

Article

Multi-Objective Plum Tree Algorithm and Machine Learning for Heating and Cooling Load Prediction

Adam Slowik¹  and Dorin Moldovan^{2,*} 

¹ Department of Electronics and Computer Science, Koszalin University of Technology, 75-453 Koszalin, Poland; adam.slowik@tu.koszalin.pl

² Independent Researcher, 405200 Dej, Romania

* Correspondence: dorin.i.moldovan@gmail.com

Abstract: The prediction of heating and cooling loads using machine learning algorithms has been considered frequently in the research literature. However, many of the studies considered the default values of the hyperparameters. This manuscript addresses both the selection of the best regressor and the tuning of the hyperparameter values using a novel nature-inspired algorithm, namely, the Multi-Objective Plum Tree Algorithm. The two objectives that were optimized were the averages of the heating and cooling predictions. The three algorithms that were compared were the Extra Trees Regressor, the Gradient Boosting Regressor, and the Random Forest Regressor of the sklearn machine learning Python library. We considered five hyperparameters which were configurable for each of the three regressors. The solutions were ranked using the MOORA method. The Multi-Objective Plum Tree Algorithm returned a root mean square error value for heating equal to 0.035719 and a root mean square error for cooling equal to 0.076197. The results are comparable to the ones obtained using standard multi-objective algorithms such as the Multi-Objective Grey Wolf Optimizer, Multi-Objective Particle Swarm Optimization, and NSGA-II. The results are also performant concerning the previous studies, which considered the same experimental dataset.

Keywords: plum tree algorithm; multi-objective optimization; energy efficiency; prediction; heating and cooling loads



Citation: Slowik, A.; Moldovan, D. Multi-Objective Plum Tree Algorithm and Machine Learning for Heating and Cooling Load Prediction. *Energies* **2024**, *17*, 3054. <https://doi.org/10.3390/en17123054>

Academic Editor: Marcin Kaminski

Received: 23 May 2024

Revised: 17 June 2024

Accepted: 19 June 2024

Published: 20 June 2024



Copyright: © 2024 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

Energy conservation and emission reduction have recently received a lot of attention in the context of the increase in energy consumption [1]. The prediction of heating and cooling loads could lead to a rational use of renewable energy to replace thermal energy systems which are based on fossil fuel. Moreover, reducing heating and cooling loads can help to decarbonize the building sector [2]. Therefore, the application of machine learning techniques has become important in the context of heating and cooling load prediction. Even though many living buildings are usually equipped with electricity metrics which provide almost complete statistics, the data are often not enough to develop complex machine learning models for the prediction of heating and cooling loads. Information about characteristics which are particular to each building, such as the wall area, the surface area, and the roof area, can complement the metrics data to develop algorithms which are more accurate and which better address the particularities of buildings.

Buildings are responsible for approximately 40% of total global energy consumption [3]. The improvement of energy efficiency and the conservation of more energy has become essential in recent years [4,5] due to the adverse effects of high-energy consumption on the environment. The estimation of heating and cooling loads depends on the characteristics of the structure. To construct energy-efficient buildings, it is helpful to develop conceptual systems that anticipate the cooling load in the residential building sector [6].

Since energy resources present limitations and have an important role in the economic development of countries, reduction in energy consumption represents a necessity [7,8]. The

modeling of heating and cooling loads represents the cornerstone of energy-efficient building design. Furthermore, as stated in [9], energy efficiency leads to both environmental and financial benefits [10]. Moreover, energy efficiency directly impacts economic competitiveness and sustainable development [11]. These facts underscore the significance of the work presented in the current manuscript.

The accurate prediction of energy consumption and the determination of the factors that influence heating energy consumption are important [12] in the context of the substantial increase in energy consumption in the case of residential buildings [13]. Therefore, the use of advanced machine learning algorithms for energy consumption prediction presents great interest for researchers. As can be seen in [14], methods based on machine learning for the prediction of energy consumption have advanced significantly in recent years. The development and examination of machine learning algorithms which can learn from patterns in the data and make predictions presented a lot of interest for many scholars and scientists [15].

Several statistics, such as the ones presented by the authors of [16], estimate that by 2040, there will be an increase of nearly 25% in global energy demand. Also, to create a sustainable and healthy economy, it is important to measure the economic and environmental effects of energy production [17].

The optimization of building energy prediction represents an important research area because of its potential to improve the efficiency of energy management systems [18]. Many studies have shown that the air conditioning system consumes up to 38% of the total energy in the building sector [19,20].

As can be seen, the energy efficiency research domain will present great interest to the research community in the years to come, and the application of novel machine learning and artificial intelligence techniques can lead to a significant improvement in the existing techniques used for energy consumption prediction.

The solution presented in the manuscript is specific to the heating and cooling load prediction optimization problem, as it aims to improve the prediction results for two objectives, namely the heating load and the cooling load. Two particularities of the proposed solution were the consideration of three machine learning algorithms which had common hyperparameters and returned good predictions for energy data characterized by a small number of features and the development of an objective function which considered a 10-fold cross-validation and averaged the heating and cooling prediction results.

The main contributions of the work presented in this paper are as follows:

- (1) A critical review of the application of machine learning methods for the prediction of heating and cooling loads;
- (2) The introduction of a novel algorithm called the Multi-Objective Plum Tree Algorithm (MOPTA) which adapts the original Plum Tree Algorithm [21] to multi-objective optimization problems;
- (3) The ranking of the solutions using the MOORA method [22];
- (4) The adaptation of the MOPTA to the hyperparameter optimization and the optimal regressor selection for a machine learning methodology used to predict heating and cooling loads, using the Energy Efficiency Dataset of the UCI Machine Learning Repository as experimental support [23,24];
- (5) The development of an objective function that considers the averages of the heating and cooling RMSE results;
- (6) The comparison and validation of the obtained results with the ones obtained by the Multi-Objective Grey Wolf Optimizer (MOGWO) [25], Multi-Objective Particle Swarm Optimization (MOPSO) [26], and NSGA-II [27].

The manuscript is structured as follows: Section 2 presents the research background, Section 3 presents the MOPTA-based machine learning methodology for the optimization of heating and cooling load prediction, Section 4 presents the results, Section 5 compares the obtained results with the ones from previous studies, and Section 6 shows the conclusions.

2. Research Background

This section reviews representative recent studies that considered the application of machine learning for the prediction of heating and cooling loads.

In [28], the authors approached the prediction of the energy efficiency of buildings using machine learning techniques considering a small data approach. The method proposed by them considered the Support Vector Regression and the K-means algorithms. The prediction of heating and cooling parameters was considered as two separate tasks. The dataset was split at 75%:25% and the metrics used to evaluate the performance were mean square error and mean absolute error. Their proposed method was better in terms of mean square error and mean absolute error than other methods, such as the classical Support Vector Regression with rbf kernel.

The review from [29] presented a selection of representative studies that used data-driven techniques, such as machine learning and artificial intelligence, for the prediction of the cooling and heating loads of residential buildings. The review considered various techniques, such as ensemble learning, Artificial Neural Network, Support Vector Machines, probabilistic models, and statistical models. As support for their experiments, the review also considered recent studies that used the same experimental dataset as the one used in our manuscript. For example, the approach presented in [30] used an ensemble machine learning model based on three Random Forest models that achieved 0.999 R^2 for the heating load prediction and 0.997 R^2 for the cooling load prediction using a 10-fold cross-validation approach. On the other hand, the authors of [31] used an approach based on the Tri-Layered Neural Network and Maximum Relevance Minimum Redundancy that led to 0.289 mean absolute error for heating load and 0.535 mean absolute error for cooling load, respectively.

The approach presented in [32] considered a novel method for energy consumption estimation using Support Vector Machine and Random Forest. The Owl Search Algorithm [33] was used to improve the performance of these two algorithms. The root mean square error values returned by the approaches based on Support Vector Machine and Random Forest were 0.85 and 1.29 for heating and 1.02 and 1.65 for cooling, respectively.

The approach presented in [34] compared four algorithms, namely, the Linear Regression, the Decision Tree, the Random Forest, and the XGBoost. Like our approach, the data were split randomly into 80% training data and 20% testing data. The hyperparameters were optimized using Bayesian Optimization [35]. The best results in terms of root mean square error for the testing data were obtained by the XGBoost algorithm, as follows: 0.3797 for the heating load and 0.7578 for the cooling load.

The mean square error results obtained by the authors of [36] were 0.201 for the heating load prediction and 2.56 for the cooling load forecast. Another approach—for example, the one presented in [37], which used the Multilayer Perceptron and Support Vector Regression algorithms—returned 0.4832 and 0.8853 root mean square error values for the heating load prediction and 2.626 and 1.7389 root mean square error values for the cooling load prediction. On the other hand, the approach presented in [38] based on the Gated Recurrent Unit returned the 0.0166 and 0.0247 root mean square error values for the heating and cooling load predictions, respectively, when hold-out was used, and 0.01 root mean square error values for both heating and cooling load predictions when 10-fold validation was used.

The authors of [39] considered an approach based on a Multi-Objective Optimization method for the tuning of the hyperparameters of a Random Forest model used for the prediction of heating and the cooling loads. The two objectives that were optimized were the averages of the heating and cooling load prediction. Compared to their approach, our method also predicts which regressor to use as part of the multi-objective optimization process.

3. Multi-Objective Plum Tree Algorithm (MOPTA) Machine Learning Methodology for Heating and Cooling Load Prediction

The original version of the Plum Tree Algorithm was introduced in [21] with the following sources of inspiration:

- The plum trees flowering at the beginning of spring;

- The transformation of the flowers, which are pollinated into plums;
- The dropping of a percentage of the plums before maturity due to various reasons;
- The continuity of the lives of the plums after the harvest for a couple of weeks.

