applied sciences

MDPI

Article

Machine Learning-Based Energy Consumption and Carbon Footprint Forecasting in Urban Rail Transit Systems

Sertaç Savaş [1,*] and Kamber Külahcı [2]

[1] Department of Mechatronics Engineering, Erciyes University, 38039 Kayseri, Türkiye
[2] Mechatronics Engineering, Graduate School of Natural and Applied Sciences, Erciyes University, 38039 Kayseri, Türkiye; kamberkulahcieem@gmail.com
* Correspondence: sertacsavas@erciyes.edu.tr

Abstract

In the fight against global climate change, the transportation sector is of critical importance because it is one of the major causes of total greenhouse gas emissions worldwide. Although urban rail transit systems offer a lower carbon footprint compared to road transportation, accurately forecasting the energy consumption of these systems is vital for sustainable urban planning, energy supply management, and the development of carbon balancing strategies. In this study, forecasting models are designed using five different machine learning (ML) algorithms, and their performances in predicting the energy consumption and carbon footprint of urban rail transit systems are comprehensively compared. For five distribution-center substations, 10 years of monthly energy consumption data and the total carbon footprint data of these substations are used. Support Vector Regression (SVR), Extreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), Adaptive Neuro-Fuzzy Inference System (ANFIS), and Nonlinear Autoregressive Neural Network (NAR-NN) models are developed to forecast these data. Model hyperparameters are optimized using a 20-iteration Random Search algorithm, and the stochastic models are run 10 times with the optimized parameters. Results reveal that the SVR model consistently exhibits the highest forecasting performance across all datasets. For carbon footprint forecasting, the SVR model yields the best results, with an R^2 of 0.942 and a MAPE of 3.51%. The ensemble method XGBoost also demonstrates the second-best performance ($R^2 = 0.648$). Accordingly, while deterministic traditional ML models exhibit superior performance, the neural network-based stochastic models, such as LSTM, ANFIS, and NAR-NN, show insufficient generalization capability under limited data conditions. These findings indicate that, in small- and medium-scale time-series forecasting problems, traditional machine learning methods are more effective than neural network-based methods that require large datasets.

Keywords: energy consumption; carbon footprint forecasting; machine learning; random search; urban rail transit systems

Check for updates

Academic Editor: Ján Dižo

Received: 9 January 2026
Revised: 27 January 2026
Accepted: 27 January 2026
Published: 29 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Due to transportation and industrial activities, the amount of carbon dioxide released into the atmosphere is rapidly increasing. This leads to climate change, global warming, and the disruption of the natural balance. One of the main indicators of these adverse effects is carbon footprint. Carbon footprint refers to the total amount of carbon dioxide (CO_2) emitted into the atmosphere as a result of the activities of individuals, institutions,

or countries. Reducing carbon footprint is crucial in minimizing environmental damage. Using energy-efficient devices, shifting toward renewable energy sources, preferring public transportation, and adopting conscious consumption habits are among the primary methods for reducing carbon footprint. Through these measures, less fossil fuel is consumed, and the amount of CO_2 released into the atmosphere decreases significantly. In addition to reducing carbon footprint, afforestation efforts also play a significant role in offsetting the carbon dioxide that has already been emitted. Trees absorb carbon dioxide and produce oxygen through photosynthesis, thereby acting as a natural carbon sink. Based on the emission amount obtained from carbon footprint calculations, planting a specific number of trees can help compensate for the environmental impact.

For these approaches to be planned effectively, energy consumption and the associated emissions must be predicted reliably. Therefore, in line with global climate change and sustainable development goals, accurately forecasting energy consumption and carbon emissions is of critical importance. The transportation sector accounts for approximately 20–30% of global CO_2 emissions, and roughly three-quarters of these emissions originate from road transport. Urban rail systems generate substantially lower carbon emissions per passenger-kilometer than road-based transportation. In the coming years, many developed countries target a 50% reduction in transport-related emissions, and optimizing the energy efficiency of rail systems is therefore critical to achieving these goals. In Türkiye, the urban rail transit network has expanded significantly over the past decade, further increasing the strategic importance of energy consumption forecasting for planning and operational decision-making. Moreover, recent advances in machine learning and artificial intelligence technologies have enabled revolutionary progress in forecasting energy consumption and carbon emissions. These algorithms have the capacity to model complex and nonlinear relationships that traditional statistical methods fail to capture.

Work on energy management and building/residential electricity forecasting typically focuses on short-to-mid-term demand estimation using time-series data. Qureshi et al. [1] studied a hospital setting using Building Energy Management Systems (BEMS), employing LSTM forecasting, and noted that optimization steps can improve performance. Elhabyb et al. [2] compared ML and deep-learning predictors for educational buildings, emphasizing that tailoring models to building-specific consumption patterns is critical. Morcillo-Jimenez et al. [3] evaluated deep-learning time-series models using real office-building sensor data, reporting that memory-based architectures can outperform stateless approaches, such as an MLP, while limited data constrains more advanced paradigms. Mahjoub et al. [4] compared LSTM, Gated Recurrent Unit (GRU), and Drop-GRU for short-term power forecasting on real data from French cities, reporting that LSTM generally yielded lower errors and could support proactive load management. Ramos et al. [5] compared several deep-learning architectures for residential load forecasting and also explored a Voting Ensemble to improve accuracy. Chan and Yeo [6] proposed a sparse Transformer approach for load forecasting, arguing that attention-based sequence modeling can achieve similar accuracy with faster inference. Chung and Jang [7] presented a hybrid approach that combines a Convolutional Neural Network (CNN) and an LSTM network for multivariate electricity consumption forecasting, aiming to improve feature extraction and regional forecasting performance. Maleki et al. [8] benchmarked multiple approaches on Internet of Things (IoT) smart-meter data. They reported that deep-learning can be competitive, while Automated Machine Learning (AutoML) can improve generalization when more data are available.

A second line of work enhances forecasting robustness by incorporating signal processing, attention, or long-term dependency modeling components. Deng et al. [9] combined a Temporal Convolutional Network (TCN) with an Attention Mechanism, and reported that

feature-reduction steps can further enhance prediction quality. Frikha et al. [10] employed the Stationary Wavelet Transform (SWT) as a denoising step before deep-learning forecasting, reporting improved accuracy compared to using deep learning alone. Some studies prioritize transparency and interpretability alongside accuracy, especially for decision support in buildings.

Gorzałczany and Rudziński [11] proposed Fuzzy Rule-Based Prediction Systems (FRBPS) for residential energy consumption and optimized the accuracy–interpretability trade-off via evolutionary optimization, highlighting improved transparency. In smart-city settings, forecasting is often paired with anomaly or theft detection to improve operational reliability. Chahardoli et al. [12] proposed a hybrid deep-learning framework for consumption forecasting and theft detection, reporting gains from data balancing and optimization, including the use of a Generative Adversarial Network (GAN) for data balancing.

Carbon emissions forecasting encompasses daily predictions, national trajectories, and assessments of policy alignment. Ajala et al. [13] compared statistical, ML, and deep-learning models for daily CO_2 emissions and highlighted the practical value of ensemble methods. Begum and Mobin [14] projected CO_2 trajectories for major emitters through 2030 and discussed alignment with Nationally Determined Contributions (NDC). Xia et al. [15] addressed industrial emissions forecasting by combining deep learning with factor-oriented analysis to support scenario-based decision-making. Meng et al. [16] forecasted China's energy consumption using SVR and analyzed structural evolution with a Markov Chain. Li et al. [17] proposed a meta-learning transfer approach to mitigate overfitting in industrial CO_2 prediction with limited data. Ghorbal et al. [18] combined deep recurrent modeling with metaheuristic optimization and utilized Principal Component Analysis (PCA) and Blind Source Separation (BSS) for data purification. Sector-specific emissions work—particularly for transportation—often adds interpretability, linking predictions to actionable factors. Alam et al. [19] predicted vehicle CO_2 emissions using deep learning and integrated eXplainable Artificial Intelligence (XAI), using SHapley Additive exPlanations (SHAP) to attribute feature importance.

Urban rail transit systems possess distinctive operational characteristics, including timetable design, passenger demand patterns, network structure, rolling stock performance, and regenerative braking mechanisms. In this domain, Lee et al. [20] developed a skip-stop strategy considering passenger choice behaviors based on smartcard data, while Zhu et al. [21] proposed train plan optimization with flexible train composition considering carbon emissions.

In this study, the forecasting performances of five different machine learning algorithms (SVR, XGBoost, LSTM, ANFIS, and NAR-NN) are compared using 10 years of energy consumption data from a rail transit system. Models are developed for the energy consumption of five distribution-center substations and for the total carbon footprint. Hyperparameter optimization is performed via Random Search, and a comprehensive performance evaluation is conducted using different metrics. The study highlights the advantages of conventional machine learning methods for small- and medium-scale time-series datasets.

2. Materials and Methods

This section describes the experimental framework employed in this study. It begins with a detailed description of the dataset and the preprocessing steps applied, including missing value imputation using the Seasonal K-NN method. Subsequently, the theoretical foundations of the five machine learning algorithms (SVR, XGBoost, LSTM, ANFIS, and NAR-NN) are presented. The section also explains the hyperparameter tuning process via Random Search and concludes with the definitions of the performance metrics used for model evaluation.

2.1. Dataset Description and Preprocessing

The dataset used in this study is obtained from Kayseri Ulaşım A.Ş., which operates in the city of Kayseri, Türkiye. The dataset consists of monthly energy consumption values from five distribution-center substations. These distribution centers feed the rail system substations and provide energy to the tram lines. The dataset spans a 10-year period from January 2016 to November 2025, comprising a total of 119 monthly observations. In addition, the total carbon footprint of these five substations is obtained for an 83-month period from January 2019 to November 2025. In recent years, the integration of renewable energy sources—such as solar panels and wind turbines (WPP—wind power plant)—in the city has enabled substantial progress in supplying energy for rail transit. Carbon footprint is calculated by multiplying the total energy consumption of the five substations by the grid emission factor for the corresponding year. The emission factor has varied across years, and official data or specific emission values are not published for the period prior to 2018. In this study, carbon footprint is treated as the primary target variable, since direct carbon emission forecasts are of critical importance for assessing the environmental impact of urban transit systems, carbon budgeting, and sizing carbon-offset projects. Figure 1 schematically illustrates the energy flow in the urban rail transit system and the carbon footprint calculation process. It visualizes the energy transferred from distribution centers to rail system substations and then to trams, along with the resulting carbon emissions.

