AI

Article

A Novel Multi-Objective Hybrid Evolutionary-Based Approach for Tuning Machine Learning Models in Short-Term Power Consumption Forecasting

Aleksei Vakhnin *, Ivan Ryzhikov, Harri Niska and Mikko Kolehmainen

Department of Environmental and Biological Sciences, University of Eastern Finland, Yliopistonranta 1E, 70210 Kuopio, Finland; ivan.ryzhikov@uef.fi (I.R.); harri.niska@uef.fi (H.N.); mikko.kolehmainen@uef.fi (M.K.)

* Correspondence: aleksei.vakhnin@uef.fi or alexeyvah@gmail.com

Abstract: Accurately forecasting power consumption is crucial important for efficient energy management. Machine learning (ML) models are often employed for this purpose. However, tuning their hyperparameters is a complex and time-consuming task. The article presents a novel multi-objective (MO) hybrid evolutionary-based approach, GA-SHADE-MO, for tuning ML models aimed at solving the complex problem of forecasting power consumption. The proposed algorithm simultaneously optimizes both hyperparameters and feature sets across six different ML models, ensuring enhanced accuracy and efficiency. The study focuses on predicting household power consumption at hourly and daily levels. The hybrid MO evolutionary algorithm integrates elements of genetic algorithms and self-adapted differential evolution. By incorporating MO optimization, GA-SHADE-MO balances the trade-offs between model complexity (the number of used features) and prediction accuracy, ensuring robust performance across various forecasting scenarios. Experimental numerical results show the superiority of the proposed method compared to traditional tuning techniques, and random search, showcasing significant improvements in predictive accuracy and computational efficiency. The findings suggest that the proposed GA-SHADE-MO approach offers a powerful tool for optimizing ML models in the context of energy consumption forecasting, with potential applications in other domains requiring precise predictive modeling. The study contributes to the advancement of ML optimization techniques, providing a framework that can be adapted and extended for various predictive analytics tasks.

Keywords: forecasting; power consumption; hybrid evolutionary algorithm; multi-objective evolutionary algorithm

Citation: Vakhnin, A.; Ryzhikov, I.; Niska, H.; Kolehmainen, M. A Novel Multi-Objective Hybrid Evolutionary-Based Approach for Tuning Machine Learning Models in Short-Term Power Consumption Forecasting. *AI* **2024**, *5*, 2461–2496. https://doi.org/10.3390/ai5040120

Academic Editor: Rafał Dreżewski

Received: 29 September 2024
Revised: 5 November 2024
Accepted: 15 November 2024
Published: 19 November 2024

1. Introduction

Forecasting power consumption is a challenging task with a significant impact on energy management, efficiency, and sustainability [1]. Accurate predictions enable better resource allocation, cost reduction, and enhanced grid stability, which are essential for both residential and industrial sectors [2]. Recent advances in machine learning (ML) have significantly impacted the field of renewable energy, especially in optimizing and forecasting wind power generation.

In the study [3], researchers developed a cyber-physical system using deep learning to support renewable energy communities. The system focuses on improving energy distribution and management for sustainable energy sources. By using artificial intelligence (AI), the system can monitor and adjust energy flow to ensure stability and efficiency. The research highlights how this AI-based approach can enhance the resilience of energy networks. This system offers the potential for more reliable energy solutions in communities relying on renewable sources. Another modern study [4] examines methods for predicting wind power using ML models. The study aims to improve the accuracy of wind energy forecasts, which are important for efficient energy management. By comparing different ML models, the researcher identified the most effective techniques for forecasting wind power. The results suggest that advanced algorithms can enhance prediction precision, helping in the integration of wind energy into power grids. This research highlights the potential of ML to support renewable energy systems and make them more reliable. Together, these studies illustrate the transformative role of ML in addressing the challenges of renewable energy, driving the development of intelligent, adaptive systems for sustainable energy solutions.

Given the complexity and variability of energy consumption patterns, developing robust predictive models is challenging, particularly when it involves selecting optimal features and tuning hyperparameters to achieve high accuracy. Numerous methods have been developed to forecast power consumption, ranging from simple statistical models to complex ML algorithms [5]. Despite the advancements, the process of tuning these models and selecting the most relevant features remains a significant challenge. Effective tuning of hyperparameters and feature selection are crucial as they directly impact any model's performance, influencing both its accuracy and computational efficiency [6].

In this context, the article introduces a novel multi-objective (MO) hybrid evolutionary-based approach, GA-SHADE-MO, designed to enhance the tuning process of ML models specifically for forecasting power consumption. In previously published research [7], we have already proposed a GA-SHADE algorithm. GA-SHADE is a single-objective algorithm for simultaneously tuning the set of features and hyperparameters of ML algorithms. In the case of using GA-SHADE, it is necessary to set the preferable number of used features in a model. To find a good solution, it is necessary to run the algorithm multiple times. The proposed GA-SHADE-MO is a logical extension of the previously proposed GA-SHADE algorithm. In this study, we applied GA-SHADE-MO separately to six distinct ML models to demonstrate its flexibility and effectiveness across different models. For each selected model, GA-SHADE-MO generates a set of tuned versions of that same model, each with different hyperparameters and selected features. These versions are ranked according to the Pareto front, providing options with varying numbers of feature inclusion and prediction errors on the validation data.

By employing MO optimization, GA-SHADE-MO addresses the trade-offs involved in ML model development. This method ensures that the models are not only accurate but also efficient, reducing unnecessary complexity by using an optimal number of features. The study focuses on predicting household power use at both hourly and daily levels, which shows the practical importance and usefulness of the proposed algorithm. Results from experiments in the article show that GA-SHADE-MO performs better than traditional tuning methods like random search, achieving higher accuracy and improved computational efficiency. This advantage makes it a valuable tool for optimizing ML models in the energy sector.

Beyond this specific application, the GA-SHADE-MO approach provides a flexible framework that can be used in different predictive tasks, not only in energy forecasting. This study thus contributes to the progress of ML optimization techniques, offering valuable insights and a reliable method for improving predictive analytics.

In summary, the major contributions of this study are as follows: (1) using the proposed GA-SHADE-MO allows obtaining different ML models with varying numbers of features and performance; (2) the proposed GA-SHADE-MO algorithm self-adapts its parameters during the optimization process. No pre-setting of parameters is required for the GA-SHADE-MO algorithm; (3) GA-SHADE-MO is not sensitive to the number of features and hyperparameters of optimized ML due to its evolution nature; (4) the results obtained in this study provide practical valuable information about the features that significantly affect the predictive ability of energy consumption prediction ML models; (5) the proposed approach can be modified and extended for solving different forecasting problems using ML models or other regression models; (6) the proposed algorithm is flexible and not limited to the algorithms currently used, such as GA and SHADE; incorporating more effective algorithms for binary and real-valued optimization would further enhance the overall efficiency of the approach; (7) existing approaches to MO optimization operate with homogeneous data, where all variables in the solution vector must be of the same type. The proposed GA-SHADE-MO approach implements MO optimization for solution vectors with mixed data types (binary, integer, and real-valued).

The rest of the paper is organized as follows. In Section 2, related work and literature review are presented. Section 3 describes in detail the proposed GA-SHADE-MO algorithm. Section 4 describes the used dataset, the set of used ML algorithms and their hyperparameters to be tuned, the settings of numerical experiments, the description of a computation cluster, and the results of the numerical experiments. In Section 5, the results of the numerical experiments are discussed. Section 6 summarizes the entire research work and proposes a direction for further studies in this field.

2. Related Work and Literature Review

Forecasting power consumption properly is a critical task in the energy sector. It helps utility companies plan for future energy demands, manage resources efficiently, and ensure a stable power supply. To achieve accurate forecasts, various regression models are employed. However, the accuracy of these models significantly depends on how well they are tuned. In recent years, the advancement of ML and statistical methods has introduced numerous approaches to improve the performance of regression models. Despite their simplicity, traditional methods, such as linear regression, demonstrate satisfactory performance [8]. More complex models, including decision trees [9], support vector regression [10], and artificial neural networks [11], have also been applied to these problems with notable success.

Hyperparameter tuning is one of the fundamental techniques used to optimize regression models. It involves adjusting model parameters that control the learning process. Methods such as grid search [12], random search [13], and more complex approaches like Bayesian optimization [14] are commonly used for this purpose. Grid search, although computationally expensive, explores a predefined set of hyperparameters to identify the best combination. Random search, on the other hand, samples a larger hyperparameter space more efficiently but with a probabilistic approach. Bayesian optimization leverages past evaluation results to model the performance landscape and make informed decisions regarding which hyperparameters to explore next.

Feature selection and extraction are also vital in optimizing regression models. Techniques such as principal component analysis (PCA) [15] and regularization methods like Lasso and Ridge [16] assist in identifying the features that most significantly influence power consumption predictions. These methods reduce the dimensionality of the data, thereby improving model performance and interpretability.

Ensemble methods, which combine multiple models to improve prediction accuracy, have gained considerable popularity in recent years. Techniques like bagging [17], boosting [18], and stacking [19] utilize the strengths of different models, leading to more robust and accurate forecasts. In summary, the field of power consumption forecasting has experienced significant advancements through the application of various regression model tuning techniques.

Table 1 shows an observation in the field of power consumption prediction. The first column shows the names of the models used. The second column shows a short description of where the data were sourced. The third column lists the authors' names and references. The second-to-last and the last columns show the types of feature selection and hyperparameter methods used. In cases of 'fixed parameters', the authors simply noted the values without providing additional details on how they were determined.

Table 1. Real-world applications in forecasting power consumption.

Model	Data Source	Authors	Feature Selection	Hyperparameters
Wavelet transform and multi-layer LSTM	The electricity consumption from U.S. Electric Power Company	D. Chi [20]	None	Fixed hyperparameters
K-means and FCM–BP	The load data of 200 users from an area of Nanjing	H. Bian, et al. [21]	None	Fixed hyperparameters
MRA-ANN	The multi-energy district boiler, La Rochelle, west coast of France	J. Eynard, et al. [22]	None	Grid search
The hybrid of RobustSTL and TCN	Hourly Power Consumption of Turkey	C. H. Lin, et al. [23]	None	Fixed hyperparameters
ConvLSTM and LSTM	NASA Battery Dataset, Individual Household Electric Power Consumption Dataset, Domestic Energy Management System Dataset	N. Khan, et al. [24]	None	Fixed hyperparameters
DB-Net	IHEPC dataset from the UCI ML repository; the Korean AICT dataset	N. Khan, et al. [25]	None	Grid search
A multiple regression model, a multiple econometric regression model and a LR model of double logarithm	The six socio-economic strata in Bogotá City	C. Peña-Guzmán, et al. [26]	None	None
DNN, RNN, CNN, LSTM	Companies B and T located in Naju, Jeollanam-do	N. Son [27]	Correlation analysis	Fixed hyperparameters
EPC-PM	The UMass Smart dataset	J. Kumar, et al. [28]	None	Fixed hyperparameters
DNN hybrid	Five real-world household power consumption datasets	K. Yan, et al. [29]	None	Fixed hyperparameters
SVR, ANN	Four building clusters in a university	J. Moon. et al. [30]	PCA and FA	Grid search
The learning ensemble of ARIMA, GARCH and PSF	The Spanish Control Centre for the Electric Vehicle	C. Gomez-Quiles, et al. [31]	None	Fixed hyperparameters
DF-CNNLSTM	PJM Hourly Energy Consumption Data	X. Shao, et al. [32]	None	Fixed hyperparameters
TL-MCLSTM	Two subsets from Pennsylvania-New Jersey Maryland	X. Shao, et. al. [33]	None	Grid search

Most models listed in Table 1 utilize fixed hyperparameters. For instance, advanced models such as the wavelet transform and multi-layer LSTM, ConvLSTM and LSTM, and the DNN hybrid model all rely on fixed hyperparameters. This approach potentially overlooks the benefits of more dynamic and adaptive tuning methods. Where parameter tuning is performed, it is predominantly achieved through straightforward techniques such as grid search, as demonstrated in the MRA-ANN and TL-MCLSTM models. Grid search, while systematic, is relatively simple and may not fully exploit the potential of more sophisticated hyperparameter optimization methods such as random search, Bayesian optimization, or evolutionary algorithms. Feature selection receives similarly insufficient emphasis. It is fair to state that in solving forecasting power consumption problems, in most real-world cases, the available set of features is limited. Practically, in most cases, only power lag and time features are available. The majority of models do not incorporate any feature selection methods, indicating a reliance on raw input data in its original form. The few exceptions include the use of Principal Component Analysis (PCA) and Factor Analysis (FA) in the SVR and ANN models, as well as correlation analysis in the study involving DNN, RNN, CNN, and LSTM. While useful, these techniques are relatively basic compared to more advanced feature selection methods that can capture non-linear relationships and interactions between features.

Overall, the table highlights a significant gap in the thorough tuning of model parameters and selection of features in power consumption prediction studies. The predominant use of fixed hyperparameters and basic feature selection techniques highlights a need for more rigorous and advanced approaches to potentially enhance model performance and robustness. The GA-SHADE-MO approach we propose aims to bridge the gap and address the shortcomings encountered during the modeling of power consumption forecasting models. GA-SHADE-MO allows the obtaining of a set of well-tuned models in terms of their complexity. By model complexity, we refer to the number of features used.

3. The Proposed GA-SHADE-MO Algorithm

Before providing a detailed description of our proposed approach, we will first give a general overview of the multi-objective optimization problem and how it is typically solved using evolutionary algorithms. This foundational understanding will help contextualize our method and demonstrate how evolutionary-based techniques are applied to find optimal solutions in complex problem spaces.

3.1. Multi-Objective Optimization

3.1.1. Problem Statement

Multi-criteria optimization, also known as multi-objective or multi-goal optimization, involves optimizing MO functions simultaneously. The formal problem statement for multi-criteria optimization can be formulated as follows in Equation (1).

$$\underset{x \in X}{min}(f_1(x), f_2(x), \ldots, f_k(x)) \tag{1}$$

The integer $k \geq 2$ defines the number of objective functions. X represents the set of feasible solutions, and $x \in \mathbb{R}^n$, where n is the search space dimension. The solution x must satisfy the set of constraints defined in Equations (2) and (3).

$$g_j(x) \leq 0, j = 1, 2, \ldots, m, \tag{2}$$

$$h_l(x) = 0, l = 1, 2, \ldots, p, \tag{3}$$

where m is the number of inequality constraints and p is the number of equality constraints. The feasible solution set is defined in Equation (4).

$$X = \left\{ x \in \mathbb{R}^n \mid g_j(x) \leq 0, j = 1, 2, \ldots, m; h_l(x) = 0, l = 1, 2, \ldots, p \right\}. \tag{4}$$

Formally, we aim to find $x^* \in X$ such that $f(x^*)$ is a non-dominated solution. We say that a vector-solution x_1 dominates a vector solution x_2 $(x_1 \preceq x_2)$ if and only if:

$$\forall\, i \in \{1, 2, \ldots, k\}, f_i(x_1) \leq f_i(x_2), \text{ and} \tag{5}$$

$$\exists\, i \in \{1, 2, \ldots, k\}, f_i(x_1) < f_i(x_2). \tag{6}$$

We say a vector-solution $x^* \in X$ is called Pareto-optimal if there does not exist another vector-solution $x \in X$ that dominates x^*. Thus, the multi-criteria optimization problem involves finding the set of Pareto-optimal solutions, which cannot be improved in any objective function without degrading at least one of the others. In this study, we rely on two criteria for tuning ML models. The first criterion is a regression model error, the second criterion is the number of features used in a regression model. Both criteria are subject to minimization.