The PTA presents similarities with other bio-inspired algorithms, such as Chicken Swarm Optimization [40], Particle Swarm Optimization, Grey Wolf Optimizer [41], and Crow Search Algorithm [42], which influenced how particular mathematical parts of the algorithm were modeled.

Table 1 summarizes the PTA’s configurable parameters.

Table 1. PTA configurable parameters.

Parameter	Description
I	the total number of iterations
D	the number of dimensions of the search space
N	the number of plums
FT	the fruitiness threshold
RT	the ripeness threshold
$[FR_{min}, FR_{max}]$	the minimum and the maximum values of the fruitiness rate
ϵ	a constant used for avoiding division by 0
$[X_{min}, X_{max}]$	the minimum and the maximum possible values of the positions
OF	the objective function used to evaluate the plums

The PTA starts with the initialization of N flowers in a search space with D dimensions, such that the values are selected randomly from the range $[X_{min}, X_{max}]$:

$$flowers = \begin{bmatrix} flower_{1,1}^0 & \dots & flower_{1,D}^0 \\ \dots & \dots & \dots \\ flower_{N,1}^0 & \dots & flower_{N,D}^0 \end{bmatrix}. \tag{1}$$

Then, N plums are initialized with the value of the flowers:

$$plums = \begin{bmatrix} plum_{1,1}^0 & \dots & plum_{1,D}^0 \\ \dots & \dots & \dots \\ plum_{N,1}^0 & \dots & plum_{N,D}^0 \end{bmatrix} = flowers. \tag{2}$$

The OF is used to calculate the fitness values of the flowers and of the plums. The $plum_{g_{best}}$ is set to the position of the plum that has the best fitness value.

Then, the PTA runs the following instructions I times.

At the beginning of each iteration, the positions of the following two plums are computed:

- $plum_{ripe}$ —the plum with the best fitness value
- $plum_{unripe}$ —the plum with the second-best fitness value

For each $flower_i^k$, where k is the iteration number and $i = \overline{1, N}$, a random number r from the range $[0, 1]$ is selected. Three cases, one for each phase, are considered further:

- Fruitiness Phase ($r \geq FT$):

In this case, the positions of the flowers are updated using the following formula:

$$flower_i^k = flower_i^{k-1} + rand(FR_{min}, FR_{max}) \times (plum_i^{k-1} - flower_i^{k-1}) \tag{3}$$

where $rand(FR_{min}, FR_{max})$ is a random number from $[FR_{min}, FR_{max}]$.

- Ripeness Phase ($FT > r \geq RT$):

The following formula is used to update the positions of the flowers in this case:

$$flower_i^k = flower_i^{k-1} + 2 \times r_1 \times (plum_{ripe} - flower_i^{k-1}) + 2 \times r_2 \times (plum_{unripe} - flower_i^{k-1}) \quad (4)$$

where r_1 and r_2 are random numbers from $[0, 1]$.

- Storeness Phase ($RT > r$):

The positions of the flowers are updated as follows:

$$flower_i^k = plum_i^{k-1} \times \left(1 + \mathcal{N}(0, \sigma^2)\right) \quad (5)$$

where $\mathcal{N}(0, \sigma^2)$ is a Gaussian distribution with mean 0 and standard deviation σ^2 , defined as follows:

$$\sigma^2 = \begin{cases} 1, & \text{if } OF(plum_i^{k-1}) < OF(plum_{ripe}) \\ e^{\frac{OF(plum_{ripe}) - OF(plum_i^{k-1})}{|OF(plum_i^{k-1})| + \epsilon}}, & \text{otherwise} \end{cases} \quad (6)$$

Then, the positions of the flowers are updated to be in $[X_{min}, X_{max}]$. For each $j = \overline{1, D}$:

- If $flower_{i,j}^k < X_{min}$ then $flower_{i,j}^k = X_{min}$;
- If $flower_{i,j}^k > X_{max}$ then $flower_{i,j}^k = X_{max}$.

After the positions of the flowers are updated, for each $plum_i^k$, where k is the iteration number and $i = \overline{1, N}$, the following formula is used:

$$plum_i^k = \begin{cases} flower_i^k, & \text{if } OF(flower_i^k) < OF(plum_i^{k-1}) \\ plum_i^{k-1}, & \text{otherwise} \end{cases} \quad (7)$$

At the end of each iteration, the position of $plum_{g_{best}}$ is updated to the position of the plum with the best fitness.

Finally, when all iterations are completed, the PTA returns the value of the $plum_{g_{best}}$.

3.1. Heating and Cooling Load Prediction

The Energy Efficiency Dataset was split randomly using a 5-fold cross-validation. For each split out of the five splits, the testing data were represented by one different fold, while the training data were represented by the remaining folds. The training data were standardized using the Z-score for the values of each column, while the testing data were standardized using the mean and the standard deviation values, which were computed for the training data.

The algorithms that were used were the Extra Trees Regressor (ETR), the Gradient Boosting Regressor (GBR), and the Random Forest Regressor (RFR). The regressors were configured with the hyperparameter values described by the plums. The eight metrics which were used to evaluate the results were the averages of the $RMSE$, R^2 , MAE , and $MAPE$ across the 5 folds for heating and cooling load prediction.

3.2. MOPTA Multi-Objective Fitness Function

Each position of a plum corresponds to an algorithm and its hyperparameter configuration. To apply the multi-objective fitness function to a plum, it is necessary to convert the values that describe the position of the plum to integers first. This is done using the floor function, which takes a real value as input and returns the greatest integer value that is less than or equal to it.

Figure 1 illustrates the high-level view of the multi-objective fitness function.

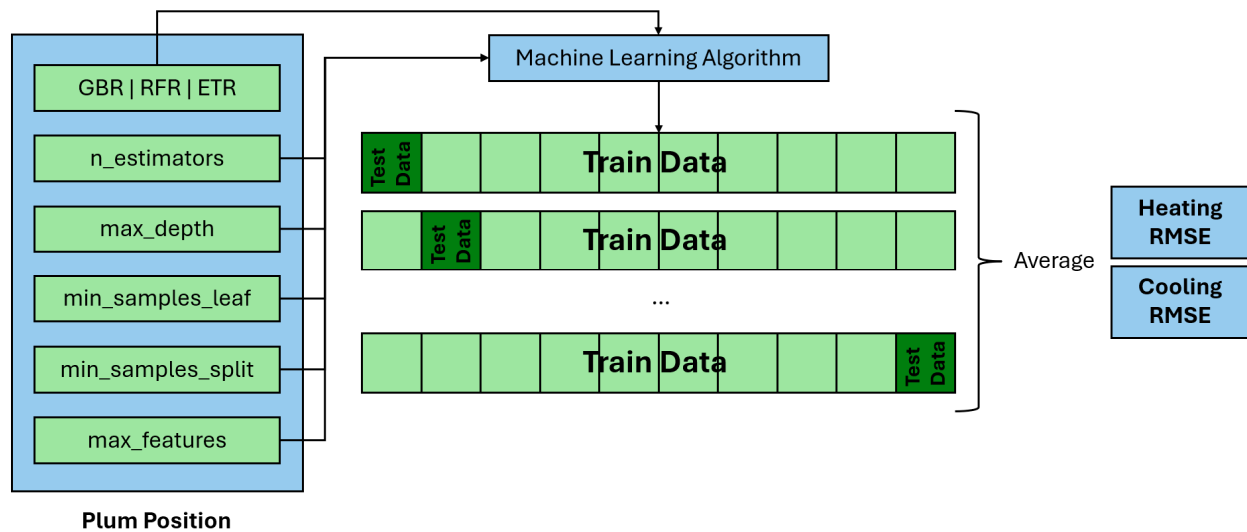


Figure 1. Multi-objective fitness function.

Compared to the approach presented in [39], we did not consider the bootstrap parameter, as we aimed to use a set of hyperparameters that can be configured for all algorithms. However, compared to that approach, we added a new dimension that describes the algorithm, namely GBR, RFR, or ETR, such that 0 corresponds to GBR, 1 to RFR, and 2 to ETR, respectively.

The inputs of the fitness function are the converted position of the plum and the train data. The first dimension describes the algorithm, while the other five dimensions describe the values of the hyperparameters.

We performed a 10-fold cross-validation on the train data, and we computed the average RMSE values. In each partition, the test data are represented by one fold, and the train data are represented by the other nine folds. The configured selected algorithm was applied twice in each partition, depending on the prediction type. The first time, it was used to predict the heating load, while the second time, it was used to predict the cooling load.

The output of the fitness function is represented by the following two values:

- The average Heating RMSE, denoted as $RMSE_H$;
- The average Cooling RMSE, denoted as $RMSE_C$.

The MOPTA aims to obtain minimal values for both values.

3.3. Multi-Objective Adaptations of PTA

The multi-objective adaptations of the PTA considered in this manuscript are similar to the ones we used in [25], and they are based on the method presented in [43]. The major adaptations introduced by the MOPTA are the application of an external archive for the saving and retrieval of the solutions that are pareto-optimal and the use of this archive for obtaining the values for the ripe and the unripe plums.

The dominance relations between $Plum_1 = (RMSE_{H_1}, RMSE_{C_1})$ and $Plum_2 = (RMSE_{H_2}, RMSE_{C_2})$ are defined as follows:

- (1) If $RMSE_{H_1} \leq RMSE_{H_2}$ and $RMSE_{C_1} \leq RMSE_{C_2}$, and at least one of the relations $RMSE_{H_1} < RMSE_{H_2}$ and $RMSE_{C_1} < RMSE_{C_2}$ is true, then $Plum_1$ dominates $Plum_2$;
- (2) If $RMSE_{H_1} \geq RMSE_{H_2}$ and $RMSE_{C_1} \geq RMSE_{C_2}$, and at least one of the relations $RMSE_{H_1} > RMSE_{H_2}$ and $RMSE_{C_1} > RMSE_{C_2}$ is true, then $Plum_2$ dominates $Plum_1$;
- (3) If neither (1) nor (2) is true, then $Plum_1$ and $Plum_2$ are non-dominated.

