or IoT is overgrowing and leads to a high volume of data generated by different IoT and considered by the location and time dependency, with various modalities and varying data quality. As a result, intelligent analyses of such data are the crucial means of developing IoT applications [11]. This study focuses on building intelligent models for the prediction of energy consumption in IoT-based smart buildings.

According to [12], buildings in cities consume 70% of the primary energy, in which the most energy-consuming part is the HVAC (heating, ventilation and air conditioning) system. Therefore, predicting and optimizing energy consumption in IoT-based buildings through machine learning algorithms is an essential human need and economic and social development factor [12], which we focus on in this study. That means that an accurate prediction of heating load and cooling load in different IoT-based buildings through the proposed model can lead to optimizing energy consumption, which implies a small but necessary step to prevent global warming. Moreover, considering factors that affect energy consumption, the *heating load* (HL) is the amount of heat energy added to an environment to keep its temperature in a satisfactory manner for the residents. The *cooling load* (CL) is the amount of heat energy removed from an environment to similarly keep its temperature satisfactorily for the residents. The heating and cooling loads which are named thermal loads, consider the construction features of buildings. Prediction of the CL and HL from simple properties of the buildings such as surface area, height, orientation, and so on, might assist in determining the energy performance of the buildings (EBP). It can also assist decision-makers in allocating resources to reconstruction measures, which can have both long-term and short-term benefits for cost savings, energy efficiency, and environmental health. The main requirements of predicting HL and CL in buildings are to reduce energy consumption, manage energy demands, reduce operational cost, and reduce emissions of harmful gases. In IoT-based buildings, air-conditioning or heating devices may handle the heating and cooling Loads smartly. This process will improve energy consumption through an efficient prediction based on building features to keep the temperature at a suitable level.

The main contributions of this study can be listed as follows. (i) It proposes novel predictive models for cooling and heating Loads in IoT-based smart buildings by applying various machine learning techniques to the data and considering features to have efficient energy consumption. (ii) It is the first study that uses both the tri-layered neural network (TNN) and maximum relevance minimum redundancy (MRMR) algorithms together to predict energy consumption in IoT-based smart buildings. The structure of the neural network was designed by considering many aspects such as the number of nodes, activation function, and symmetry property. (iii) Our study is also original in that it proposes a multitarget learning solution, unlike the traditional single-target learning studies. (iv) The experimental results showed that our method outperformed the state-of-the-art methods on the same dataset.

The rest of the paper is organized as follows. In the following section, a recent literature review on machine learning for IoT systems is given. In Section 3, the proposed model is described. Section 4 explains the experiments that were carried out in this study. In the next section, the obtained results are presented. In Section 6, the related conclusions and future works are described, respectively.

2. Literature Review

In the recent past, some machine learning studies have also been conducted with or without taking into account the symmetry concept. Gaber et al. (2022) proposed an intrusion detection method based on machine learning to distinguish the injection attacks

in smart-city IoT for security. As indoor wireless networks include more than 80% of the IoT networks for smart cities, security and privacy challenges have become a serious concern for intelligent IoT devices. Thus, they applied SVM, RF, DT, recursive feature elimination, and constant removal algorithms to the public AWID dataset, and used a t-test to analyze the results. According to the results, the decision tree method could be used to recognize injection attacks by utilizing just eight features with 99% accuracy [13].

Mondal et al. (2021) implemented a machine learning model with IoT devices to provide a smart healthcare ecosystem, which can lead to improvement in the healthcare industry. They gathered the dataset from wearable sensors and used various wearable devices and cloud computing technologies. Therefore, this investigation conquers the challenges of wearable and implanted healthcare body network connections [14].

Siaterlis et al. (2022) designed and developed a framework to monitor the condition of harsh operating environments by means of IoT, including a knowledge graph in industrial production procedures for condition monitoring and predictive maintenance of assets, which can support personnel in decision-making and supervision processes. In their study, they aimed to apply semantic artificial intelligence and machine learning for approximating the remaining useful life of the monitored assets. Furthermore, they used a real dataset over five years from an aluminum-producing company and proved the usefulness of the proposed solution for practical applications [15].

Junior et al. (2022) proposed a method in the field of IoT smart agriculture to reduce the data on machine learning algorithms for fog computing because of cloud disconnections that usually occur in the countryside. Their proposed approach collects and stores data in a fog-based intelligent agricultural surrounding. Moreover, various data-reduction approaches were used to preserve the data's time-series nature. Furthermore, the k-means and latent classification model (LCM) algorithms were applied to two real datasets. They achieved higher reduction results than the previous works [16].

Tiwari et al. (2021) established an ensemble machine learning approach for ocean IoT attack detection on the basis of the improved light gradient boosting machine algorithm. Their model was proposed to protect the marine IoT environment from cyberattacks and destructive activities. As a result, the dispersed IoT attacks could be controlled in more profound marine environments with lower computational costs, and higher accuracy was achieved and evaluated with various metrics. Their method presents a hopeful future for IoT applications in the ocean environment [17].

Fard and Hosseini (2022) aimed to investigate the properties of a building that influence the amount of energy consumption inside it by means of IoT concept and machine learning algorithms, namely univariate linear regression, RF, KNN, AdaBoost, and neural network. They utilized the energy efficiency dataset, and as a result, the overall height of buildings was introduced as the most important feature impacting energy consumption. Moreover, the AdaBoost algorithm was introduced as the best algorithm for heating and cooling loads [18].

Cakir et al. (2021) created an industrial IoT-based condition monitoring system at a low cost. As it is crucial to detect defective bearings earlier than reaching a critical level, it was predicted by machine learning algorithms, including SVM, DT, RF, and KNN. Furthermore, their system can notify the related maintenance team to take the necessary measures in critical events [19].

Rahman et al. (2022) presented a machine learning and IoT-based farming system that enables intelligent control to categorize poisonous and edible mushrooms. As automation was an essential need for farmers, they preferred to move from traditional methods to modern ones. In their method, remote monitoring and management (RMM) and sensor technologies had been included. Additionally, various machine learning algorithms have been used, including DT, SVM, KNN, and RF. The accuracy of their model is very high, which can be efficient in mushroom farming [20].

Meghana et al. (2021) proposed an approach to collect the data on social IoT. Moreover, the performance of different machine learning algorithms on its data was investigated. The

result of their study revealed that artificial neural networks and decision tree algorithms achieved a good performance in comparison with other machine learning algorithms. In contrast, KNN was shown to have the weakest performance in most cases. Therefore, it resulted that applying machine learning algorithms to data aggregation led to better network performance in comparison with the entire dataset [21].

Khan and Al-Badi (2020) investigated the various open-source machine learning platforms from the programming language, implementation, and usage aspects. Nowadays, industries need machine learning methods to analyze huge amounts of datasets, which are generated through applications, smart devices, industrial systems, and sensors. Such generated data have their specific properties, and thus, it may be difficult to understand and use newly developed models for machine learning. In their work, different types of machine learning algorithms (linear regression, support vector machines, decision tree, and random forest) and related frameworks (Tensorflow, H2O, Caffe, PyTorch, Microsoft Cognitive Toolkit, etc.) were examined by the data of IoT systems. The optimal selection of the machine learning frameworks for applying various models was PyTorch and Tensorflow, among the others [22].

Our work differs from the previous studies in four important aspects. (i) It proposes novel predictive models to predict energy consumption in IoT-based smart buildings. (ii) It is the first study that uses both the tri-layered neural network (TNN) and maximum relevance minimum redundancy (MRMR) algorithms together for the prediction of cooling and heating loads in buildings. (iii) Our study is also original in that it proposes a multitarget learning solution, unlike the traditional single-target learning studies. (iv) Our method achieved better performance than the state-of-the-art methods on the same dataset.

3. Proposed Model

3.1. Description

This study proposes novel machine learning models for the prediction of cooling and heating loads in IoT-based smart buildings. It is the first study that uses both the tri-layered neural network (TNN) and maximum relevance minimum redundancy (MRMR) algorithms together to predict energy consumption in buildings. Our study is also original in that it proposes a multitarget learning solution that predicts two outputs: heating load (Y1) and cooling load (Y2), unlike the traditional single-target learning studies.

Figure 1 shows the general overview of the proposed model. An energy efficiency dataset is analyzed by using some data-preprocessing techniques. Although the concept of symmetry is widely used in many topics, it is almost not discussed related to the distribution of building features for the prediction of energy consumption based on cooling and heating loads. After data analysis, the feature-selection algorithms, namely maximum relevance minimum redundancy (MRMR), F-test, and Regressional Relief version-F (RReliefF), are used for the mentioned dataset features. Based on [23], MRMR was finally chosen as the feature-selection algorithm in all experiments of this work, which uses an incremental greedy strategy. After the feature-selection step in the proposed model, Bayesian optimization is used to tune the hyperparameters of a model on a validation dataset, e.g., in GPR, for fitting the model. The improvement of the acquisition function is expected per second plus. It is regarded for a number of iterations in the implementation of this model. Moreover, in the next step, the *k*-fold cross-validation technique is used to partition the related data into folds and estimate the accuracy of each fold to decrease the risk of underfitting or overfitting.