3.1.2. State-of-the-Art Approaches for Multi-Objective Optimization

MO optimization has evolved significantly over the decades. In the 1950s and 1960s, the foundation was laid with the concept of Pareto optimality, introduced by Vilfredo Pareto [34], which defines a state where no objective can be improved without worsening another objective. Early methods such as the weighted sum method [35] emerged during this period, simplifying multi-objective problems by combining multiple objectives into a single one. In the 1970s, linear programming techniques [36] were adapted to handle multiple objectives, often utilizing the weighted sum method. Goal programming [37] was also introduced to address the limitations of the weighted sum method, setting specific targets for each objective and minimizing the deviation from these targets. The 1980s saw the advent of evolutionary algorithms, with John Holland's work on genetic algorithms (GAs) introducing a novel approach by simulating natural evolution processes [38]. Schaffer's Vector Evaluated Genetic Algorithm (VEGA) [39] was one of the first attempts to extend GAs for multi-objective optimization, by dividing the population based on different objectives. During the 1990s, advanced evolutionary techniques were further developed. Multi-Objective Genetic Algorithms (MOGAs) emerged [40], incorporating Pareto ranking to better handle multiple objectives. The Niched Pareto Genetic Algorithm (NPGA) [41] introduced niching methods to maintain diversity in the population, ensuring a better spread of Pareto optimal solutions. Srinivas and Deb's Non-dominated Sorting Genetic Algorithm (NSGA) [42] improved the selection process using non-dominated sorting and sharing functions. The 2000s marked further refinements and the introduction of hybrid methods. NSGA-II, a major refinement of NSGA developed by Deb et al. [43], addressed computational complexity, elitism, and diversity maintenance, becoming widely adopted. Particle Swarm Optimization (PSO) was adapted for multi-objective optimization [44], utilizing a population of solutions influenced by their own and their neighbors' best positions. Hybrid methods, combining evolutionary algorithms with other optimization techniques like local search and mathematical programming, enhanced performance and robustness [45]. From the 2010s to the present, advances and applications in multi-objective optimization have continued to expand. The Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), introduced by Zhang and Li [46], decomposes a MO problem into single-objective subproblems. Indicator-based methods, such as the Indicator-Based Evolutionary Algorithm (IBEA), use performance indicators to guide the search process [47]. Recent approaches have also integrated ML techniques to enhance the efficiency and effectiveness of multi-objective optimization. Throughout its history, the field has evolved from simple linear methods to sophisticated evolutionary algorithms, continuously improving solution quality, computational efficiency, and application breadth.

In recent years, many approaches have been proposed for solving MO problems [48]. However, most real-world applications are viewed as black-box optimization problems [49]. It means that only the quality of the solution can be evaluated. There is no additional information regarding the connection between variables. Sometimes, based on system analysis, a problem can be viewed as a gray-box optimization problem, but it does not provide enough information for solving. Evolutionary algorithms have achieved significant success in solving MO optimization problems. One of the state-of-the-art evolutionary approaches for solving MO optimization problems, as previously mentioned, is MOEA/D [46]. The underlying principle of this approach is as follows: MOEA/D decomposes the MO problem into a set of single-objective problems. These decomposed problems must be optimized simultaneously within the same population. The number of single-objective problems is defined as the population size (the number of individuals). The pseudo-code of the MOEA/D algorithm, including its main steps, is presented below:

```
Step 1. Initialize population $P$ and $W$ weights for each individual.
Step 2. Evaluate the population $P$.
Step 3. Generate a set of neighbors $T$ for each weight vector.
Step 4. While the termination condition is not met do Step 5, otherwise go to Step 9.
Step 5. Generate mutant vectors for each individual using a mutation strategy.
```

Step 6. Perform crossover operator using mutant and parent vectors to create a trial vector.

Step 7. Evaluate trial vectors and update a solution if a new scalar objective value is better.

Step 8. If the termination criterion is not met, go to Step 4, otherwise go to Step 9.

Step 9. Return the non-dominated solutions of P.

In Step 1, based on the principles of evolutionary-based optimization algorithms, it is necessary to randomly generate a population P consisting of N vectors (possible solutions) $x_i, i = 1, 2, \ldots, N$. Every solution should be generated within the feasible decision space. In other words, every solution should be feasible. The set of weights vectors, $W = \left(w^1, w^2, \ldots, w^k\right)$, is generated based on the following rules, $\sum\limits_{i=1}^{k} w_i = 1,\ and\ \forall\ i,\ w_i \geq 0$. These vector weights are used for decomposing the MO problem. The set of weights can be generated uniformly or designed to cover a specific area of the searching space. In Step 2, it is needed to perform scalarization. Here, scalarization is a method used to transform each subproblem into a one-dimensional (scalar) problem that can be more easily optimized. For example, using weighted sums, Chebyshev's method, or other scalarization methods, each of the subproblems becomes a function whose optimization leads to finding solutions to the original multicriterial problem. This approach allows optimizing multiple goals simultaneously by decomposing them into problems that are easier to solve. Thus, in terms of MOEA/D, decomposition refers to the partitioning of the problem, while scalarization is the method of transforming each resulting subproblem into a form suitable for optimization. In Steps 5 to 7, the population is evolved. If the algorithm finds a better solution than the trial vector, then it should be replaced, and the optimization process continues until a termination criterion is met. There are two commonly used termination criteria in EA-based heuristics. The first is based on a fitness budget: if the EA has exhausted the predefined number of fitness evaluations, the optimization process should be terminated. The second criterion is based on the evaluation of changes within the population. For example, a predefined number of generations may be established without any changes to the global optimum or the average fitness value across the population. Once this number is reached, the search process is terminated. In this study, we used the first approach, which involves a predefined maximum number of fitness evaluations.

Because we use MOEA/D principles, we have to decompose a MO problem into many single-objective problems. Several common methods exist for performing scalarization: weighted sum, normalized weighted sum, minimax method, and linear interpolation method. Each of these will be examined in detail.

The weighted sum method, one of the simplest and most widely used scalarization methods, consists of summing all the objective functions according to their weight coefficients, as shown in Equation (7).

$$g(x, w) = \sum\nolimits_{j=1}^{k} w_j \cdot f_j(x), \tag{7}$$

where w_j is the weight coefficient for j-th objective problem, $f_j(x)$ is a value of j-th objective problem.

In the case of normalized weighted sum, the objective problem should be normalized according to Equation (8).

$$g(x, w) = \sum\nolimits_{j=1}^{k} w_j \frac{f_j(x) - z_j^*}{r_j}, \tag{8}$$

where z_j^* is the current ideal point (the best-found value for j-th objective problem), and r_j is the difference between the highest and lowest values found for the objective.

The minimax method focuses on minimizing the maximum deviation from the ideal point and is defined by Equation (9),

$$g(x) = \max_{1 \le j \le k} \left(\frac{f_j(x) - z_j^*}{r_j} \right), \tag{9}$$

where z_j^* and r_j are the same as in the normalized weighted sum method.

One of the most commonly used methods when two criteria exist is the linear interpolation method. It is defined as Equation (10),

$$g(x, w) = w_1 f_1(x) + (1 - w_1) f_2(x). \tag{10}$$

Although this method works well for two objectives, it can be extended by techniques such as uniform coverage of the search space. However, it is important to consider the 'curse of dimensionality', as increasing the number of criteria can make this approach computationally expensive.

3.2. GA-SHADE-MO

We propose a hybrid population-based multi-objective GA-SHADE-MO algorithm for simultaneous hyperparameter optimization and feature selection. In our study, the Genetic Algorithm (GA) [50] is used for optimizing the feature set, as it performs well in optimization problems where solutions are represented as binary vectors (0 s and 1 s). In this representation, features that are used are marked as 1, and those that are not used are marked as 0. The SHADE (success-history-based parameter adaptation for differential evolution) [51] algorithm is utilized for hyperparameter optimization of ML models.

An example of a solution's representation can be seen in Figure 1. The first part of the decision vector is a set of hyperparameters $\theta_l \epsilon \mathbb{R}^n$ of an ML model, the second part represents the used feature, $\varphi_k \epsilon [0;1]$ by the ML model. If $\varphi_k = 0$, it means that the k-th feature is not used. And if $\varphi_k = 1$, it means that the k-th feature is used. Every solution should be feasible, and $\varphi \neq \varnothing$. At least one feature from the set of features must be equal to 1.

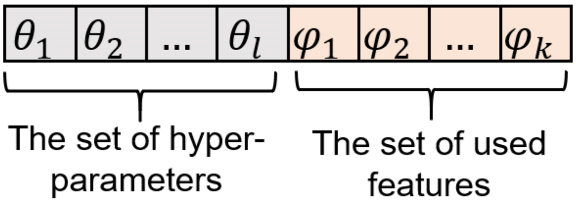

Figure 1. Representation of a solution in the GA-SHADE-MO algorithm.

Because our goal is to simplify the model, we will employ two criteria. Every individual in a population is represented as an individual which decomposes the MO problem into single-objective problems. When a multi-objective optimization problem with two criteria is decomposed into a single-objective problem, as in MOEA/D, the mathematical formulation can be expressed as follows. Let $f_1(x)$ and $f_2(x)$ be the two objective functions to be minimized. In our specific context, $f_1(x)$ represents the mean absolute error (*MAE*) on the validation dataset, and $f_2(x)$ represents the number of features used in the ML model. Since these two criteria are different in terms of scale, we modified the fitness function calculation, transitioning from Equations (10) to (11).

$$fitness = w_1 \frac{f_1(x) - f_1(x)_{min}}{f_1(x)_{max} - f_1(x)_{min}} + (1 - w_1) \frac{f_2(x) - f_2(x)_{min}}{f_2(x)_{max} - f_2(x)_{min}}, \tag{11}$$

where $f_1(x)_{max}$ and $f_1(x)_{min}$ are maximum and minimum possible values that the criterion can take. The same is for the $f_2(x)$ criterion. These values must be predefined. Defining minimum values is straightforward. In our case, $f_1(x)$ relates to the *MAE* on the validation set; thus, the minimum possible value is 0.0. The second criterion is related to the number of features, making the minimum possible value for $f_2(x)$ is 1. To determine the $f_1(x)_{max}$ value after randomly generating the population, we evaluate all solutions. We then find the maximum value among them and set $f_1(x)_{max}$ to this value. $f_2(x)_{max}$ is defined as the total number of features in the used dataset.

As discussed in the previous section, the simultaneous tuning of parameters and feature selection presents challenges due to the unique characteristics of these approaches. The common characteristic of these methods is the need to adjust the model hyperparameters and select specific features for prediction. The proposed GA-SHADE-MO algorithm effectively tunes an ML model over an adequate number of experiments. In practice, the SHADE algorithm has proven its effectiveness in parameter optimization of black-box problems. Additionally, hyperparameters of SHADE, such as the scale factor *F* and the crossover rate *CR*, self-adapt during the optimization process. For feature optimization, we employ a crossover operator from the GA to generate new solution candidates. Originally, SHADE used the current-to-*p*best/1 strategy, as shown in Equation (12), to perform the mutation operator and create a trial solution. This strategy uses the *p*best index. *p*best defines a randomly selected individual from the top *p*% of individuals in the population. However, when solving MO problems, it is not possible to define a single best solution in the population, because the solution to a MO problem consists of a set of non-dominated solutions. Because of this, we replaced the current-to-*p*best/1 strategy with current-to-rand/1, Equation (13). This eliminates the need to define the best or a set of best solutions in the population during the optimization process.

$$v_{i,j} = x_{i,j} + F \cdot \left(x_{pbest,j} - x_{i,j} \right) + F \cdot \left(x_{r_1,j} - x_{r_2,j} \right), \tag{12}$$

$$v_{i,j} = x_{i,j} + F \cdot \left(x_{r_3,j} - x_{i,j} \right) + F \cdot \left(x_{r_1,j} - x_{r_2,j} \right), \tag{13}$$

where x_i is the *i*-th individual from the current population; x_{pbest} is a randomly chosen individual from the top *p*% of the best individuals in the population. r_1, r_2, r_3 are randomly generated indices from the population, $r_1 \neq r_2 \neq r_3$. Additionally, GA-SHADE-MO does not use an external archive. In the original SHADE, the external archive is used to store replaced individuals, which are later used with low probability to maintain population diversity. However, when solving multi-objective (MO) problems with MOEA/D, the algorithm allows the creation of new individuals based on those with the closest distances according to the weight vectors. Since maintaining diversity by storing replaced individuals in an external archive is challenging and not the main goal of our study, we excluded the external archive from GA-SHADE-MO. To conserve space in the paper, we do not provide detailed descriptions of GA and SHADE, as these were addressed in a previous study (see [7]).

One of the primary aims of the study is to simplify ML models. By simplification, we refer to achieving a balance between the number of used features and predictive accuracy. Obviously, increasing the number of dependent features with high impact in an ML model typically increases accuracy. Figure 2 shows an example of existing ML models. The *X*-axis denotes the number of used features, *y*-axis denotes a model error. The example of a Pareto front in a multi-objective optimization problem consists of a series of feasible and infeasible points representing the majority of solutions. The Pareto front is composed of a set of equally optimal solutions, where no single solution is superior to another when considering all objectives simultaneously.

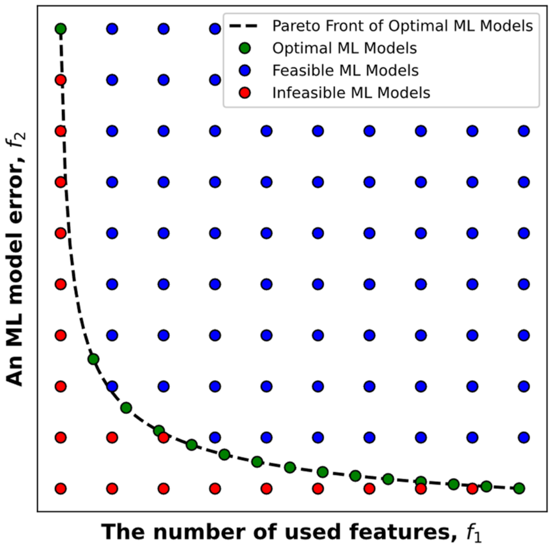

Figure 2. Example of solutions of MO problem in the context of building ML models.

A complete pseudo-code of the proposed GA-SHADE-MO algorithm is presented below. To execute the GA-SHADE-MO algorithm, it is necessary to define an ML model, the search range for the model's hyperparameters (lower and upper bounds), and the set of features from a dataset. The main steps of the GA-SHADE-MO algorithm, without loss of generality, can be described as follows.

Requirements: an ML model, the set of hyperparameters Θ and their searching ranges, the set of features Φ, the population size, and the maximum number of fitness evaluations must be defined.

Step 1. Initialize population *P* and *W* weights for each individual, set the initial values for the historical memory of *F* and *CR* to 0.5, define the closest set of individuals from the population based on weights, and set the closest set *T* of individuals for each one.

Step 2. Evaluate the population *P* based on a decomposition method.

Step 3. While the termination condition is not met, proceed to Step 4; otherwise, go to Step 8.

Step 4. Generate mutant vectors for each individual using a mutation strategy.

Step 5. Perform crossover operator using mutant and parent vectors to create a trial vector.

Step 6. Evaluate trial vectors and update the solution if the new scalar objective value is better.

Step 7. Update the historical memory *H*.

Step 8. If the termination criterion is not met, go to Step 3, otherwise go to Step 9.

Step 9. Return non-dominated solutions of *P*.

In Step 2 and Step 6, it is needed to train the selected model and evaluate its performance on the validation set. Since any population-based optimization algorithm uses the calculation of the fitness function to evolve the population. Here, the modified error on the validation dataset acts as the fitness function (Equation (11)). As previously stated, this iteration of the algorithm optimizes a single model while identifying distinct model variants characterized by differing feature sets and validation error metrics.

The algorithm includes population initialization, solution evaluation using a decomposition method, mutation and crossover operations, memory updating, and returning non-dominated solutions. This process aims to find an optimal balance between model accuracy and the number of features used.

4. The Experimental Setup and Results

4.1. Power Consumption Forecasting Problem

The forecasting problem we are addressing focuses on predicting next-day power consumption in a single household with a primary emphasis on optimizing energy use. Optimizing energy use in residential settings is crucial not only for reducing costs but also for improving energy efficiency and sustainability, which both are increasingly important in the context of global energy challenges. This type of energy demand forecasting is typically performed on daily, hourly, or even 15-min intervals to ensure precise control over energy consumption and enable proactive decision-making. The higher the resolution, the more accurately energy management systems can react to fluctuations in demand, enhancing energy optimization for individual households. Predicting demand at such granular levels allows the implementation of dynamic pricing, load shifting, and demand response strategies, all of which are vital for both the grid's stability and the home's energy efficiency [52].