A set that contains two solutions is non-dominated if neither solution dominates the other one.

3.3.1. Plum Matrix Grid Computation

Figure 2 presents the methodology for the computation of the plum matrix grid.

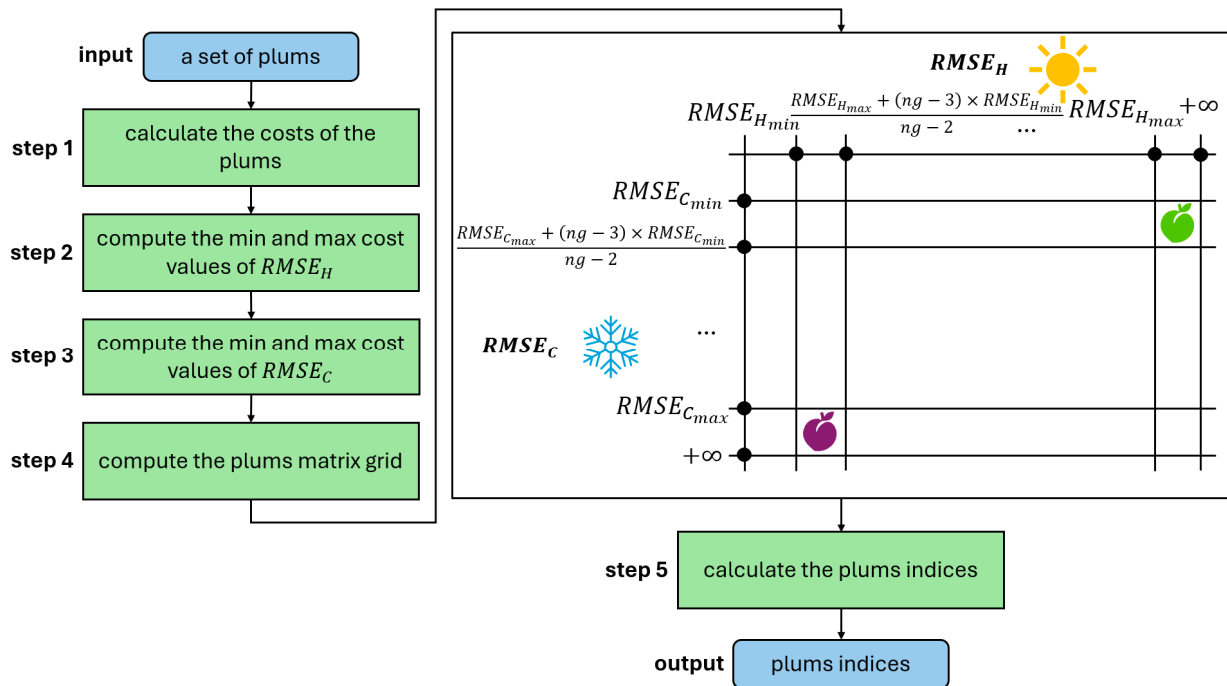


Figure 2. Plum matrix grid computation methodology.

The input is represented by a set of plums $\{plum_1, \dots, plum_{N_{plums}}\}$, where N_{plums} is the total number of plums.

step 1: The costs of the plums are computed using the multi-objective fitness function. The result is the following matrix:

$$Cost = \begin{bmatrix} RMSE_{H_1} & RMSE_{C_1} \\ \dots & \dots \\ RMSE_{H_{N_{plums}}} & RMSE_{C_{N_{plums}}} \end{bmatrix} \tag{8}$$

step 2: The minimum and the maximum cost values of $RMSE_H$ are computed using the $Cost$ matrix and the grid's inflation parameter (ϵ):

$$\begin{bmatrix} min_{RMSE_H} \\ max_{RMSE_H} \end{bmatrix} = \begin{bmatrix} min(RMSE_{H_1}, \dots, RMSE_{H_{N_{plums}}}) \\ max(RMSE_{H_1}, \dots, RMSE_{H_{N_{plums}}}) \end{bmatrix} \tag{9}$$

$$\Delta RMSE_H = \epsilon \times (max_{RMSE_H} - min_{RMSE_H}) \tag{10}$$

$$\begin{bmatrix} RMSE_{H_{min}} \\ RMSE_{H_{max}} \end{bmatrix} = \begin{bmatrix} min_{RMSE_H} - \Delta RMSE_H \\ max_{RMSE_H} + \Delta RMSE_H \end{bmatrix} \tag{11}$$

step 3: Similarly, the minimum and the maximum cost values of $RMSE_C$ are computed as follows:

$$\begin{bmatrix} min_{RMSE_C} \\ max_{RMSE_C} \end{bmatrix} = \begin{bmatrix} min(RMSE_{C_1}, \dots, RMSE_{C_{N_{plums}}}) \\ max(RMSE_{C_1}, \dots, RMSE_{C_{N_{plums}}}) \end{bmatrix} \tag{12}$$

$$\Delta RMSE_C = \epsilon \times (max_{RMSE_C} - min_{RMSE_C}) \tag{13}$$

$$\begin{bmatrix} RMSE_{C_{min}} \\ RMSE_{C_{max}} \end{bmatrix} = \begin{bmatrix} \min_{RMSE_C} - \Delta RMSE_C \\ \max_{RMSE_C} + \Delta RMSE_C \end{bmatrix} \quad (14)$$

step 4: The plum matrix *Grid* was defined using the number of grids *ng*, such that the *x*-axis presents the endpoints of the $RMSE_H$ minimization objective, and the *y*-axis presents the endpoints of the $RMSE_C$ minimization objective.

step 5: The formulas that are used for the computation of the index of $plum_i$, with the cost $(RMSE_{H_i}, RMSE_{C_i})$ such that $i = \overline{1, N_{plums}}$, are as follows:

$$index_H = \min(i | RMSE_{H_i} < Grid_{i,1}) \quad (15)$$

$$index_C = \min(i | RMSE_{C_i} < Grid_{i,1}) \quad (16)$$

$$index(plum_i) = index_H(plum_i) + ng \times (index_C(plum_i) - 1) \quad (17)$$

The output is represented by the plum indices set $\{index(plum_1), \dots, index(plum_{N_{plums}})\}$.

3.3.2. Plum Selection Methodology

The ripe and unripe plums were selected using the archive of plums $archive_{plums}$. The selection of these two plums was performed at each iteration of the MOPTA for each plum.

Figure 3 presents a high-level view of the methodology for the plum selection.

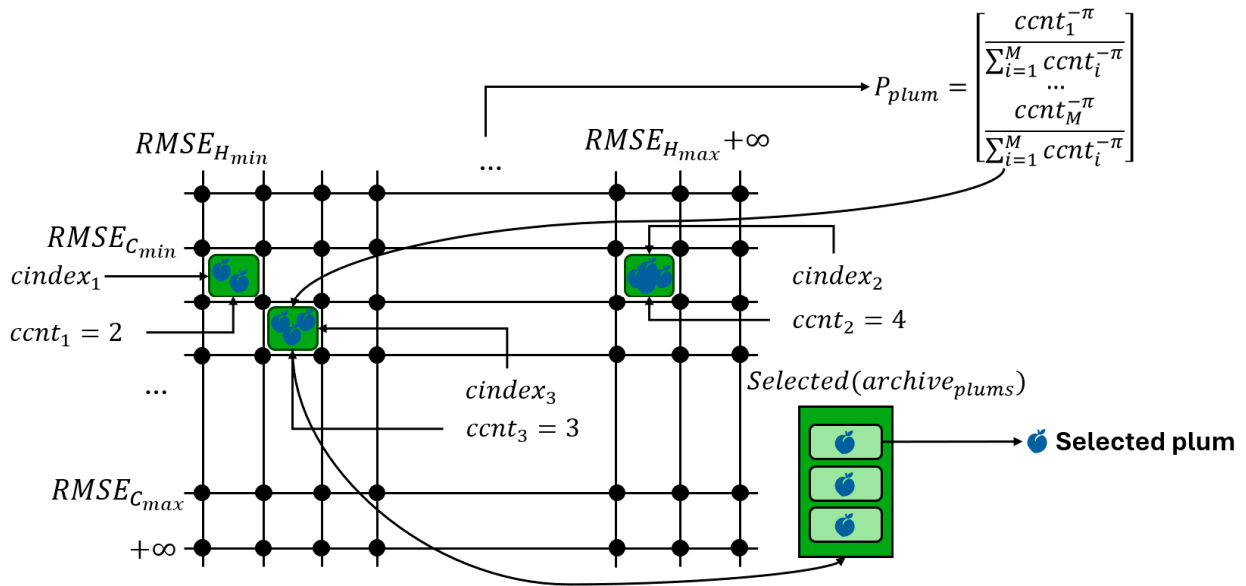


Figure 3. Plum selection methodology.

The set *OccIndex* of occupied indices for $archive_{plums}$ is calculated as follows:

$$OccIndex(archive_{plums}) = Set(index(plum_1), \dots, index(plum_{as})) \quad (18)$$

such that the function *Set* converts the numbers received as input into a list of unique numbers sorted in increasing order. Suppose that there are *M* cells that are occupied, and each one of them is defined by the cell index *cindex*. Then:

$$OccIndex(archive_{plums}) = \{cindex_1, \dots, cindex_M\} \quad (19)$$

The following vector that stores the cell count of each plum is defined:

$$OccCnt(archive_{plums}) = \{ccnt_1, \dots, ccnt_M\} \tag{20}$$

such that for each $i = \overline{1, M}$, the value $ccnt_i$ represents how many plums are present at the location $cindex_i$.