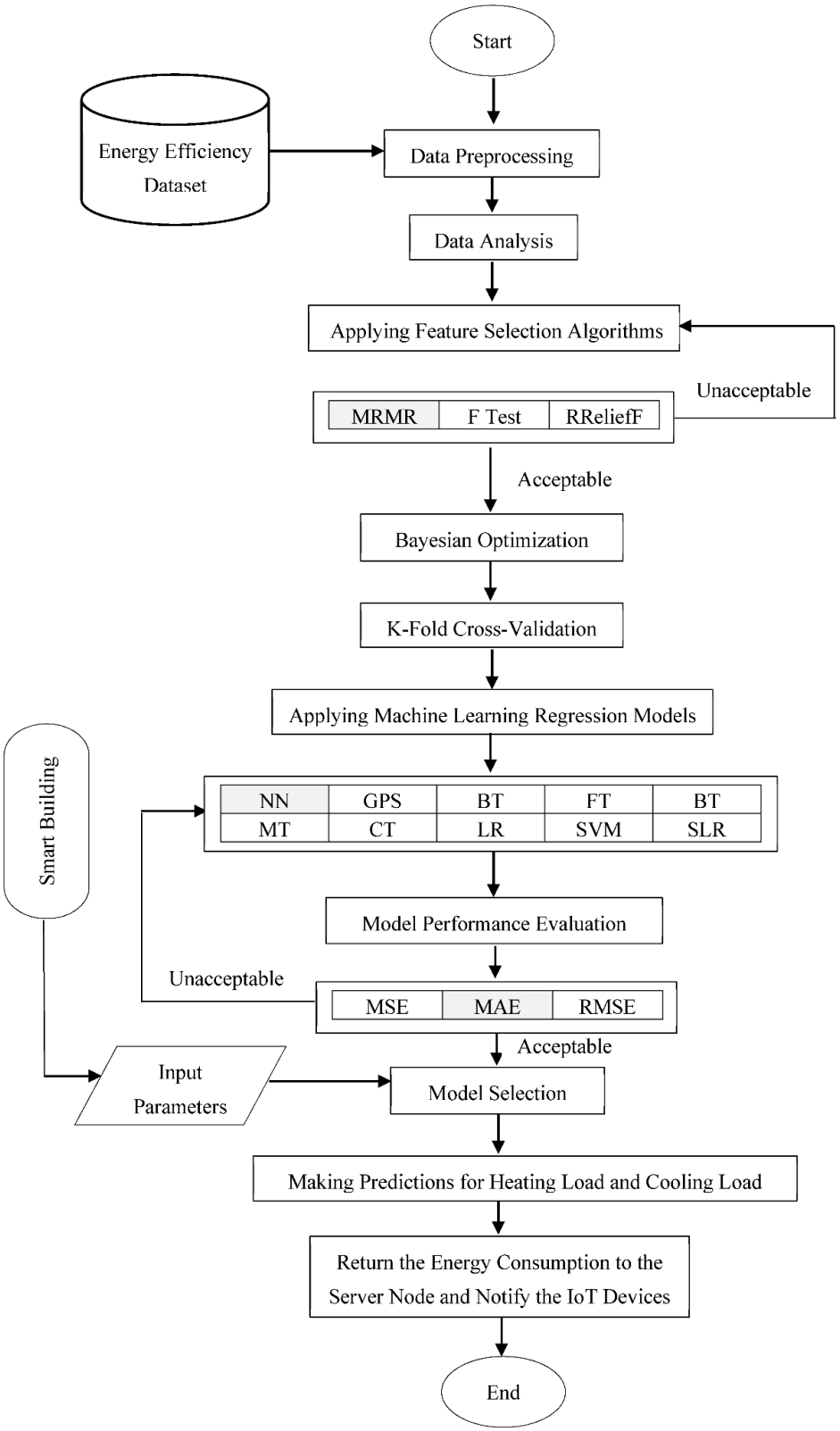


Figure 1. The proposed model in the IoT environment.

The k-fold cross-validation is a technique that randomly divides the dataset into k equal-sized subparts (called folds). At each step, the k-th part of the dataset is regarded as the validation data for testing the model, and the remaining $k - 1$ subparts are used as training data to construct a classifier. This process is repeated k times such that all the

subparts are successively employed for validation. In the end, the *k* results from the folds are averaged to determine performance.

The proposed approach assesses ten different machine learning regression algorithms, namely bagged tree (BaT), fine tree (FT), boosted tree (BoT), coarse tree (CT), medium tree (MT), tri-layered neural network (TNN), Gaussian process regression (GPR), stepwise linear regression (SLR), linear regression (LR), and support vector regression (SVR) with various parameters by training 60 models in several experiments. After that, performance evaluations of these algorithms are made in terms of different metrics, including mean-square error (MSE), MAE, and root-mean-square error (RMSE). MAE takes the absolute difference between the actual and predicted values and averages it across the dataset. Hence the lower MAE means the higher accuracy of a model. The TNN model is selected as the best predictor to make predictions for cooling load and heating load in IoT-based smart buildings, returning the energy consumption to the server node and notifying the IoT devices.

To have a better understanding of the proposed model, an example architecture is shown in Figure 2. In this model, by connecting the IoT devices and communication modules inside a smart building, the extracted knowledge from data can be delivered to the cloud through the Wi-Fi module to generate notifications and maybe alarms for smart devices (especially air-conditioning systems and IoT heating) and also for occupants (by e-mail and SMS) through different IoT devices such as a smartwatch, smartphone, laptops, PDAs, and so on. Here, symmetrical connections are assumed. After the prediction of heating load and cooling load by an intelligence model, the energy-consumption estimation is returned to the server node to notify the IoT devices, e.g., IoT air conditioning and IoT heating, and then take the necessary actions for balancing energy consumption inside the building.

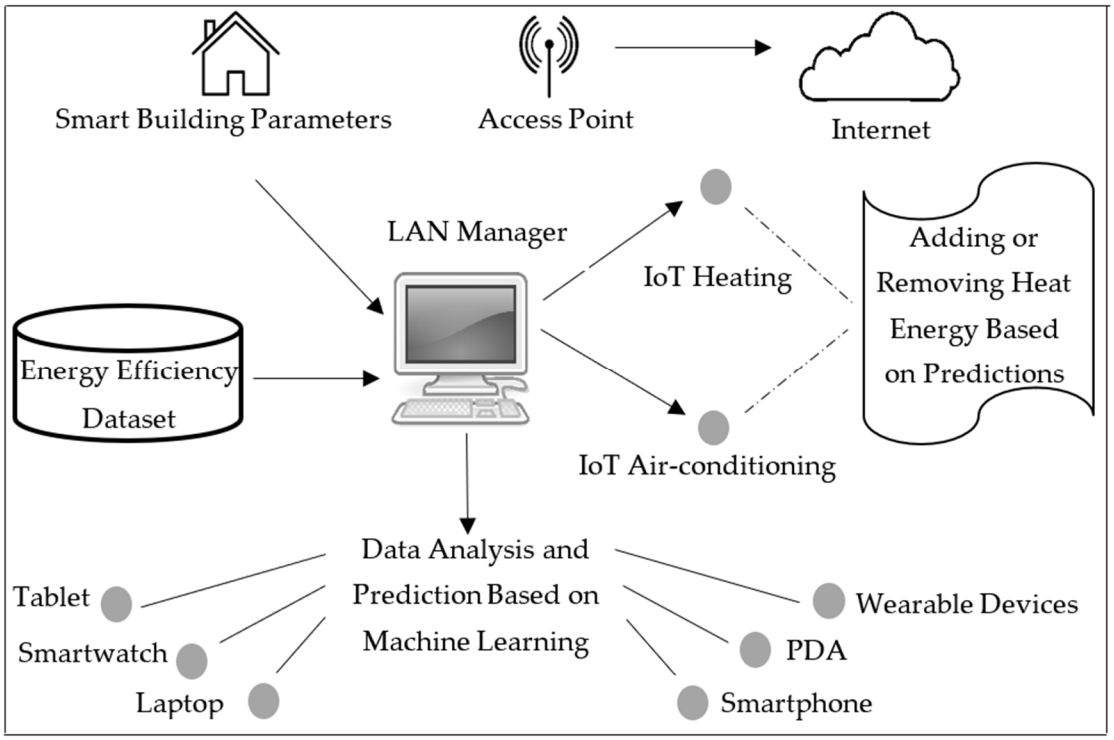


Figure 2. Model architecture in the IoT environment.

3.2. Properties

Machine learning is one of the most important techniques that implements symmetry in computer science. The mentioned problem in this research is considered as a regression problem since the output attributes (heating load and cooling load) contain continuous data.

In machine learning, regression is concerned with the prediction of a continuous target variable based on the set of input variables. Therefore, as one of the most common statistical methods, regression analysis was performed in this study to determine the relationship between independent and dependent variables. Different machine learning algorithms (TNN, FT, CT, MT, BaT, BoT, GPR, LR, SLR, and SVR) were applied to the energy efficiency dataset. Among these algorithms, the tri-layered neural network was selected as the best algorithm for the current work so it could be efficiently used for future predictions. The parameter values of TNN are given in Table 1. The structure of the neural network was designed by taking into account many aspects such as the activation function, number of nodes, and symmetry property. An optimal design of NN architecture is important to speed up the training process and strengthen the generalization ability of the model, which means better fitting of the network to new (unknown) samples. In addition to TNN, the MRMR feature-selection algorithm was applied to select the features with the most impact. Moreover, Bayesian optimization and k-fold cross-validation techniques were involved in this research. Moreover, the MAE metric was used to evaluate the performance of the proposed model.

Table 1. Parameter settings.