Traditional statistical models for power consumption prediction, however, are often based on data from groups of similar buildings or aggregated demand profiles. These group-based models, while useful for understanding general consumption patterns, fail to account for the distinct characteristics and behaviors of individual homes. Averaged models derived from groups of buildings miss critical local variables such as weather conditions, which can vary even within short distances, affecting heating and cooling demands in homes with electric HVAC (Heating, Ventilation, and Air Conditioning) systems. The physical attributes of the building itself—such as insulation, construction materials, and window placement—further influence energy consumption in ways that generic models cannot capture. Studies have shown that group-based forecasting approaches tend to generalize behaviors, leading to reduced accuracy when applied to specific buildings, especially those that deviate from the norm in terms of construction or occupant behavior [53].

Moreover, these generalized models often omit the specific behavioral patterns of the occupants when using domestic appliances, which can significantly impact daily energy demand. For example, one household may operate washing machines and dishwashers during the day, while another may prefer night-time usage, which would lead to vastly different consumption profiles. Family-specific habits, such as the use of electric vehicles, home offices, or smart appliances, are typically overlooked in averaged models. Without tuning these models to a specific building or household, it is challenging to implement demand-side management effectively. Personalized energy forecasting models that integrate local environmental data, building-specific parameters, and occupant behavior are therefore necessary for accurate demand prediction and energy optimization. This approach aligns with recent research advocating for more granular, context-sensitive models in the smart grid domain, enabling a more efficient and personalized approach to managing energy consumption at the household level [54].

4.2. Measurement Data

In this paper, the data were collected from a private family house in Kuopio, a city in Eastern Finland. The power consumption data were collected from 2015 to 2018. The values were recorded every minute. In some intervals, the data contain missing values. The pre-processed dataset is shown in Figure 3 below. The x-axis presents the time. The y-axis denotes the power in kWh.

In this study, ML models forecast power consumption based on weather data. Weather data were obtained from the Savilahti observation station. The location of the observation station is shown in Figure 4. The map view was made using the OpenStreetMap® service. OpenStreetMap® is open data, licensed under the Open Data Commons Open Database License (ODbL) by the OpenStreetMap Foundation (OSMF), https://www.openstreetmap.org/copyright/en, accessed on 30 September 2024. For privacy reasons, the exact location of the private house cannot be disclosed. However, it is confirmed that the house is located within a 10 km radius of the weather station. The approximate location of the house is indicated by a dotted black circle in the upper part of Figure 4. In the top-left corner, there is a map of Finland with the area of interest marked by a red square. This map is included for illustrative purposes and to provide additional context regarding the geographical location where the data were collected.

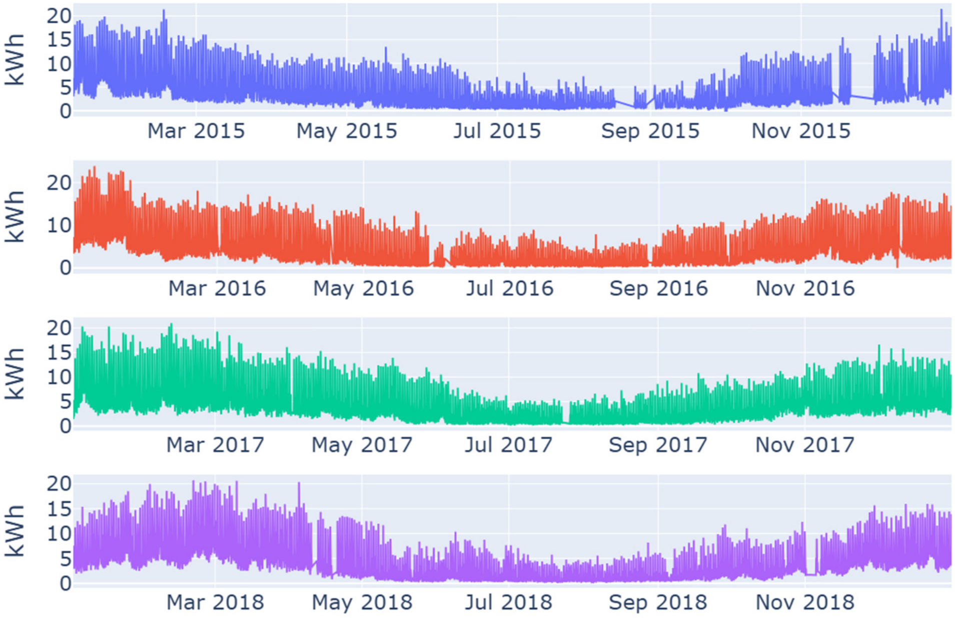

Figure 3. Visualization of the obtained power consumption in the private house.

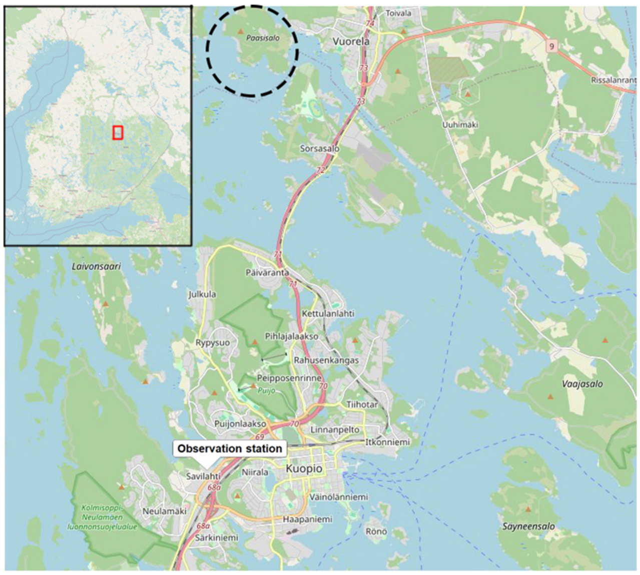

Figure 4. The view on the map of the observation station and an area of the house.

4.3. Modelling Schemes and Input Variables

Table 2 shows the description of the set of features, where the first column indicates the feature abbreviation, and the second column contains the feature description. Since time features are periodic, such as hourly, daily, weekly, and monthly, they have to be extracted from the time series using sine and cosine trigonometric functions. It is important to note that we consider two temporal levels, daily and hourly. We use two different abbreviations for power and ambient temperature lags depending on the forecasting level. For instance, T1lag and T24lag represent the power consumption from one day ago and 24 h ago, corresponding to the daily and hourly levels, respectively. Our numerical experiments have shown that more distant lags, such as four or more days ago, do not lead to improvements in predictive error values. We base our numerical experiments on actual weather data, as they represent an optimistic scenario for future real-time forecasts that would rely on predicted weather data.

Table 2. Feature variables used as the inputs of the ML models.

Abbreviation of the Feature	Description of the Feature
T, Tlag1, Tlag2, Tlag3	Averaged ambient temperature at the day, one day ago, two days ago, and three days ago.
Tlag24, Tlag48, Tlag72	Averaged ambient temperature, 24 h ago, 48 h ago, and 72 h ago.
P1lag, P2lag, P3lag	Power consumption one day ago, two days ago, and three days ago, respectively.
P24lag, P48lag, P72lag	Power consumption 24 h ago, 48 h ago, and 72 h ago, respectively.
Pr	Atmospheric pressure
Rel	Relative humidity
Dew	Dew point
Cl	Cloud cover level
windcos	Direction of the wind transformed in cos
windsin	Direction of the wind transformed in sin
windsp	Wind speed
hcos	cos(hour·2π/24) transformation of hours
hsin	sin(hour·2π/24) transformation of hours
dcos	cos(day·2π/7) transformation of days
dsin	sin(day·2π/7) transformation of days
wcos	cos(week·2π/52) transformation of weeks
wsin	sin(week·2π/52) transformation of weeks
mcos	cos(month·2π/12) transformation of months
msin	sin(month·2π/12) transformation of months

Figure 5 presents two correlation matrices illustrating the relationships among features at the daily (left subplot) and hourly (right subplot) levels on the training dataset. The color scale ranges from −1.0 (strong negative correlation) to +1.0 (strong positive correlation). The X- and Y-axes display the set of features, showing the relationships between all features with each other.

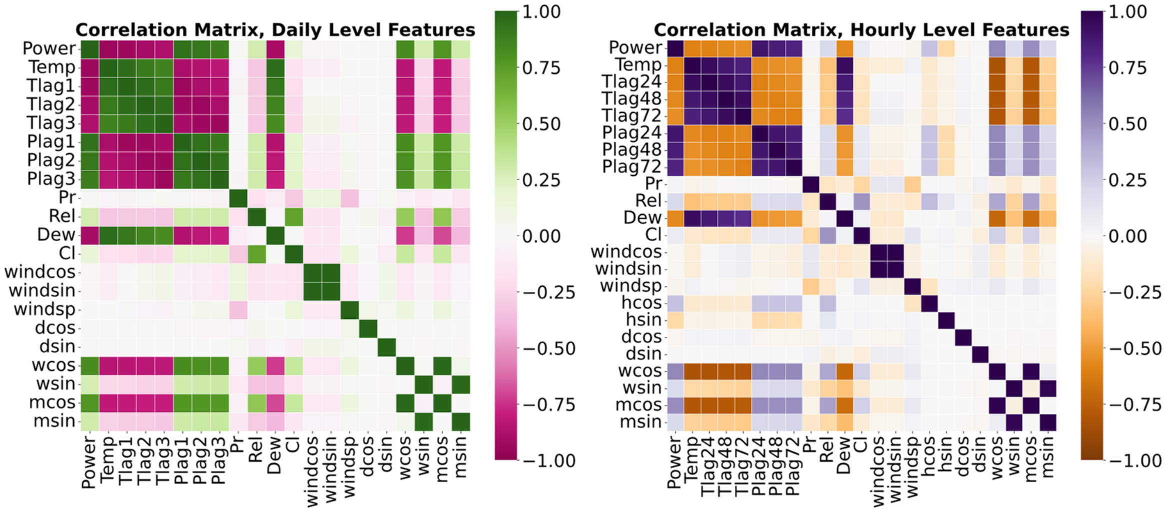

Figure 5. Correlation matrices of Daily (**left**) and Hourly levels (**right**) on train datasets.

To evaluate the performance of ML models tuned by GA-SHADE-MO, we used the following scheme for splitting data, known as time series cross-validation [55,56], as illustrated in Figure 6. The *X*-axis represents the time intervals. The first five rows on the *Y*-axis represent the folds used to evaluate the model's performance on validation data. At each subsequent fold, the validation data from the previous fold is added to the training set, and a new time period is selected as the validation set. We modified the classic approach of calculating validation errors by adding weights for each fold. Equation (14) shows the calculation of the weighted *MAE* on the validation set. Here, *n* is the number of folds.

$$MAE_{weighted} = \sum_{i=1}^{n} w_i \cdot MAE_i,\ w_i = \frac{i}{\sum_{j=1}^{n} j}, \tag{14}$$

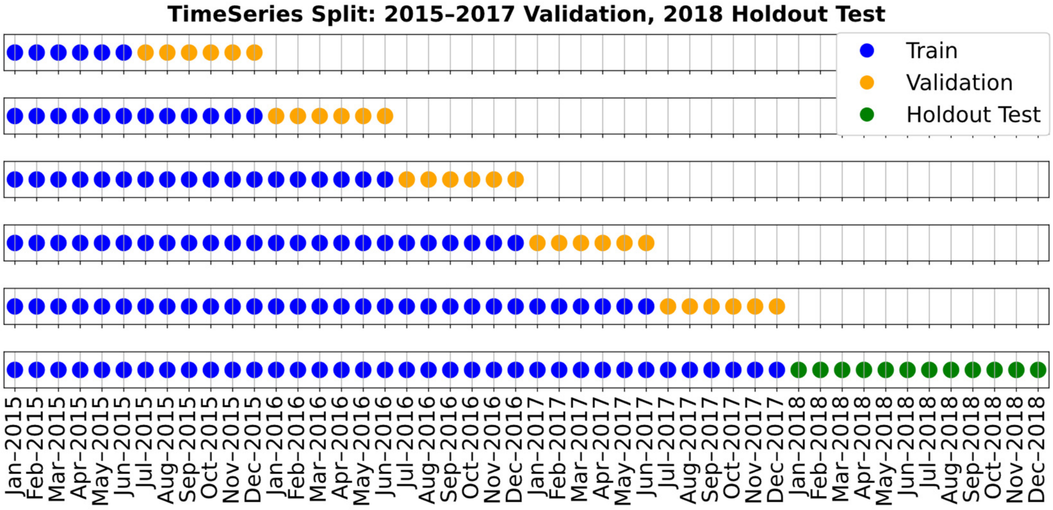

Figure 6. Splitting the data using time series cross-validation.

In this study, we use weights for metric evaluations on validation data for the following reasons. In time series problems, the data are time-dependent, meaning the order of observations is important. However, more recent data often contain more relevant information for predicting future values accurately. In this context, it is important to pay more attention to the validation results at later time intervals than at earlier ones. Applying weights during the validation process, where later time periods are given more weight, has several important advantages. Data that are closer to the prediction time may better reflect current trends and states of the system, since time series may include various seasonal changes, trends, and other time dependencies. Applying greater weights to later folds allows us to account for the fact that these data are more relevant to the model.

For example, if a long-term trend emerges later in the data, the model must capture this trend for accurate future predictions. In time series forecasting problems, there may be situations where older data are no longer relevant, and newer data carry more meaningful information for decision-making. Weighting errors on validation data assists in guiding the model to improve forecasts based on more recent information, ensuring that the model generalizes better to future data. If the time series exhibits a changing trend, a model trained on uniformly weighted data may struggle to adapt. Error weighting, where later periods are weighted more heavily as they contain more relevant information, helps the model better adapt to trend changes because the greater importance of recent observations is considered during the optimization of the model parameters. Errors made later in the time series can have a significant impact on the final forecast, especially if trends or seasonal effects are strong. By increasing the weight of recent folds, we ensure that the model pays more attention to these periods, which can improve forecasting accuracy in real-world scenarios.

To mitigate the effect of features with different range values on the learning process, the values of all features should be mapped to the same scale. All features are normalized using Z-score normalization, where $new\_value = (x - \mu)/\sigma$, with x as the original value, μ as the mean value, and σ as the standard deviation of the data. Each model is tuned using GA-SHADE-MO on validation data, after which the tuned ML model is evaluated on holdout test data.

We also evaluated the model's performance based on different feature sets. We chose three scenarios: (1) using all features, (2) excluding ambient temperature-related features, and (3) using power consumption-related features and time features. Finally, the performance of the ML models was evaluated at both hourly and daily levels, aiming to forecast power consumption for the next 24 h or a day, respectively.

4.4. Model Optimization Using GA-SHADE-MO and Settings

We investigated the performance of the proposed GA-SHADE-MO algorithm for automatically building various well-known ML models using a real-world dataset. Evolutionary algorithms (EAs), including GA-SHADE, are stochastic and incorporate randomness in their search process. Random factors such as mutation, crossover, and initial population generation can lead to different outcomes with each run. Therefore, running the algorithm multiple times helps evaluate its average performance. In this study, the GA-SHADE-MO algorithm was run independently five times. After each independent run, the population of the last generation was recorded. After five independent runs, solutions based on the Pareto front were selected from all five populations. Each run is limited to a maximum of 5000 fitness evaluations. Based on our numerical experiments, after about 4500 evaluations, on average, across all ML models and scenarios, GA-SHADE-MO can no longer improve individuals and reaches a plateau. It is important to note that, the value of maximum fitness evaluations strongly depends on the problem and used ML models. We set the population size to 100 and the historical memory size H to 10. When identifying the nearest set T of individuals for mutation, we selected the closest 10% of individuals based on their weights. The distance of weights was measured by the Euclidean metric. The GA-SHADE-MO performance was also compared with the random search approach at both daily and hourly levels in the all-features scenario. A model was selected based on the obtained results on the validation dataset. The random search approach also had five independent runs with 5000 fitness evaluations in each run, generating a total of 25,000 random solutions.