Then, a random number r is selected from $\{1, \dots, M\}$ using a roulette wheel selection mechanism defined by the following:

$$P_{plum} = \begin{bmatrix} \frac{ccnt_1^{-\pi}}{\sum_{i=1}^M ccnt_i^{-\pi}} \\ \dots \\ \frac{ccnt_M^{-\pi}}{\sum_{i=1}^M ccnt_i^{-\pi}} \end{bmatrix} \tag{21}$$

The set $Selected(archive_{plums})$ was defined by the formula:

$$Selected(archive_{plums}) = \{plum_i | index(plum_i) = cindex_r\} \tag{22}$$

such that $i = \overline{1, as}$, where as is the archive size, was used to select the ripe or the unripe plum, randomly considering the uniform probability.

3.3.3. Plum Removal Methodology

The size of $archive_{plums}$ was adjusted during each iteration of the algorithm if it was greater than the maximum archive size (mas). Figure 4 presents a high-level view of the methodology for the removal of the plums.

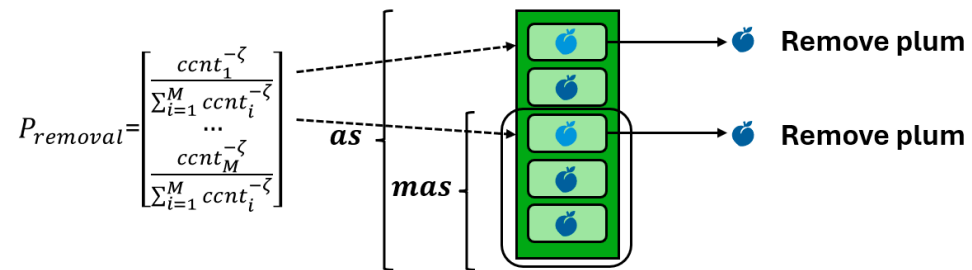


Figure 4. Plum removal methodology.

Therefore, a number of $(as - mas)$ plums were removed from the archive using the steps similar to the ones presented in the plum selection methodology, with the following two adaptations:

Adaptation 1: the plums removal probability, which was used instead of the P_{plum} probability, was defined using the formula:

$$P_{removal} = \begin{bmatrix} \frac{ccnt_1^{-\zeta}}{\sum_{i=1}^M ccnt_i^{-\zeta}} \\ \dots \\ \frac{ccnt_M^{-\zeta}}{\sum_{i=1}^M ccnt_i^{-\zeta}} \end{bmatrix} \tag{23}$$

such that the parameter ζ describes the plum selection pressure.

Adaptation 2: the removal of the selected plum from $archive_{plums}$, as the last step of the methodology.

3.3.4. MOPTA for Heating and Cooling Prediction

Algorithm 1 presents the MOPTA for heating and cooling prediction.

Algorithm 1 MOPTA for Heating and Cooling Prediction

```

1: Input  $I, D, N, FT, RT, FR_{min}, FR_{max}, \epsilon, X_{min}, X_{max}, OF, mas, ng, \epsilon, \pi, \zeta$ 
2: Output  $archive_{plums}$ 
3: initialize  $N$  flowers in the  $D$ -dimensional space with values from  $[X_{min}, X_{max}]$ ;
4: initialize  $N$  plums to the positions of the  $N$  flowers;
5: adapt the positions  $plum_i$  and  $flower_i$  ( $i = 1, \dots, N$ ) to arrays of integers;
6:  $(RMSE_H, RMSE_C) = OF(plum_i)$  ( $i = 1, \dots, N$ );
7:  $plums = GetDomination(plums)$ ;
8:  $archive_{plums} = GetArchive(plums)$ ;
9:  $grids = GetGrids(archive_{plums}, ng, \epsilon)$ ;
10: for  $iter = 1$  to  $I$  do
11:   for  $i = 1$  to  $N$  do
12:      $as = SizeOf(archive_{plums})$ ;
13:     determine  $plum_{ripe}$  and  $plum_{unripe}$  from  $archive_{plums}$  using  $as, \pi, grids$ ;
14:     update  $r$  to a random number from  $[0, 1]$ ;
15:     update  $flower_i$  according to  $FT, RT, FR_{min}, FR_{max}, r$ ;
16:     adjust the flowers to be in the range  $[X_{min}, X_{max}]$ ;
17:   end for
18:   for  $i = 1$  to  $N$  do
19:     update  $plum_i$  and  $flower_i$  to arrays of integers;
20:     use  $OF$  to get the fitness values  $Plum_i$  and  $Flower_i$  of  $plum_i$  and  $flower_i$ ;
21:     if  $dominates(Flower_i, Plum_i)$  then
22:        $plum_i = flower_i$ ;
23:        $Plum_i = Flower_i$ ;
24:     end if
25:   end for
26:    $plums = GetDomination(plums)$ ;
27:    $nplums = NonDominated(plums)$ ;
28:    $nplums = archive_{plums} \cup nplums$ ;
29:    $nplums = GetDomination(nplums)$ ;
30:    $archive_{plums} = GetArchive(nplums)$ ;
31:    $grids = GetGrids(archive_{plums}, ng, \epsilon)$ ;
32:    $as = SizeOf(archive_{plums})$ ;
33:    $ResizeArchive(as > mas, as, mas, archive_{plums}, \zeta)$ ;
34: end for
35: return  $archive_{plums}$ ;

```

The input parameters of the MOPTA consist of the standard input parameters of the PTA, which are presented in Table 1, and the following additional parameters:

- mas —the maximum archive size of the repository that contains the nondominated solutions;
- ng —the number of grids per objective;
- ϵ —the grid's inflation parameter;
- π —the pressure parameter used during the plum selection;
- ζ —the plum selection pressure parameter used during the plum removal.

The OF is a multi-objective function that returns the average values of the $RMSE_H$ and $RMSE_C$ of the ML algorithm trained and validated according to the position of the plum. The range $[X_{min}, X_{max}]$ was adapted such that the first dimension describes the selected regressor, while the other dimensions describe the limits of the hyperparameters considered in the training of the ML algorithm.

The output of the MOPTA is $archive_{plums}$, which consists of the non-dominant plums after I iterations.

The N flowers were initialized with random values from $[X_{min}, X_{max}]$ in the D -dimensional search space (line 3), while the N plums were initialized to the positions of the N flowers in line 4. Then, both the positions of the plums and of the flowers were adapted to arrays of integers in line 5 using the floor function. The multi-objective fitness function OF presented in Section 3.2 was applied to each $plum_i$ ($i = 1, \dots, N$) (line 6).

Using the conditions presented at the beginning of Section 3.3, the dominance relation was determined in line 7 of the algorithm. The $archive_{plums}$ of non-dominated plums was created in line 8. Then, using the steps presented in Section 3.3.1, the grid matrix and the indices of the plums were computed.

The instructions from lines 11–33 were repeated for I iterations. For each $flower_i$ ($i = 1, \dots, N$), the instructions from lines 12–16 were performed. The current size of the archive as was updated in line 12 to the total number of plums from $archive_{plums}$. The values of $plum_{ripe}$ and $plum_{unripe}$ were computed in line 13 using the values of as , π , and $grids$ and the plum selection methodology presented in Section 3.3.2.

Initially, the ripe and the unripe plums were selected randomly from $archive_{plums}$. If $as > 1$, then $plum_{ripe}$ was selected from $archive_{plums} - \{plum_{unripe}\}$ following the steps from Section 3.3.2.

The positions $flower_i$ ($i = 1, \dots, N$) were updated in line 15 using Equations (3)–(6) for the three phases: fruitiness phase, ripeness phase, and storeness phase. The equations for the storeness phase were adapted for the multi-objective optimization using a procedure adapted after the one from [44]. Equation (6) was adapted to the equation:

$$\sigma^2 = \begin{cases} 1, & \text{if } OF(plum_i^{k-1}) < OF(plum_{ripe}) \\ \frac{F(plum_{ripe}) - F(plum_i^{k-1})}{e^{|F(plum_i^{k-1})| + \epsilon}}, & \text{otherwise} \end{cases} \quad (24)$$

such that the function F was defined as follows:

$$F(plum) = \frac{RMSE_H + RMSE_C}{2} \quad (25)$$

where $(RMSE_H, RMSE_C) = OF(plum)$.

Then, the positions of the flowers were updated to be in $[X_{min}, X_{max}]$ (line 16).

The instructions from (lines 19–24) were performed for each $plum_i$ ($i = 1, \dots, N$). First, the $plum_i$ and the corresponding $flower_i$ were updated to arrays of integers using the floor function (line 19). Then, the OF was used to compute the fitness values $Plum_i$ and $Flower_i$ of $plum_i$ and $flower_i$, respectively (line 20). If $Flower_i$ dominated $Plum_i$, then the position $plum_i$ and the fitness value $Plum_i$ were updated to $flower_i$ and $Flower_i$, respectively.

The plum dominance relation was determined again in line 26. The non-dominated plums $nplums$ were computed in line 27. Then, in line 28, the plums from $archive_{plums}$ were appended to $nplums$ (line 29) and the $nplums$ were updated in line 30.

The matrix $grids$ was computed in line 31, while the value as was computed in line 32. If the value as was greater than mas , then $(as - mas)$ plums were removed from the archive, according to the methodology presented in Section 3.3.3.

Finally, the MOPTA returned the $archive_{plums}$ as output in line 35.

3.3.5. Solution Ranking Using MOORA

The solutions which were returned by the MOPTA were ranked using an adaptation of MOORA [22,45].

The matrix D was defined as follows:

$$D = \begin{bmatrix} RMSE_{H_1} & RMSE_{C_1} \\ \dots & \dots \\ RMSE_{H_m} & RMSE_{C_m} \end{bmatrix} \quad (26)$$

where m is the size of the plums archive and $(RMSE_{H_i}, RMSE_{C_i})$ represent the $RMSE$ values predicted by the model trained according to the position of the i -th plum, where $i = 1, \dots, m$.

