sustainability

MDPI

Article

Partially Linear Component Support Vector Machine for Primary Energy Consumption Forecasting of the Electric Power Sector in the United States

Xin Ma [1,*], Yubin Cai [1,2], Hong Yuan [1,3] and Yanqiao Deng [1,3]

[1] School of Mathematics and Physics, Southwest University of Science and Technology, Mianyang 621010, China
[2] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China
[3] School of Management Science and Real Estate, Chongqing University, Chongqing 400045, China
* Correspondence: cauchy7203@gmail.com

Abstract: Energy forecasting based on univariate time series has long been a challenge in energy engineering and has become one of the most popular tasks in data analytics. In order to take advantage of the characteristics of observed data, a partially linear model is proposed based on principal component analysis and support vector machine methods. The principal linear components of the input with lower dimensions are used as the linear part, while the nonlinear part is expressed by the kernel function. The primal-dual method is used to construct the convex optimization problem for the proposed model, and the sequential minimization optimization algorithm is used to train the model with global convergence. The univariate forecasting scheme is designed to forecast the primary energy consumption of the electric power sector of the United States using real-world data sets ranging from January 1973 to January 2020, and the model is compared with eight commonly used machine learning models as well as the linear auto-regressive model. Comprehensive comparisons with multiple evaluation criteria (including 19 metrics) show that the proposed model outperforms all other models in all scenarios of mid-/long-term forecasting, indicating its high potential in primary energy consumption forecasting.

Keywords: support vector machines; principal component analysis; partially linear models; primary energy consumption

Citation: Ma, X.; Cai, Y.; Yuan, H.; Deng, Y. Partially Linear Component Support Vector Machine for Primary Energy Consumption Forecasting of the Electric Power Sector in the United States. *Sustainability* **2023**, *15*, 7086. https://doi.org/10.3390/su15097086

Academic Editor: Sergio Nardini

Received: 21 February 2023
Revised: 26 March 2023
Accepted: 13 April 2023
Published: 23 April 2023

1. Introduction

Energy forecasting has long been a hot spot in this era of energy revolution. In most recent years, energy forecasting is already available to bring profits to enterprises by helping them make more reasonable financial plans [1]. On the other hand, energy consumption is not only an indicator for economics or finance but also an important factor for environmental issues, especially for carbon-related issues [2]. With the more diverse impact of real-world problems, energy forecasting is appealing to many researchers and engineers to make their own contributions. Topics of energy forecasting are also broadened to wider areas, such as energy consumption [3], energy production [4], energy price [5], the relationship between energy and economics and the environment [6], etc.

There is a long history of the application of primary for industrial production. Accurate forecasts of primary energy forecasting are still of great importance for making decisions in energy marketing, management, and also in the policies for pollution emissions. However, our investigation of the existing literature on energy forecasting (presented in Section 2) indicates that there are still issues in existing methods and implies that there is a research gap in the application of partially linear models for primary energy forecasting. In actuality, the time series of primary energy consumption often has very clear patterns of

variation, especially for the stable economical entities in mid–short periods; therefore, it is suitable to use the deterministic models to fit such properties. On the other hand, with the development of data-capturing technologies, it is much easier to obtain more data sets to build forecasting models. Thus, it is natural to consider using machine learning models to further improve forecasting accuracy. Above all, it is more reasonable to combine the merits of these models for better practice and higher accuracy in real-world applications.

The partially linear model is a typical example of the practice of combining models with deterministic and indeterministic formulations. The earliest work using the partially linear model should be credited to Engle et al., in which a very simple combination of linear regression and a nonlinear function was used [7]. The semi-parametric support vector machines (SVM) presented by Smola and Schölkopf was the first work that used machine learning models to build such a partially linear structure [8] in a uniform way, and the linear kernel was used to represent the linear part. Espinoza et al. presented another version of a kernel-based partially linear model based on the framework of least squares support vector machines (LSSVMs) [9]. Conversely, this work uses the nonlinear kernel to represent the nonlinear part, and it also presented an analytic way of training the model for the first time. Such properties of analytical solutions make them much easier to implement more models, and several models have been developed for function estimation and system identification [10–12]. In the last several years, Ma et al. used a simplified formulation to build the kernel-based grey system models by regularizing all linear parameters and parameters in the feature space [13–15], which actually also shares the philosophy of Hammerstein system models. The work by [16] Matí also uses the method of regularizing all parameters and made it easier to train a partially linear SVM. Within the different specific ways for implementation, all of these works have proven that the kernel-based partially linear models are much more efficient in the cases in which prior knowledge is available, such as a known linear relationship between the input and output.