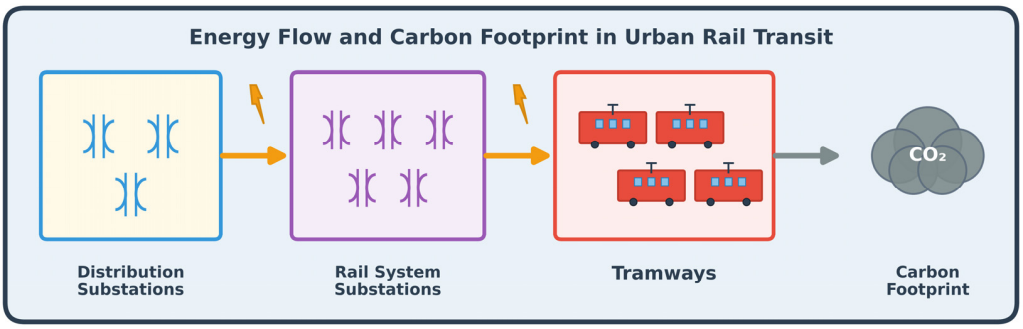

Figure 1. Energy flow and carbon footprint schematic in an urban rail transit system.

The dataset contains missing values arising from outage periods. To complete these missing values, the Seasonal K-NN (Seasonal K-Nearest Neighbors) method is used [22,23]. The K-NN imputation algorithm identifies the nearest k neighbors for an observation with a missing value and estimates the missing entry by taking the average of the corresponding values of these neighbors. Using the Euclidean distance, the distance between two observations is computed as in Equation (1).

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2} \quad (1)$$

where x_{ik} and x_{jk} denote the k-th feature of samples x_i and x_j, respectively, p is the number of features, and $d(\mathrm{x}_i, \mathrm{x}_j)$ is the Euclidean distance. In the Seasonal K-NN method, only observations within the same season or the same month are considered as neighbors. This approach preserves seasonal patterns in energy consumption. To prevent data leakage, the Seasonal K-NN imputation is applied only to the training set portion. The test set portion contains no missing data. Figure 2 presents the train–test split for the five substations and for the carbon footprint. The dataset is divided chronologically into 80% training and 20% testing.

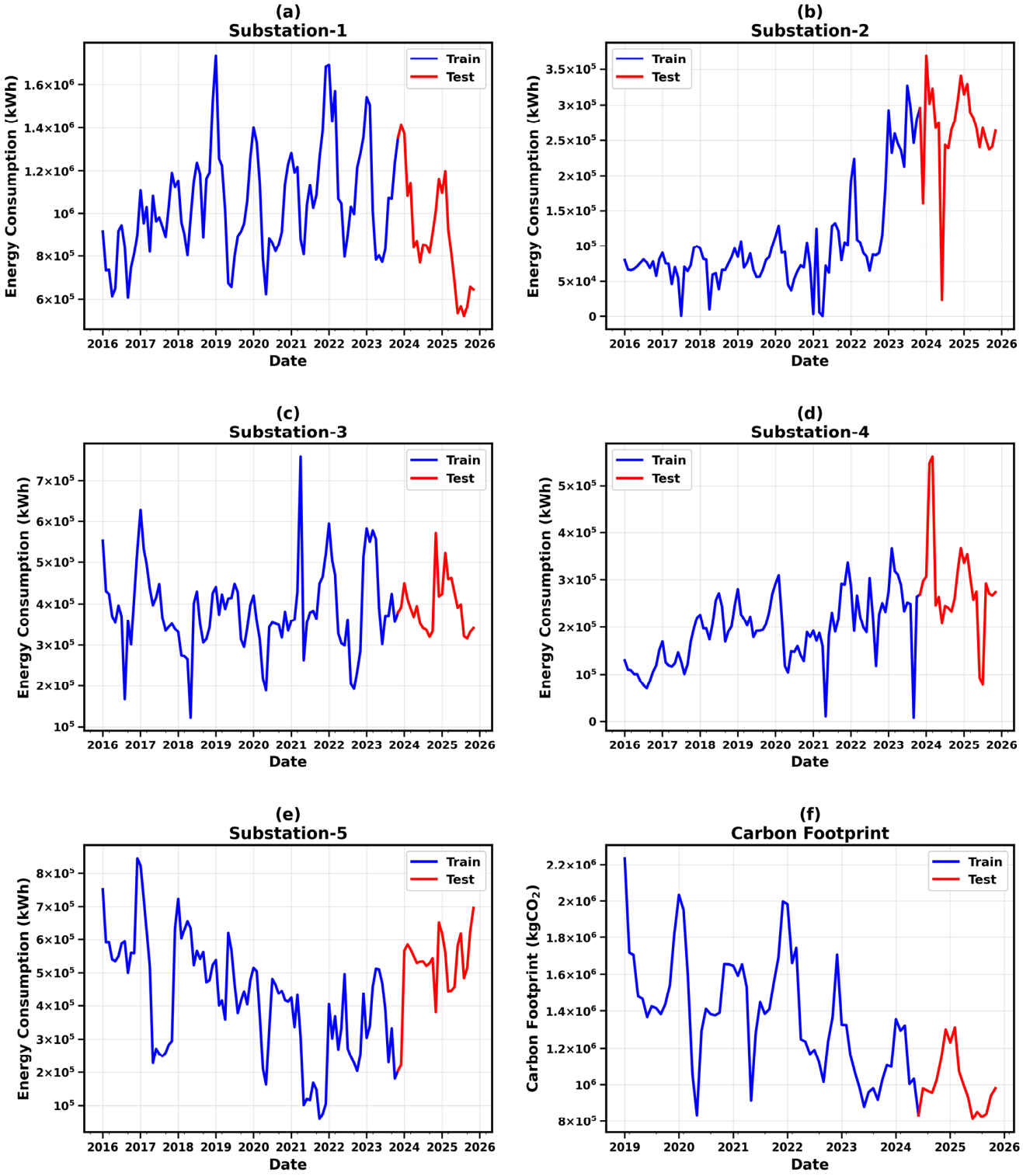

Figure 2. Time series of energy consumption of five substations and carbon footprint, with the train–test split.

As seen in Figure 2, pronounced seasonal patterns are observed across all substations. The carbon footprint data are calculated by multiplying the total energy consumption of the five substations by the emission factor.

2.2. Machine Learning Methods

This subsection presents the theoretical background of the five machine learning algorithms employed in this study. For each model, the underlying principles, mathemati-

cal formulations, and key equations are provided. The methods include two traditional machine learning approaches (SVR and XGBoost) and three neural network-based architectures (LSTM, ANFIS, and NAR-NN), enabling a comprehensive comparison between deterministic and stochastic forecasting models.

2.2.1. Support Vector Regression (SVR)

Support Vector Regression is a regression method developed by Vapnik and based on the principle of structural risk minimization [24]. SVR uses the ε-insensitive loss function, meaning that prediction errors within a specified tolerance are not penalized. The main objective of SVR is to find the optimal function given in Equation (2).

$$f(x) = w^{\top}\varphi(x) + b \tag{2}$$

Here, w denotes the weight vector, $\varphi(x)$ is the mapping function that transforms the input space into a high-dimensional feature space, and b is the bias term. The optimization problem is given in Equation (3).

$$min\left\{\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)\right\} \tag{3}$$

Here, C is the regularization parameter, N is the number of training samples, and ξ_i and ξ_i^* are slack variables. In this study, the Radial Basis Function (RBF) kernel, as defined in Equation (4), is employed.

$$K(x_i, x_j) = exp\left(-\gamma\|x_i - x_j\|^2\right) \tag{4}$$

where γ denotes the kernel width parameter.

2.2.2. Extreme Gradient Boosting (XGBoost)

XGBoost is an ensemble learning algorithm developed by Chen and Guestrin [25]. The algorithm sequentially combines weak learners (decision trees), such that each new tree corrects the errors of the previous model. The objective function is shown in Equation (5).

$$L(\phi) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{5}$$

Here, $l(y_i, \hat{y}_i)$ denotes the loss function, N is the number of training samples, K is the number of trees, and $\Omega(f_k)$ is the regularization term. The regularization term is defined in Equation (6).

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \tag{6}$$

where T is the number of leaf nodes, w_j are the leaf weights, and γ and λ are regularization parameters.

2.2.3. Long Short-Term Memory (LSTM)

LSTM is a specialized recurrent neural network architecture developed by Hochreiter and Schmidhuber to address the vanishing gradient problem [26]. An LSTM cell consists of three gates: the forget gate (f_t), input gate (i_t), and output gate (o_t). The gate equations are given in Equations (7)–(9).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{7}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{8}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{9}$$

Here, σ is the sigmoid function, W are the weight matrices, h is the hidden state, x is the input, and b are the bias vectors. The cell-state and hidden-state updates are given in Equations (10) and (11).

$$C_t = f_t \odot C_{t-1} + i_t \odot tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{10}$$

$$h_t = o_t \odot tanh(C_t) \tag{11}$$

where $\odot$ denotes the Hadamard (element-wise) product. In this study, a two-layer LSTM architecture with dropout regularization is used [27].