Evaluating the performance of the proposed GA-SHADE-MO algorithm based on a single randomly chosen ML model is inadequate. We evaluated the performance of six ML models that were previously used for forecasting power consumption, such as Linear Regression (LR) [57], ElasticNetCV (ENCV) [58], Decision Tree (DT) [59], Random Forest (RF) [60], multi-layer perceptron (MLP) [61], and XGBoost [62]. ML models vary in their approach and applicability, which makes it important to understand their unique strengths. LR is a linear model that predicts the target variable by fitting a linear relationship between the input features and the target variable, making it suitable for datasets with linear dependencies. ENCV is a regularized linear model that combines L1 (Lasso) and L2 (Ridge) regularization with cross-validation to reduce overfitting and select important features, making it effective for datasets with many correlated features. DT models are tree-based, splitting data into subsets based on feature values to make predictions. They are effective for understanding and visualizing decision processes but are prone to overfitting complex datasets. To address this, RF models use an ensemble approach (bagging), constructing multiple decision trees and aggregating their predictions to improve accuracy and control overfitting. This makes RF suitable for a wide range of real-world applications and robust for large datasets. MLP is a neural network model with multiple layers of neurons and non-linear activation functions, allowing it to model complex relationships in data. It is effective for both regression and classification tasks, especially when the relationship between features and the target is highly non-linear. Finally, XGBoost is an ensemble model that uses gradient boosting to combine the predictions of multiple weak learners, usually decision trees, to create a strong predictive model. XGBoost is known for its high performance and efficiency, making it popular in ML competitions and structured data applications. Each model has unique strengths and is suited to different types of data and problems, making it essential to choose the appropriate model based on the specific characteristics of the dataset and the task at hand. The hyperparameter set and corresponding search boundaries are provided in Table A1.

The experimental analysis of GA-SHADE-MO for building ML models is computationally intensive, especially for more complex models. The method was implemented in Python using the scikit-learn open-source ML libraries [63,64]. Our computational cluster consisted of eight AMD Ryzen Pro 2700 CPUs, offering 128 threads for parallel processing. The cluster runs on Ubuntu 22.04 LTS. We uploaded the algorithm's source code and the results of the numerical experiments. Further details, including the source code of GA-SHADE-MO and experimental results, are available in the repository https://github.com/VakhninAleksei/GA-SHADE-MO (accessed 30 September 2024).

In the DT, RF, MLP, and XGBoost models, we fixed the random state parameter to ensure repeatability in evaluating the models' performance. Fixing the random state ensures the repeatability of performance evaluations, preventing fluctuations in results caused by different initial conditions. This is critical for fair comparisons and reliable hyperparameter selection. Assessing model stability becomes challenging, as changes in the random state can lead to significant performance variations that may skew evaluations of the model's reliability. In our study, we implemented early stopping for the MLP, terminating the training process if the model error did not decrease for five consecutive epochs.

4.5. Performance Evaluation

To evaluate forecast predictions in our study, we employed traditional and commonly used metrics, such as mean absolute error (MAE), Equation (15); mean squared error (MSE), Equation (16); the coefficient of determination (R^2), Equation (17); and the index of agreement (IA), Equation (18). In Equations (15)–(18), y represents the set of observed values, $\hat{y}$ represents the set of predicted values. Based on the values of the metrics, we can evaluate and better understand how a particular model works.

MAE is a widely used metric for understanding the average difference between observed and predicted values. It retains the same units as the original predicted value. MSE is similar to MAE, but MSE is more sensitive to large errors, which makes it effective in scenarios where minimizing large deviations is essential. R^2 measures the proportion of variance in the target variable that the model can explain. The values of the metric range from 0.0 to 1.0. An R^2 value close to 1 indicates that the model explains most of the variance in the target variable, implying strong predictive performance. The index of agreement (IA) measures the agreement between predicted and observed values. Like R^2, it ranges from 0.0 to 1.0, with 1.0 indicating the best match between model predictions and actual observations.

To form a comprehensive evaluation of model performance, it is essential to consider these metrics collectively rather than in isolation.

$$MAE = \frac{1}{n}\sum\nolimits_{i=1}^{n}|y_i - \hat{y}_i|, \tag{15}$$

$$MSE = \frac{1}{n}\sum\nolimits_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{16}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}, \tag{17}$$

$$IA = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(|\hat{y}_i - \overline{y}| + |y_i - \overline{y}|)^2}. \tag{18}$$

4.6. Numerical Results

In this subsection, we first present the numerical results at the daily level, followed by the numerical results at the hourly level. Here, the best-found models refer to those with the lowest MAE error on the validation data. Therefore, they use the maximum number of features and have the minimum error on the validation data.

4.6.1. Forecasting Daily Energy

Table 3 contains errors of the best-found tuned models by GA-SHADE-MO. It presents values for different ML models across various experimental conditions. The table shows validation and test errors according to the considered metrics and is divided into two sections. Columns define different metrics, while rows list the different ML models that were tuned. The cells contain numeric error values representing the metric error for each model. Tables 4 and 5 have the same structure as Table 3, except that they show numerical results in different scenarios.

Table 3. The best-found tuned models by GA-SHADE-MO, validation performance, daily level. All-features scenario.

The Best ML Tuned Model	**Validation**				**Test**			
	MAE	**MSE**	**IA**	**R^2**	**MAE**	**MSE**	**IA**	**R^2**
LR	7.144	124.351	0.934	0.983	8.122	133.879	0.957	0.989
ENCV	7.152	123.971	0.934	0.983	8.156	134.737	0.957	0.989
DT	8.796	182.728	0.907	0.976	9.556	180.824	0.942	0.985
RF	7.434	133.028	0.93	0.982	8.312	135.63	0.956	0.989
MLP	6.685	115.374	0.939	0.984	7.527	115.959	0.963	0.990
XGBoost	7.424	130.873	0.931	0.982	8.101	132.323	0.957	0.989

Table 4. The best-found tuned models by GA-SHADE-MO, validation performance, daily level. Excluded ambient temperature scenario.

The Best ML Tuned Model	**Validation**				**Test**			
	MAE	**MSE**	**IA**	**R^2**	**MAE**	**MSE**	**IA**	**R^2**
LR	7.338	126.076	0.933	0.983	8.270	137.162	0.956	0.989
ENCV	7.232	123.059	0.935	0.983	8.511	141.938	0.954	0.988
DT	9.355	187.305	0.901	0.974	10.516	209.288	0.933	0.983
RF	7.926	146.398	0.923	0.980	8.682	152.150	0.951	0.988
MLP	6.824	117.734	0.939	0.984	7.903	124.932	0.960	0.989
XGBoost	7.646	135.900	0.927	0.981	8.577	149.363	0.952	0.988

Table 5. The best-found tuned models by GA-SHADE-MO, validation performance, daily level. Power lags and time scenario.

The Best ML Tuned Model	**Validation**				**Test**			
	MAE	**MSE**	**IA**	**R^2**	**MAE**	**MSE**	**IA**	**R^2**
LR	9.541	196.054	0.899	0.973	9.984	196.182	0.937	0.983
ENCV	9.493	195.795	0.899	0.973	9.962	196.618	0.937	0.983
DT	10.274	228.431	0.879	0.967	12.647	292.150	0.906	0.973
RF	9.803	217.283	0.886	0.969	11.240	225.884	0.927	0.980
MLP	9.525	201.665	0.893	0.971	11.211	226.656	0.927	0.980
XGBoost	9.712	205.661	0.892	0.971	11.382	236.580	0.924	0.979

Figure 7 consists of two subplots, each illustrating the *MAE* performance of six tuned ML models across varying numbers of features (found by GA-SHADE-MO). The left and right subplots correspond to validation and test data, respectively. The *X*-axis represents the number of features used by the tuned models, and the *Y*-axis displays the *MAE* values. To address the overlap of dots when displaying multiple ML models, we randomly shifted the dots near the number of used features (*X*-axis) to enhance the readability of the numerical results.

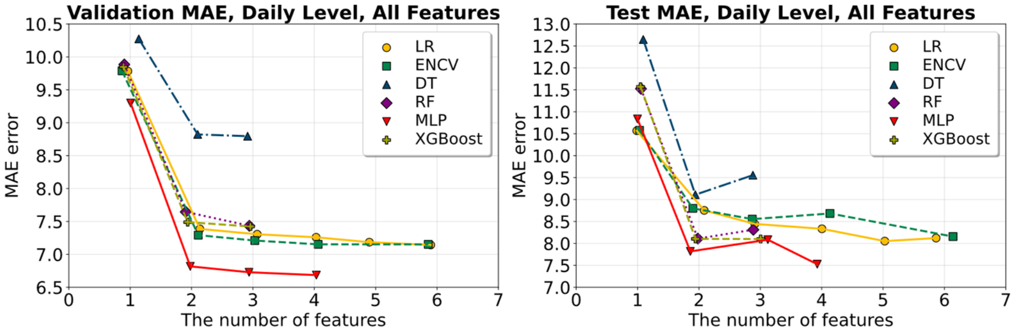

Figure 7. Found Pareto front of tuned ML models using GA-SHADE-MO. Daily level. All-features scenario.

Figure 8 is a hexagon marker plot that visualizes the relationship between different tuned ML models and the features they utilize at the daily level. The horizontal axis shows the names of various tuned ML models, with the number of used features indicated next to each name in brackets. The vertical axis lists the features used by these models. Each hexagon marker represents the use of a particular feature by a corresponding model. The color of the hexagon corresponds to the model group, allowing for easy visual identification of feature usage patterns across different models. The chart provides a comprehensive overview of the features utilized by each model, with color coding and hexagon markers facilitating quick visual comparisons.

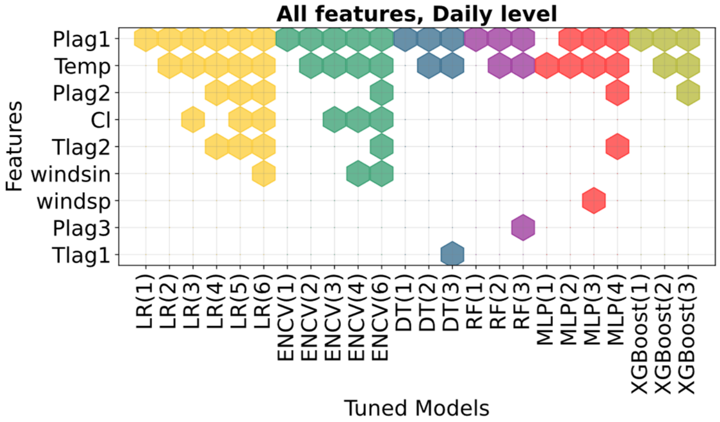

Figure 8. The set of used features for each tuned ML model. Daily level. All-features scenario.

Figures 9 and 10 show the found Pareto front and the corresponding feature sets for excluded ambient temperature scenario at daily level, respectively. Figures 11 and 12 show the found the Pareto front and the corresponding feature sets for power lags and time scenario at daily level, respectively.

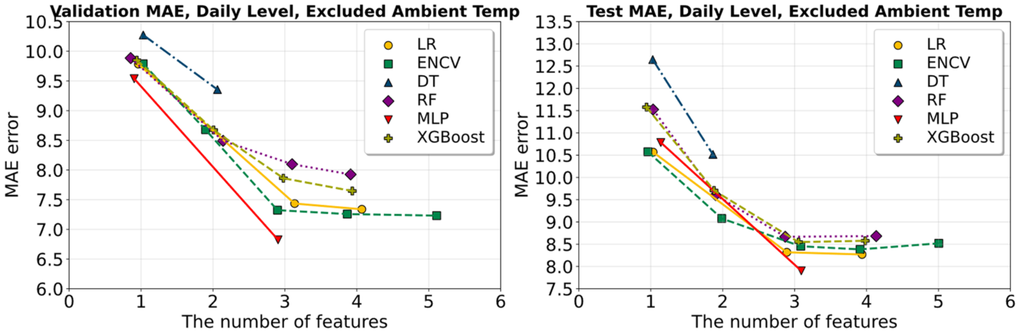

Figure 9. Found Pareto front of tuned ML models using GA-SHADE-MO. Daily level. Excluded ambient temperature scenario.

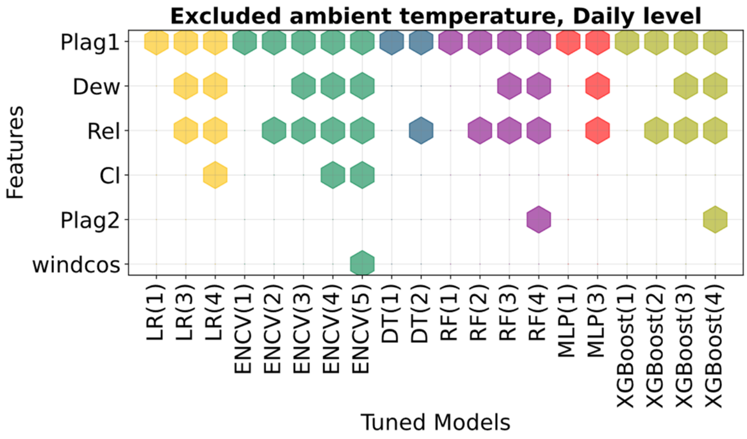

Figure 10. The set of used features for each tuned ML model. Daily level. Excluded ambient temperature scenario.

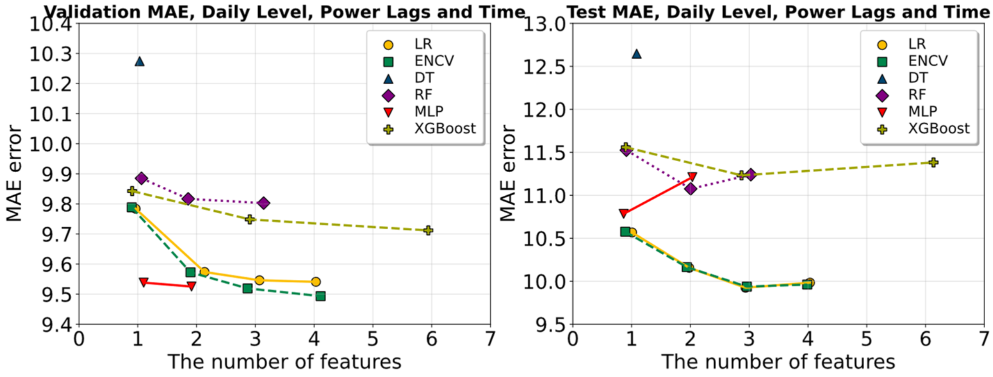

Figure 11. Found Pareto front of tuned ML models using GA-SHADE-MO. Daily level. Power lags and time scenario.

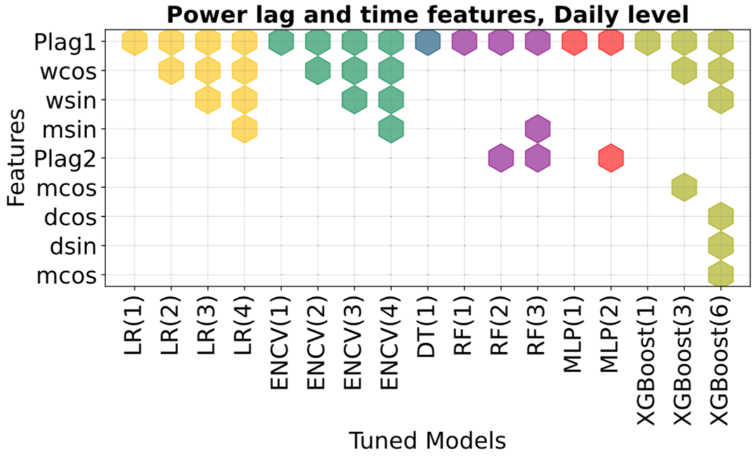

Figure 12. The set of used features for each tuned ML model. Daily level. Power lags and time scenario.

Figure 13 is a line plot that compares actual and predicted values over time at the daily level of the best-found model, MLP with 4 features in a case of using all features from the dataset. The *X*-axis represents the time period from January 2018 to December 2018, while the *Y*-axis is the values of power consumption indicating the magnitude of the actual and predicted data. The plot features two lines, the actual values are depicted by a solid blue line, and the predicted values are shown by a dashed red line.