Parameter Type	Parameter Value
Number of layers	3
Number of neurons in each layer	30
Activation	Rectified Linear Unit (ReLU)
Momentum	0.9000
Iteration limit	1200
Iteration (epochs)	30
Regularization strength (lambda)	0
Initial learn rate	0.0100
Learn rate schedule	Piecewise
Learn rate drop factor	0.2000
Learn rate drop period	5

The *heating load* (HL) is the amount of heat energy that is considered for an environment to keep its temperature in a satisfactory manner for the residents. The *cooling load* (CL) of a building is the amount of energy that is caused by energy transferred through the building envelope (walls, floor, roof, etc.) and energy generated by occupants, lights, and equipment. They are based on the principle that the energy required for space cooling and heating primarily depends on the difference in temperatures between outdoors and indoors. Both are very sensitive to the design and the operation of the buildings and are to be managed based on several physical parameters such as temperature, relative humidity, and air velocity within the environment. The HL and CL are also named thermal loads and are influenced by different physical factors, especially the construction features of buildings. Each building is regarded as a whole block from the viewpoint of a heat network, which means the heating and cooling loads of a building are influenced by several physical factors such as the building itself (i.e., geometry, layout, construction, mechanical equipment), the location, the climate, and the residents. They play major roles in the financial cost according to the different seasons. If the heating and cooling loads of a building are to be predicted, it is important to know the influence of these factors. The prediction of the HL and CL of a building is essential for planning the efficient next-day operation of air conditioning, ventilation, and heating equipment. In this context, the objective of this study is to build an intelligent model that predicts HL and CL under different input assumptions such as surface area, height, and orientation of buildings.

3.3. Algorithm

Algorithm 1 presents the pseudocode of the proposed model for the prediction of the cooling load and heating load. First, the data are prepared by considering the smart

building parameters. After that, data preprocessing and analysis are undertaken using the dataset such that irrelevant, redundant, and noisy data are eliminated. Next, a feature rating is determined for each feature by using the MRMR algorithm. The most important features are selected and data are prepared for learning. After that, the predictive models are built by using the TNN algorithm separately for heating and cooling loads. Finally, the outputs are predicted by the models for each of the test query data.

Algorithm 1: Proposed Model (TNN + MRMR)

Inputs: N: Number of IoT-based smart buildings
GA: Glazing areas
GAD: Glazing area distributions
O: Orientations
Outputs: $O_{HL} = \{o_1, o_2, \ldots, o_N\}$ a set of heating load predictions
$O_{CL} = \{o_1, o_2, \ldots, o_N\}$ a set of cooling load predictions
Begin:
for $i = 1$ **to** $N * GA * GAD * O$ **do**
insert $Data_i$
end
apply data preprocessing
perform data analysis
for each feature f_i **in** Data // determining feature importance
rank(f_i) = MRMR(f_i)
end
$D = \underset{1 \le x \le m}{\text{argmax}}\ rank(x)$ // feature selection
Bayesian optimization
$\text{Model}_{\text{HL}} = \text{TNN}(D)$ // training
$\text{Model}_{\text{CL}} = \text{TNN}(D)$
for each testdata t_i **do** // testing
$o_i = \text{Model}_{\text{HL}}(t_i)$ // obtain heating load prediction
$O_{HL} = O_{HL} \cup o_i$
$o_i = \text{Model}_{\text{CL}}(t_i)$ // obtain cooling load prediction
$O_{CL} = O_{CL} \cup o_i$
end
Return O_{HL} *and* O_{CL}
End

4. Experimental Studies

4.1. Experiments

In this study, we designed four experiments in order to provide a deep analysis. The first experiment is related to predicting heating load (Y1) considering 70% training set and 30% testing set from the original data. The second experiment also focused on the prediction of heating load (Y1), but in this case, the 5-fold cross-validation technique was used. Similarly, in the third experiment, for the prediction of cooling load (Y2), 70% of the dataset and 30% of the dataset were considered as the training set and testing set, respectively. In addition, for predicting cooling load (Y2) in the fourth experiment, 5-fold cross-validation was used.

The proposed model was implemented in MATLAB® Online™ R2022a, which is accessible from a web browser, is automatically updateable to the latest version, is a consistent platform with the latest features, and is fully integrated with drives.

As evaluation criteria, mean absolute error (MAE), mean-squared error (MSE), and root-mean-square error (RMSE) were utilized. MAE depends on the mean of the difference between predictions and real values, as given in Equation (1). MSE is the sum of the square error between the predicted output and actual output, as given in Equation (2). RMSE is another index reflecting the difference between actual and predicted values, as given in

Equation (3). Based on these evaluation metrics, the best model was selected and used for the prediction.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|P_i - O_i| \quad (1)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(P_i - O_i)^2 \quad (2)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - O_i)^2} \quad (3)$$

where n is the number of samples, P_i is the predicted value, and O_i is the observed value.

4.2. Dataset Description

In this study, the "Energy Efficiency" dataset [24], which is available in the UCI (University of California Irvine) dataset repository, was used. It is a popular dataset that has been used by many studies [18,24–57], has a high hit value in the repository, and has made a significant contribution to the field of energy. The dataset information is given in Table 2. Energy analysis was performed by using 12 different building shapes, which differ from each other considering the building parameters. This dataset consists of 768 samples and eight features (X1, X2, . . . , X8) to predict real-valued responses (Y1 and Y2).

Table 2. Dataset information.

Dataset	Attribute	Problem	Number of Instances	Number of Attributes	Missed Value	Field	Year	Hit
Multivariate	Real Integer	Regression Classification	768	8	Not Available	Computer	2012	418,111

The features, their descriptions, and statistical information are included in Table 3. Relative compactness (RC) indicates the ratio of the surface area (A) to the corresponding volume (V) in the building and is calculated by the following formula: RC = $6V^{2/3}/A$. The shapes of the buildings with their corresponding RC values are shown in Figure 3. The glazing area (GA) represents the overall area measured through the rough opening, including the glazing, sash, and frame. In other words, GA is the total area of the wall, which is glass. GA affects the cooling and heating conditions of the building since it is exposed to external factors such as sun, wind, snow, and others. In the dataset, there are four kinds of glazing areas with different percentages of the floor area: 0%, 10%, 25%, and 40%. Glazing area distribution (GAD) indicates the distribution of the GA within the whole building. The dataset has six different distribution scenarios for each glazing area: (i) uniform: with 25% glazing on each side; (ii–v) north, east, south, and west: 55% in the corresponding direction and 15% on the remaining sides; (vi) no glazing areas. Skewness in Table 3 is a measure of the symmetry of the distribution for the related feature.

Table 3. Dataset features and their properties.

Features	Descriptions	Unit	Type	Min	Max	Mean	Mode	Median	Std. Dev.	Skewness
X1	Relative Compactness	-	Input	0.620	0.980	0.7642	0.980	0.750	0.106	0.496
X2	Surface Area	m^2	Input	514.500	808.500	671.708	514.500	673.750	88.086	−0.130
X3	Wall Area	m^2	Input	245.000	416.500	318.500	294.000	318.500	43.626	0.533
X4	Roof Area	m^2	Input	110.250	220.500	176.604	220.500	183.750	45.166	−0.163

Table 3. *Cont.*

Features	Descriptions	Unit	Type	Min	Max	Mean	Mode	Median	Std. Dev.	Skewness
X5	Overall Height	m	Input	3.500	7.000	5.250	7.000	5.250	1.751	0.000
X6	Orientation	-	Input	2.000	5.000	3.500	2.000	3.500	1.119	0.000
X7	Glazing Area	m^2	Input	0.000	0.400	0.234	0.100	0.250	0.133	−0.060
X8	Glazing Area Distribution	-	Input	0.000	5.000	2.812	1.000	3.000	1.551	−0.089
Y1	Heating load	kWh/m^2	Output	6.010	43.100	22.307	15.160	18.950	10.090	0.360
Y2	Cooling load	kWh/m^2	Output	10.900	48.030	24.588	21.330	22.080	9.513	0.400

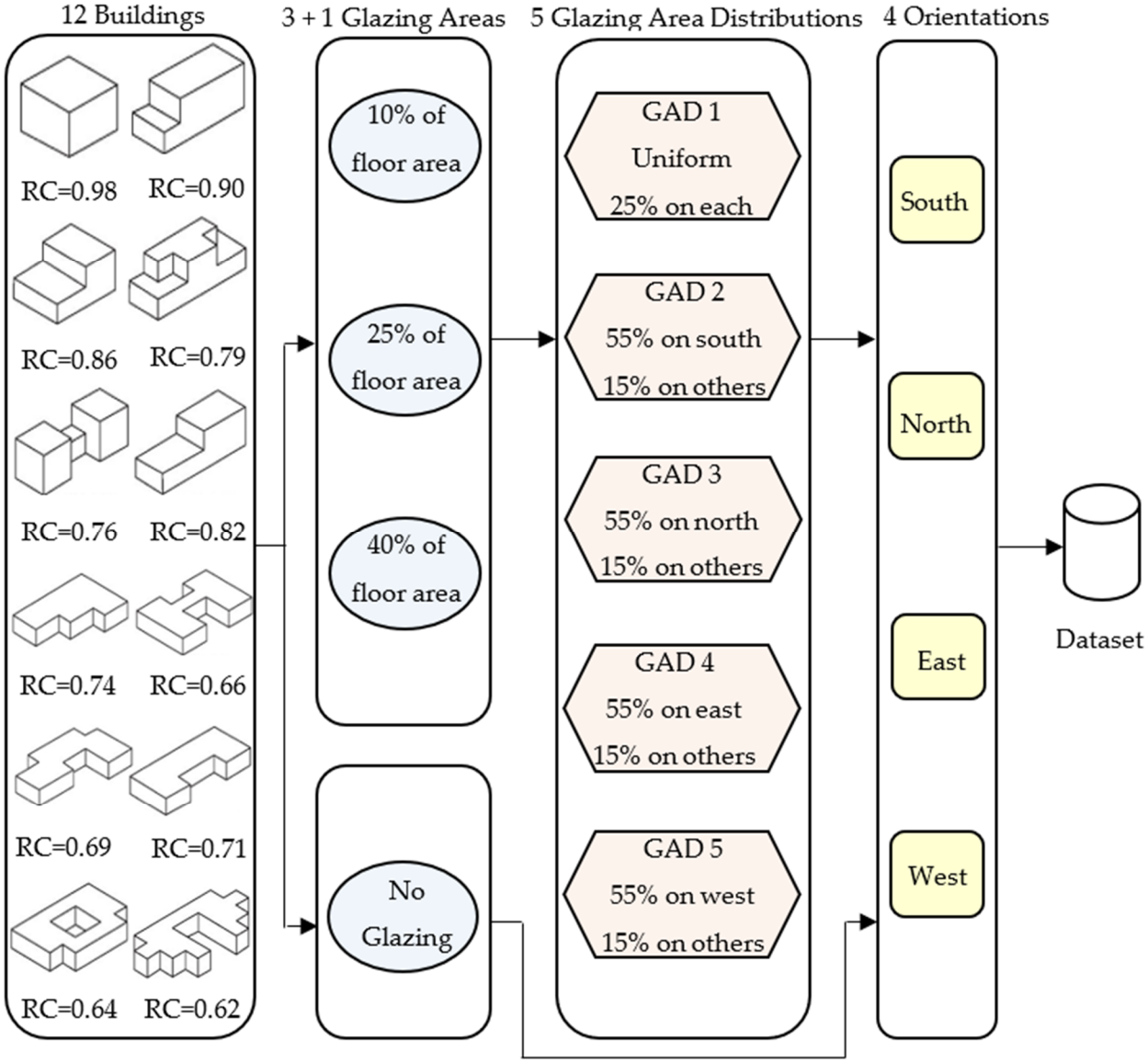


Figure 3. Block diagram depiction of the dataset.