The values of D were normalized as follows:

$$\begin{bmatrix} RMSE_{H_j}^* \\ RMSE_{C_j}^* \end{bmatrix} = \begin{bmatrix} \frac{RMSE_{H_j}}{\sqrt{\sum_{i=1}^m RMSE_{H_i}^2}} \\ \frac{RMSE_{C_j}}{\sqrt{\sum_{i=1}^m RMSE_{C_i}^2}} \end{bmatrix} \quad (j = 1, \dots, m) \quad (27)$$

The MOORA scores of the plums from the archive of size m were finally computed as follows:

$$S_{MOORA} = \begin{bmatrix} RMSE_{H_1}^* + RMSE_{C_1}^* \\ \dots \\ RMSE_{H_m}^* + RMSE_{C_m}^* \end{bmatrix} \quad (28)$$

The most dominant plum was the one with the lowest MOORA score.

3.3.6. MOPTA Methodology for Heating and Cooling Prediction

Figure 5 presents the high-level view of the MOPTA methodology which was used for the prediction of the heating and cooling loads.

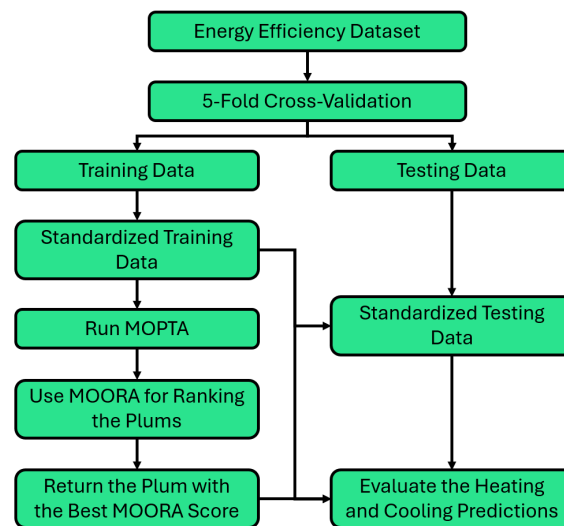


Figure 5. MOPTA methodology for heating and cooling load prediction.

The input of the methodology was represented by the Energy Efficiency Dataset. The data were split into Training Data and Testing Data considering a 5-Fold Cross-Validation approach, such that five splits were performed. Each time, one fold was used for testing and the remaining ones for training. Then, the Standardized Training Data and the Standardized Testing Data were obtained. The MOPTA was run using the Standardized Training Data as input, with a 10-fold cross-validation to evaluate the plums. The archive returned by the algorithm was evaluated using MOORA, and the plum with the best MOORA score was further considered to evaluate the predictions. The predictions were evaluated using the $MAPE$, $RMSE$, R^2 , and MAE metrics.

4. Results

The experiments were performed in Python version 3.12.3 using the sklearn library on a machine with the following properties:

- Processor: Intel Core i9-13900 K 3.00 GHz;
- RAM: 64 GB;
- Operating System: Windows 11 Pro;
- SDD: 1 TB.

All the computations were CPU-based.

4.1. Energy Efficiency Dataset

The Energy Efficiency Dataset used in the experiments was characterized by 768 samples, eight attributes, and two responses. The dataset was obtained considering 12 building shapes, simulated in Ecotect. Table 2 presents the summary of the features.

Table 2. Energy Efficiency Dataset features summary.

Feature	Variable	Minimum Value	Maximum Value	Type
X1	Relative Compactness	0.62	0.98	attribute
X2	Surface Area	514.5	808.5	
X3	Wall Area	245	416.5	
X4	Roof Area	110.25	220.5	
X5	Overall Height	3.5	7	
X6	Orientation	2	5	
X7	Glazing Area	0	0.4	
X8	Glazing Area Distribution	0	5	
Y1	Heating Load	6.01	43.1	response
Y2	Cooling Load	10.9	48.3	

The dataset was split randomly into five folds of an approximately equal size, such that the Testing Data were represented by one of the folds while the Training Data were represented by the other four folds.

4.2. Hyperparameters Configuration

Table 3 presents the ranges of the hyperparameters used in the experiments. The values were inspired by the ones used by the authors of [39].

Table 3. Hyperparameter ranges.

Hyperparameter	Minimum Value	Maximum Value
n_estimators	200	600
max_depth	10	100
min_samples_leaf	1	10
min_samples_split	2	10
max_features	1	8

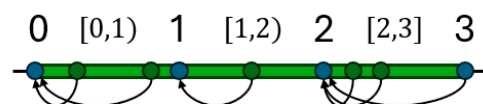
4.3. MOPTA Configuration Parameters

Table 4 presents the MOPTA configuration parameters used in our experiments.

As a remark, in the case of X_{max} , the table also adds the value 1 to the upper limit since the search space is represented by continuous values. However, if the upper limit is obtained, then the value 1 is subtracted from that value. Figure 6 presents this adjusting transformation more clearly for the first dimension.

Table 4. MOPTA configuration parameters summary.

Configuration Parameter	Value
I (iterations)	50
D (number of dimensions)	6
N (number of plums)	30
FT (fruitiness threshold)	0.8
RT (ripeness threshold)	0.2
FR_{min} (minimum fruitiness rate)	0.5
FR_{max} (maximum fruitiness rate)	1
ϵ (avoiding division by 0 constant)	10^{-300}
X_{min} (minimum position values)	[0, 200, 10, 1, 2, 1]
X_{max} (maximum position values)	[3, 601, 101, 11, 11, 9]
OF (objective function)	the multi-objective function presented in Section 3.2
mas (maximum archive size)	30
ng (number of grids)	10
ϵ (grid's inflation parameter)	0.1
π (pressure parameter)	4
ζ (plum selection pressure parameter)	2

**Figure 6.** Position adjusting representative example.

As can be seen in the figure, the values from $[0, 1)$ are adjusted to 0, the values from $[1, 2)$ are adjusted to 1, and the values from $[2, 3]$ are adjusted to 2, respectively.

4.4. MOPTA Prediction Results

Table 5 presents the results obtained by the MOPTA for each of the five folds and the mean results.

Table 5. MOPTA prediction results summary.

	Metric	RMSE	R2	MAPE	MAE
Fold 1	Heating	0.043057	0.99797	0.012089	0.03
	Cooling	0.091637	0.991753	0.047623	0.06
Fold 2	Heating	0.032117	0.999037	0.006547	0.02
	Cooling	0.071462	0.994942	0.058136	0.05
Fold 3	Heating	0.038437	0.998473	0.029722	0.03
	Cooling	0.075794	0.993825	0.023327	0.05
Fold 4	Heating	0.029834	0.999056	-0.001601	0.02
	Cooling	0.058583	0.99617	0.044078	0.04
Fold 5	Heating	0.035152	0.998872	0.007234	0.03
	Cooling	0.083508	0.99387	0.165621	0.05
Mean	Heating	0.035719	0.998682	0.010798	0.026
	Cooling	0.076197	0.994112	0.067757	0.05

As can be seen in the table, in the case of the MAPE metric, the result was negative for Fold 4. The negative value is justified by the fact that after the standardization operation, the labels had both positive and negative values. In all five cases, the selected algorithm was the GBR.

4.5. Comparison to the Prediction Results Obtained Using the Default Parameters

Table 6 compares the results obtained by the MOPTA approach to the ones obtained by each of the algorithms GBR, RFR, and ETR when the default values were used. We considered these three algorithms in the comparison because they are used by the MOPTA as part of the optimization process. Moreover, the best solution returned by the MOPTA describes which of the three algorithms is applied and the optimal values of the hyperparameters. Basically, in this table, each MOPTA result corresponds to one of the three algorithms, depending on the value of the first dimension of the best plum, tuned according to the values of the remaining dimensions of the best plum. To get reproducible results, each of the algorithms was initialized with a random_state equal to 42.

Table 6. Comparison of the MOPTA results to the ones obtained by each algorithm when the default parameters were used.

		Metric	RMSE	R2	MAPE	MAE
Fold 1	GBR	Heating	0.058349	0.996273	0.044175	0.04
		Cooling	0.183215	0.967036	0.119231	0.12
	RFR	Heating	0.059156	0.996169	0.046786	0.04
		Cooling	0.18708	0.965631	0.124183	0.12
ETR	Heating	0.054964	0.996693	0.036841	0.04	
	Cooling	0.197788	0.961584	0.14311	0.12	
MOPTA	Heating	0.043057	0.99797	0.012089	0.03	
	Cooling	0.091637	0.991753	0.047623	0.06	
Fold 2	GBR	Heating	0.042297	0.99833	0.018927	0.03
		Cooling	0.161946	0.974028	0.27774	0.11
	RFR	Heating	0.045063	0.998104	0.020777	0.03
		Cooling	0.172825	0.970421	0.269599	0.11
ETR	Heating	0.043534	0.998231	0.018459	0.03	
	Cooling	0.185254	0.966014	0.208484	0.11	
MOPTA	Heating	0.032117	0.999037	0.006547	0.02	
	Cooling	0.071462	0.994942	0.058136	0.05	
Fold 3	GBR	Heating	0.052812	0.997118	0.244372	0.04
		Cooling	0.128911	0.982138	0.053806	0.09
	RFR	Heating	0.052765	0.997123	0.151096	0.04
		Cooling	0.156191	0.973779	0.07873	0.1
ETR	Heating	0.0502	0.997396	0.11682	0.03	
	Cooling	0.172112	0.968161	0.069121	0.09	
MOPTA	Heating	0.038437	0.998473	0.029722	0.03	
	Cooling	0.075794	0.993825	0.023327	0.05	
Fold 4	GBR	Heating	0.041434	0.99818	0.006423	0.03
		Cooling	0.138274	0.978668	0.275395	0.1
	RFR	Heating	0.042305	0.998103	0.003031	0.03
		Cooling	0.152909	0.973914	0.37535	0.09
ETR	Heating	0.039679	0.998331	0.013102	0.03	
	Cooling	0.163703	0.970101	0.426203	0.09	
MOPTA	Heating	0.029834	0.999056	−0.001601	0.02	
	Cooling	0.058583	0.99617	0.044078	0.04	
Fold 5	GBR	Heating	0.039741	0.998559	0.011487	0.03
		Cooling	0.167864	0.975231	0.469739	0.11
	RFR	Heating	0.042071	0.998385	0.015281	0.03
		Cooling	0.1788	0.971899	0.501039	0.11
ETR	Heating	0.04125	0.998447	0.018558	0.03	
	Cooling	0.197247	0.965802	0.511074	0.12	
MOPTA	Heating	0.035152	0.998872	0.007234	0.03	
	Cooling	0.083508	0.99387	0.165621	0.05	
Mean	GBR	Heating	0.046927	0.997692	0.065077	0.034
		Cooling	0.156042	0.97542	0.239182	0.105999
	RFR	Heating	0.048272	0.997577	0.047394	0.034
		Cooling	0.169561	0.971129	0.26978	0.105999
ETR	Heating	0.045925	0.997819	0.040756	0.032	
	Cooling	0.183221	0.966332	0.271598	0.105999	
MOPTA	Heating	0.035719	0.998682	0.010798	0.026	
	Cooling	0.076197	0.994112	0.067757	0.05	