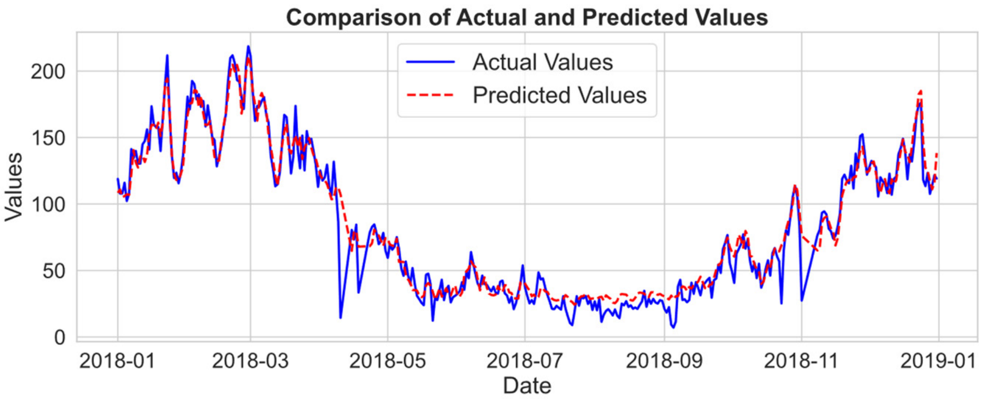

Figure 13. Comparison graph of the performance of the tuned MLP model and the actual values. Daily level. All-features scenario.

Figure 14 consists of three side-by-side subplots, each illustrating a different aspect of the model's performance. The numerical results are the same as for Figure 13, the performance of tuned MLP with 4 features. The first subplot on the left is a scatter plot that compares predicted values to actual values. The *X*-axis shows predicted values, and the *Y*-axis shows actual values. A red dashed line is drawn to represent the ideal scenario where the predicted values perfectly match the actual values. The center subplot displays a scatter plot of the residuals, which are the differences between actual and predicted values, plotted against the predicted values. A horizontal line at zero represents where residuals would lie if the model predictions were perfect. The third subplot on the right shows a histogram that represents the distribution of residuals. The histogram bars indicate how frequently certain residual values occur, with a line overlay likely representing the density estimate of the residuals. This combined visualization provides a comprehensive overview of the model's predictive accuracy, the nature of the residuals, and their distribution, offering insights into the model's performance.

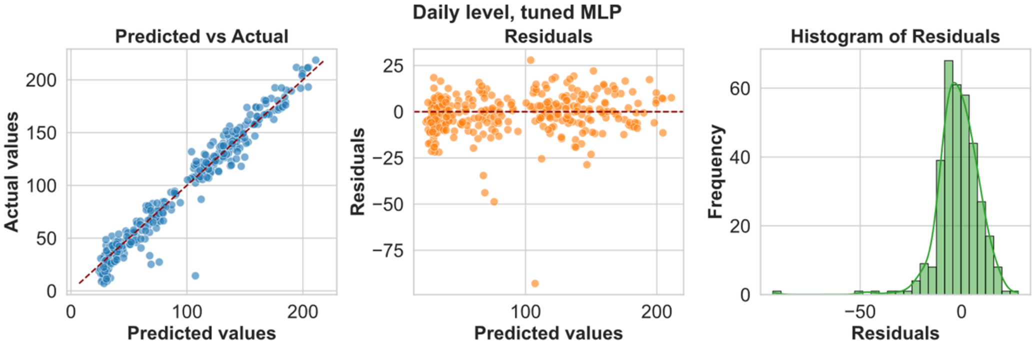

Figure 14. Scatter plot (**left**), residuals plot (**center**), and histogram of residuals values (**right**) of tuned MLP model on a daily level. All-features scenario.

Based on the numerical results, we can see that the best performance on a daily level with all features has been obtained on the MLP model with 4 features. We performed a random search on this scenario. The random search had the same amount of resources (maximum fitness evaluations) as GA-SHADE-MO. Figure 15 shows the difference in the Pareto front using GA-SHADE-MO and Random Search. The structure of Figure 15 is the same as Figure 7.

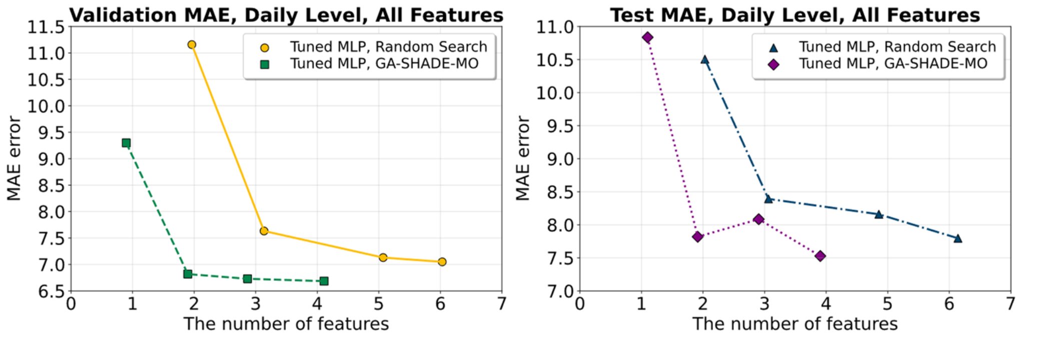

Figure 15. Found Pareto fronts using the proposed GA-SHADE-MO and Random Search. Daily level. All-features scenario.

4.6.2. Forecasting Hourly Energy

Figures 16 and 17 show the found Pareto front and the corresponding feature sets for all features scenario at hourly level, respectively. Figures 18 and 19 show the found Pareto front and the corresponding feature sets for excluded ambient temperature scenario at hourly level, respectively. Figures 20 and 21 show the found Pareto front and the corresponding feature sets for power lags and time scenario at hourly level, respectively.

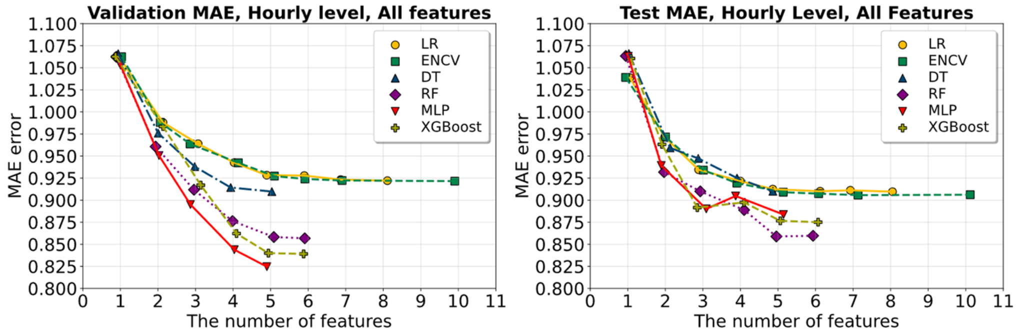

Figure 16. Found Pareto front of tuned ML models using GA-SHADE-MO. Hourly level. All-features scenario.

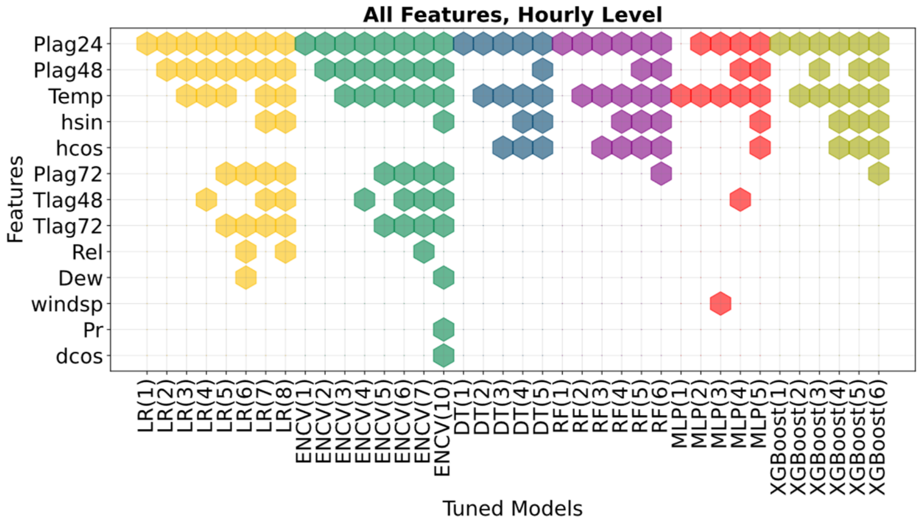

Figure 17. The set of used features for each tuned ML model. Hourly level. All-features scenario.

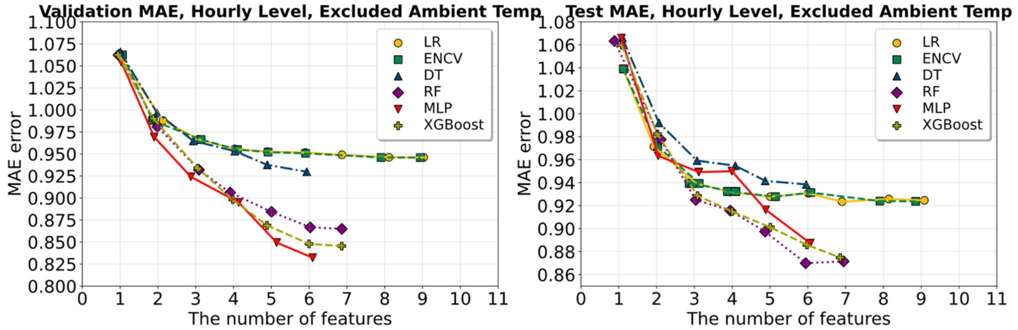

Figure 18. Found Pareto front of tuned ML models using GA-SHADE-MO. Hourly level. Excluded ambient temperature scenario.

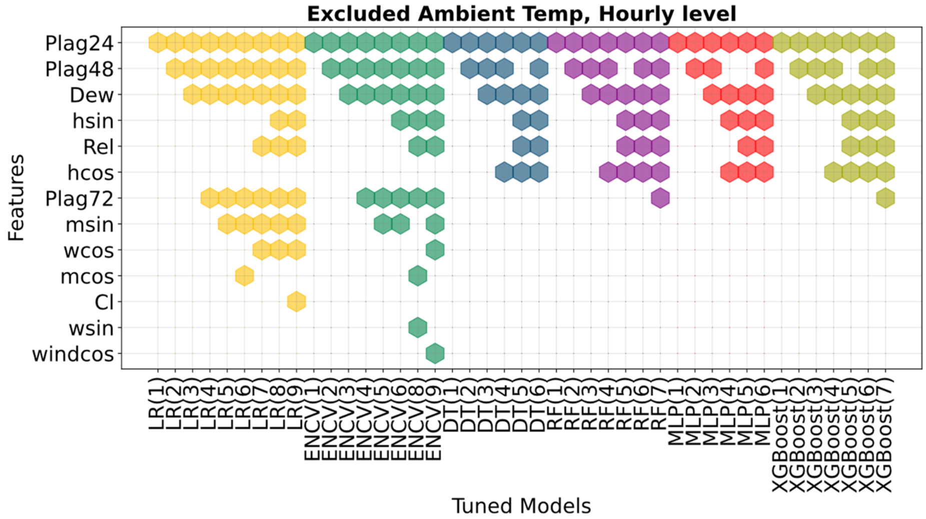

Figure 19. The set of used features for each tuned ML model. Hourly level. Excluded ambient temperature scenario.

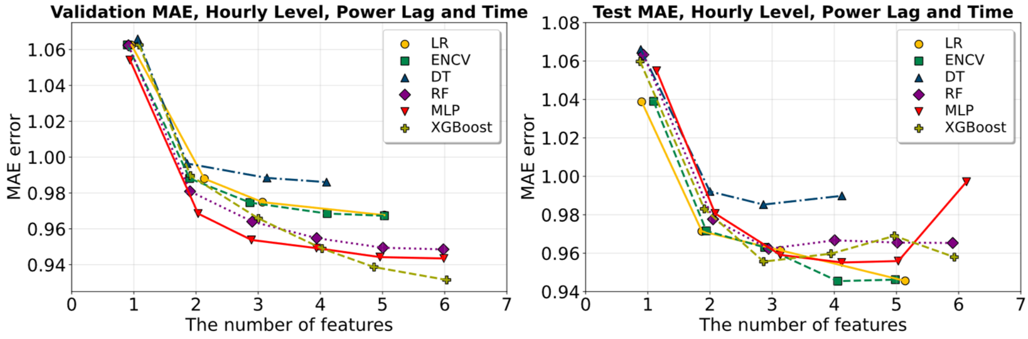

Figure 20. Found Pareto front of tuned ML models using GA-SHADE-MO. Hourly level. Power lags and time scenario.

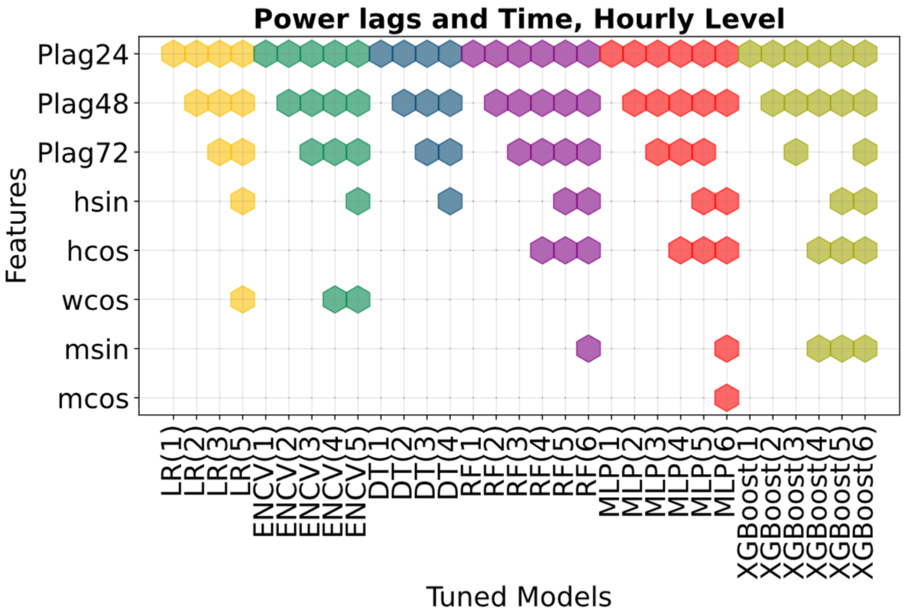

Figure 21. The set of used features for each tuned ML model. Hourly level. Power lags and time scenario.

Tables 6–8 have the same structure as Table 3, except that they show numerical results in different scenarios for hourly level.

Table 6. The best-found tuned models by GA-SHADE-MO, validation performance, hourly level. All-features scenario.

The Best ML Tuned Model	Validation				Test			
	MAE	**MSE**	**IA**	**R^2**	**MAE**	**MSE**	**IA**	**R^2**
LR	0.922	1.709	0.811	0.946	0.910	1.722	0.835	0.954
ENCV	0.922	1.709	0.811	0.946	0.906	1.716	0.836	0.954
DT	0.910	1.660	0.817	0.949	0.910	1.783	0.829	0.952
RF	0.857	1.434	0.840	0.955	0.860	1.533	0.853	0.959
MLP	0.825	1.272	0.858	0.961	0.884	1.970	0.811	0.951
XGBoost	0.839	1.335	0.852	0.959	0.875	1.677	0.839	0.956

Table 7. The best-found tuned models by GA-SHADE-MO, validation performance, hourly level. Excluded ambient temperature scenario.

The Best ML Tuned Model	Validation				Test			
	MAE	**MSE**	**IA**	**R^2**	**MAE**	**MSE**	**IA**	**R^2**
LR	0.946	1.781	0.804	0.943	0.925	1.776	0.830	0.953
ENCV	0.946	1.780	0.804	0.943	0.924	1.775	0.830	0.953
DT	0.930	1.710	0.811	0.947	0.938	1.843	0.823	0.950
RF	0.865	1.468	0.837	0.954	0.871	1.563	0.850	0.958
MLP	0.835	1.317	0.853	0.960	0.895	1.821	0.826	0.952
XGBoost	0.847	1.365	0.848	0.958	0.874	1.606	0.846	0.958

Table 8. The best-found tuned models by GA-SHADE-MO, validation performance, hourly level. Power lags and time scenario.