It can be learned from the previous works that an efficient partially linear model can be developed if the features of the data are properly treated. For instance, Xu et al. [17] pointed out that it is also reasonable to separate the linear and nonlinear functions of the input, where the partially linear LSSVM based on this idea can then outperform the other models. Enlightened by this pattern, a new partially linear SVM using principal linear components extracted using a principal component analysis (PCA) is developed, and its related theoretical and computational problems will be discussed in detail. The real-world applications of forecasting the monthly primary energy consumption of electric power sector in the US will be presented, and the proposed model will be compared with several other machine learning models that have been very popular in recent research studies.

The rest of this work is organized as follows: literature studies are presented in Section 2; preliminary examinations on the specific formulation of the partially linear model, with the related theoretical basis, and the computational details of the PCA are introduced in Section 3; a complete representation of the proposed partially linear component support vector machine (PLC-SVM) is presented in Section 4, including its formulation in primal and dual spaces and its computational details for univariate time series forecasting; the case study forecasting the monthly primary energy consumption of the electric power sector in the US based on a data set with 565 months of real-world data is presented in Section 5, along with a comprehensive comparison between different models and a detailed discussion; the conclusions are drawn in Section 6.

2. Literature Study

In this section, some recent literature on energy forecasting will be reviewed, and the details of the most commonly used structured and non-structured models for energy forecasting will be briefly summarized. A short discussion on the findings and research gaps will also be presented in the last subsection. For convenience, an overview of the main models for energy forecasting reviewed in this section is presented in Figure 1.

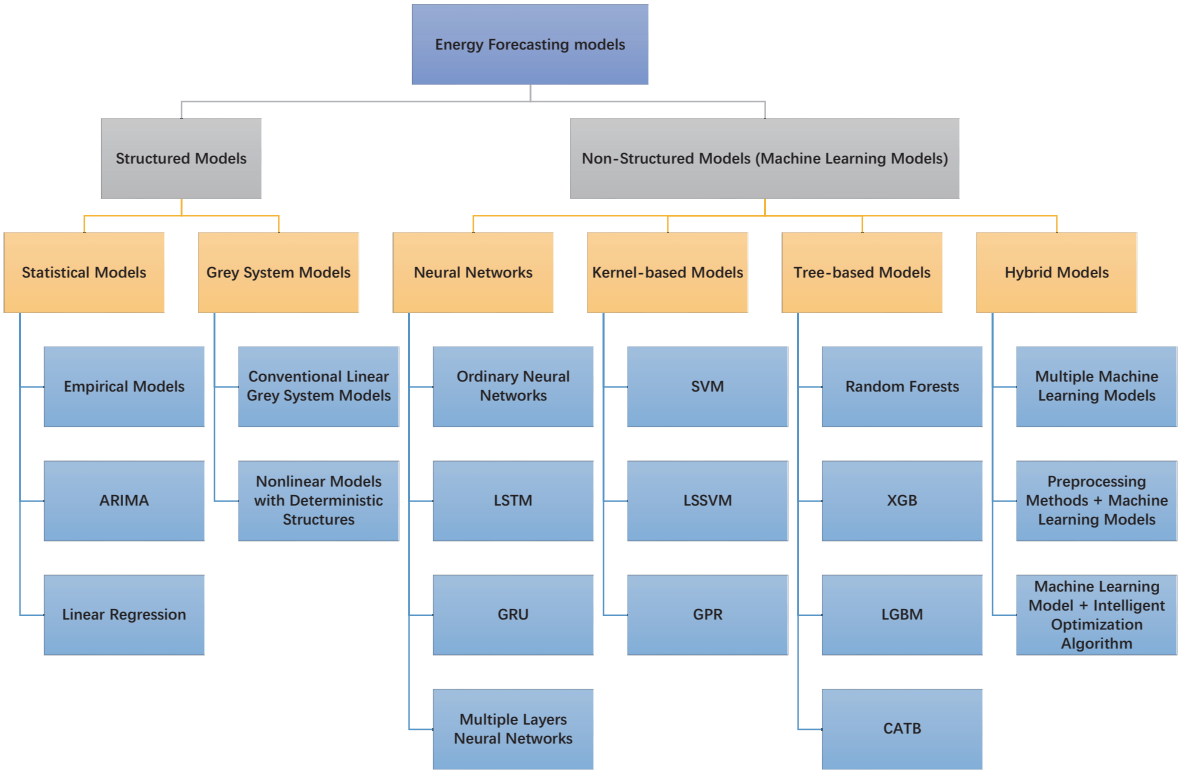

Figure 1. Overview of the models for energy foecasting in recent years.