2.2.4. Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS is a hybrid model proposed by Jang that combines artificial neural networks and fuzzy logic systems [28]. ANFIS employs a Takagi–Sugeno-type fuzzy inference system and consists of five layers. The membership function is given in Equation (12).

$$\mu_i(x) = \frac{1}{1 + |\frac{x-c_i}{a_i}|^{2b_i}} \tag{12}$$

Here, c_i is the center, a_i is the width, and b_i is the slope parameter. The normalized firing strength and final output are given in Equations (13) and (14).

$$\overline{w}_i = \frac{w_i}{\sum_{j=1}^{M} w_j} \tag{13}$$

$$f = \sum_{i=1}^{M} \overline{w}_i \left(\sum_{k=1}^{d} p_{ik} x_k + r_i \right) \tag{14}$$

In Equation (13), w_i is the firing strength of the rule i, $\overline{w}_i$ is the normalized firing strength, and M is the number of rules. In Equation (14), d is the number of inputs (lookback length), p_{ik} and r_i are the consequent parameters of the rule i, and x_k denotes the k-th input variable. ANFIS optimizes both antecedent and consequent parameters using a hybrid learning algorithm.

2.2.5. Nonlinear Autoregressive Neural Network (NAR-NN)

NAR-NN is a feedforward neural network architecture used for time-series forecasting [29]. The model predicts future values using past observations, and its mathematical representation is given in Equation (15).

$$y_t = F(y_{t-1}, y_{t-2}, \ldots, y_{t-d}) + \varepsilon_t \tag{15}$$

Here, d is the number of delays (lookback), $F(\cdot)$ is the nonlinear mapping function, and ε_t is the error term. In this study, the Adam optimization algorithm [30] and an early stopping strategy are employed.

2.3. Hyperparameter Tuning

For each model, hyperparameter optimization is performed using the Random Search algorithm. Random Search explores the hyperparameter space by drawing random samples to identify an optimal configuration. Bergstra and Bengio [31] showed that this approach can be more efficient than grid search. In this study, 20 iterations are conducted for each

model–series pair, and the best hyperparameters are selected based on the validation RMSE. Hyperparameter optimization employs a temporal hold-out strategy, reserving the last 12 months of the training period for validation RMSE calculation. Although walk-forward cross-validation would be preferable, the limited sample size (119 observations) constrains this approach; this is acknowledged as a limitation. Table 1 presents the hyperparameters and search spaces used for the five models. Optimized hyperparameters for all models across the six time series (five substations and the carbon footprint) are provided in Table A1 in Appendix A.

Table 1. Hyperparameters and the search space for forecasting models.

Model	Parameter	Search Range	Description
SVR	C	[0.1, 100]	Regularization parameter
	ε (epsilon)	[0.01, 0.5]	Epsilon tube width
	γ (gamma)	[0.001, 1.0]	Kernel coefficient
	kernel	{rbf, poly, sigmoid}	Kernel function type
XGBoost	n_estimators	[30, 200]	Number of boosting trees
	max_depth	[2, 5]	Maximum tree depth
	learning_rate	[0.05, 0.3]	Boosting learning rate
	subsample	[0.7, 1.0]	Row sampling ratio
	colsample_bytree	[0.7, 1.0]	Column sampling ratio
	min_child_weight	[3, 10]	Minimum child weight
	gamma	[0.1, 0.5]	Minimum loss reduction
	reg_alpha	[0.1, 1]	L1 regularization
	reg_lambda	[0.1, 1]	L2 regularization
LSTM	units_1	[8, 32]	First-LSTM-layer neurons
	units_2	[0, 16]	Second-LSTM-layer neurons
	dropout	[0.2, 0.5]	Dropout rate
	learning_rate	[0.001, 0.01]	Adam optimizer learning rate
	batch_size	[8, 16]	Training batch size
	lookback	[2, 6]	Input sequence length (months)
ANFIS	n_mfs	[2, 4]	Number of membership functions
	learning_rate	[0.01, 0.1]	Training learning rate
	lookback	[2, 6]	Input sequence length (months)
NAR-NN	hidden_units	[5, 20]	Hidden-layer neurons
	n_layers	[1, 2]	Number of hidden layers
	activation	{relu, tanh, sigmoid}	Activation function
	learning_rate	[0.01, 0.1]	Training learning rate
	lookback	[2, 6]	Input sequence length (months)

When Table 1 is examined, it can be observed that four hyperparameters (C, ε, γ, kernel) are optimized for the SVR model. For XGBoost, nine hyperparameters are optimized; the number of boosting trees (91) and the maximum tree depth (3) indicate that relatively shallow trees are preferred. For the LSTM model, the number of neurons in the first LSTM layer is set to 17, the number of neurons in the second LSTM layer to 4, the dropout rate to 0.4486, and the lookback value (input sequence length) to 4 months. For ANFIS, the optimal number of membership functions is found to be 3. For NAR-NN, 17 hidden-layer neurons, one hidden layer, and the tanh activation function yield the best performance. To capture annual seasonality beyond the lookback window, all models receive cyclically encoded month features (sin/cos transformation). For SVR and XGBoost, lag-12 features (same month of the previous year) are additionally included to explicitly model year-over-year patterns.

2.4. Performance Metrics

The performance metrics used in this study are RMSE (Root Mean Squared Error), which is the square root of the mean squared tracking error; MAE (Mean Absolute Error), which averages the absolute tracking error; MAPE (Mean Absolute Percentage Error), which reports the average absolute percentage deviation; and R^2 (Coefficient of Determination), which quantifies the proportion of variance explained by the predictions. Their mathematical definitions are given in Equations (16)–(19).

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathrm{e}_i\|^2} \tag{16}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} \|\mathrm{e}_i\| \tag{17}$$

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N} \|\frac{\mathrm{e}_i}{\mathrm{r}_i}\| \tag{18}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \|y_i - r_i\|^2}{\sum_{i=1}^{N} \|r_i - \bar{r}\|^2} \tag{19}$$

In the equations, N denotes the number of samples, $e_i = r_i - y_i$ is the error at sample i, r_i is the reference value, y_i is the predicted value, and $\bar{r}$ is the mean of the reference values. $\|\cdot\|$ denotes the Euclidean norm.

3. Results

This section presents the performance evaluation of the forecasting models. Comparative metrics are summarized through tables and heatmaps, followed by actual versus predicted value analyses with graphical illustrations for each substation and the carbon footprint.

In the study, five different machine learning models are designed, and their future forecasting performances are evaluated for the energy consumption of five substations and the total carbon footprint. These forecasting experiments are conducted using the obtained hyperparameters. A single run is performed for deterministic models, such as SVR and XGBoost, whereas 10 independent runs are conducted for neural network-based models (LSTM, ANFIS, and NAR-NN) to better characterize their stochastic behavior. Table 2 presents the performance metrics (RMSE, MAE, MAPE, and R^2) of all models. For stochastic models, the metrics are reported as mean $\pm$ standard deviation, with $[min,\ max]$ values provided in brackets.

When the results in Table 2 are examined, it can be observed that the SVR model shows the highest performance across all datasets. For the five substations datasets, SVR demonstrates consistent superiority by achieving the highest R^2 values at all stations (0.896–0.993). XGBoost performs well for Substation-1 ($R^2 = 0.925$) and Substation-5 ($R^2 = 0.825$), but it remains low for Substation-2 ($R^2 = 0.254$) and Substation-4 ($R^2 = 0.499$). The neural network-based models show poor performance at most stations; in Substation-2 and Substation-5, LSTM, ANFIS, and NAR-NN produce negative R^2 values, which means they perform worse than the mean baseline. The best forecasts occur in Substation-1 and Substation-3, which have regular seasonal patterns, while the most difficult forecasts occur in Substation-2, which has high variance.

Table 2. Comparison of model performance metrics.