The Best ML Tuned Model	Validation				Test			
	MAE	MSE	IA	R^2	MAE	MSE	IA	R^2
LR	0.968	1.878	0.793	0.941	0.946	1.879	0.820	0.948
ENCV	0.967	1.876	0.794	0.941	0.946	1.881	0.820	0.948
DT	0.986	1.943	0.787	0.939	0.990	2.004	0.808	0.944
RF	0.949	1.774	0.805	0.945	0.965	1.885	0.819	0.946
MLP	0.943	1.685	0.817	0.950	0.997	2.032	0.805	0.943
XGBoost	0.932	1.693	0.815	0.948	0.953	1.862	0.822	0.947

Figure 22 is the same structure as Figure 13. Figure 22 shows a line plot that compares actual and predicted values over time at the hourly level of the best-found model, MLP with five features.

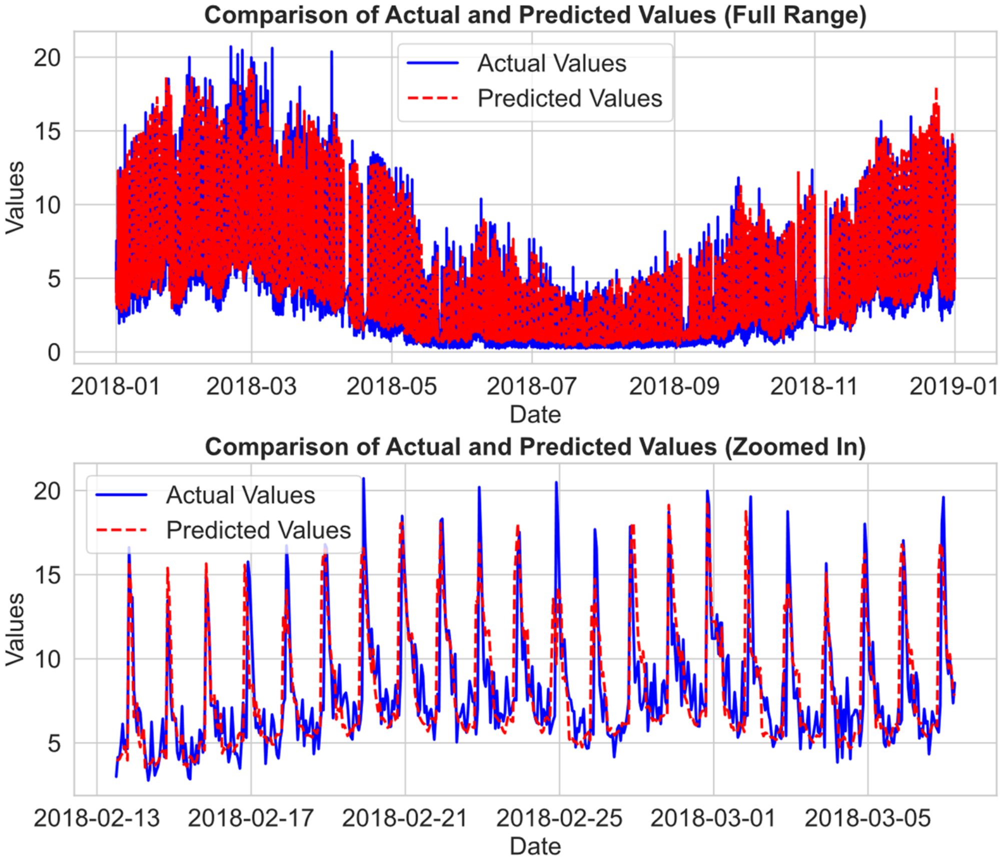

Figure 22. Comparison graph of the performance of the tuned MLP model and the actual values, hourly level. All-features scenario.

Figure 23 has the same structure as Figure 14, but it shows the numerical results on an hourly level for tuned MLP with five features.

As it was in the previous section, Section 4.6.1, we evaluated the searching performance of random search on the MPL model and compared it with the GA-SHADE-MO performance at the hourly level with all feature scenarios, Figure 24.

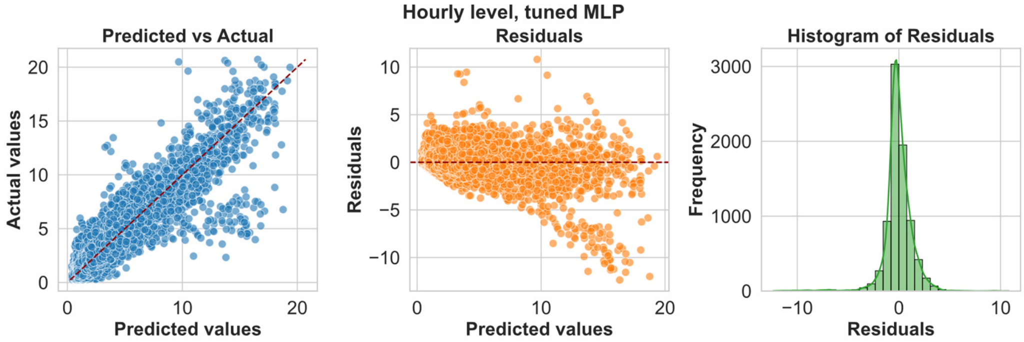

Figure 23. Scatter plot (**left**), residuals plot (**center**), and histogram of residuals values (**right**) of tuned MLP model on hourly level. All-features scenario.

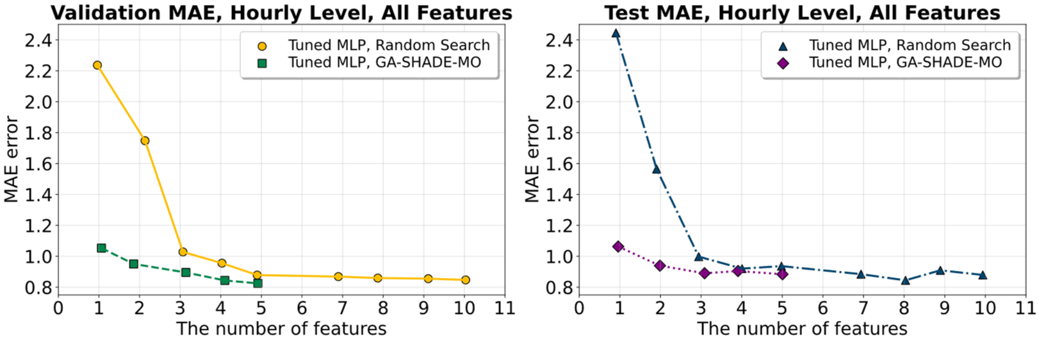

Figure 24. Found Pareto fronts using the proposed GA-SHADE-MO and Random Search. Hourly level, all-features scenario.

5. Discussion

In this section, we will thoroughly examine and analyze the numerical results obtained from our study in Section 4.6. We will discuss the significance of our findings, identify potential limitations, and provide an interpretation of the results in terms of their practical and theoretical implications. We will first discuss the results at the daily level and then proceed to analyze the hourly level results.

5.1. Discussion of Daily Level of Forecasting

5.1.1. All-Features Scenario on Daily Level

On the validation dataset (Table 3), the MLP model with 4 features shows the best *MAE* performance at 6.685, followed by the LR model with 7.144. XGBoost achieves the lowest *MSE* of 130.873, effectively minimizing larger errors. Both MLP and XGBoost have high *IA* and R^2 values (0.939/0.984 for MLP and 0.931/0.982 for XGBoost), indicating a strong correlation between predicted and actual values. On the test dataset, MLP again performs best with an *MAE* of 7.527, confirming its robustness. MLP also achieves the lowest *MSE* of 115.959. RF, MLP, and XGBoost models all show the highest *IA* and R^2 values, reflecting prediction accuracy.

Figure 7 reveals that the test *MAE* trends are similar to the validation *MAE*. MLP, LR, and ENCV maintain low *MAE* values as the number of features increases on validation data. In contrast, DT consistently has higher *MAE* across all feature counts. LR and ENCV perform well with 2 to 3 features but show limited improvement with more. These observations suggest that the optimal number of features for most models is around 3 to 4, where the *MAE* is minimized. Simpler models like LR and ENCV perform adequately with fewer features, but more complex models like XGBoost and MLP benefit significantly from additional information on test data. Overall, increasing features improves accuracy, with XGBoost and MLP performing best with more features on test data.

Figure 8 shows the specific features used by each model. Obviously, models like LR and ENCV rely on a greater number of features, while MLP and XGBoost achieve good performance with fewer features, underscoring their efficiency. As we can assume, in daily forecasting, the target variable can be described well by linear partitions (using LR and ENCV models). Features such as "Temp", "Plag1", and "Plag2" are frequently selected simultaneously across models, highlighting their critical importance for the power prediction task.

5.1.2. Excluded Ambient Temperature Scenario on Daily Level

Excluding ambient temperature leads to the following observations. On the validation dataset, Table 4, the MLP model continues to excel, achieving an *MAE* of 6.824, slightly better than ENCV's 7.232. MLP also has the lowest *MSE* of 117.734, demonstrating its robustness even without the significant feature of ambient temperature. MLP leads in *IA* and R^2 (0.939 and 0.984), consistently aligning with actual values and explaining data variance. On the test dataset, MLP again shows superior performance with an *MAE* of 7.903 and the lowest *MSE* of 124.932, confirming its robustness across datasets. LR and ENCV models also perform well, though MLP slightly outperforms them in *IA* and R^2. DT shows the highest errors, indicating its struggle when key features, such as ambient temperature, are missing. As we can see, in general, the performance of this scenario is worse in comparison to the scenario with all features.

The Pareto front analysis in Figure 9 shows how model accuracy (*MAE*) improves with the number of features, especially from one to two and from two to three. On the test data, all models, except DT, demonstrate notable improvements with just a few features, highlighting their adaptability.

Figure 10 reveals that features like "Plag1", "Dew", and "Rel" are frequently selected across models, emphasizing their importance in the absence of ambient temperature. Models like LR and ENCV rely on a larger set of features to maintain accuracy, while MLP and XGBoost perform well with fewer features, showcasing their efficiency in feature selection. These results suggest that although ambient temperature is significant, models can still perform well by leveraging other key features effectively.

5.1.3. Power Lag and Time Scenario on Daily Level

On the validation dataset, Table 5, the ENCV model achieves the best *MAE* of 9.493, closely followed by MLP with an *MAE* of 9.525. ENCV also leads in *MSE* with a value of 195.795, demonstrating its effectiveness in reducing larger errors. The highest *IA* and R^2 values are seen in MLP and ENCV, reflecting their strong correlation with actual values and ability to explain data variance. On the test dataset, ENCV maintains its top performance with an *MAE* of 9.962, slightly outperforming LR's *MAE* of 9.984. Both models have similar *MSE* values (ENCV at 196.618 and LR at 196.182). In contrast, DT performs again poorly, with an *MAE* of 12.647 and an *MSE* of 292.150, indicating it struggles with power lags and time features.

The Pareto front analysis in Figure 11 shows how model accuracy, measured by *MAE*, improves with the number of features, especially when moving from one to two features. However, the improvements are less significant compared to previous scenarios. LR and ENCV show the most noticeable gains, suggesting they leverage additional features more effectively for improved accuracy. In contrast, RF and DT exhibit smaller improvements, indicating lower sensitivity to additional features in this scenario.

Figure 12 highlights the specific features frequently selected by models in the context of power lags and time, such as "Plag1", "wcos", and "wsin", which are critical across multiple models. LR and ENCV continue to rely on a larger set of features, while MLP demonstrates a good performance with fewer features on validation data, showcasing its efficiency in feature utilization. As we can see, the scenario with power and time features is the worst scenario in terms of prediction performance on a daily level.

5.1.4. Detailed Analysis of the Tuned ML Model on Daily Level and All-Features Scenario

The visualizations provide valuable insights into the MLP model's performance with 4 features in predicting power consumption for a private house. In Figure 13, a comparison of actual and predicted values for 2018 shows that the predicted values (red dashed line) closely follow the actual values (solid blue line), indicating the model effectively captures overall trends and seasonal variations. The strong correlation suggests the model is well-tuned and can generalize to unseen data. However, some deviations occur during extreme spikes and drops in usage, pointing to areas for further refinement or the influence of external factors not included in the model.

Figure 14 provides additional analysis with a scatter plot, residuals plot, and histogram of residuals. The scatter plot shows predictions are closely clustered around the diagonal line, confirming a strong correlation between predicted and actual values. Minor errors are visible in the scatter and residual plots, where residuals are evenly distributed around the horizontal axis, indicating no systematic bias. Most residuals are small, though a few outliers highlight occasional inaccuracies. The histogram of residuals, with a slight leftward shift, shows most errors are centered around zero, reinforcing the model's accuracy and consistency.

Overall, the model performs well in predicting household power consumption, accurately capturing daily patterns. However, occasional deviations are linked to unusual consumption behaviors, which may be influenced by unpredictable human factors such as vacations or changes in household routines. These factors are difficult to model and may explain discrepancies between actual and predicted values. While the model is a powerful forecasting tool, its predictions should be considered in the context of potential human-related variations in energy usage.

5.1.5. GA-SHADE-MO vs. Ransom Search on Daily Level and All-Features Scenario

Figure 15 provides several important insights into the performance of the GA-SHADE-MO and Random Search algorithms in optimizing MLP models. GA-SHADE-MO achieves lower *MAE* compared to Random Search. This suggests that GA-SHADE-MO is more effective in finding optimal model configurations, particularly with fewer features. The validation graph clearly shows that GA-SHADE-MO reaches lower *MAE* with fewer features, demonstrating its efficiency. Furthermore, as the number of features decreases, there is a noticeable reduction in *MAE* for both methods, though this effect is more pronounced with GA-SHADE-MO. This indicates that the well-tuned models using GA-SHADE-MO do not require a large number of features to perform well. The best performance is observed with only three or four features. Increasing the number of features beyond this point does not lead to significant improvement and, in some cases, even worsens the model's performance on test data. The similarity between the validation and test results, with only slightly higher *MAE* values on the test data, suggests good generalization capabilities of the models. This consistency between the validation and test graphs indicates that the findings are robust. In conclusion, GA-SHADE-MO demonstrates enhanced efficiency in utilizing features to achieve lower *MAE* compared to Random Search, making it a more suitable method for tuning MLP models in this context. These insights are critical for informing the choice of optimization method and the number of features to use in further studies.

5.2. Discussion of Hourly Level of Forecasting

5.2.1. All-Features Scenario

In the validation phase, Table 6, the MLP model performs the lowest *MAE* of 0.825 and *MSE* of 1.272, indicating its high prediction accuracy. It also achieves the highest *IA* of 0.858 and R^2 value of 0.961, demonstrating its superior ability to match actual data and explain variance. XGBoost performs similarly, with an *MAE* of 0.839, slightly higher *MSE* of 1.335, and comparable *IA* (0.852) and R^2 (0.959) to MLP, making it a strong alternative in terms of accuracy and reliability. On the test dataset, MLP performs well with an *MAE* of 0.884. XGBoost and RF perform better than MLP on test data, with *MAE* of 0.875 and

0.860, respectively. Notably, the RF model excels with the lowest test *MSE* of 1.533 and the highest *IA* of 0.853, indicating its strength in capturing trends despite not having the lowest *MAE* on validation data.

The Pareto front analysis in Figure 16 shows that as the number of features increases, *MAE* decreases across all models, with MLP and XGBoost benefiting the most from 2 to 5 features, after which improvements plateau. This suggests that these models are effective in leveraging additional features up to a point. RF and DT also improve with more features but plateau earlier, indicating less sensitivity to additional features. As seen in daily-level scenarios, LR and ENCV perform better with more features but underperform on the hourly level, indicating the need for more complex, nonlinear models for hourly predictions. DT shows quite the same performance as LR and ENCV on test data.

Figure 17 highlights commonly selected features like “Plag24”, “Plag48”, “Temp”, and “hsin” across all models, emphasizing their importance for accurate hourly predictions. MLP, XGBoost, and RF despite using fewer features, maintain high performance, underscoring their efficiency in feature selection and utilization.