Figure 3 illustrates the general structure of the dataset, which varies in size and has four glazing regions with five distribution scenarios and four orientations. Note that the orientation consists of the north, east, south, and west. Each building form is composed of 18 elements (elementary cubes). The buildings were constructed with the most prevalent, newest, and similar materials, as well as the lowest U-value: floors (0.860 W/m^2K; walls

(1.780 W/m^{2}K), windows (2.260 W/m^{2}K), and roofs (0.500 W/m^{2}K). The buildings are used for sedentary purposes (70 W) and are residential with a maximum of seven persons. The interior design has the following properties: 60% moistness, 0.6 clothing, 300 Lux illumination intensity, and 0.30 m/s airspeed. While the infiltration rate is 0.5 for air change rate with a wind sensitivity of 0.25 air changes/h, internal gains were set at latent (2 W/m^2) and sensible gain (5). Thermal characteristics were defined by a thermostat between 19 and 24 °C, a mixed mode with a 95% efficiency, 10–20 h of operation on weekends, and 15–20 h on weekdays.

It should be mentioned that splitting data into training and testing sets is an essential step for evaluating a machine learning-based model. Typically, in such separations, a great amount of data are used for training, and a small amount of data are used for testing. This process can reduce the effect of data discrepancies and lead to a better understanding of the model characteristics.

In the implementation, approximately 500 instances were used as training data, while the remaining instances were considered as testing data in the first and third experiments for predicting heating load and cooling load, respectively. Because there was no priority among the original dataset rows and having the same underlying distribution, this work used the common rule of 70% for training data and 30% for testing data in the preprocessing phase of splitting in the first and third experiments. This ratio was preferred, with the aim of providing comparability since some previous studies [25,36,39,40,42] used it. Moreover, k-fold cross-validation was used in the second and fourth experiments for the evaluation of the performances of the models.

4.3. Feature Selection

Some features in the dataset are more significant than the other ones. This study used three different feature-selection algorithms (MRMR, F Test, and RReliefF) in order to cross-check results and ensure the robustness of the selected feature set. The results are in Tables 4–6 for experiment 1, Tables 7–9 for experiment 2, Tables 10–12 for experiment 3, and Tables 13–15 for experiment 4, respectively. These tables show weight values obtained by the algorithms to examine the importance of each predictor. A large weight value indicates that the corresponding predictor is more important. The parameter setting of the F Test was determined as follows: the number of bins for binning continuous predictors was set to 10, missing values are discarded, and the weights of all features were equally set to 1. For the RReliefF algorithm, the nearest-neighbor parameter (k) was assigned to 10, so the algorithm found the nearest objects to a query instance from both the same class and the other different classes, called *hits* and *misses*, respectively. The verbosity level parameter of MRMR, which controls the amount of diagnostic information, was set to zero. The MRMR feature-selection technique distinguishes the features that impact the amount of energy consumption. According to this algorithm, "relative compactness" and "glazing area" properties affect prediction the most. According to the results of the MRMR, F Test, and RReliefF algorithms, "orientation" and "glazing area distribution" are less correlated with the output variables than other features. This conclusion has also been supported by previous studies with different methods such as random forest [36], gradient boosting machines [36], Pearson correlation [25], and Spearman rank correlation coefficient [12,38]. Therefore, when constructing the models in this study, we used the feature subset that includes X1, X2, X3, X4, X5, and X7 variables, corresponding to relative compactness, surface area, wall area, roof area, overall height, and glazing area, respectively.

Table 4. F Test for Y1 in experiment 1.

Select	Features	F Test (Weight Value)
1	X2	597.2962
2	X5	396.4874
3	X4	392.4925
4	X1	280.5078
5	X3	132.4942
6	X7	13.8447
7	X8	3.2846
8	X6	0.0004

Table 5. RReliefF for Y1 in experiment 1.

Select	Features	RReliefF (Weight Value)
1	X2	597.2962
2	X5	396.4874
3	X4	392.4925
4	X1	280.5078
5	X3	132.4942
6	X7	13.8447
7	X8	3.2846
8	X6	0.0004

Table 6. MRMR for Y1 in experiment 1.

Select	Features	MRMR (Weight Value)
1	X1	1.5395
2	X7	1.0968
3	X5	0.0004
4	X2	0.0003
5	X4	0.0003
6	X3	0.0003
7	X6	0
8	X8	0

Table 7. F Test for Y1 in experiment 2.

Select	Features	F Test (Weight Value)
1	X1	Inf
2	X2	Inf
3	X5	603.1448
4	X4	600.1703
5	X3	202.7149
6	X7	24.7248
7	X8	1.4203
8	X6	0.0006

Table 8. RReliefF for Y1 in experiment 2.

Select	Features	RReliefF (Weight Value)
1	X7	0.0528
2	X3	0.0407
3	X1	0.0254
4	X2	0.0247
5	X4	0.0032
6	X5	0
7	X8	−0.0271
8	X6	−0.0646

Table 9. MRMR for Y1 in experiment 2.

Select	Features	MRMR (Weight Value)
1	X1	1.1764
2	X7	0.8875
3	X5	0.1959
4	X6	0.1920
5	X4	0.1401
6	X8	0.1374
7	X2	0.1053
8	X3	0.0995

Table 10. F Test for Y2 in experiment 3.

Select	Features	F Test (Weight Value)
1	X2	613.0155
2	X5	403.2400
3	X4	399.7051
4	X1	295.1582
5	X3	142.3529
6	X7	9.5858
7	X8	1.5007
8	X6	0.0506

Table 11. RReliefF for Y2 in experiment 3.

Select	Features	RReliefF (Weight Value)
1	X3	0.0368
2	X1	0.0252
3	X2	0.0240
4	X7	0.0112
5	X4	0.0063
6	X5	0
7	X8	−0.0182
8	X6	−0.0478

Table 12. MRMR for Y2 in experiment 3.

Select	Features	MRMR (Weight Value)
1	X2	1.2353
2	X7	0.9096
3	X5	0.0004
4	X1	0.0003
5	X4	0.0003
6	X3	0.0002
7	X6	0
8	X8	0

Table 13. F Test for Y2 in experiment 4.

Select	Features	F Test (Weight Value)
1	X1	Inf
2	X2	Inf
3	X5	624.5357
4	X4	620.3089
5	X3	217.5311
6	X7	15.1747
7	X8	0.2737
8	X6	0.0771

Table 14. RReliefF for Y2 in experiment 4.

Select	Features	RReliefF (Weight Value)
1	X3	0.0317
2	X2	0.0192
3	X1	0.0189
4	X7	0.0047
5	X4	0.0022
6	X5	0
7	X8	−0.0089
8	X6	−0.0341

Table 15. MRMR for Y2 in experiment 4.

Select	Features	MRMR (Weight Value)
1	X1	1.1521
2	X7	0.8652
3	X5	0.2004
4	X6	0.1872
5	X4	0.1412
6	X8	0.1305
7	X2	0.1090
8	X3	0.1030

5. Experimental Results

5.1. Results

The comparison of different machine learning models based on RMSE, MSE, and MAE are shown in Tables 16–19 for four experiments of this study. The results revealed that the tri-layered neural network (TNN) algorithm performed better than other machine learning algorithms for the prediction of cooling load and heating load. For example, in the first

experiment, TNN made predictions with small error values (kWh/m^2): 0.43101, 0.18577, and 0.28993 in terms of RMSE, MSE, and MAE, respectively. The TNN algorithm with the MRMR feature-selection method obtained the best scores for heating load and cooling load predictions in the first and fourth experiments, with 0.28993 and 0.53527 MAE values (kWh/m^2), respectively.

Table 16. Model comparison for heating load (Y1) in experiment 1.

Trained Models	Heating Load (kWh/m^2)		
	RMSE	**MSE**	**MAE**
Tri-Layered Neural Network	**0.43101**	**0.18577**	**0.28993**
Gaussian Process Regression	0.43094	0.18571	0.30279
Boosted Trees	0.57863	0.33481	0.39011
Fine Tree	0.69002	0.47613	0.42614
Bagged Trees	1.01200	1.02410	0.62627
Medium Tree	1.28930	1.66220	0.62704
Stepwise Linear Regression	1.06580	1.13600	0.85654
Linear Regression	1.08520	1.17770	0.87552
Support Vector Machine	1.81500	3.29410	1.34740
Coarse Tree	2.55120	6.50880	1.82770

Table 17. Model comparison for heating load (Y1) in experiment 2.