2.1. The Structured Models for Energy Forecasting

In this subsection, the structured models are roughly categorized into empirical models, linear models, and grey system models.

The empirical models are often presented as specific functions (see [18]), which are often built with engineering experience and are directly validated in practice. These models are often easy to use but are not suitable for very complex data sets. Recent works have paid significantly fewer attention to such models.

The linear regression (LR) and autoregressive integrated moving average (ARIMA) models both share linear structures. While the LR model only simulates a simple linear correlation between the input and output variables [19,20], the ARIMA model mainly considers the auto-correlation of the time series. The linear models are quite popular in the application of energy forecasting and have been used to forecast oil consumption [19], electricity consumption [21,22], demand [20,23], wind generation [24], total energy demand and supply [25], etc. However, the ARIMA model often suffers from "overdifferece" [26], and both of these linear models are limited in describing nonlinear data sets.

Grey system models are increasingly popular in energy forecasting. There are several techniques used in the recent literature, including designing new structures to fit the data (e.g., nonlinear whitening equations [27], time-delayed terms [28], and periodic terms [29]), using complex accumulation operators (e.g., Hausdorff fractional order accumulation [30] and buffer operators [31]), and combining grey system models with other methods (e.g., Kalman filter [32] and Markov model [33]). Researchers often use intelligence optimizers when new methods contain nonlinear parameters [27,29–31]. One advantage of grey system models is their ability to make reliable predictions with limited data. However, for more complex forecasting applications, the proper structure or preprocessing methods still require the experience of researchers.

2.2. The Non-Structured Models for Energy Forecasting

Non-structured models do not have deterministic structures; a complete formulation can only be determined by the data sets. Machine learning is one of the most popular non-

structured models, and recent literature has shown considerable interest in the application of these models. The most popular machine learning models for energy forecasting are neural networks, support vector machines, and regression trees.

Neural networks, particularly multilayer perceptrons, remain popular for energy forecasting, with applications in areas such as electricity [34–36] and building energy consumption [37], ocean wave energy and photovoltaic plants generation forecasting [38], etc. Deep learning has led to the development of more complex models, such as LSTM-based networks with fully connected layers [39,40] or convolutional layers [41–43]. Other types of layers, such as bagged echo state networks [44], echo state networks [45], and radial belief networks [46], are also used. While these complex networks improve flexibility, they increase computational costs and require expert knowledge for the design. Thus, developing general models for energy forecasting remains challenging.

Kernel-based machine learning models, especially SVMs, remain popular for energy forecasting. Recent studies have focused on combining SVMs with evolutionary algorithms such as particle swarm optimization (PSO) [47], differential evolution (DE) [48], improved chicken swarm optimization (ICSO) [49], covariance matrix adaptation evolutionary strategy (CMAES) [50], improved fruit fly optimization (IFFO) [51], and Harris Hawks optimization [52], to optimize the hyperparameters automatically. These models are less time-consuming and have higher generality. However, partially linear kernel-based models have not been used in recent energy forecasting studies.

Many new models based on the basic regression trees have been developed in the past decade and are also widely adopted in energy forecasting, such as in carbon trading volume and price [53], building energy consumption [54], solar radiation [55], hydro-energy [56], etc. One significant merit of the regression tree-based models is that the ones with shallow structures are generally explainable. However, efficient regression trees usually become deeper with larger or more complex data sets, and a large amount of hyperparameters may also make the overall forecasting process too complex.

Hybrid models are gaining more interest in energy forecasting in both the literature and in competitions [57]. The main schemes found in the literature can be categorized into three classes. The first class is to combine the machine learning models and the preprocessing methods, such as variational mode decomposition (VMD), autoencoder [58], singular spectrum analysis (SSA) [59], wavelet transform [60], etc. The second class is to combine different machine learning models using the ensemble learning scheme [61–63] or multiple combining scheme [64,65], among other schemes. The third class is actually the integration of the above two schemes. In these works, the decomposition methods are often adopted, such as empirical mode decomposition (EMD) [66] and complete ensemble empirical mode decomposition (CEEMD) [67]. Despite being simple and effective, these hybrid models are more complex than other machine learning models and can lead to longer training times, less explainability, and the need for better hardware.