5.2.2. Excluded Ambient Temperature Scenario

This scenario provides insights into how the models adapt when a crucial feature, ambient temperature, is removed. Based on the numerical results, in Table 7, we can see the following. In the validation phase, the MLP model demonstrates the best performance with an *MAE* of 0.835 and the lowest *MSE* of 1.317. The MLP model also achieves the highest *IA* of 0.853 and an R^2 value of 0.960, indicating its strong capability to accurately predict and explain the variance in the data, even without the ambient temperature feature. The XGBoost model follows closely, with an *MAE* of 0.847 and an *MSE* of 1.365, with a high *IA* of 0.848 and R^2 of 0.958, showcasing its robustness and reliability in the scenario without a key feature. On the test dataset, MLP continues to perform well, achieving an *MAE* of 0.895. The XGBoost model again shows comparable performance with an *MAE* of 0.874 and an *MSE* of 1.606, alongside an *IA* of 0.846 and an R^2 of 0.958, confirming its effectiveness. Notably, the RF model demonstrates strong performance on the test data with the lowest *MAE* of 0.871 and *MSE* of 1.563 and the highest *IA* of 0.850, suggesting that it can capture trends effectively despite the exclusion of ambient temperature. However, the DT model, while performing well in the validation phase, better than LR and ENCV, with an *MAE* of 0.930, shows slightly weaker performance on the test data with an *MAE* of 0.938, indicating that it is more sensitive to the absence of ambient temperature.

The Pareto front analysis in Figure 18 illustrates the relationship between the number of features and *MAE* across different models when ambient temperature is excluded. For both validation and test datasets, MLP and XGBoost demonstrate significant improvement in *MAE* as the number of features increases from one to five, after which the benefits plateau. This pattern suggests that these models are adept at compensating for the lack of ambient temperature by effectively utilizing other features. The RF model also shows improvement, in both phases, indicating its adaptability in this scenario. Notably, RF increases greatly its performance from 4 to 5 features.

Figure 19 provides a detailed view of the features selected by each model. Commonly selected features across models include “Plag24”, “Plag48”, and “Dew”, which are crucial for maintaining accuracy in the absence of ambient temperature. MLP is able to maintain high performance with a small number of features, underscoring its efficiency in feature utilization. On the other hand, models like LR and ENCV require a broader set of features. However, their performance is significantly worse in comparison with MLP.

5.2.3. Power Lag and Time Scenario

This scenario, Table 8, highlights how models perform in power prediction when power and time features are emphasized. In the validation phase, XGBoost shows the best performance with an *MAE* of 0.932 and *MSE* of 1.693, along with high *IA* (0.815) and R^2 (0.948), indicating its strong predictive capability. The MLP model closely follows, achieving the lowest *MSE* of 1.685, with the highest *IA* (0.817) and R^2 (0.950), showing superior accuracy in capturing data patterns. On the test dataset, LR and ENCV show the best performance with an *MAE* of 0.946 for both tuned models. On the other hand, MLP shows the worst performance using the maximum find features. Low performance on the test dataset may be explained by the fact that the model requires additional variables to generalize effectively. Adding even one more feature can reduce the performance of the model on test data.

The Pareto front analysis in Figure 20 shows that RF and MLP benefit significantly from adding features consistently, especially from one to four, after which gains plateau. In contrast, LR, ENCV, and DT show less sensitivity to feature increases. However, in the test phase, LR and ENCV show the best performance using five features. It can be explained that the power lag features are linearly connected to the target feature.

Figure 21 reveals that common features like "Plag24", "Plag48", "Plag72", and "hsin" are critical for accurate power predictions. XGBoost and MLP achieve high performance with fewer features, while LR and ENCV require more inputs for lower accuracy on validation data. Overall, XGBoost and MLP emerge as the most effective models, while RF also performs well in trend prediction. DT struggles, suggesting it is less suited for power and time-based tasks. The analysis highlights the importance of selecting key features, with most models benefiting from starting from up to three features.

5.2.4. Detailed Analysis of the Tuned ML Model on Hourly Level and All-Features Scenario

The visualizations provide a detailed analysis of the MLP model's performance in predicting hourly power consumption for a private house. In Figure 22, the predicted values closely follow the actual consumption trends, capturing seasonal patterns and fluctuations. However, there are some discrepancies, especially during winter peaks, likely due to unaccounted factors like random family behavior.

Additional plots provide further insights, as shown in Figure 23. The scatter plot shows a strong correlation between predicted and actual values; however, some dispersion is observed at higher consumption levels, indicating areas of reduced accuracy. The residuals plot displays errors that are randomly distributed around the horizontal axis, suggesting that the model is well-calibrated and exhibits no systematic bias. However, outliers occur during periods of higher consumption. The same situation with unsystematic outliers can be seen at the daily level, as we described in and show in Figure 14. The histogram of residuals, Figure 23, shows most errors are small and centered around zero, with a slight leftward skew, confirming the model's overall accuracy.

While the tuned model performs well in capturing power consumption trends, occasional discrepancies and outliers highlight the impact of unpredictable human behavior, such as vacations or changes in household routines. These factors are inherently difficult to predict and should be considered when interpreting the model's forecasts.

5.2.5. GA-SHADE-MO in Comparison with Random Search, Hourly Level

The provided Figure 24 offers valuable insights into the performance of the GA-SHADE-MO and Random Search algorithms when tuning MLP models across varying numbers of features. Notably, GA-SHADE-MO consistently outperforms Random Search, as indicated by its generally lower *MAE* values on both the validation and test datasets using fewer features. This advantage is especially evident in the validation data, where GA-SHADE-MO achieves significantly lower *MAE* with a smaller set of features, illustrating its efficiency in model configuration. Upon reaching eight features through Random Search, the performance of both methods on the test data is quite similar. However, GA-SHADE-MO finds a solution with just five features. Both methods show a pronounced reduction in *MAE* as the number of features increases from one to around five, after which the performance improvement plateaus. This trend suggests that a feature set consisting of five features is sufficient for achieving optimal performance, highlighting the potential for feature reduction without a loss in model efficacy. The test results reinforce the validation outcomes, with GA-SHADE-MO maintaining lower *MAE* values than Random Search across most feature counts. Moreover, the close alignment of *MAE* values between the validation and test scenarios underscores the models' robust generalization capabilities, as both algorithms produce similar performance trends on unseen data.

Overall, the GA-SHADE-MO not only demonstrates a consistent edge over Random Search in feature utilization but also confirms the feasibility of using fewer features to achieve low *MAE*. These findings strongly support the use of GA-SHADE-MO for efficient MLP model tuning, particularly when aiming to streamline feature sets without compromising predictive accuracy. This analysis is critical for guiding future research and the application of optimization methods in similar settings.

5.3. Discussion on the Role of Correlation in Feature Selection vs. GA-SHADE-MO

In this subsection, a comparative analysis of the results based on the correlation matrix, Figure 5, and feature selection, Figures 8 and 17, performed using the proposed GA-SHADE-MO algorithm for forecasting power consumption using all features from the dataset was performed.

From the correlation matrices, Figure 5, at the daily and hourly levels, it can be seen that features such as temperature lags and previous values of power consumption lags show high correlations with the target variable "Power". In addition, it is clearly visible that some meteorological parameters, such as "Pr", "Rel", and "Cl", show a low correlation with Power. This suggests that their direct linear impact on energy consumption is minimal, which may make these features less meaningful when using linear forecasting methods.

The optimization using GA-SHADE-MO, or other heuristic-based algorithms, does not consider the correlations between features, but focuses solely on optimizing forecasting accuracy and the number of features in possible solutions. As a result, the algorithm selected feature sets that allowed each selected ML model to achieve high performance. Based on the feature selections provided for different models, Figures 8 and 17, several of the following key points can be noted. The key features selected simultaneously by the proposed GA-SHADE-MO algorithm include "Temp", "Plag1", "Plag2" for daily level and "Temp", "Plag24", "Plag48" for hourly level, comparable with the observations from the correlation analysis. These variables were selected in almost all models, confirming their critical importance for the "Power" forecasting task. As we can see from the study [65], related to forecasting and analyzing the impact of lag features. Based on their numerical results it is clearly seen that the use of feature lags significantly increases the efficiency of ML algorithms in forecasting time series, as it allows one to capture time dependencies and identify data patterns. Lags allow the model to consider the influence of previous states on the current one, which improves the accuracy of predictions and allows one to solve forecasting problems more effectively even in the presence of nonlinear dependencies.

Nonlinear features and time-harmonic components, such as "hsin", "hcos" for the hourly level have been also found to be useful in some models. These features, which do not have a strong linear correlation with "Power" (Figure 5), helped improve the forecast accuracy by highlighting the presence of nonlinear or seasonal effects that were not obvious from the correlation matrices. Based on extensive research [66], the authors have shown that time features are frequently used in energy consumption forecasting tasks.

Regarding selecting features based on the correlation matrix, GA-SHADE-MO demonstrated that the found feature sets vary across models. For instance, RF and XGBoost models often selected all available lags and time components, while Linear Regressions (LR) had a more limited feature selection. This highlights the flexibility of the proposed approach, which allows features to be selected based on the specific characteristics of the selected ML model, which directly affects the error in the validation sample.

A comparison of correlation analysis and the evolutionary algorithm approach highlights the importance of using methods that can reveal nonlinear and hidden dependencies between features through numerical experiments. Correlation matrices, although providing useful information about linear relationships between features, do not capture complex multivariate dependencies. GA-SHADE-MO focused on improving forecasting quality is able to identify such dependencies, providing a more accurate feature selection.

The obtained results confirm that evolutionary algorithms can significantly improve the feature selection process for time series forecasting problems, such as forecasting power consumption. The proposed approach not only allows identifying features with high significance, but also adapting their set to specific models, which leads to an increase in the overall forecast accuracy. This method is especially useful in situations where there are nonlinear effects and seasonal patterns that are not detected using standard correlation analysis.

6. Conclusions

In this paper, we proposed the hybrid GA-SHADE-MO evolutionary algorithm for simultaneously tuning the set of hyperparameters and features of an ML model. The comprehensive analysis of the ML models optimized using the proposed GA-SHADE-MO algorithm across different scenarios—daily and hourly levels with various feature combinations—provides significant insights into the models' predictive capabilities and robustness. Accurate forecasting is critical for managing energy demand, improving grid stability, and more effectively integrating renewable energy sources. By providing a method that enhances the accuracy of predictive models, our approach enables energy providers to make more informed decisions regarding energy distribution and consumption. This, in turn, supports broader goals of increasing energy efficiency, reducing costs, and promoting sustainability within energy communities. The methodology we propose also offers a scalable solution that can be adapted across different households, regions, and energy systems, ensuring its relevance and utility in diverse energy markets. Based on numerical experiments, this article provides practical insights into which variables should be used for forecasting household energy consumption at daily and hourly intervals.

Across all scenarios, the MLP and XGBoost models consistently outperform in both daily and hourly scenarios. Notably, the MLP model shows the lowest *MAE* and *MSE* in the majority of scenarios, indicating its strong accuracy and reliability. For instance, in the daily all-features scenario, MLP achieves an *MAE* of 6.685 on validation, which is approximately 9.95% lower than the next best model, XGBoost (with an *MAE* of 7.424). On the test data, MLP's performance remains robust, with an *MAE* of 7.527, about 7.09% lowe^{r} than XGBoost's *MAE* of 8.101. In the hourly all-features scenario, MLP again outperforms other models, with an *MAE* of 0.825 on validation, which is about 1.67% lower than XGBoost (0.839). On the test data, the *MAE* for MLP increases to 0.884, but it remains competitive, with a difference of only 1.02% compared to XGBoost (0.875).

The RF model also shows strong performance in capturing trends, as indicated by its high *IA* values. However, its *MAE* is higher, on average, compared to MLP and XGBoost. However, for instance, in the hourly excluded ambient temperature scenario, RF's *MSE* on the test data is 1.563, which is 2.68% lower than XGBoost's *MAE* of 1.606. Decision Tree (DT) models, while performing reasonably well in some scenarios, exhibit significant variability in their performance, particularly in test scenarios. For example, in the power and time hourly scenario, DT's *MAE* on the test data is 2.004, about 7.63% higher than XGBoost's *MAE* of 1.862, indicating that DT models are less reliable in more complex scenarios involving time lags and power features. Percentage differences in errors across scenarios further highlight these trends.

These findings lead to general recommendations for model usage in daily and hourly scenarios. MLP is generally the best model for daily-level predictions across different feature sets, providing the lowest errors and showing robust performance across both validation and test datasets. XGBoost also performs well, particularly when fewer features are used, making it a viable alternative, especially in scenarios where computational efficiency is a priority. RF models can be considered for their strong trend-predicting capabilities, though they might not match the precision of MLP and XGBoost in absolute terms. MLP remains a strong candidate for hourly predictions, especially when maximum accuracy is required. RF should be considered when the model's ability to capture trends is critical, as it shows good generalization but may require more features to achieve comparable accuracy to MLP and XGBoost.

The high computational complexity of the proposed GA-SHADE-MO approach is due to the need to perform optimization in a MO context and to obtain an estimate of one potential solution, it is necessary to calculate the effectiveness of the model on a validation dataset. At the same time, our approach already includes several measures to reduce computational costs. First, GA-SHADE-MO self-adapts its internal parameters during the optimization process, eliminating the need for manual tuning, which in itself reduces the total number of required runs and thereby optimizes resource use. Second, the algorithm is aimed at optimizing the number of features, which also reduces the dimensionality of datasets, which can significantly speed up training for complex models. Especially if we consider cases where a small number of features are used.

The use of data from only one household may limit how well the findings apply to other situations, as unique household characteristics and environmental factors might affect the results. Collecting data from a wider range of households in different environments could make the findings more relevant and reliable. A larger dataset, including many households, could also help show differences in behavior and external factors, making the study's conclusions more broadly useful.

Interpretability and transparency of the model are important aspects, especially in practical applications for energy management. However, interpretability depends on the type of model. For example, linear regressions and decision trees are usually easier to interpret, as they allow you to clearly see the influence of each parameter on the final result. At the same time, complex models, such as deep neural networks, have high predictive ability but may be less transparent due to their multi-layered structure and large number of parameters. For real-world energy management problems, where it is important not only to predict the outcome but also to explain it, it may be useful to use a combination of models that provides a balance between accuracy and interpretability. We plan to take these aspects into account in further studies to improve their practical applicability.

In future studies, we will extend the proposed GA-SHADE-MO approach to tune both hyperparameters and feature sets of ensemble methods, such as Stacking and Voting, along with their included algorithms.

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

Funding: This work was supported by the Academy of Finland within limits of [350696 The Harvest project, 2022–2026].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We provide only the source code of our proposed GA-SHADE-MO algorithm and the obtained results of the numerical experiments via https://github.com/VakhninAleksei/GA-SHADE-MO (accessed 30 September 2024). We are not able to upload the original dataset because the previous owner of the house did not give consent to the publication of the data.

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

Abbreviations

AI	Artificial intelligence
ANN	Artificial neural network
ARIMA	AutoRegressive Integrated Moving Average
CNN	Convolutional Neural Network
ConvLSTM	Convolutional Long Short-Term Memory
CR	Crossover rate
DB-Net	Hybrid network model by incorporating a dilated convolutional neural network
DE	Differential Evolution
DF-CNNLSTM	Domain fusion of Convolutional Neural Networks and Long Short-Term Memory (LSTM) networks
DNN	Deep Neural Network
DT	Decision Tree
EA	Evolutionary Algorithm
ENCV	ElasticNetCV
EPC-PM	Ensemble learning based power consumption prediction model
F	Scale factor
FA	Factor Analysis
FCM–BP	Fuzzy C-Mean clustering BP Neural Network
GA	Genetic Algorithm
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
GA-SHADE-MO	The hybrid evolutionary-based multi-objective algorithm, combined SHADE and GA
H	Historical Memory
HVAC	Heating, Ventilation, and Air Conditioning
IA	Index of Agreement
IBEA	The Indicator-Based Evolutionary Algorithm
LR	Linear Regression
LSTM	Long short-term memory
MAE	Mean absolute error
ML	Machine learning
MLP	Multi-layer perceptron
MO	Multi-objective
MOEA/D	The Multi-Objective Evolutionary Algorithm based on Decomposition
MOGAs	Multi-Objective Genetic Algorithms
MRA-ANN	Multiple Regression Analysis-Artificial Neural Network
MSE	Mean square error
NPGA	The Niched Pareto Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
PCA	principal component analysis
PSF	Pattern Sequence Forecasting
PSO	Particle Swarm Optimization
R^2	Coefficient of determination
RF	Random Forest
RNN	Recurrent Neural Network
RobustSTL	A robust seasonal-trend decomposition algorithm for long time series
SHADE	Success-history-based parameter adaptation for differential evolution
SVR	Support vector regression
TCN	Temporal convolutional network
TL-MCLSTM	Deep model named multi-channel long short-term memory with time location genetic algorithms
VEGA	Vector Evaluated Genetic Algorithm
XGBoost	Extreme Gradient Boosting

Appendix A

Table A1. Regression models and their hyperparameters to be tuned.