Series	Model	RMSE	MAE	MAPE	R^2
Substation-1	SVR	21,599.48	17,662.95	2.33	0.9925
	XGBoost	68,299.08	50,895.30	7.25	0.9254
	LSTM	155,191.60 ± 11,436.16 [138,025.37, 170,070.82]	135,673.58 ± 11,586.49 [114,312.54, 149,836.81]	18.18 ± 1.77 [15.00, 20.33]	0.6126 ± 0.0567 [0.5372, 0.6952]
	ANFIS	209,438.81 ± 63,972.16 [157,263.01, 310,265.45]	176,343.84 ± 56,124.21 [133,731.33, 264,501.84]	24.46 ± 8.00 [17.94, 37.08]	0.2327 ± 0.4840 [−0.5402, 0.6043]
	NAR-NN	157,985.13 ± 28,459.15 [116,653.59, 202,780.03]	135,148.81 ± 24,940.76 [98,305.49, 173,593.29]	17.80 ± 3.98 [11.98, 23.97]	0.5877 ± 0.1463 [0.3421, 0.7823]
Substation-2	SVR	16,865.56	11,560.38	14.30	0.9339
	XGBoost	56,641.67	35,641.17	44.93	0.2543
	LSTM	79,534.98 ± 875.60 [78,241.21, 80,977.70]	52,568.16 ± 1237.46 [50,675.46, 54,225.54]	61.91 ± 2.54 [58.49, 65.29]	−0.4706 ± 0.0324 [−0.5242, −0.4229]
	ANFIS	98,435.09 ± 38,844.81 [73,507.97, 176,563.45]	78,505.43 ± 46,488.22 [44,804.84, 170,444.80]	60.08 ± 6.56 [55.63, 73.25]	−1.6030 ± 2.2854 [−6.2463, −0.2560]
	NAR-NN	91,594.20 ± 8872.72 [78,487.76, 105,433.78]	75,760.94 ± 14,072.37 [50,409.93, 96,917.79]	59.75 ± 1.59 [57.12, 62.56]	−0.9684 ± 0.3815 [−1.5839, −0.4319]
Substation-3	SVR	8262.74	6178.73	1.46	0.9835
	XGBoost	31,913.56	22,434.98	5.43	0.7533
	LSTM	58,542.59 ± 1122.29 [57,410.18, 60,551.14]	37,234.43 ± 318.04 [36,550.83, 37,588.60]	8.66 ± 0.18 [8.36, 8.87]	0.1694 ± 0.0321 [0.1118, 0.2015]
	ANFIS	65,135.65 ± 6644.51 [58,414.13, 82,150.90]	46,170.16 ± 7842.64 [35,946.46, 56,964.77]	10.95 ± 1.97 [8.27, 12.92]	−0.0385 ± 0.2249 [−0.6350, 0.1734]
	NAR-NN	69,564.57 ± 12,615.23 [58,477.52, 102,970.07]	51,671.38 ± 15,398.50 [40,400.17, 96,979.43]	12.66 ± 4.59 [9.28, 26.26]	−0.2109 ± 0.4925 [−1.5687, 0.1716]
Substation-4	SVR	33,346.45	22,771.74	9.43	0.8963
	XGBoost	73,261.88	42,046.45	14.92	0.4993
	LSTM	100,030.24 ± 3147.34 [96,440.70, 106,937.19]	76,155.37 ± 3888.11 [72,626.71, 84,118.12]	31.31 ± 0.80 [30.43, 32.95]	0.0655 ± 0.0598 [−0.0669, 0.1323]
	ANFIS	136,129.45 ± 10,809.75 [117,779.85, 154,636.97]	106,819.06 ± 13,769.74 [82,376.38, 129,388.28]	39.57 ± 3.52 [33.48, 45.44]	−0.7398 ± 0.2747 [−1.2310, −0.2942]
	NAR-NN	97,974.43 ± 7127.85 [90,328.29, 113,004.48]	72,900.80 ± 10,556.17 [61,336.33, 92,456.26]	30.52 ± 2.03 [28.33, 34.31]	0.0997 ± 0.1344 [−0.1914, 0.2388]
Substation-5	SVR	19,454.83	14,990.49	2.67	0.9579
	XGBoost	39,697.80	30,124.83	5.79	0.8249
	LSTM	114,885.11 ± 4519.64 [105,424.16, 121,603.17]	88,280.44 ± 5563.59 [75,818.18, 95,184.05]	16.22 ± 0.91 [14.19, 17.36]	−0.4688 ± 0.1140 [−0.6430, −0.2349]
	ANFIS	118,577.12 ± 22,955.94 [94,469.65, 171,229.28]	88,801.44 ± 26,098.44 [62,359.06, 144,884.41]	16.17 ± 4.46 [11.87, 25.74]	−0.6208 ± 0.6756 [−2.2577, 0.0084]
	NAR-NN	126,552.50 ± 25,121.28 [95,403.72, 175,977.97]	105,205.63 ± 28,547.49 [69,915.27, 161,802.63]	19.80 ± 4.90 [14.08, 30.23]	−0.8496 ± 0.7394 [−2.4409, −0.0113]
Carbon Footprint	SVR	36,453.30	32,957.69	3.51	0.9420
	XGBoost	89,872.04	63,528.38	6.71	0.6478
	LSTM	155,814.77 ± 11,389.00 [136,773.98, 179,909.05]	135,883.70 ± 10,731.47 [118,432.45, 159,547.63]	14.45 ± 1.18 [12.51, 17.01]	−0.0644 ± 0.1568 [−0.4115, 0.1842]
	ANFIS	211,620.99 ± 87,646.75 [131,336.01, 376,235.93]	189,984.87 ± 82,757.54 [114,000.12, 344,417.14]	20.22 ± 8.96 [11.84, 36.94]	−1.2880 ± 1.9896 [−5.1732, 0.2478]
	NAR-NN	130,322.23 ± 24,819.48 [103,474.85, 169,522.05]	112,519.97 ± 23,610.53 [86,281.51, 149,907.69]	11.73 ± 2.68 [8.70, 15.95]	0.2325 ± 0.2961 [−0.2533, 0.5331]

For carbon footprint forecasting, SVR again consistently achieves the best results with RMSE = 36,453.30, MAE = 32,957.69, MAPE = 3.51%, and $R^2 = 0.942$. XGBoost ranks second ($R^2 = 0.648$), while LSTM ($R^2 = -0.064$), ANFIS ($R^2 = -1.288$), and NAR-NN ($R^2 = 0.232$) show inadequate performance.

Figures 3 and 4 present RMSE and R^2 heatmaps, respectively, to visually compare model performances.

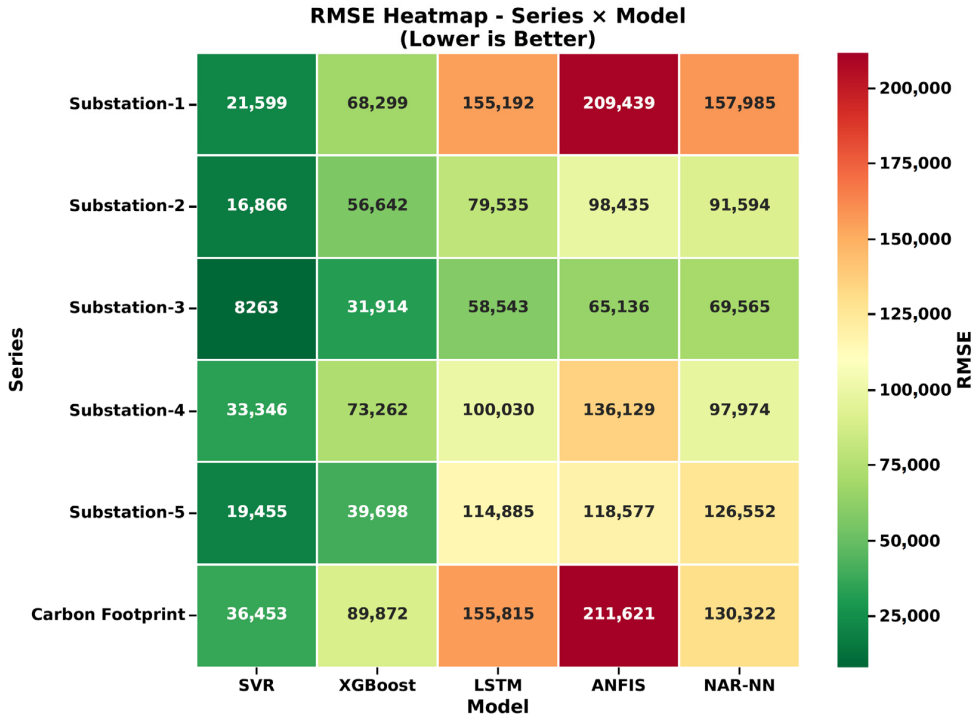

Figure 3. RMSE heatmap by model and time series.

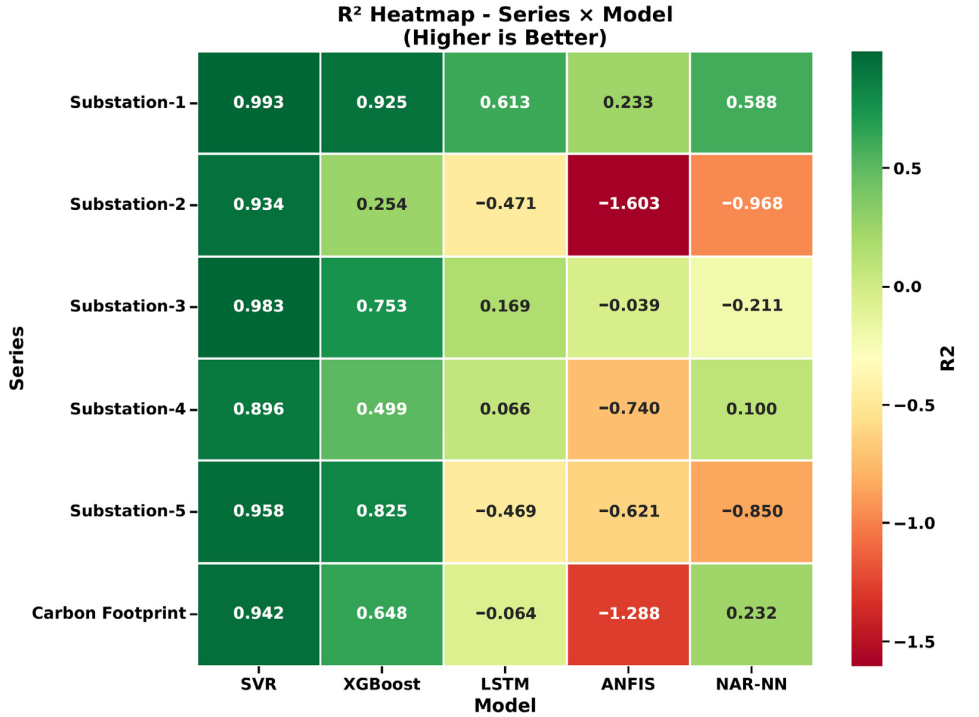

Figure 4. R^2 heatmap by model and time series.

When the RMSE heatmap in Figure 3 is examined, the SVR model shows the lowest RMSE values for all series. In this heatmap, lower RMSE values indicate higher performance. Green tones represent low error, and red tones represent high error; the SVR column remains consistently dark green. The XGBoost column shifts from light green to yellow, which reflects moderate error levels. The LSTM, ANFIS, and NAR-NN columns are concentrated in orange and red tones, which visually indicate large forecasting errors. In the carbon footprint row, SVR and XGBoost also remain in green tones, while the other three models stay in the red region.

The R^2 heatmap in Figure 4 shows the series-wise variation in model performance more clearly. R^2 values close to 1 indicate high performance. The SVR model achieves R^2 values between 0.896 and 0.993 for all series, and shows superior performance with dark-green tones. XGBoost appears in green tones for Substation-1 ($R^2 = 0.925$) and Substation-5 ($R^2 = 0.825$), but it moves to the light-green region for Substation-2 and Substation-4. For the neural network-based models, the heatmap presents a striking picture: the LSTM, ANFIS, and NAR-NN columns are mostly orange, red, and dark red, which reflects near-zero or negative R^2 values. The heatmap represents negative R^2 values with darker-red tones. This visual comparison demonstrates that, with a limited dataset of 119 monthly observations, deterministic models (SVR and XGBoost) outperform neural network-based stochastic models.

Figures 5–10 show, for each substation and for the carbon footprint, the comparison between actual values and model predictions, as well as the time-series plots of prediction errors.