2.3. A Brief Summary of Literature Study

According to the literature study presented above, the research gaps can be briefly summarized in two parts: (1) In terms of methodology, machine learning models are becoming more popular in recent works for energy forecasting. However, along with the higher performance of more complex models, it raises other issues such as higher computational complexity and an incomplete framework of appropriate models in real-world applications. (2) In terms of applications, more complex models often need larger-sized data sets, and many works only present good performance in mid-/short-term predictions. The PLSVM method illustrates a new way of combining the linearity and nonlinearity of the data sets but has not been used in energy forecasting applications based on our investigation.

To fill the above research gaps, this work presents a new machine learning model for energy forecasting in real-world applications, and the main contributions can be summarized as follows:

- A partially linear component support vector machine is developed, which uses the principal linear features of the input data set obtained by a PCA. This way will reduce the risk of multi-collinearity and keep the model as simple as possible.
- A theoretical analysis is also presented, showing that the computational complexity of the main training process of the proposed model is in the same order as the existing SVM model.
- A complete partially linear auto-regression scheme for out-of-sample time series forecasting is presented in a real-world application with different scenarios on forecasting the primary energy consumption of the electric power sector of the United States, showing that the proposed model outperforms the cutting-edge models, especially in mid-/long-term forecasting.

3. Preliminaries

In this section, the main idea of the partially linear model and key steps of the principal component analysis (PCA) will be briefly summarized.

3.1. Main Idea of the Partially Linear Model

One typical definition of the partially linear model is [68]

$$y = \beta^T x^{\text{lin}} + g(x^{\text{nonl}}), \tag{1}$$

where x^{lin} consists of the linear dimensions of the input x, x^{nonl} consists of the nonlinear dimensions, and $g(\cdot)$ is an unknown nonlinear function. However, it has been argued that this formulation only separates the linear dimensions of the input vectors, and a more reasonable approach is to separate the linear functions of the input vector [17]. Enlightened by this idea, a simpler formulation is considered

$$y = \beta^T x + g(x), \tag{2}$$

where $\beta^T x$ is the linear function of x and $g(x)$ is an unknown nonlinear function of x.

Remark 1. *It is well known that any differentiable real function can be written by the formulation*

$$f(x) = f(x_0) + Df(x_0)(x - x_0) + R(x - x_0) \tag{3}$$

according to Taylor's theorem [69], where D is a differential operator (For multivariable functions, the differential operator can be written as $Df = \left(\frac{\partial f}{x_1}, \frac{\partial f}{x_2}, ..., \frac{\partial f}{x_d}\right)^T$*, and the products of between the vectors are inner products). This formulation can be transformed compactly by*

$$f(x) = Df(x_0)x + [R(x - x_0) + f(x_0) - Df(x_0)x_0]. \tag{4}$$

It is clear that the first term is a linear function of x and the second term is a nonlinear function (with constant bias). It is obvious that this formulation is mathematically equivalent to (2).

Based on this idea, the linear function of the input will be treated in a more direct way, and this will make it more stable than treating the linear part in a fully nonlinear way. For example, if the real nonlinearity follows a polynomial function such as

$$F(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3, \tag{5}$$

it will be unstable to approximate it using a full nonlinear function as the linear term $a_1 x$ will be over-estimated. Above all, the formulation in (2) is considered to build the partially linear model in this paper.

3.2. Principal Component Analysis

As described above, a partially linear model (2) has a linear function of the input. But in real-world applications, the elements in such a linear input may have high multicollinearity, which may lead to ill-posed problems and higher computational complexity. In this work, the principal component analysis (PCA) is used to reduce the dimension of the input.