Regression Model	Hyperparameters	Value Ranges	Type
Linear Regression	None	None	None
ElasticNetCV	l1_ratio (ratio of L1 regularization, controls the balance between L1 and L2 regularization)	[0.0; 1.0]	Real
Decision Tree	max_depth (maximum depth of the tree)	[2; 20]	Integer
	min_samples_split (minimum number of samples required to split a node)	[2; 20]	Integer
	min_samples_leaf (minimum number of samples required in a leaf node)	[2; 20]	Integer
Random Forest	n_estimators (number of trees in the forest)	[1; 300]	Integer
	max_depth (maximum depth of the trees)	[2; 20]	Integer
	min_samples_split (minimum number of samples required to split a node)	[2; 20]	Integer
	min_samples_leaf (minimum number of samples required in a leaf node)	[2; 20]	Integer
Multi-layer perceptron	hidden_layers (number of hidden layers)	[1; 5]	Integer
	hidden_layer_sizes (size of each hidden layer)	[2; 50]	Integer
	batch_size (size of the mini-batch for training)	[1; 200]	Integer
XGBoost	colsample_bytree (fraction of features to be selected for each tree)	[0.001; 1.0]	Real
	learning_rate (learning rate, controls the weight updates)	[0.001; 1.0]	Real
	max_depth (maximum depth of the decision tree)	[1; 20]	Integer
	alpha (L1 regularization on weights)	[1; 10]	Integer
	n_estimators (number of trees in the ensemble)	[1; 300]	Integer

References

1. Sharma, M.; Mittal, N.; Mishra, A.; Gupta, A. Survey of electricity demand forecasting and demand side management techniques in different sectors to identify scope for improvement. *Smart Grids Sustain. Energy* **2023**, *8*, 9. [CrossRef]
2. Barthelmie, R.J.; Murray, F.; Pryor, S.C. The economic benefit of short-term forecasting for wind energy in the UK electricity market. *Energy Policy* **2008**, *36*, 1687–1696. [CrossRef]
3. Cicceri, G.; Tricomi, G.; D'Agati, L.; Longo, F.; Merlino, G.; Puliafito, A. A Deep Learning-Driven Self-Conscious Distributed Cyber-Physical System for Renewable Energy Communities. *Sensors* **2023**, *23*, 4549. [CrossRef] [PubMed]
4. Karaman, Ö.A. Prediction of Wind Power with Machine Learning Models. *Appl. Sci.* **2023**, *13*, 11455. [CrossRef]
5. Wei, N.; Li, C.; Peng, X.; Zeng, F.; Lu, X. Conventional models and artificial intelligence-based models for energy consumption forecasting: A review. *J. Pet. Sci. Eng.* **2019**, *181*, 106187. [CrossRef]
6. Huang, H.; Jia, R.; Shi, X.; Liang, J.; Dang, J. Feature selection and hyper parameters optimization for short-term wind power forecast. *Appl. Intell.* **2021**, *51*, 6752–6770. [CrossRef]
7. Vakhnin, A.; Ryzhikov, I.; Brester, C.; Niska, H.; Kolehmainen, M. Weather-Based Prediction of Power Consumption in District Heating Network: Case Study in Finland. *Energies* **2024**, *17*, 2840. [CrossRef]
8. Moletsane, P.P.; Motlhamme, T.J.; Malekian, R.; Bogatmoska, D.C. Linear regression analysis of energy consumption data for smart homes. In Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 395–399.
9. Tso, G.K.; Yau, K.K. Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy* **2007**, *32*, 1761–1768. [CrossRef]
10. Vinagre, E.; Pinto, T.; Ramos, S.; Vale, Z.; Corchado, J.M. Electrical energy consumption forecast using support vector machines. In Proceedings of the 2016 27th International Workshop on Database and Expert Systems Applications (DEXA), Porto, Portugal, 5–8 September 2016; pp. 171–175.
11. Azadeh, A.; Ghaderi, S.F.; Sohrabkhani, S. Annual electricity consumption forecasting by neural network in high energy consuming industrial sectors. *Energy Convers. Manag.* **2008**, *49*, 2272–2278. [CrossRef]
12. Salam, A.; El Hibaoui, A. Comparison of machine learning algorithms for the power consumption prediction:-case study of tetouan city–. In Proceedings of the 2018 6th International Renewable and Sustainable Energy Conference (IRSEC), Rabat, Morocco, 5–8 December 2018; pp. 1–5.
13. Reddy, S.; Akashdeep, S.; Harshvardhan, R.; Kamath, S. Stacking Deep learning and Machine learning models for short-term energy consumption forecasting. *Adv. Eng. Inform.* **2022**, *52*, 101542.
14. Sultana, N.; Hossain, S.Z.; Almuhaini, S.H.; Düştegör, D. Bayesian optimization algorithm-based statistical and machine learning approaches for forecasting short-term electricity demand. *Energies* **2022**, *15*, 3425. [CrossRef]
15. Li, K.; Hu, C.; Liu, G.; Xue, W. Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis. *Energy Build.* **2015**, *108*, 106–113. [CrossRef]

16. Li, J.; Chen, H.; Yang, J.; Liu, S.; Nie, Y.; Li, J. Power Consumption Forecast Based on Ridge Regression Model. In Proceedings of the 5th International Conference on Information Technologies and Electrical Engineering, Changsha, China, 4–6 November 2022; pp. 297–302.
17. Musleh, D.A.; Al Metrik, M.A. Machine Learning and Bagging to Predict Midterm Electricity Consumption in Saudi Arabia. *Appl. Syst. Innov.* **2023**, *6*, 65. [CrossRef]
18. Zhou, J.; Wang, Q.; Khajavi, H.; Rastgoo, A. Sensitivity analysis and comparative assessment of novel hybridized boosting method for forecasting the power consumption. *Expert Syst. Appl.* **2024**, *249*, 123631. [CrossRef]
19. Divina, F.; Gilson, A.; Goméz-Vela, F.; García Torres, M.; Torres, J.F. Stacking ensemble learning for short-term electricity consumption forecasting. *Energies* **2018**, *11*, 949. [CrossRef]
20. Chi, D. Research on electricity consumption forecasting model based on wavelet transform and multi-layer LSTM model. *Energy Rep.* **2022**, *8*, 220–228. [CrossRef]
21. Bian, H.; Zhong, Y.; Sun, J.; Shi, F. Study on power consumption load forecast based on K-means clustering and FCM–BP model. *Energy Rep.* **2020**, *6*, 693–700. [CrossRef]
22. Eynard, J.; Grieu, S.; Polit, M. Wavelet-based multi-resolution analysis and artificial neural networks for forecasting temperature and thermal power consumption. *Eng. Appl. Artif. Intell.* **2011**, *24*, 501–516. [CrossRef]
23. Lin, C.H.; Nuha, U.; Lin, G.Z.; Lee, T.F. Hourly power consumption forecasting using robuststl and tcn. *Appl. Sci.* **2022**, *12*, 4331. [CrossRef]
24. Khan, N.; Haq, I.U.; Ullah, F.U.M.; Khan, S.U.; Lee, M.Y. CL-net: ConvLSTM-based hybrid architecture for batteries' state of health and power consumption forecasting. *Mathematics* **2021**, *9*, 3326. [CrossRef]
25. Khan, N.; Haq, I.U.; Khan, S.U.; Rho, S.; Lee, M.Y.; Baik, S.W. DB-Net: A novel dilated CNN based multi-step forecasting model for power consumption in integrated local energy systems. *Int. J. Electr. Power Energy Syst.* **2021**, *133*, 107023. [CrossRef]
26. Peña-Guzmán, C.; Rey, J. Forecasting residential electric power consumption for Bogotá Colombia using regression models. *Energy Rep.* **2020**, *6*, 561–566. [CrossRef]
27. Son, N. Comparison of the deep learning performance for short-term power load forecasting. *Sustainability* **2021**, *13*, 12493. [CrossRef]
28. Kumar, J.; Gupta, R.; Saxena, D.; Singh, A.K. Power consumption forecast model using ensemble learning for smart grid. *J. Supercomput.* **2023**, *79*, 11007–11028. [CrossRef]
29. Yan, K.; Wang, X.; Du, Y.; Jin, N.; Huang, H.; Zhou, H. Multi-step short-term power consumption forecasting with a hybrid deep learning strategy. *Energies* **2018**, *11*, 3089. [CrossRef]
30. Moon, J.; Park, J.; Hwang, E.; Jun, S. Forecasting power consumption for higher educational institutions based on machine learning. *J. Supercomput.* **2018**, *74*, 3778–3800. [CrossRef]
31. Gomez-Quiles, C.; Asencio-Cortes, G.; Gastalver-Rubio, A.; Martinez-Alvarez, F.; Troncoso, A.; Manresa, J.; Riquelme, J.C.; Riquelme-Santos, J.M. A novel ensemble method for electric vehicle power consumption forecasting: Application to the Spanish system. *IEEE Access* **2019**, *7*, 120840–120856. [CrossRef]
32. Shao, X.; Pu, C.; Zhang, Y.; Kim, C.S. Domain fusion CNN-LSTM for short-term power consumption forecasting. *IEEE Access* **2020**, *8*, 188352–188362. [CrossRef]
33. Shao, X.; Kim, C.S. Multi-step short-term power consumption forecasting using multi-channel LSTM with time location considering customer behavior. *IEEE Access* **2020**, *8*, 125263–125273. [CrossRef]
34. Nagy, M.; Mansour, Y.; Abdelmohsen, S. Multi-objective optimization methods as a decision making strategy. *Int. J. Eng. Res. Technol.* **2020**, *9*, 516–522.
35. Zadeh, L. Optimality and non-scalar-valued performance criteria. *IEEE Trans. Autom. Control* **1963**, *8*, 59–60. [CrossRef]
36. Seo, T.; Asakura, Y. Multi-objective linear optimization problem for strategic planning of shared autonomous vehicle operation and infrastructure design. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 3816–3828. [CrossRef]
37. Mohseny-Tonekabony, N.; Sadjadi, S.J.; Mohammadi, E.; Tamiz, M.; Jones, D.F. Robust, extended goal programming with uncertainty sets: An application to a multi-objective portfolio selection problem leveraging DEA. *Ann. Oper. Res.* **2024**, 1–56. [CrossRef]
38. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1975.
39. Schaffer, J.D. Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms. Ph.D. Thesis, Vanderbilt University, Nashville, TN, USA, 1985.
40. Fonseca, C.M.; Fleming, P.J. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. *Icga* **1993**, *93*, 416–423.
41. Horn, J.; Nafpliotis, N.; Goldberg, D.E. A niched Pareto genetic algorithm for multiobjective optimization. In Proceedings of the First IEEE Conference on Evolutionary Computation, Orlando, FL, USA, 27–29 June 1994; pp. 82–87.
42. Srinivas, N.; Deb, K. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **1994**, *2*, 221–248. [CrossRef]
43. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.A.M.T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]

44. Coello, C.A.C.; Pulido, G.T.; Lechuga, M.S. Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 256–279. [CrossRef]
45. Purshouse, R.C.; Deb, K.; Mansor, M.M.; Mostaghim, S.; Wang, R. A review of hybrid evolutionary multiple criteria decision making methods. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation, Beijing, China, 6–11 July 2014; pp. 1147–1154.
46. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
47. Falcón-Cardona, J.G.; Coello, C.A.C. Indicator-based multi-objective evolutionary algorithms: A comprehensive survey. *ACM Comput. Surv.* **2020**, *53*, 29. [CrossRef]
48. Gunantara, N. A review of multi-objective optimization: Methods and its applications. *Cogent Eng.* **2018**, *5*, 1502242. [CrossRef]
49. Pereira, J.L.J.; Oliver, G.A.; Francisco, M.B.; Cunha Jr, S.S.; Gomes, G.F. A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems. *Arch. Comput. Methods Eng.* **2022**, *29*, 2285–2308. [CrossRef]
50. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef] [PubMed]
51. Tanabe, R.; Fukunaga, A. Success-history based parameter adaptation for differential evolution. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 71–78.
52. Sangswang, A.; Konghirun, M. Optimal Strategies in Home Energy Management System Integrating Solar Power, Energy Storage, and Vehicle-to-Grid for Grid Support and Energy Efficiency. *IEEE Trans. Ind. Appl.* **2020**, *56*, 5716–5728. [CrossRef]
53. Yuan, X.; Cai, Q.; Deng, S. Power consumption behavior analysis based on cluster analysis. In Proceedings of the International Symposium on Artificial Intelligence and Robotics 2021, Fukuoka, Japan, 21–22 August 2021; Volume 11884, pp. 476–486.
54. de Lemos Martins, T.A.; Faraut, S.; Adolphe, L. Influence of context-sensitive urban and architectural design factors on the energy demand of buildings in Toulouse, France. *Energy Build.* **2019**, *190*, 262–278. [CrossRef]
55. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*; OTexts: Melbourne, Australia, 2018.
56. Cerqueira, V.; Torgo, L.; Mozetič, I. Evaluating time series forecasting models: An empirical study on performance estimation methods. *Mach. Learn.* **2020**, *109*, 1997–2028. [CrossRef]
57. Fumo, N.; Biswas, M.R. Regression analysis for prediction of residential energy consumption. *Renew. Sustain. Energy Rev.* **2015**, *47*, 332–343. [CrossRef]
58. Liu, W.; Dou, Z.; Wang, W.; Liu, Y.; Zou, H.; Zhang, B.; Hou, S. Short-term load forecasting based on elastic net improved GMDH and difference degree weighting optimization. *Appl. Sci.* **2018**, *8*, 1603. [CrossRef]
59. Cody, C.; Ford, V.; Siraj, A. Decision tree learning for fraud detection in consumer energy consumption. In Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications, Miami, FL, USA, 9–11 December 2015; pp. 1175–1179.
60. Zogaan, W.A. Power Consumption prediction using Random Forest model. *Int. J. Mech. Eng.* **2022**, *7*, 329–341.
61. Wahid, F.; Kim, D.H. Short-term energy consumption prediction in Korean residential buildings using optimized multi-layer perceptron. *Kuwait J. Sci.* **2017**, *44*, 67–77.
62. 62. Abbasi, R.A.; Javaid, N.; Ghuman, M.N.J.; Khan, Z.A.; Ur Rehman, S.; Amanullah. Short term load forecasting using XGBoost. In *Web, Artificial Intelligence and Network Applications, Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications, Matsue, Japan, 27–29 March 2019*; Springer: Cham, Switzerland, 2019; pp. 1120–1131.
63. Tran, M.K.; Panchal, S.; Chauhan, V.; Brahmbhatt, N.; Mevawalla, A.; Fraser, R.; Fowler, M. Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery. *Int. J. Energy Res.* **2022**, *46*, 786–794. [CrossRef]
64. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
65. Surakhi, O.; Zaidan, M.A.; Fung, P.L.; Hossein Motlagh, N.; Serhan, S.; AlKhanafseh, M.; Ghoniem, R.M.; Hussein, T. Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm. *Electronics* **2021**, *10*, 2518. [CrossRef]
66. Mystakidis, A.; Koukaras, P.; Tsalikidis, N.; Ioannidis, D.; Tjortjis, C. Energy Forecasting: A Comprehensive Review of Techniques and Technologies. *Energies* **2024**, *17*, 1662. [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.