Trained Models	Heating Load (kWh/m^2)		
	RMSE	**MSE**	**MAE**
Tri-Layered Neural Network	**0.45689**	**0.20875**	**0.32360**
Gaussian Process Regression	0.46479	0.21603	0.32875
Fine Tree	0.66731	0.44530	0.41115
Medium Tree	1.02460	1.04970	0.52673
Boosted Trees	0.79549	0.63280	0.56936
Bagged Trees	1.11810	1.25010	0.71605
Stepwise Linear Regression	1.09030	1.18880	0.85486
Support Vector Machine	2.15290	4.63490	1.49260
Coarse Tree	2.32150	5.38930	1.61410
Linear Regression	2.94610	8.67920	2.09680

Table 18. Model comparison for cooling load (Y2) in experiment 3.

Trained Models	Cooling Load (kWh/m^2)		
	RMSE	**MSE**	**MAE**
Tri-Layered Neural Network	**0.92695**	**0.85924**	**0.58471**
Bagged Trees	1.04970	1.10190	0.69217
Boosted Trees	1.14220	1.30470	0.76579
Gaussian Process Regression	1.59690	2.55000	1.00870
Medium Tree	1.82170	3.31840	1.21790
Fine Tree	2.05060	4.20510	1.26150
Linear Regression	1.99200	3.96800	1.56330
Support Vector Machine	2.54350	6.46930	1.81400
Stepwise Linear Regression	2.25080	5.06610	1.85550
Coarse Tree	2.70090	7.29490	1.97750

Table 19. Model comparison for cooling load (Y2) in experiment 4.

Trained Models	Cooling Load (kWh/m^2)		
	RMSE	**MSE**	**MAE**
Tri-Layered Neural Network	**0.81391**	**0.66245**	**0.53527**
Gaussian Process Regression	1.31090	1.71850	0.85299
Boosted Trees	1.63640	2.67770	1.08790
Medium Tree	1.80640	3.26310	1.18900
Fine Tree	1.99780	3.99130	1.24190
Bagged Trees	1.87700	3.52320	1.28010
Linear Regression	1.93530	3.74520	1.51460
Support Vector Machine	2.30970	5.33480	1.67010
Stepwise Linear Regression	2.19510	4.81860	1.78440
Coarse Tree	2.61780	6.85290	1.88240

Figure 4 shows the critical difference (CD) diagram, which illustrates the average rank of each model over four experiments. In the ranking process, each algorithm is rated according to its MAE value on the corresponding dataset. This process is performed by assigning rank 1 to the most accurate algorithm, rank 2 to the second best, and so on. In the case of ties, the average of the ranks is assigned to each algorithm. Figure 4 is useful to show the differences among various machine learning algorithms. The lower the rank (further to the left), the better performance of a model under the MAE metric compared to the others on average. In Figure 4, we observe that the TNN algorithm acquired the lowest average ranking (1) on MAE, indicating that it is the best among all comparative algorithms. TNN significantly outperformed its competitors on the MAE metric regarding predictive accuracy. Therefore, we can safely say that TNN is superior to the others with the lowest average ranking. The BaT and MT methods are tried, and similarly, the performances of LR and SLR are the same. In fact, the CT method was not performing well compared to other methods.

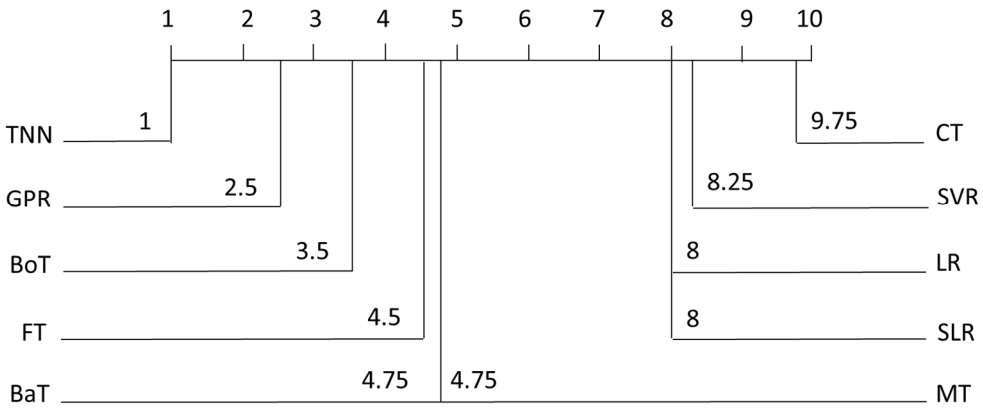


Figure 4. The critical difference diagram on the MAE metric. (TNN: tri-layered neural network, GPR: Gaussian process regression, BoT: boosted tree, FT: fine tree, BaT: bagged tree, CT: coarse tree, SVR: support vector regression, LR: linear regression, SLR: stepwise linear regression, and MT: medium tree).

The "true response" versus "predicted response" graphs are presented in four experiments in Figures 5–8 for heating load and cooling load prediction. A perfect regression model has a true response equal to the predicted response; hence, all points lie on a diagonal line. The vertical distance of any point from the line indicates the error of prediction for this point. In this study, the predictions were scattered no farther from the line. Therefore, it can be concluded that the models have small errors in all the experiments.

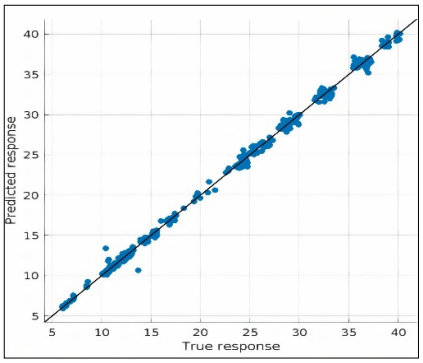


Figure 5. Predicted response versus true response for Y1 in experiment 1.

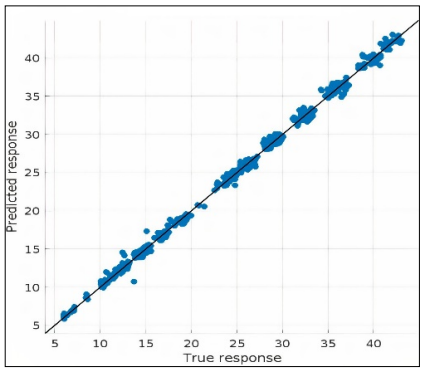


Figure 6. Predicted response versus true response for Y1 in experiment 2.

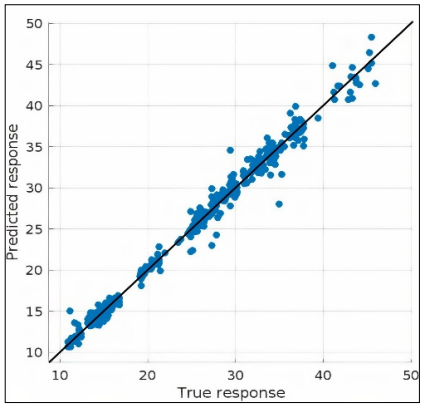


Figure 7. Predicted response versus true response for Y2 in experiment 3.

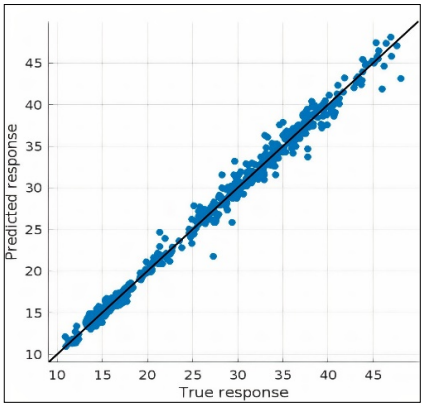


Figure 8. Predicted response versus true response for Y2 in experiment 4.

5.2. Comparison with the State-of-the-Art Studies

In order to show the superiority of our method, we compared it with the state-of-the-art methods in the literature [18,24–57]. Some of them are tree-based methods that build the model in the tree architecture by splitting the dataset into various subsets, consisting of decision nodes and leaf nodes, such as the mathematical programming tree (MPtree) [50], model tree regression (M5P) [44], conditional inference tree (CTree) [33], evolutionary tree (Evtree) [50], StatTree [33], classification and regression tree (CART) [34], and reduced error pruning tree (REPTree) [44]. Some of the methods were combined with an optimization algorithm to build one optimal model for predicting the target, such as particle swarm optimization (PSO) [28], optics-inspired optimization (OIO) [26], teaching–learning-based optimization (TLBO) [30], whale optimization algorithm (WAO) [41], ant colony optimization (ACO) [12], and Harris hawks optimization (HHO) [43]. Ensemble learning-based methods have also been used for predicting heating and cooling loads, such as AdaBoost [18], random forest [18,24,32,36,39,45,49,51,53], and regression tree ensemble [25]. When we applied the random tree (RT) algorithm [58] to the same dataset, the MAE values of 0.3780 and 0.9349 were obtained for heating load (kWh/m^2) and cooling load (kWh/m^2), respectively. Therefore, the proposed method in this study is also more efficient than RT.

Table 20 presents the previous works along with the methods and the corresponding MAE values. Since the researchers used the same dataset as our study, the results were directly taken from the referenced study. According to the results, our model achieved lower MAE values than the previous models built on the same dataset. Therefore, it can be concluded from Table 20 that the proposed method outperformed the other methods. While the differences between outputs are small for some methods [18,25,34], the improvement provided by the proposed method over some state-of-the-art methods [26–31,37,41,43,46,54] is rather significant.