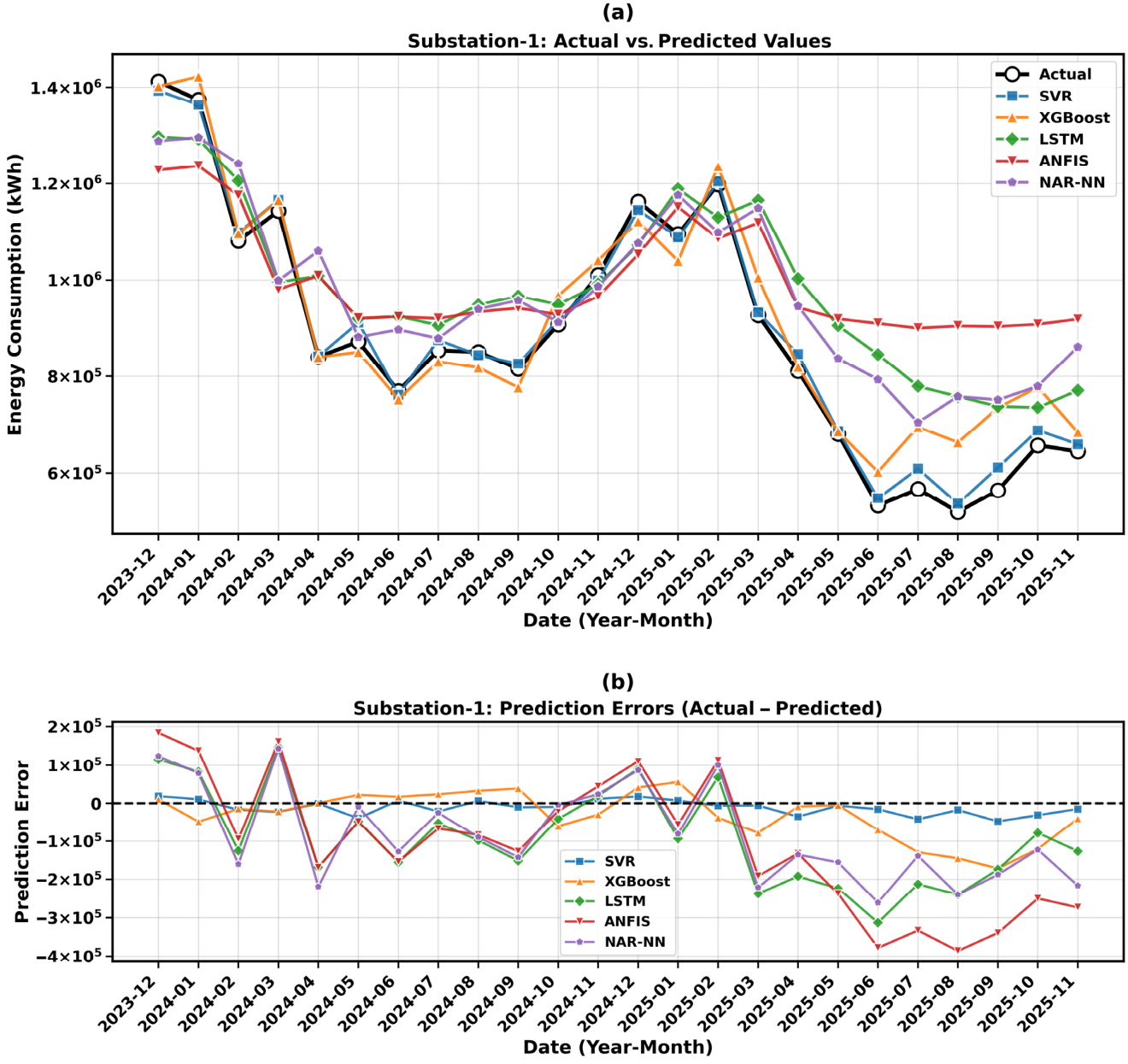

Figure 5. Substation-1: (**a**) Actual and predicted values, (**b**) prediction errors.

When Figure 5 (Substation-1) is examined, the SVR predictions closely follow the actual values. Since this station has a regular seasonal pattern, the XGBoost predictions also remain at an acceptable level. The LSTM and NAR-NN predictions capture the overall trend, but they deviate at peak points. In contrast, ANFIS shows a decline in performance over time, especially after spring 2025. In the error plot, the SVR errors remain well below the ±50,000 kWh range, while ANFIS errors reach up to 400,000 kWh.

Figure 6 (Substation-2) represents the most challenging forecasting scenario. Due to high variance and irregular consumption patterns, only the SVR model accurately tracks the actual values. Although XGBoost predictions remain acceptable in some periods, they fail to capture sudden fluctuations. In contrast, the LSTM, ANFIS, and NAR-NN predictions diverge substantially from the actual values; for ANFIS and NAR-NN in particular, the predictions become almost flat after the fall of 2024. In the error plot, the neural network-based models show systematic errors that exceed ±150,000 kWh.

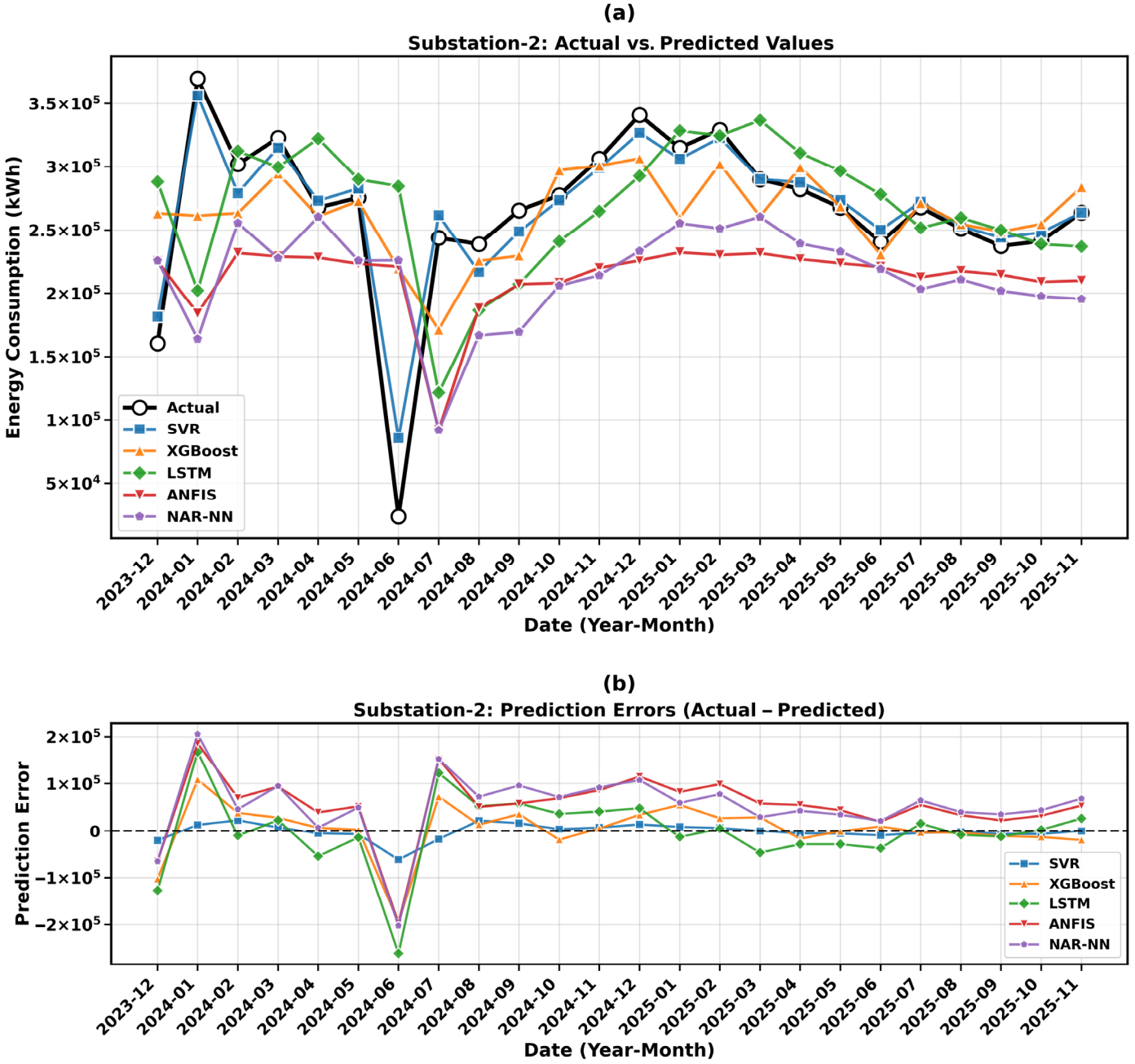

Figure 6. Substation-2: (**a**) Actual and predicted values, (**b**) prediction errors.

In Figure 7, for Substation-3, the SVR model achieves its lowest MAPE value at this station (1.46%), and the prediction curve almost overlaps with the actual values. XGBoost captures the seasonal pattern but underestimates the amplitude. The LSTM predictions follow the general trend, but they remain low during peak periods, while ANFIS and NAR-NN show substantial deviations, especially in high-consumption intervals. In the error plot, SVR errors stay within a narrow band, whereas the other models produce errors reaching up to 200,000 kWh.