The obtained results show that the MOPTA RMSE results were better than the ones returned by the GBR, the RFR, and the ETR, both for the heating predictions and the cooling predictions in all cases.

4.6. Comparison to Other Multi-Objective Optimization Approaches

The results obtained using the MOPTA were compared to the ones obtained by the MOGWO, the MOPSO, and the NSGA-II. We considered these algorithm methods in the comparison because a part of the mathematical equations of the Plum Tree Algorithm were inspired by the Grey Wolf Optimizer and the Particle Swarm Optimization, while the Genetic Algorithms, which are at the base of the NSGA-II, are one of the most popular evolutionary algorithms. Moreover, the Particle Swarm Optimization and the Grey Wolf Optimizer are some of the most popular swarm intelligence algorithms. Therefore, we considered the multi-objective implementations of these three benchmark algorithms to validate our results. In the case of the MOGWO algorithm, we used the implementation from our previous work [25], as it was used for this type of problem. The MOPSO and NSGA-II were also used in [25] to validate our results. For NSGA-II, we considered the implementation from the DEAP (Distributed Evolutionary Algorithms in Python) framework [46].

Some of the configuration parameters have the same values as in the case of the MOPTA, while other parameters were specific to each algorithm. The common configuration parameters for all four algorithms were the number of iterations I , the number of dimensions D , the population size N , the minimum and the maximum position values X_{min} and X_{max} , and the objective function OF .

The configuration parameters that were common to the MOPTA, the MOGWO, and the MOPSO were the maximum archive size mas , the number of grids ng , the grid's inflation parameter ϵ , the pressure parameter π , and the selection pressure parameter ζ .

Table 7 presents the specific configuration parameter values for each algorithm.

Table 7. Specific configuration parameters values.

Algorithm	Configuration Parameter	Value
MOGWO	a (exploration parameter)	2
MOPSO	V_{min} (minimum velocity value)	−6
	V_{max} (maximum velocity value)	6
	w_{min} (minimum inertia weight)	0.2
	w_{max} (maximum inertia weight)	0.9
	c_1 (cognitive component)	2
	c_2 (social component)	2
NSGA-II	CP (crossover probability)	0.8

Table 8 compares the MOPTA results to the ones obtained by the other three multi-objective optimization algorithms.

The MOGWO returned the best RMSE for cooling for Fold 1 and Fold 2 and the best RMSE for heating for Fold 2 and Fold 3. The MOPSO returned the best RMSE for heating for Fold 1 and the best RMSE for cooling for Fold 4. The NSGA-II returned the best RMSE for cooling for Fold 4. The MOPTA returned the best RMSE for heating for all folds except for Fold 1, and the best RMSE for cooling for Fold 1, Fold 2, and Fold 5. Also, the MOPTA obtained the best mean RMSE values both for the heating predictions and for the cooling predictions. Another remark is that all of the multi-objective algorithms selected the GBR, even though, as can be seen in Table 6, the GBR does not always return the best RMSE results when compared to RFR and ETR when the default parameter values are used.

Table 9 describes how many times each algorithm was the best with respect to the five folds.

Table 8. Comparison of the MOPTA results to the ones obtained by the MOGWO, the MOPTA, and the NSGA-II.

		Metric	RMSE	R2	MAPE	MAE
Fold 1	MOGWO	Heating	0.043057	0.99797	0.012089	0.03
		Cooling	0.091637	0.991753	0.047623	0.06
	MOPSO	Heating	0.042969	0.997979	0.012419	0.03
		Cooling	0.092244	0.991644	0.044089	0.06
	NSGA-II	Heating	0.043012	0.997975	0.010445	0.03
		Cooling	0.091747	0.991734	0.047607	0.06
	MOPTA	Heating	0.043057	0.99797	0.012089	0.03
		Cooling	0.091637	0.991753	0.047623	0.06
Fold 2	MOGWO	Heating	0.032117	0.999037	0.006547	0.02
		Cooling	0.071462	0.994942	0.058136	0.05
	MOPSO	Heating	0.034733	0.998874	0.00745	0.02
		Cooling	0.074935	0.994439	0.078264	0.05
	NSGA-II	Heating	0.03464	0.99888	0.007623	0.02
		Cooling	0.075657	0.994331	0.078111	0.05
	MOPTA	Heating	0.032117	0.999037	0.006547	0.02
		Cooling	0.071462	0.994942	0.058136	0.05
Fold 3	MOGWO	Heating	0.038437	0.998473	0.029722	0.03
		Cooling	0.075794	0.993825	0.023327	0.05
	MOPSO	Heating	0.038646	0.998456	0.052916	0.03
		Cooling	0.076207	0.993758	0.020738	0.05
	NSGA-II	Heating	0.041798	0.998194	0.050196	0.03
		Cooling	0.072592	0.994336	0.016488	0.05
	MOPTA	Heating	0.038437	0.998473	0.029722	0.03
		Cooling	0.075794	0.993825	0.023327	0.05
Fold 4	MOGWO	Heating	0.03216	0.998903	0.001763	0.02
		Cooling	0.060117	0.995967	0.055273	0.04
	MOPSO	Heating	0.032466	0.998882	−0.000460	0.02
		Cooling	0.056647	0.996419	0.049985	0.04
	NSGA-II	Heating	0.031941	0.998918	−0.001230	0.02
		Cooling	0.057837	0.996267	0.05597	0.04
	MOPTA	Heating	0.029834	0.999056	−0.001601	0.02
		Cooling	0.058583	0.99617	0.044078	0.04
Fold 5	MOGWO	Heating	0.035853	0.998827	−0.002201	0.02
		Cooling	0.114958	0.988383	0.12386	0.06
	MOPSO	Heating	0.035628	0.998841	0.007137	0.03
		Cooling	0.084536	0.993718	0.171809	0.05
	NSGA-II	Heating	0.036782	0.998765	0.008075	0.03
		Cooling	0.089095	0.993022	0.250595	0.06
	MOPTA	Heating	0.035152	0.998872	0.007234	0.03
		Cooling	0.083508	0.99387	0.165621	0.05
Mean	MOGWO	Heating	0.036325	0.998642	0.009584	0.024
		Cooling	0.082794	0.992974	0.061644	0.052
	MOPSO	Heating	0.036888	0.998606	0.015892	0.026
		Cooling	0.076914	0.993995	0.072977	0.05
	NSGA-II	Heating	0.037635	0.998546	0.015022	0.026
		Cooling	0.077385	0.993938	0.089754	0.052
	MOPTA	Heating	0.035719	0.998682	0.010798	0.026
		Cooling	0.076197	0.994112	0.067757	0.05

Table 9. Summary of the algorithm ranking.

Algorithm	Heating	Cooling	Summary
MOGWO	2	2	4
MOPSO	1	1	2
NSGA-II	0	1	1
MOPTA	4	3	7

We can see that in seven cases, the MOPTA was the best. The second algorithm, the MOGWO, was the best in only four cases.

4.7. Computational Load Analysis

This section presents a computational load analysis of the algorithms used in our experiments from the perspective of the running time. Table 10 summarizes the total running time expressed in milliseconds for each algorithm across all five folds.

Table 10. Running time comparison.

Algorithm	Running Time (ms)
MOPTA	70,475,987
MOGWO	34,962,471
MOPSO	31,266,781
NSGA-II	5,457,951
GBR	297
RFR	797
ETR	657

As can be seen in the table, the GBR, the RFR, and the ETR had the best running time. The running time of the MOPTA, which was approximately 19.5 h, was almost double compared to the running time of the MOGWO and MOPSO algorithms. The running time of the NSGA-II was almost 12 times better than the one of the MOPTA.

However, we also want to point out that a grid search which searches through all combinations of hyperparameters, namely $3 \times 401 \times 91 \times 10 \times 9 \times 8 = 78,820,560$, and which would need an average of around 200 ms per experiment, a value which is slightly less than that of the GBR algorithm when it is tuned with the default parameters (e.g., 297 ms), would need around 15,764,112,000 ms to complete, or around 182 days. With respect to these remarks, we can conclude that the MOPTA has a much better running time compared to the running time of the standard grid search.