The results were validated by using a statistical test to ensure the differences in performance are statistically significant. We used the Wilcoxon Test, which is a well-known nonparametric statistical test for comparing two groups. The p-values obtained for heating load and cooling load are $0.00459264 \times 10^{-21}$ and $0.00305302 \times 10^{-20}$, respectively. Therefore, it can be concluded that the results are statistically significant since the p-values are smaller than the significance level (0.05).

Table 20. Comparison of the proposed method against the state-of-the-art methods on the same dataset.

Reference	Year	Method	Heating Load (MAE) (kWh/m^2)	Cooling Load (MAE) (kWh/m^2)
Pachauri and Ahn [25]	2022	Stepwise Regression (STR)	0.997	1.631
		Squared Exponential Gaussian Process Regression (SEGPR)	0.627	2.685
		Exponential Gaussian Process Regression (EGPR)	1.323	1.065
		Matern 5/2 Exponential Gaussian Process Regression (M52GPR)	0.866	2.690
		Rational Quadratic Exponential Gaussian Process Regression (RQGPR)	0.736	2.694
		Bayesian Optimized GPR (BGPR)	0.497	0.739
		Shuffled Frog Leaping Optimization—Regression Tree Ensemble (SRTE)	0.332	0.536
Almutairi et al. [26]	2022	Firefly Algorithm—Multi-Layer Perceptron (FA-MLP)	1.797	-
		Optics-Inspired Optimization—Multi-Layer Perceptron (OIO-MLP)	1.927	-
		Shuffled Complex Evolution—Multi-Layer Perceptron (SCE-MLP)	1.607	-
		Teaching–Learning-Based Optimization—Multi-Layer Perceptron (TLBO-MLP)	1.580	-
Zheng et al. [27]	2022	Shuffled Complex Evolution—Multi-Layer Perceptron (SCE-MLP)	-	1.8124

Table 20. *Cont.*

Reference	Year	Method	Heating Load (MAE) (kWh/m^2)	Cooling Load (MAE) (kWh/m^2)
Xu et al. [28]	2022	Biogeography-Based Optimization (BBO)	2.350	2.460
		Genetic Algorithm (GA)	2.730	2.410
		Particle Swarm Optimization (PSO)	3.720	3.010
		Population-Based Incremental Learning (PBIL)	5.580	4.170
		Evolution Strategy (ES)	6.650	4.490
		Ant Colony Optimization (ACO)	8.790	6.650
Fard and Hosseini [18]	2022	K-Nearest Neighbors	1.512	1.339
		AdaBoost	0.292	0.911
		Random Forest	0.361	1.129
		Neural Network	2.744	3.192
Yildiz et al. [29]	2022	Decision Tress	2.520	2.400
Zhou et al. [30]	2021	Teaching–Learning-Based Optimization—Multi-Layer Perceptron (TLBO-MLP)	-	1.829
Moayedi and Mosavi [31]	2021	Multi-Layer Perceptron Neural Network (MLPNN)	-	2.457
		Grasshopper Optimization Algorithm—Artificial Neural Network (GOA-ANN)	-	1.895
		Firefly Algorithm—Artificial Neural Network (FA-ANN)	-	2.026
		Stochastic Fractal Search—Artificial Neural Network (SFS–ANN)	-	1.583
Huang and Li [12]	2021	Wavelet Neural Network (WNN)	4.825	4.617
		Ant Colony Optimization—Wavelet Neural Network (ACO-WNN)	3.516	3.144
		Improved ACO-WNN (I-ACO-WNN)	0.847	0.700
Hosseini and Fard [32]	2021	Decision Tree	0.725	1.274
		Random Forest	0.404	1.128
		K-Nearest Neighbors	1.692	1.512
		StatTree	0.367	1.175
Gkioulekas and Papageorgiou [33]	2021	Mathematical Programming Tree (MPtree)	0.354	0.891
		Cubist	0.347	0.938
		Classification and Regression Tree (CART)	2.011	2.400
		Model Tree (M5P)	0.693	1.210
		Conditional Inference Tree (CTree)	0.665	1.403
Chou et al. [34]	2021	Artificial Neural Network (ANN)	0.360	0.799
		ANN + Classification and Regression Tree (CART)	0.352	0.900
		Bagging ANN	0.291	0.556
Altay et al. [35]	2021	Linear Ridge Regression (LRR)	3.226	3.619
		Linear Regression (LR)	2.087	2.264
		Support Vector Regression (SVR)	2.043	2.244
		Discrete-time Chaotic Systems-based Extreme Learning Machine (DCS-ELM)	0.803	1.074
Goyal and Pandey [36]	2021	Multiple Linear Regression (MLR)	2.610	2.620
		K-Nearest Neighbours (KNN)	1.960	1.540
		Support Vector Regression (SVR)	3.190	2.250
		Random Forest	0.360	1.390
		Gradient Boosting Machines	0.380	1.250
		Extreme Gradient Boosting	0.370	1.270
Zhou et al. [37]	2020	Multi-Layer Perceptron (MLP)	2.460	2.427
		Artificial Bee Colony—Multi-Layer Perceptron (ABC-MLP)	1.911	2.176
		Particle Swarm Optimization—Multi-Layer Perceptron (PSO-MLP)	1.863	2.136
Xudong et al. [38]	2020	Media Loss Rate (MLR)	2.253	2.277
		Support Vector Regression (SVR)	1.207	1.546
		Extreme Learning Machine (ELM)	0.659	1.211
		Long Short-Term Memory (LSTM)	0.453	1.170
		Improved Particle Swarm Optimization—Long Short-Term Memory (IPSO-LSTM)	0.375	1.166
		Improved Particle Swarm Optimization—Convolution Long Short-Term Memory (IPSO-CLSTM)	0.343	1.020
Rashidifar and Chen [39]	2020	Random Forest	0.36	1.24
Moradzadeh et al. [40]	2020	Multi-Layer Perceptron (MLP)	0.411	2.097
		Support Vector Regression (SVR	0.778	1.476

Table 20. *Cont.*

Reference	Year	Method	Heating Load (MAE) (kWh/m^2)	Cooling Load (MAE) (kWh/m^2)
Guo et al. [41]	2020	Wind-Driven Optimization—Multi-Layer Perceptron (WDO-MLP)	1.986	2.242
		Whale Optimization Algorithm—Multi-Layer Perceptron (WOA-MLP)	2.192	2.539
		Spotted Hyena Optimization—Multi-Layer Perceptron (SHO-MLP)	3.109	4.593
		Salp Swarm Algorithm—Multi-Layer Perceptron (SSA-MLP)	1.917	2.183
Akgundogdu [42]	2020	Linear Regression	1.970	2.146
		Multi-Layer Perceptron (MLP)	1.406	1.635
		Radial Basis Function Network (RBFN)	1.794	2.001
		Support Vector Machine (SVM)	1.892	2.066
		Gaussian Processes (GP)	1.958	2.150
		Adaptive Neuro-Fuzzy Inference System (ANFIS)	0.460	1.260
Moayedi et al. [43]	2020	Ant Colony Optimization (ACO)—Multi-Layer Perceptron (ACO-MLP)	-	2.601
		Harris Hawks Optimization—Multi-Layer Perceptron (HHO-MLP)	-	2.326
		Elephant Herding Optimization—Multi-Layer Perceptron (EHO-MLP)	-	2.128
Namli et al. [44]	2019	Multi-Layer Perceptron (MLP)	0.840	1.838
		Support Vector Regression (SVR)	2.040	2.205
		Instance-based Learning (IBk)	3.326	3.580
		Locally Weighted Learning (LWL)	3.303	3.009
		Model Trees Regression (M5P)	0.649	1.186
		Reduced Error Pruning Tree (REPTree)	0.386	1.179
Le et al. [45]	2019	Particle Swarm Optimization—Extreme Gradient Boosting Machine (PSO-XGBoost)	0.615	-
		Extreme Gradient Boosting Machine (XGBoost)	0.720	-
		Support Vector Machine (SVM)	0.910	-
		Random Forest	0.557	-
		Genetic Programming (GP)	0.798	-
		Classification and Regression Tree (CART)	0.773	-
Bui et al. [46]	2019	Artificial Neural Network (ANN)	2.938	3.283
		Genetic Algorithm—Artificial Neural Network (GA-ANN)	2.062	2.098
		Imperialist Competition Algorithm—Artificial Neural Network (ICA-ANN)	2.008	2.105
Gkioulekas and Papageorgiou [47]	2019	Piecewise Regression with Iterative Akaike (PRIA)	0.820	1.337
		Piecewise Regression with Iterative Bayesian (PRIB)	0.909	1.342
		Piecewise Regression with Optimised Akaike (PROA)	0.806	1.275
		Piecewise Regression with Optimised Bayesian (PROB)	0.906	1.351
Le et al. [48]	2019	Genetic Algorithm—Artificial Neural Network (GA-ANN)	0.798	-
		Particle Swarm Optimization—Artificial Neural Network (PSO-ANN)	1.027	-
		Imperialist Competitive Algorithm—Artificial Neural Network (ICA-ANN)	0.980	-
		Artificial Bee Colony—Artificial Neural Network (ABC-ANN)	0.957	-
Razali et al. [49]	2018	Radial Basis Function Neural Network (RBFNN)	0.320	0.890
		Random Forest (RF)	0.510	1.420
		Mathematical Programming Tree (MPTree)	0.350	0.800
Yang et al. [50]	2017	Classification and Regression Tree (CART)	2.000	2.380
		Conditional Inference Tree (Ctree)	0.630	1.400
		Evolutionary Tree (Evtree)	0.560	1.590
		M5P Tree	0.690	1.210
		Cubist	0.350	0.890
Peker et al. [51]	2017	Support Vector Machine (SVM)	2.439	3.186
		Linear Regression	2.074	2.240
		Random Forest	0.422	1.339
Altun et al. [52]	2017	K-Nearest Neighbors (KNN)	1.512	1.313
		Artificial Neural Network (ANN)	0.350	0.800