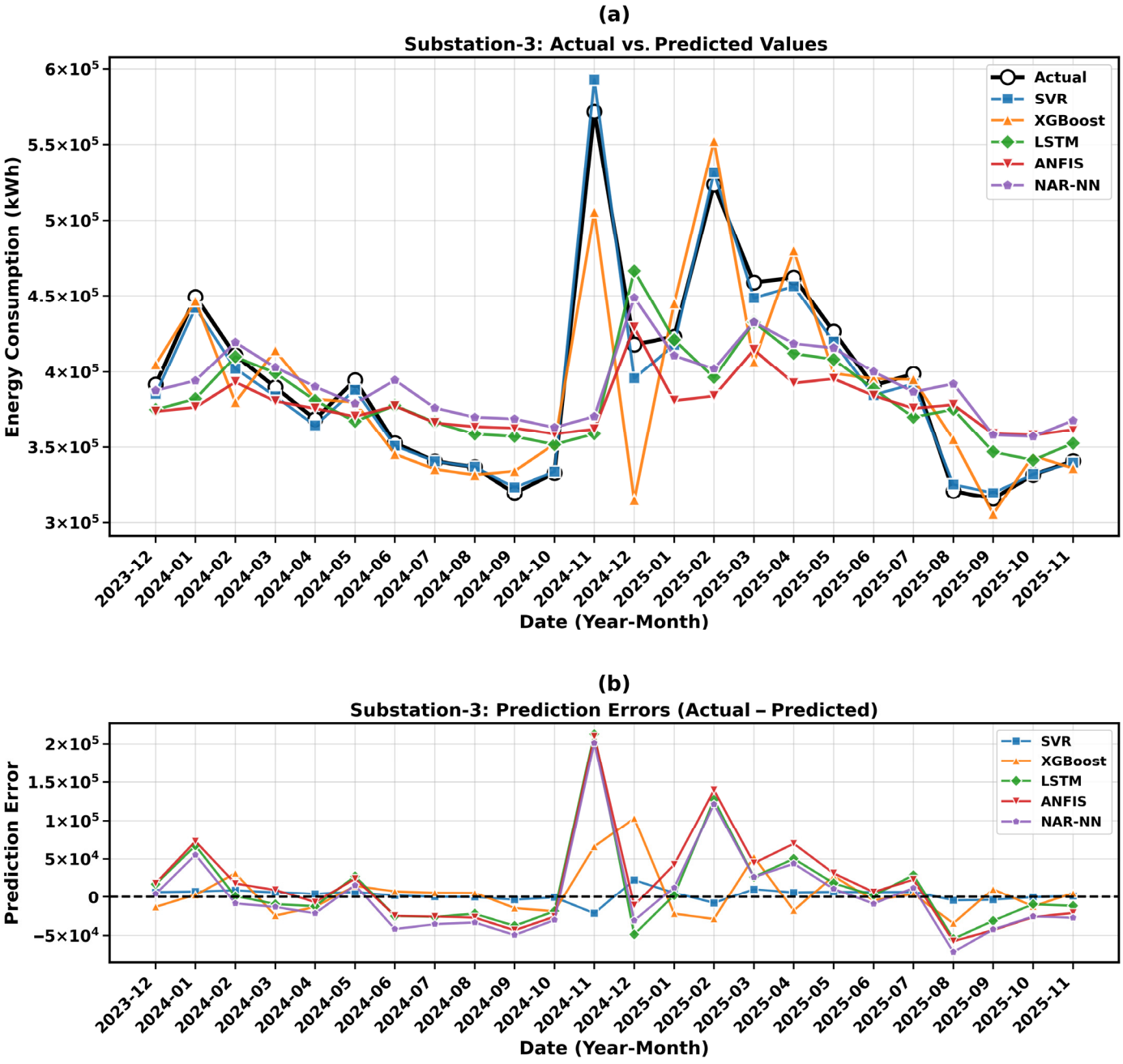

Figure 7. Substation-3: (**a**) Actual and predicted values, (**b**) prediction errors.

When Figure 8 (Substation-4) is examined, it can be observed that the SVR model accurately captures both sudden increases and decreases. XGBoost follows the overall trend but remains inadequate for extreme values. At this station, NAR-NN performs relatively better than LSTM; however, both models show clear deviations at peak points. ANFIS predictions stay far below the actual values and fail to capture seasonal fluctuations.

As is seen in the error graph, ANFIS performs quite poorly, with error values reaching 300,000 kWh.

In Figure 9, for Substation-5, the SVR and XGBoost models successfully capture the seasonal pattern. SVR predictions remain close to the actual values throughout the entire period, while XGBoost shows slight deviations, particularly at the 2025 peaks. In contrast, the LSTM, ANFIS, and NAR-NN models fail altogether at this station, with predictions staying well below the actual values. This behavior clearly corresponds to the negative R^2 values (LSTM: −0.469, ANFIS: −0.621, NAR-NN: −0.850). In the error plot, the neural network-based models show persistent errors because of underestimation of the actual values.

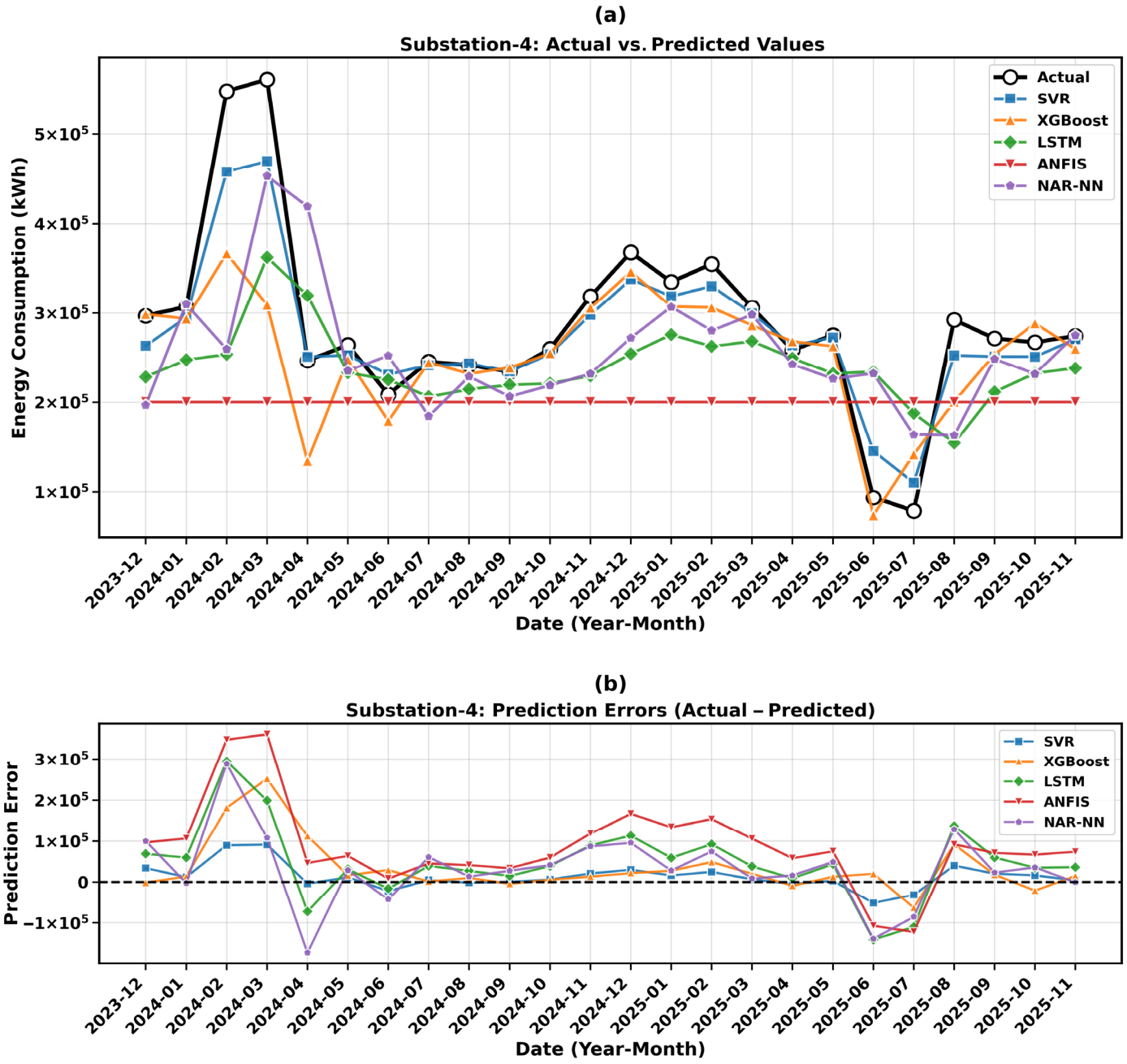

Figure 8. Substation-4: (**a**) Actual and predicted values, (**b**) prediction errors.

Figure 10 shows carbon emission forecasts computed from the total energy consumption of the five substations. The SVR model accurately predicts the seasonal fluctuations in the total carbon footprint, with a MAPE of 3.51%. XGBoost captures the overall trend but

its predictions fluctuate and produce overestimation in some intervals and underestimation in others. NAR-NN shows limited success ($R^2 = 0.232$), whereas LSTM and especially ANFIS show the worst performance, with predictions that diverge substantially from the actual values. In the error plot, SVR errors remain well below 100,000 $kgCO_2$, while ANFIS errors reach 300,000 $kgCO_2$, showing significant deviations.

The limited sample size constrains the robustness of the single train–test split evaluation. Although stochastic model variability is characterized through 10 independent runs, repeated temporal back-testing across multiple folds would provide stronger evidence; this is identified as a direction for future work with expanded datasets.