4.8. Robustness and Convergence Analysis

In this section, we discuss the robustness and the convergence of the MOPTA, and we compare the results to the ones corresponding to the MOGWO, the MOPSO, and the NSGA-II.

To compute the robustness values, we consider the heating and cooling RMSE results obtained for each fold and calculate the standard deviation. Table 11 summarizes the comparison of the standard deviation (std) values obtained for each multi-objective optimization algorithm.

Table 11. Standard deviation results comparison summary.

Algorithm	Heating (std)	Cooling (std)
MOPTA	0.005224	0.012496
MOGWO	0.004611	0.021239
MOPSO	0.004057	0.013306
NSGA-II	0.004700	0.013706

If we are to consider that a lower variability means a better robustness value, then the MOPSO returned the best result for the heating standard deviation, while the MOPTA returned the best result for the cooling standard deviation.

For the analysis of the convergence of the MOPTA, we identified the first iteration, which returned the best value for each fold. We performed similar calculations for the other multi-objective algorithms. Table 12 summarizes the convergence analysis results.

Table 12. Convergence analysis results summary.

Algorithm	Fold	Best Position	Iteration
MOPTA	Fold 1	[0, 600, 10, 10, 6, 8]	3
	Fold 2	[0, 600, 91, 10, 9, 4]	2
	Fold 3	[0, 600, 10, 10, 5, 6]	5
	Fold 4	[0, 600, 10, 7, 8, 5]	1
	Fold 5	[0, 600, 11, 10, 2, 8]	20
MOGWO	Fold 1	[0, 600, 10, 10, 7, 8]	27
	Fold 2	[0, 600, 68, 10, 2, 4]	11
	Fold 3	[0, 600, 10, 10, 4, 6]	17
	Fold 4	[0, 600, 62, 10, 9, 6]	47
	Fold 5	[0, 600, 35, 5, 8, 8]	22
MOPSO	Fold 1	[0, 567, 42, 10, 10, 8]	33
	Fold 2	[0, 589, 73, 6, 2, 5]	43
	Fold 3	[0, 576, 35, 10, 2, 6]	23
	Fold 4	[0, 584, 67, 9, 2, 6]	43
	Fold 5	[0, 548, 48, 10, 4, 8]	2
NSGA-II	Fold 1	[0, 599, 56, 11, 3, 7]	29
	Fold 2	[0, 537, 17, 6, 3, 5]	32
	Fold 3	[0, 587, 22, 9, 7, 5]	23
	Fold 4	[0, 589, 86, 9, 2, 8]	46
	Fold 5	[0, 567, 13, 11, 8, 4]	36

The MOPTA converged relatively fast compared to the other algorithms except for Fold 5, where it obtained the best result in Iteration 20. In the case of the other algorithms, the best results were obtained after more iterations, except for the PSO and Fold 5, when the best result was obtained in Iteration 2.

5. Discussions

This section compares our results to the ones obtained by recent studies in the literature. Table 13 presents a summary of the result comparison building upon the comparison results presented in [21]. The articles presented in the table were selected so that a 5-fold cross-validation was used, and the models performed two predictions, one for the heating load and the other one for the cooling load.

Table 13. Comparison to the literature results.

Year	Approach	Method	Heating (RMSE)	Cooling (RMSE)
2024	this	A method based on Multi-Objective Plum Tree Algorithm which optimized both the heating and the cooling root mean square error.	0.035719	0.076197
2024	Guo et al. [47]	A method based on consolidated artificial neural network paradigms and metaheuristic algorithms.	2.4417	2.8031

Table 13. Cont.

Year	Approach	Method	Heating (RMSE)	Cooling (RMSE)
2023	Abdelkader et al. [48]	A hybrid parameterized deep learning model based on the Bayesian Optimization.	1.69	2.79
2023	Moldovan [21]	A method based on an ensemble of four regressors which was optimized using the Plum Tree Algorithm.	0.043124	0.159903
2023	Salami et al. [34]	An XGBoost hybrid machine learning methodology for the prediction of the energy loads.	0.3797	0.7578
2022	Ghasemkani et al. [31]	A Tri-Layer Neural Network and Maximum Relevance Minimum Redundancy-based method.	0.45689	0.81391
2019	Prasetiyo et al. [49]	A Naïve Bayes Classifier-based method.	0.1491	0.1412
2019	Bui et al. [50]	A method that uses the Imperialist Competition Algorithm to optimize the weights and the biases of an Artificial Neural Network.	2.7819	2.7795

The results presented in [47,48] are better than the ones presented in [50], but they are not directly comparable because no standardization was performed. For articles [31,49,50], we presented only the best values. In the case of article [34], which used a 5-fold cross-validation like our approach, we presented only the best results. Even though the results from [34] are not directly comparable, as we also used standardization, they were better than the ones from [31]. However, the root mean square error results of the current approach based on the Multi-Objective Plum Tree Algorithm were significantly better than the results obtained by the Plum Tree Algorithm-based ensemble.

Even though they are not directly comparable to our results because of different pre-processing configurations or cross-validation settings, the recent studies based on the latest deep learning methods returned promising results. For example, the approach presented in [51] based on deep neural networks returned a root mean square error equal to 0.0137. However, compared to our approach, the problem was converted into an image processing problem by transforming the data into image datasets. Rounding the heating and cooling load values to the closest integer, the issue was converted into a multi-class classification issue.

6. Conclusions

The manuscript presented a novel approach based on the Multi-Objective Plum Tree Algorithm for the prediction of the heating and cooling loads. The dataset that was used for the testing and validation of the approach was the Energy Efficiency Dataset from the UCI Machine Learning Repository. The solutions were ranked using the MOORA method. The results are better than the ones returned by the individual predictors Gradient Boosting Regressor, Random Forest Regressor, and Extra Trees Regressor, respectively. The results were also comparable to the ones returned by other multi-objective optimization approaches such as the Multi-Objective Grey Wolf Optimizer, the Multi-Objective Particle Swarm Optimization, and the NSGA-II, respectively. Compared to the results obtained in one of the previous research studies, where the Plum Tree Algorithm was used to tune an ensemble of predictors, the results were better. Also, the Multi-Objective Plum Tree Algorithm results were compared to the ones from the literature. The following directions are proposed for future research work:

- The improvement of the performance of the proposed algorithm through hybridization or the use of concepts such as Levy flights;

- The comparison of the results obtained by the Multi-Objective Plum Tree Algorithm to the ones obtained using other multi-objective optimization approaches that were not considered in the current paper;
- The application of the Multi-Objective Plum Tree Algorithm to more engineering problems from the field of energy efficiency prediction;
- The adaptation of the prediction methodology presented in the manuscript to a larger class of energy-consuming buildings, the ones which belong to the heavy industry, considering more data about the characteristics of the buildings, which complement characteristics such as the surface area, the overall height, and the orientation.

Author Contributions: Conceptualization, A.S. and D.M.; methodology, A.S. and D.M.; software, A.S.; validation, A.S.; formal analysis, D.M.; investigation, A.S. and D.M.; resources, A.S. and D.M.; data curation, D.M.; writing—original draft preparation, D.M.; writing—review and editing, A.S.; supervision, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset which supports this article is from the UCI machine learning repository, which can be found at "<https://archive.ics.uci.edu/ml/index.php> (accessed on 24 March 2024)".

Acknowledgments: The authors want to express appreciation for the editors and the anonymous reviewers for their beneficial suggestions, which improved the quality of this manuscript significantly.

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

References

1. Guo, J.; Yun, S.; Meng, Y.; He, N.; Ye, D.; Zhao, Z.; Jia, L.; Yang, L. Prediction of heating and cooling loads based on light gradient boosting machine algorithms. *Build. Environ.* **2023**, *236*, 110252. [[CrossRef](#)]
2. Lu, C.; Li, S.; Penaka, S.R.; Olofsson, T. Automated machine learning-based framework of heating and cooling load prediction for quick residential building design. *Energy* **2023**, *274*, 127334. [[CrossRef](#)]
3. Kucuktopcu, E. Comparative Analysis of Data-Driven Techniques to Predict Heating and Cooling Energy Requirements of Poultry Buildings. *Buildings* **2023**, *13*, 142. [[CrossRef](#)]
4. Acikklap, E.; Kandemir, S.Y. A method for determining optimum insulation thickness: Combined economic and environmental method. *Therm. Sci. Eng. Prog.* **2019**, *11*, 249–253. [[CrossRef](#)]
5. Kucuktopcu, E.; Cemek, B. A study on environmental impact of insulation thickness of poultry building walls. *Energy* **2018**, *150*, 583–590. [[CrossRef](#)]
6. Zheng, S.; Xu, H.; Mukhtar, A.; Yasir, A.S.H.M.; Khalilpoor, N. Estimating residential buildings' energy usage utilising a combination of Teaching-Learning-Based Optimization (TLBO) method with conventional prediction techniques. *Eng. Appl. Comp. Fluid.* **2023**, *17*, 2276347. [[CrossRef](#)]
7. Baheri, A.; Najafi, M.; Azimi, A.; Aghanajafi, C. A simplified model to predict and optimize energy consumption of residential buildings in the cold climate regions of iran. *Energy Source Part A* **2020**, 1–19. [[CrossRef](#)]
8. Khaligh Fard, S.; Ahmadi, H.; Alizadeh Elizei, M.H. Electricity Supply Model of Conventional Residential Buildings in Tehran with Priority on Renewable Energy Using Adaptive Fuzzy-neural Inference System. *Int. J. Eng.* **2023**, *36*, 1793–1814. [[CrossRef](#)]
9. Zurakowska-Sawa, J.; Pyra, M. Assessment and Forecasting of Energy Efficiency in Economic Sectors in Poland. *Energies* **2024**, *17*, 2128. [[CrossRef](#)]
10. Kruszelnicka, W.; Baldowska, P.; Tomporowski, A.; Piasecka, I.; Mrozinski, A. Analysis of energy benefits from energy carriers grinding. *Chem. Eng. Appar.* **2018**, *1*, 7–8.
11. Tang, T.; Kim, H. Linking energy policy, energy insecurity, and health outcomes. *Front. Sustain. Energy Policy* **2023**, *2*, 1231821. [[CrossRef](#)]
12. Dinmohammadi, F.; Han, Y.; Shafiee, M. Predicting Energy Consumption in Residential Buildings Using Advanced Machine Learning Algorithms. *Energies* **2023**, *16*, 3748. [[CrossRef](#)]
13. Krarti, M.; Aldubyan, M. Review analysis of COVID-19 impact on electricity demand for residential buildings. *Renew. Sustain. Energy Rev.* **2021**, *143*, 110888. [[CrossRef](#)] [[PubMed](#)]
14. Zhao, H.-X.; Magoules, F. A review on the prediction of building energy consumption. *Renew. Sustain. Energy Rev.* **2012**, *16*, 3586–3592. [[CrossRef](#)]
15. Alizamir, M.; Kim, S.; Kisi, O.; Zounemat-Kermani, M. A comparative study of several machine learning based non-linear regression methods in estimating solar radiation: Case studies of the USA and Turkey regions. *Energy* **2020**, *197*, 117239. [[CrossRef](#)]
16. Divina, F.; Torres, M.G.; Gomez Vela, F.A.; Noguerra, J.L.V. A Comparative Study of Time Series Forecasting Methods for Short Term Electric Energy Consumption Prediction in Smart Buildings. *Energies* **2019**, *12*, 1934. [[CrossRef](#)]