Table 20. *Cont.*

Reference	Year	Method	Heating Load (MAE) (kWh/m^2)	Cooling Load (MAE) (kWh/m^2)
Yang et al. [53]	2016	Linear Regression	2.089	2.266
		Multi-Layer Perceptron	0.993	1.924
		Kriging	1.788	2.044
		Support Vector Regression (SVR)	2.036	2.191
		K-Nearest Neighbors (KNN)	1.937	2.148
		Random Forest	1.435	1.644
		Multivariate Adaptive Regression Splines (MARS)	0.796	1.324
		Pace Regression	2.089	2.261
		Automated Learning of Algebraic Models for Optimization (ALAMO)	2.722	2.765
		Optimal Piecewise Linear Regression Analysis (OPLRA)	0.810	1.278
Ertugrul and Kaya [54]	2016	Extreme Learning Machine (ELM)	2.031	1.726
		Artificial Neural Network (ANN)	2.304	1.946
		Linear Regression (LR)	2.880	2.450
		K-Nearest Neighbor Regression (KNNR)	2.558	1.990
		Ridge Regression (Ridger)	2.127	2.293
		Kernel Smoother (kSmooth)	2.332	1.916
		Pseudo-Inverse Regression (PINVR)	2.091	2.269
		Partial Least Squares Regression (PLSR)	2.160	2.320
Castelli et al. [55]	2015	Geometric Semantic Genetic Programming (GSGP)	1.310	1.470
		GSGP with Local Search (HYBRID)	1.260	1.370
		HYBRID Approach Integrated with Linear Scaling (HYBRID-LIN)	0.510	1.180
		Evolutionary Multivariate Adaptive Regression Splines (EMARS)	0.350	0.710
Cheng and Cao [56]	2014	Multivariate Adaptive Regression Splines (MARS)	0.530	1.120
		Back-Propagation Neural Network (BPNN)	1.610	1.920
		Radial Basis Function Neural Network (RBFNN)	0.510	1.300
		Classification And Regression Tree (CART)	0.730	1.310
		Support Vector Machine (SVM)	2.190	2.100
Nebot and Mugica [57]	2013	Adaptive Neuro-Fuzzy Inference System (ANFIS)	0.520	1.060
		Fuzzy Inductive Reasoning (FIR)	0.350	1.090
Tsanasa and Xifarab [24]	2012	Random Forest	0.510	1.420
		Iteratively Reweighted Least Squares (IRLS)	2.140	2.210
		Average	1.506	1.893
Proposed Method		**Tri-Layered Neural Network (TNN) + Maximum Relevance Minimum Redundancy (MRMR)**	**0.289**	**0.535**

6. Conclusions and Future Works

This paper focuses on the consumption of energy in IoT-based smart buildings, some of which have the symmetry property. This study's main aim is to predict cooling and heating loads in buildings. For this purpose, it proposes novel machine learning models by selecting the best features. It is the first study that uses both the tri-layered neural network (TNN) and maximum relevance minimum redundancy (MRMR) algorithms together to predict energy consumption in smart buildings. Our study is also original in that it proposes a multitarget learning solution, unlike the traditional single-target learning studies.

The experimental studies were conducted on an energy-efficiency dataset. The building-related features in the dataset were investigated in terms of skewness to determine whether their distributions are symmetric or asymmetric. As lower MAE means the higher accuracy of a model, and the lower training time (between 10 and 19 s) in all experiments is also an important factor for assessing predictive models, the results show the efficiency of the proposed method. The results also show that the relative compactness (X1) and glazing area (X7) are the features of buildings that have the highest effect on the amount of energy consumption inside the buildings. Moreover, the orientation (X6) and glazing area distribution (X8) are the other features that have the least effect on the energy consumption in buildings. The best mean absolute error was calculated as 0.28993 for heating load (kWh/m^2) prediction and 0.53527 for cooling load (kWh/m^2) prediction. The

experimental results showed that our method outperformed the state-of-the-art methods on the same dataset.

For future works, the proposed model can be combined with thermal sensors inside the smart buildings to predict energy consumption not only based on the building features but also considering the temperature from different areas of the building. Moreover, as this study aims to balance the energy consumption in buildings precisely based on machine learning predictions, it can be developed into a smart energy recycling system to trade off cooling load and heating load in different areas of the building according to related features. As another trend, it can be advised to present a novel system that applies the security measurements for saving the related appliances of the building by considering the threshold temperatures. In addition, mobile phone apps can be implemented for real-time remote monitoring and controlling the energy consumption inside the buildings. In addition, generating daily, weekly, or monthly reports is possible through IoT-based buildings to have an efficient building energy management system (BEMS) through predictive models.

Author Contributions: Conceptualization, B.G. and D.B.; methodology, B.G., R.Y., D.B. and R.A.K.; software, B.G.; validation, B.G., R.Y., D.B. and R.A.K.; formal analysis, D.B.; investigation, B.G., D.B. and R.A.K.; resources, B.G., R.Y. and D.B.; data curation, R.Y. and R.A.K.; writing—original draft preparation, B.G. and D.B.; writing—review and editing, B.G. and D.B.; visualization, B.G.; supervision, R.A.K., D.B. and R.Y.; project administration, R.A.K.; funding acquisition, R.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The "Energy Efficiency" dataset [24] is publicly available in the UCI (University of California Irvine) dataset repository (https://archive.ics.uci.edu/ml/datasets/Energy+efficiency, accessed on 30 June 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saravanan, G.; Parkhe, S.; Thakar, C.; Kulkarni, V.; Mishra, H.; Gulothungan, G. Implementation of IoT in production and manufacturing: An Industry 4.0 approach. *Mater. Today Proc.* **2022**, *51*, 2427–2430. [CrossRef]
2. Chaudhary, S.; Johari, R.; Bhatia, R.; Gupta, K.; Bhatnagar, A. CRAIOT: Concept, Review and Application(S) of IoT. In Proceedings of the 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 18–19 April 2019; pp. 1–4.
3. Dholu, M.; Ghodinde, K. Internet of Things (IoT) for precision agriculture application. In Proceedings of the 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 11–12 May 2018; pp. 339–342.
4. Thakur, N.; Han, C.Y. Indoor localization for personalized ambient assisted living of multiple users in multi-floor smart environments. *Big Data Cogn. Comput.* **2021**, *5*, 42. [CrossRef]
5. Parvathi Sangeetha, B.; Kumar, N.; Ambalgi, A.; Abdul Haleem, S.; Thilagam, K.; Vijayakumar, P. IOT based smart irrigation management system for environmental sustainability in India. *Sustain. Energy Technol. Assess.* **2022**, *52*, 101973. [CrossRef]
6. Anand, L.; Kotha, M.; Kannan, N.; Kumar, S.; Meera, M.; Shawl, R.; Ray, A. Design and development of IoT based health monitoring system for military applications. *Mater. Today Proc.* **2022**, 1–4. [CrossRef]
7. Kaur, J.; Santhoshkumar, N.; Nomani, M.; Kumar Sharma, D.; Pai Maroor, J.; Dhiman, V. Impact of internets of things (IOT) in retail sector. *Mater. Today Proc.* **2022**, *51*, 26–30. [CrossRef]
8. Nithya, N.; Nallavan, G.; Sriabirami, V. A study on surface electromyography in sports applications using IoT. In *Intelligent Data Communication Technologies and Internet of Things*; Hemanth, D.J., Pelusi, D., Vuppalapati, C., Eds.; Lecture Notes on Data Engineering and Communications Technologies; Springer: Singapore, 2022; Volume 101, pp. 855–867.
9. Motlagh, N.H.; Mohammadrezaei, M.; Hunt, J.; Zakeri, B. Internet of things (IoT) and the energy sector. *Energies* **2020**, *13*, 1–27.
10. Khanna, A.; Kaur, S. Internet of things (IoT), applications and challenges: A comprehensive review. *Wirel. Pers. Commun.* **2020**, *114*, 1687–1762. [CrossRef]
11. Mahdavinejad, M.; Rezvan, M.; Barekatain, M.; Adibi, P.; Barnaghi, P.; Sheth, A. Machine learning for Internet of things data analysis: A survey. *Digit. Commun. Netw.* **2018**, *4*, 161–175. [CrossRef]
12. Huang, Y.; Li, C. Accurate heating, ventilation and air conditioning system load prediction for residential buildings using improved ant colony optimization and wavelet neural network. *J. Build. Eng.* **2021**, *35*, 101972. [CrossRef]