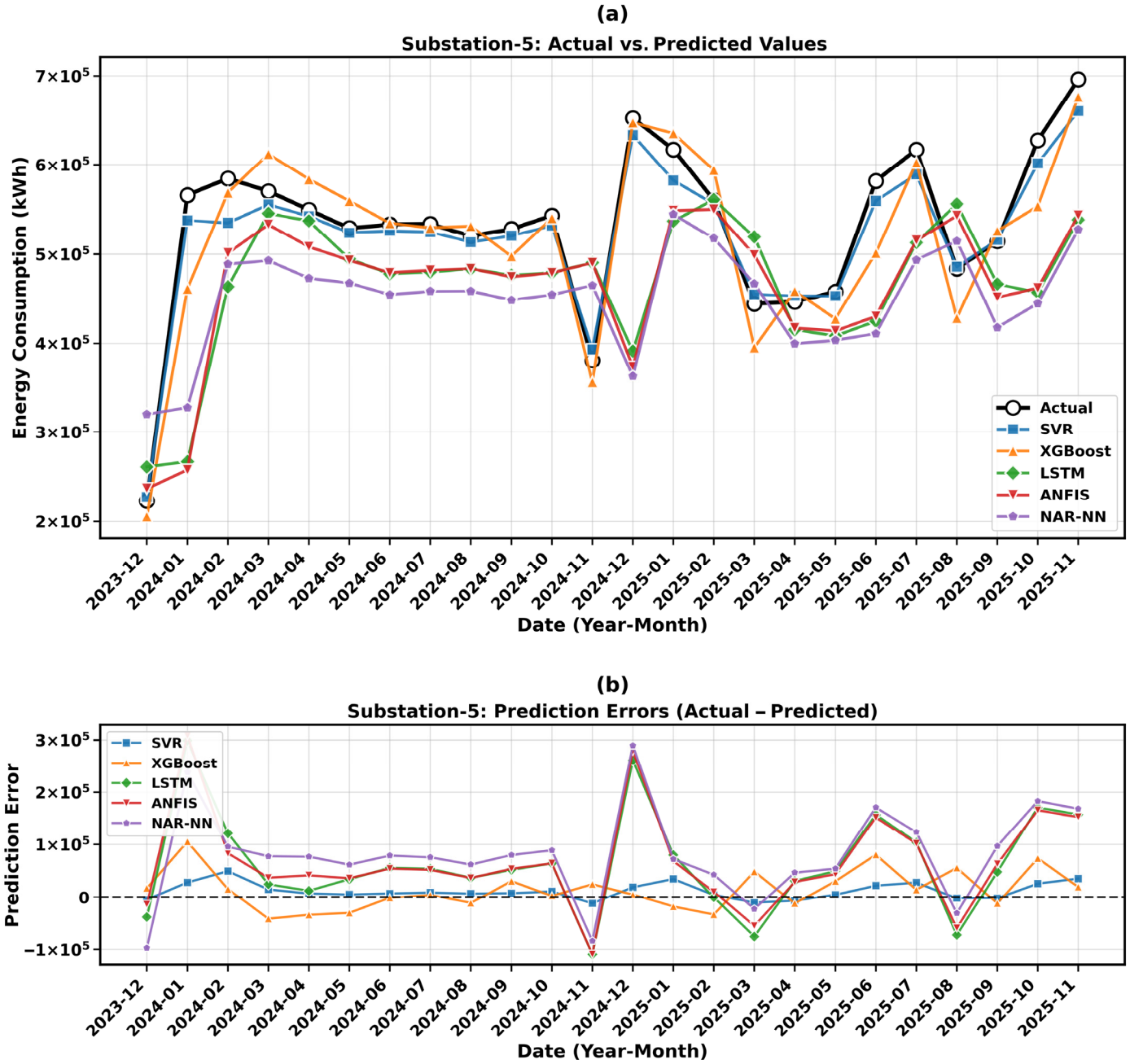

Figure 9. Substation-5: (**a**) Actual and predicted values, (**b**) prediction errors.

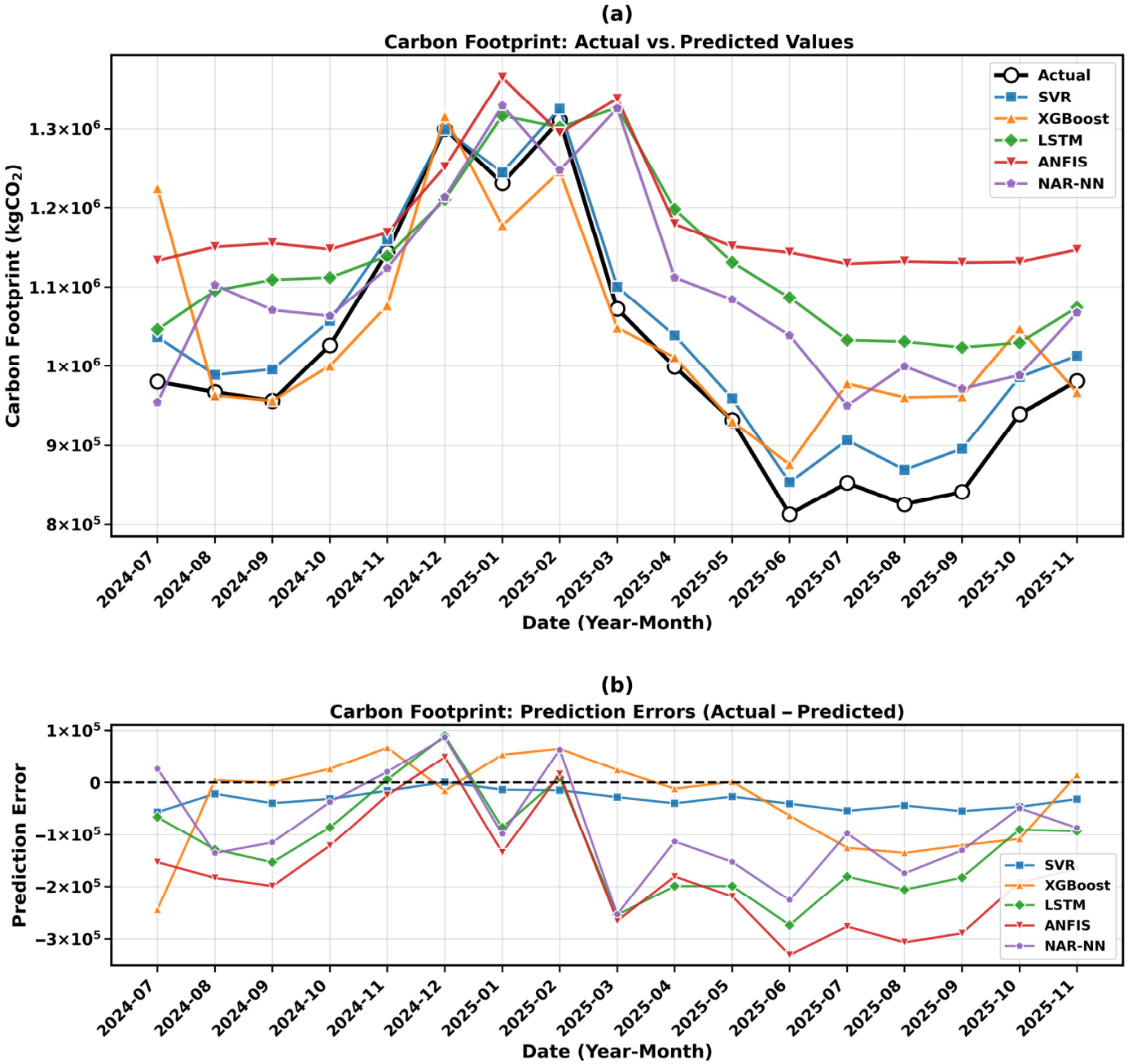

Figure 10. Carbon footprint: (**a**) Actual and predicted values, (**b**) prediction errors.

Overall, across all figures, SVR predictions stay closest to the actual values, XGBoost predictions generally remain acceptable, and the neural network-based LSTM, ANFIS, and NAR-NN models show large deviations, particularly during sudden changes and seasonal peaks. The prediction error plots indicate that SVR errors are often symmetrical and small around the zero line, whereas the neural network-based models produce large-magnitude errors.

4. Conclusions and Discussion

In this study, five different machine learning algorithms are comprehensively evaluated for forecasting the energy consumption and carbon footprint of urban rail transit systems. Ten years of monthly energy consumption data obtained from five distribution substations of a tram rail system are used, missing values are imputed using the Seasonal K-NN method, and hyperparameters are optimized via Random Search.

The results show that SVR, among the other machine learning methods, consistently achieves the highest performance across all datasets. For carbon footprint forecasting, the SVR model yields a superior performance with an R^2 of 0.942 and achieves low prediction error with RMSE = 36,453.30. These findings indicate that SVR has strong generalization capability for small- and medium-scale time series. The XGBoost algorithm shows the second-best performance for carbon footprint data with $R^2 = 0.648$. As an ensemble method, XGBoost performs particularly well on datasets with regular patterns, such as Substation-1 ($R^2 = 0.925$) and Substation-5 ($R^2 = 0.825$). In contrast, the neural network-based models (LSTM, ANFIS, and NAR-NN) exhibit inadequate performance, contrary to expectations. The LSTM model yields $R^2 = -0.064$ for carbon footprint forecasting, and ANFIS ($R^2 = -1.288$) and NAR-NN ($R^2 = 0.232$) similarly show low performance. This outcome is consistent with the fact that deep-learning methods typically require large datasets to perform well; the 119-month dataset is insufficient to train complex architectures such as LSTM effectively. Similar conclusions also emerge for the substation datasets.

From a practical perspective, SVR-based forecasting models are recommended for medium-term energy planning and carbon footprint monitoring in urban rail transit systems. A MAPE of 3.51% provides acceptable accuracy for annual budget allocation, energy procurement planning, and carbon-offset project sizing. These monthly forecasts can support renewable energy capacity planning, seasonal demand preparation, and policy-level carbon accounting. To quantify the practical implications of forecasting accuracy improvements: The annual total energy consumption of the five substations is approximately 25 million kWh, with a corresponding carbon footprint of approximately 11,000 tons of CO_2. While the SVR model achieves a MAPE of 3.51%, the average MAPE of LSTM is 14.45%. This 10.94% accuracy difference corresponds to a forecasting deviation of approximately 2.7 million kWh annually, and consequently, a carbon budgeting error of approximately 1200 tons of CO_2. A limitation of this study is the absence of classical statistical baselines (e.g., Seasonal Naive, SARIMA). Future studies may address such benchmarks to determine the relative advantage of ML methods for this specific data regime. Also in future work, exogenous variables (e.g., weather conditions, passenger counts, holidays, and substation-specific meta-variables) could be incorporated into the models. In addition, whether transfer learning approaches can improve the performance of deep-learning models on small datasets is also considered as a future research direction.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request, with the permission of the data provider.