17. Medina, A.; Camara, A.; Monrobel, J.R. Measuring the Socioeconomic and Environmental Effects of Energy Efficiency Investments for a More Sustainable Spanish Economy. *Sustainability* **2016**, *8*, 1039. [CrossRef]
18. Alghamadi, S.; Tang, W.; Kanjanabootra, S.; Alterman, D. Optimising Building Energy and Comfort Predictions with Intelligent Computational Model. *Sustainability* **2024**, *16*, 3432. [CrossRef]
19. Yi, Z.; Lv, Y.; Xu, D.; Xu, J.; Qian, H.; Zhao, D.; Yang, R. Energy Saving Analysis of a Transparent Radiative Cooling Film for Buildings with Roof Glazing. *Energy Built Environ.* **2021**, *2*, 214–222. [CrossRef]
20. Budiman, A.S.; Illya, G.; Anbazhagan, S.; Tippabhotla, S.K.; Song, W.J.; Sahay, R.; Tay, A.A.O. Enabling Lightweight Polycarbonate-Polycarbonate (PC-PC) Photovoltaics Module Technology—Enhancing Integration of Silicon Solar Cells into Aesthetic Design for Greener Building and Urban Structures. *Sol. Energy* **2022**, *235*, 129–139. [CrossRef]
21. Moldovan, D. Plum Tree Algorithm and Weighted Aggregated Ensembles for Energy Efficiency Estimation. *Algorithms* **2023**, *16*, 134. [CrossRef]
22. Brauers, W.K. *Optimization Methods for a Stakeholder Society: A Revolution in Economic Thinking by Multi-Objective Optimization*; Springer: New York, NY, USA, 2004.
23. UCI Machine Learning Repository. Available online: <https://archive.ics.uci.edu/> (accessed on 17 April 2024).
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. Moldovan, D.; Slowik, A. Energy consumption prediction of appliances using machine learning and multi-objective binary grey wolf optimization for feature selection. *Appl. Soft Comput.* **2021**, *111*, 107745. [CrossRef]
26. Coello Coello, C.A.; Lechuga, M.S. MOPSO: A proposal for multiple objective particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), Honolulu, HI, USA, 12–17 May 2002; pp. 1051–1056.
27. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
28. Izonin, I.; Tkachenko, R.; Mitoulis, S.A.; Faramarzi, A.; Tsmots, I.; Mashtalir, D. Machine learning for predicting energy efficiency of buildings: A small data approach. In Proceedings of the 14th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2023), Almaty, Kazakhstan, 7–9 November 2023; pp. 72–77.
29. Abdel-Jaber, F.; Dirks, K.N. A Review of Cooling and Heating Loads Predictions of Residential Buildings Using Data-Driven Techniques. *Buildings* **2024**, *14*, 752. [CrossRef]
30. Chaganti, R.; Rustam, F.; Daghiri, T.; de la Torre Diez, I.; Mazon, J.L.V.; Rodriguez, C.L.; Ashraf, I. Building Heating and Cooling Load Prediction Using Ensemble Machine Learning Model. *Sensors* **2022**, *22*, 7692. [CrossRef] [PubMed]
31. 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. [CrossRef]
32. Lin, M.; Peng, L.; Liu, T.; Zhang, L. Novel approach to energy consumption estimation in smart homes: Application of data mining and optimization techniques. *Front. Energy Res.* **2024**, *12*, 1361803. [CrossRef]
33. Jain, M.; Maurya, S.; Rani, A.; Singh, V. Owl search algorithm: A novel nature-inspired heuristic paradigm for global optimization. *J. Intell. Fuzzy Syst.* **2018**, *34*, 1573–1582. [CrossRef]
34. Salami, B.A.; Abba, S.I.; Adewumi, A.A.; Dodo, U.A.; Otukogbe, G.K.; Oyedele, L.O. Building energy loads prediction using bayesian-based metaheuristic optimized-explainable tree-based model. *CSCM* **2023**, *19*, e02676. [CrossRef]
35. Eggenesperger, K.; Feurer, M.; Hutter, F.; Bergstra, J.; Snoek, J.; Hoos, H.; Leyton-Brown, K. Towards an Empirical Foundation for Assessing Bayesian Optimization of Hyperparameters. In Proceedings of the NIPS Workshop on Bayesian Optimization in Theory and Practice, Lake Tahoe, NV, USA, 10 December 2013; pp. 1–5.
36. Liu, J.; Zeng, K.; Wang, H.; Du, B.; Tang, Y. Generalized Prediction of Commercial Buildings Cooling and Heating Load Based on Machine Learning Technology. *IOP Conf. Ser. Earth Environ. Sci.* **2020**, *610*, 12004. [CrossRef]
37. 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]
38. Sajjad, M.; Khan, S.U.; Khan, N.; Haq, I.U.; Ullah, A.; Lee, M.Y.; Baik, S.W. Towards Efficient Building Designing: Heating and Cooling Load Prediction via Multi-Output Model. *Sensors* **2020**, *20*, 6419. [CrossRef] [PubMed]
39. Seyedzadeh, S.; Pour Rahimian, F.; Rastogi, P.; Oliver, S.; Glesk, I.; Kumar, B. Multi-Objective Optimisation for Tuning Building Heating and Cooling Loads Forecasting Models. In Proceedings of the 36th CIB W78 2019 Conference, Newcastle, UK, 18–20 September 2019.
40. Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A New Bio-Inspired Algorithm: Chicken Swarm Optimization. In *Advances in Swarm Intelligence. ICSI 2014. Lecture Notes in Computer Science*; Tan, Y., Shi, Y., Coello, C.A.C., Eds.; Springer: Cham, Switzerland, 2014; Volume 8794, pp. 86–94.
41. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
42. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow Search Algorithm. *Comput. Struct.* **2016**, *169*, 1–12. [CrossRef]
43. Mirjalili, S.; Saremi, S.; Mirjalili, S.M.; Coelho, L.d.S. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert. Syst. Appl.* **2016**, *47*, 106–119. [CrossRef]
44. Zouache, D.; Arby, Y.O.; Nouioua, F.; Abdelaziz, F.B. Multi-objective chicken swarm optimization: A novel algorithm for solving multi-objective optimization problems. *Comput. Ind. Eng.* **2019**, *129*, 377–391. [CrossRef]

45. Ic, Y.T. A multi-objective credit evaluation model using moora method and goal programming. *Arab. J. Sci. Eng.* **2020**, *45*, 2035–2048. [[CrossRef](#)]
46. Fortin, F.-A.; De Rainville, F.-M.; Gardner, M.-A.; Parizeau, M.; Gagne, C. DEAP: Evolutionary Algorithms Made Easy. *JMLR* **2012**, *13*, 2171–2175.
47. Guo, G.; Liu, P.; Zheng, Y. Early energy performance analysis of smart building by consolidated artificial neural network paradigms. *Heliyon* **2024**, *10*, e25848. [[CrossRef](#)] [[PubMed](#)]
48. Abdelkader, E.M.; Elshaboury, N.; Ali, E.; Alfalah, G.; Mansour, A.; Al-Sakkaf, A. A Hyper Parameterized Deep Learning Model for Analyzing Heating and Cooling Loads in Energy Efficiency Buildings. In Proceedings of the International Conference on New Trends in Applied Sciences (ICONTAS'23), Konya, Turkey, 1–3 December 2023; pp. 54–60.
49. Prasetyo, B.; Alamsyah; Muslim, M.A. Analysis of building energy efficiency dataset using naïve bayes classification classifier. *J. Phys. Conf. Ser.* **2019**, *1321*, 032016. [[CrossRef](#)]
50. Bui, D.T.; Moayedi, H.; Anastasios, D.; Foong, L.K. Predicting Heating and Cooling Loads in Energy-Efficiency Buildings Using Two Hybrid Intelligent Models. *Appl. Sci.* **2019**, *9*, 3543. [[CrossRef](#)]
51. Rani, U.; Dahiya, N.; Kundu, S.; Kanungo, S.; Kathuria, S.; Rakesh, S.K.; Sharma, A.; Singh, P. Deep learning-based urban energy forecasting model for residential building energy efficiency. *Energy Explor. Exploit.* **2024**; *ahead of print*. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.