13. Gaber, T.; El-Ghamry, A.; Hassanien, A. injection attack detection using machine learning for smart Iot applications. *Phys. Commun.* **2022**, *52*, 101685. [CrossRef]
14. Mondal, T.; Mugaloremutt Jayadeva, S.; Pani, R.; Subramanian, M.; Ashokkumar, P.; Sumana, B. E Marketing strategy in health care using IoT and machine learning. *Mater. Today Proc.* **2022**, *56*, 2087–2091. [CrossRef]
15. Siaterlis, G.; Franke, M.; Klein, K.; Hribernik, K.; Papapanagiotakis, G.; Palaiologos, S.; Antypas, G.; Nikolakis, N.; Alexopoulos, K. An IIoT approach for edge intelligence in production environments using machine learning and knowledge graphs. *Procedia CIRP* **2022**, *106*, 282–287. [CrossRef]
16. Ribeiro Junior, F.; Bianchi, R.; Prati, R.; Kolehmainen, K.; Soininen, J.; Kamienski, C. Data reduction based on machine learning algorithms for fog computing in IoT smart agriculture. *Biosyst. Eng.* **2022**, 1–17. [CrossRef]
17. Tiwari, D.; Bhati, B.; Nagpal, B.; Sankhwar, S.; Al-Turjman, F. An Enhanced intelligent model: To protect marine IoT sensor environment using ensemble machine learning approach. *Ocean. Eng.* **2021**, *242*, 110180. [CrossRef]
18. Fard, R.; Hosseini, S. Machine Learning algorithms for prediction of energy consumption and IoT modeling in complex networks. *Microprocess Microsyst.* **2022**, *89*, 104423. [CrossRef]
19. Cakir, M.; Guvenc, M.; Mistikoglu, S. The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system. *Comput. Ind. Eng* **2021**, *151*, 106948. [CrossRef]
20. Rahman, H.; Faruq, M.; Abdul Hai, T.; Rahman, W.; Hossain, M.; Hasan, M.; Islam, S.; Moinuddin, M.; Islam, M.; Azad, M. IoT enabled mushroom farm automation with Machine Learning to classify toxic mushrooms in Bangladesh. *J. Agric. Res.* **2022**, *7*, 100267. [CrossRef]
21. Meghana, J.; Hanumanthappa, J.; Prakash, S. Performance comparison of machine learning algorithms for data aggregation in social internet of things. *Glob. Transit. Proc.* **2021**, *2*, 212–219. [CrossRef]
22. Khan, A.; Al-Badi, A. Open source machine learning frameworks for industrial internet of things. *Procedia Comput. Sci.* **2020**, *170*, 571–577. [CrossRef]
23. Zhao, Z.; Anand, R.; Wang, M. Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform. In Proceedings of the 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Washington, DC, USA, 5–8 October 2019; pp. 442–452.
24. Tsanas, A.; Xifara, A. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy Build.* **2012**, *49*, 560–567. [CrossRef]
25. Pachauri, N.; Ahn, C. Regression tree ensemble learning-based prediction of the heating and cooling loads of residential buildings. *Build. Simul.* **2022**, 1–15. [CrossRef]
26. Almutairi, K.; Algarni, S.; Alqahtani, T.; Moayedi, H.; Mosavi, A. A TLBO-Tuned neural processor for predicting heating load in residential buildings. *Sustainability* **2022**, *14*, 5924. [CrossRef]
27. Zheng, S.; Lyu, Z.; Foong, L. Early prediction of cooling load in energy-efficient buildings through novel optimizer of shuffled complex evolution. *Eng. Comput.* **2020**, *38*, 105–119. [CrossRef]
28. Xu, Y.; Li, F.; Asgari, A. Prediction and optimization of heating and cooling loads in a residential building based on multi-layer perceptron neural network and different optimization algorithms. *Energy J.* **2022**, *240*, 122692. [CrossRef]
29. Yildiz, M.; Beyhan, F.; Uçar, M. Enerji verimli bina tasarımında kural tabanlı yöntem yardımıyla ısıtma ve soğutma yüklerinin tahminlemesi. *OJAD* **2021**, *10*, 73–80.
30. Zhou, G.; Moayedi, H.; Foong, L. Teaching–learning-based metaheuristic scheme for modifying neural computing in appraising energy performance of building. *Eng. Comput.* **2021**, *37*, 3037–3048. [CrossRef]
31. Moayedi, H.; Mosavi, A. Suggesting a stochastic fractal search paradigm in combination with artificial neural network for early prediction of cooling load in residential buildings. *Energies* **2021**, *14*, 1649. [CrossRef]
32. Hosseini, S.; Fard, R. Machine learning algorithms for predicting electricity consumption of buildings. *Wirel. Pers. Commun.* **2021**, *121*, 3329–3341. [CrossRef]
33. Gkioulekas, I.; Papageorgiou, L. Tree regression models using statistical testing and mixed integer programming. *Comput. Ind. Eng.* **2021**, *153*, 107059. [CrossRef]
34. Chou, J.; Truong, D.; Tsai, C. Solving regression problems with intelligent machine learner for engineering informatics. *Mathematics* **2021**, *9*, 686. [CrossRef]
35. Altay, O.; Ulas, M.; Alyamac, K. DCS-ELM: A novel method for extreme learning machine for regression problems and a new approach for the SFRSCC. *PeerJ Comput. Sci.* **2021**, *7*, 411. [CrossRef]
36. Goyal, M.; Pandey, M. A systematic analysis for energy performance predictions in residential buildings using ensemble learning. *Arab. J. Sci. Eng.* **2021**, *46*, 3155–3168. [CrossRef]
37. Zhou, G.; Moayedi, H.; Bahiraei, M.; Lyu, Z. Employing artificial bee colony and particle swarm techniques for optimizing a neural network in prediction of heating and cooling loads of residential buildings. *J. Clean. Prod.* **2020**, *254*, 120082. [CrossRef]
38. Xudong, L.; Shuo, L.; Fan, Q. Prediction of building energy consumption based on IPSO-CLSTM neural network. *J. Auton. Intell.* **2021**, *3*, 11–22.
39. Rashidifar, R.; Chen, F. Estimation of energy performance of buildings using machine learning tools. *Eng. Engrxiv Arch.* **2020**, 1–6.
40. Moradzadeh, A.; Mansour-Saatloo, A.; Mohammadi-Ivatloo, B.; Anvari-Moghaddam, A. Performance evaluation of two machine learning techniques in heating and cooling loads forecasting of residential buildings. *Appl. Sci.* **2020**, *10*, 3829. [CrossRef]

41. Guo, Z.; Moayedi, H.; Foong, L.; Bahiraei, M. Optimal modification of heating, ventilation, and air conditioning system performances in residential buildings using the integration of metaheuristic optimization and neural computing. *Energy Build.* **2020**, *214*, 109866. [CrossRef]
42. Akgundogdu, A. Comparative analysis of regression learning methods for estimation of energy performance of residential structures. *Erzincan Univ. JST* **2020**, *13*, 600–608.
43. Moayedi, H.; Mu'azu, M.; Foong, L. Novel swarm-based approach for predicting the cooling load of residential buildings based on social behavior of elephant herds. *Energy Build.* **2020**, *206*, 109579. [CrossRef]
44. Namlı, E.; Erdal, H.; Erdal, H. Artificial intelligence-based prediction models for energy performance of residential buildings. In *Recycling and Reuse Approaches for Better Sustainability*, 2nd ed.; Balkaya, N., Guneysu, S., Eds.; Springer: Cham, Switzerland, 2018; Volume 1, pp. 141–149.
45. Le, L.; Nguyen, H.; Zhou, J.; Dou, J.; Moayedi, H. Estimating the heating load of buildings for smart city planning using a novel artificial intelligence technique PSO-XGBOOST. *Appl. Sci.* **2019**, *9*, 2714. [CrossRef]
46. Tien Bui, D.; Moayedi, H.; Anastasios, D.; Kok Foong, L. Predicting heating and cooling loads in energy-efficient buildings using two hybrid intelligent models. *Appl. Sci.* **2019**, *9*, 3543. [CrossRef]
47. Gkioulekas, I.; Papageorgiou, L. Piecewise regression analysis through information criteria using mathematical programming. *Expert Syst. Appl.* **2019**, *121*, 362–372. [CrossRef]
48. Le, L.T.; Nguyen, H.; Dou, J.; Zhou, J. A comparative study of PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in estimating the heating load of buildings' energy efficiency for smart city planning. *Appl. Sci.* **2019**, *9*, 2630. [CrossRef]
49. Razali, C.M.C.; Faruq, A. Comparative study between radial basis function neural network and random forest algorithm for building energy estimation. In Proceedings of the Fifth Malaysia-Japan Joint International Conference (MJJIC 2018), Sepang, Malaysia, 17–18 October 2018; pp. 3–4.
50. Yang, L.; Liu, S.; Tsoka, S.; Papageorgiou, L. A regression tree approach using mathematical programming. *Expert Syst. Appl.* **2017**, *78*, 347–357. [CrossRef]
51. Peker, M.; Özkaraca, O.; Kesimal, B. Enerji tasarruflu bina tasarımı için ısıtma ve soğutma yüklerini regresyon tabanlı makine öğrenmesi algoritmaları ile modelleme. *JIT* **2017**, *10*, 443–449.
52. Altun, M.; Ersöz, A.B.; Akçamete Güngör, A.; Pekcan, O. Application of artificial neural networks on building energy estimation. In Proceedings of the International Conference on Engineering Technologies, Konya, Turkey, 7–9 December 2017; pp. 1–4.
53. Yang, L.; Liu, S.; Tsoka, S.; Papageorgiou, L. Mathematical programming for piecewise linear regression analysis. *Expert Syst. Appl.* **2016**, *44*, 156–167. [CrossRef]
54. Ertugrul, Ö.F.; Kaya, Y. Smart city planning by estimating energy efficiency of buildings by extreme learning machine. In Proceedings of the 2016 4th International Istanbul Smart Grid Congress and Fair (ICSG), Istanbul, Turkey, 20–21 April 2016; pp. 1–5.
55. Castelli, M.; Trujillo, L.; Vanneschi, L.; Popovič, A. Prediction of energy performance of residential buildings: A genetic programming approach. *Energy Build.* **2015**, *102*, 67–74. [CrossRef]
56. Cheng, M.; Cao, M. Accurately predicting building energy performance using evolutionary multivariate adaptive regression splines. *Appl. Soft Comput.* **2014**, *22*, 178–188. [CrossRef]
57. Nebot, À.; Mugica, F. Fuzzy approaches improve predictions of energy performance of buildings. In Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Reykjavík, Iceland, 29–31 July 2013; pp. 504–511.
58. Witten, I.H.; Frank, E. *Data Mining: Practical Machine Learning Tools with Java Implementations*, 3rd ed.; Morgan Kaufmann: San Francisco, CA, USA, 2000; pp. 191–303.