Acknowledgments: The authors gratefully acknowledge Kayseri Ulaşım A.Ş. for granting access to the dataset used in this study.

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

Abbreviations

The following abbreviations are used in this manuscript:

ML	Machine Learning
SVR	Support Vector Regression
XGBoost	Extreme Gradient Boosting
LSTM	Long Short-Term Memory
ANFIS	Adaptive Neuro-Fuzzy Inference System
NAR-NN	Nonlinear Autoregressive Neural Network
MAPE	Mean Absolute Percentage Error
CO_2	Carbon Dioxide
BEMS	Building Energy Management Systems
GRU	Gated Recurrent Unit
CNN	Convolutional Neural Network
IoT	Internet of Things
AutoML	Automated Machine Learning
TCN	Temporal Convolutional Network
SWT	Stationary Wavelet Transform
FRBPS	Fuzzy Rule-Based Prediction Systems
GAN	Generative Adversarial Network
NDC	Nationally Determined Contributions
PCA	Principal Component Analysis
BSS	Blind Source Separation
XAI	eXplainable Artificial Intelligence
SHAP	SHapley Additive exPlanations
WPP	Wind Power Plant
K-NN	K-Nearest Neighbors
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
kWh	Kilowatt-Hour

Appendix A

Table A1. Optimized hyperparameters for all models across the six time series (five substations and carbon footprint).

Model	Parameter	Substation-1	Substation-2	Substation-3	Substation-4	Substation-5	Carbon Footprint
SVR	C	1.9762	1.9762	2.2465	1.9762	1.9762	1.9762
	ε (epsilon)	0.0312	0.0312	0.0105	0.0312	0.0312	0.0312
	γ (gamma)	0.0685	0.0685	0.6708	0.0685	0.0685	0.0685
	kernel	rbf	rbf	poly	rbf	rbf	rbf
XGBoost	n_estimators	138	94	94	150	31	91
	max_depth	4	5	5	2	4	3
	learning_rate	0.0726	0.2330	0.2330	0.2925	0.2267	0.1857
	subsample	0.9506	0.8796	0.8796	0.9497	0.9187	0.7423
	colsample_bytree	0.7962	0.7468	0.7468	0.7637	0.9314	0.9407
	min_child_weight	4	4	4	4	4	4
	gamma	0.1163	0.1232	0.1232	0.1734	0.2434	0.4948
	reg_alpha	0.6318	0.8796	0.8796	0.3738	0.2043	0.7950
	reg_lambda	0.7098	0.6410	0.6410	0.5723	0.8768	0.2788

Table A1. *Cont.*

Model	Parameter	Substation-1	Substation-2	Substation-3	Substation-4	Substation-5	Carbon Footprint
LSTM	units_1	9	23	23	9	17	17
	units_2	14	3	3	14	4	4
	dropout	0.3803	0.2195	0.2195	0.3803	0.4486	0.4486
	learning_rate	0.005106	0.008890	0.008890	0.005106	0.002274	0.002274
	batch_size	8	32	32	8	8	8
	lookback	6	5	5	6	4	4
ANFIS	n_mfs	2	3	3	3	3	3
	learning_rate	0.0201	0.0205	0.0148	0.0893	0.0132	0.0196
	lookback	4	4	2	5	4	4
NAR-NN	hidden_units	15	18	10	14	17	17
	n_layers	1	2	2	1	1	1
	activation	tanh	relu	tanh	relu	relu	tanh
	learning_rate	0.0352	0.0116	0.0771	0.0232	0.0483	0.0391
	lookback	3	3	4	4	4	2

References

1. Qureshi, M.; Arbab, M.A.; Rehman, S. Deep learning-based forecasting of electricity consumption. *Sci. Rep.* **2024**, *14*, 6489. [CrossRef]
2. Elhabyb, K.; Baina, A.; Bellafkih, M.; Deifalla, A.F. Machine Learning Algorithms for Predicting Energy Consumption in Educational Buildings. *Int. J. Energy Res.* **2024**, *2024*, 6812425. [CrossRef]
3. Morcillo-Jimenez, R.; Mesa, J.; Gómez-Romero, J.; Vila, M.A.; Martin-Bautista, M.J. Deep learning for prediction of energy consumption: An applied use case in an office building. *Appl. Intell.* **2024**, *54*, 5813–5825. [CrossRef]
4. Mahjoub, S.; Chrifi-Alaoui, L.; Marhic, B.; Delahoche, L. Predicting Energy Consumption Using LSTM, Multi-Layer GRU and Drop-GRU Neural Networks. *Sensors* **2022**, *22*, 4062. [CrossRef] [PubMed]
5. Ramos, P.V.B.; Villela, S.M.; Silva, W.N.; Dias, B.H. Residential energy consumption forecasting using deep learning models. *Appl. Energy* **2023**, *350*, 121705. [CrossRef]
6. Chan, J.W.; Yeo, C.K. A Transformer based approach to electricity load forecasting. *Electr. J.* **2024**, *37*, 107370. [CrossRef]
7. Chung, J.; Jang, B. Accurate prediction of electricity consumption using a hybrid CNN-LSTM model based on multivariable data. *PLoS ONE* **2022**, *17*, e0278071. [CrossRef]
8. Maleki, N.; Lundström, O.; Musaddiq, A.; Jeansson, J.; Olsson, T.; Ahlgren, F. Future energy insights: Time-series and deep learning models for city load forecasting. *Appl. Energy* **2024**, *374*, 124067. [CrossRef]
9. Deng, Y.; Yue, Z.; Wu, Z.; Li, Y.; Wang, Y. TCN-Attention-BIGRU: Building energy modelling based on attention mechanisms and temporal convolutional networks. *Electron. Res. Arch.* **2024**, *32*, 2160–2179. [CrossRef]
10. Frikha, M.; Taouil, K.; Fakhfakh, A.; Derbel, F. Predicting Power Consumption Using Deep Learning with Stationary Wavelet. *Forecasting* **2024**, *6*, 864–884. [CrossRef]
11. Gorzałczany, M.B.; Rudziński, F. Energy Consumption Prediction in Residential Buildings—An Accurate and Interpretable Machine Learning Approach Combining Fuzzy Systems with Evolutionary Optimization. *Energies* **2024**, *17*, 3242. [CrossRef]
12. Chahardoli, M.; Osati Eraghi, N.; Nazari, S. An energy consumption prediction approach in smart cities by CNN-LSTM network improved with game theory and Namib Beetle Optimization (NBO) algorithm. *J. Supercomput.* **2025**, *81*, 403. [CrossRef]
13. Ajala, A.A.; Adeoye, O.L.; Salami, O.M.; Jimoh, A.Y. An examination of daily CO_2 emissions prediction through a comparative analysis of machine learning, deep learning, and statistical models. *Environ. Sci. Pollut. Res.* **2025**, *32*, 2510–2535. [CrossRef]
14. Begum, A.M.; Mobin, M.A. A machine learning approach to carbon emissions prediction of the top eleven emitters by 2030 and their prospects for meeting Paris agreement targets. *Sci. Rep.* **2025**, *15*, 19469. [CrossRef]
15. Xia, X.; Zhu, D.; Sha, J.; Ma, R.; Kang, W. Research on industrial carbon emission prediction method based on CNN–LSTM under dual carbon goals. *Int. J. Low-Carbon. Technol.* **2025**, *20*, 580–589. [CrossRef]
16. Meng, Z.; Sun, H.; Wang, X. Forecasting energy consumption based on SVR and Markov model: A case study of China. *Front. Environ. Sci.* **2022**, *10*, 883711. [CrossRef]
17. Li, F.; Sun, M.; Xian, Q.; Feng, X. MDL: Industrial carbon emission prediction method based on meta-learning and diff long short-term memory networks. *PLoS ONE* **2024**, *19*, e0307915. [CrossRef]
18. Ghorbal, A.B.; Grine, A.; Elbatal, I.; Almetwally, E.M.; Eid, M.M.; El-Kenawy, E.M. Predicting carbon dioxide emissions using deep learning and Ninja metaheuristic optimization algorithm. *Sci. Rep.* **2025**, *15*, 4021. [CrossRef] [PubMed]

19. Alam, G.M.I.; Arfin Tanim, S.; Sarker, S.K.; Watanobe, Y.; Islam, R.; Mridha, M.F.; Nur, K. Deep learning model based prediction of vehicle CO_2 emissions with eXplainable AI integration for sustainable environment. *Sci. Rep.* **2025**, *15*, 3655. [CrossRef]
20. Lee, E.H.; Lee, I.; Cho, S.-H.; Kho, S.-Y.; Kim, D.-K. A Travel Behavior-Based Skip-Stop Strategy Considering Train Choice Behaviors Based on Smartcard Data. *Sustainability* **2019**, *11*, 2791. [CrossRef]
21. Zhu, C.; Yang, X.; Wang, Z.; Fang, J.; Wang, J.; Cheng, L. Optimization for the Train Plan with Flexible Train Composition Considering Carbon Emission. *Eng. Lett.* **2023**, *31*, 562–573.
22. Troyanskaya, O.; Cantor, M.; Sherlock, G.; Brown, P.; Hastie, T.; Tibshirani, R.; Botstein, D.; Altman, R.B. Missing value estimation methods for DNA microarrays. *Bioinformatics* **2001**, *17*, 520–525. [CrossRef]
23. Beretta, L.; Santaniello, A. Nearest neighbor imputation algorithms: A critical evaluation. *BMC Med. Inform. Decis. Mak.* **2016**, *16*, 74. [CrossRef]
24. Vapnik, V.N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, NY, USA, 1999. [CrossRef]
25. 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 (KDD '16), San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
26. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
27. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
28. Jang, J.-S.R. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [CrossRef]
29. Narendra, K.S.; Parthasarathy, K. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Netw.* **1990**, *1*, 4–27. [CrossRef]
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015. [CrossRef]
31. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.

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.