PCA is one of the most popular classical linear methods, which can efficiently extract the linear features of the input vector and make it more stable for linear function estimations. For the original input $x = \left(x^1, x^2, \ldots, x^d\right)^T$, where $x^i (i = 1, 2, \ldots, d)$ represent the elements (features) of the input, the main goal of the PCA is to find a linear transformation A that transforms the original input x into a new vector z, of which the features are linearly independent to each other. For convenience, a set of an input is denoted by

$$X = \left(x_1, x_2, \cdots, x_N\right) \tag{6}$$

and the objective of the PCA is to find a linear matrix that satisfies:

$$A_{d\times d}(X_{d\times N} - U_{d\times N}) = Z_{d\times N} \tag{7}$$

where U is the matrix of mean values of X, of which the elements are $u_{ij} = \frac{1}{N}\sum_{k=1}^{N} x_k^j$ $(i = 1, \ldots, N, j = 1, \ldots, d)$. The transformation matrix A can be denoted by

$$A = \left(\xi_1, \xi_2, \cdots, \xi_d\right) \tag{8}$$

where ξ_i are the eigenvectors of the auto-covariance matrix $(X - U)(X - U)^T$, i.e.,

$$(X - U)(X - U)^T \xi_i = \lambda \xi_i. \tag{9}$$

The order of the eigenvectors is coincidental with the descending order of the corresponding eigenvalues λ_i of the auto-covariance matrix of X.

The contribution ratio of the k-th linear component in the new features Z is calculated by

$$r_k = \frac{\lambda_k}{\sum_{i=1}^{d} \lambda_i}. \tag{10}$$

The total contributions of the first k components are the sum of the first k ratios defined in (10). As the auto-covariance matrix $(X - U)(X - U)^T$ is a positive semi-definite symmetric matrix, all eigenvalues are non-negative; thus, the contribution ratios r_k are all non-negative. Furthermore, the total contributions of the first k components are in the range $[0, 1]$. Usually, if the contributions of some components are larger than a threshold r_p, they can contain almost all of the information of the original samples, and these components are called the principal components.

4. The Proposed Partially Linear Component Support Vector Machines

The modeling procedures and some key notes on the theoretical basis of the proposed partially linear component support vector machines for regression will now be presented.

4.1. Partially Linear Component Model in the Feature Space

A support vector machine model for regression essentially estimates a nonlinear function in a feature space, which is defined by

$$y = w^T \varphi(x) + b, \tag{11}$$

where

$$\varphi : \mathcal{R}^d \to \mathcal{F} \tag{12}$$

is a feature mapping that maps the vector in $\mathcal{R}^d$ space to a feature space, and $w^T\varphi(x)+b$ is a linear approximation in the feature space of a nonlinear function, i.e., $g(x)$ in (2) can be approximated in this way. Based on this idea, it is very natural to rewrite the partially linear function (2) into the following formulation

$$y = \beta^T z + w^T\varphi(x) + b, \tag{13}$$

where z is a vector only containing the principal linear components corresponding to x. According to the basic principles of functional analysis, it is very easy to build a new feature space using

$$\widetilde{\mathcal{F}} = \left\{ \begin{pmatrix} z \\ \varphi(x) \end{pmatrix} \mid z \in \mathcal{R}^p, \varphi(x) \in \mathcal{F}; x \in \mathcal{R}^d \right\}, \tag{14}$$

where p is the number of principal linear components and d is the dimension of x. Thus, it is very easy to define a new feature mapping $\phi : R^d \to \widetilde{\mathcal{F}}$ using

$$\phi(x) = \begin{pmatrix} z \\ \varphi(x) \end{pmatrix}. \tag{15}$$

The linear weights can then be concatenated using

$$\omega = \begin{pmatrix} \beta \\ w \end{pmatrix}. \tag{16}$$

The partially linear model can then be compactly written in the new feature space $\widetilde{\mathcal{F}}$ by

$$y = \omega^T\phi(x) + b. \tag{17}$$

4.2. Partially Linear Component Support Vector Machines in Primal and Dual Formulations

Within Formula (17), the primal problem of the partially linear component support vector machine for regression (PLC-SVM) can be defined as

$$\min_{\omega} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{18}$$

$$s.t. \begin{cases} y_i - \omega^T\phi(x) - b \le \varepsilon + \xi_i \\ \omega^T\phi(x) + b - y_i \le \varepsilon + \xi_i^*. \\ \xi_i, \xi_i^* \ge 0 \end{cases}$$

This formulation shares the same primal problem of the support vector regression, which is often known as the ε-insensitive formulation. However, this formulation is not available for computation use; thus, its corresponding dual problem should be used, which is defined by Smola et al. [70]

$$\begin{aligned} \max_{\alpha,\alpha^*} J(\alpha,\alpha^*) = &-\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\phi^T(x_i)\phi(x_j) \\ &-\varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \end{aligned} \tag{19}$$

$$s.t. \begin{cases} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}.$$

It is very important to notice that the linear weight ω in the feature space can be expressed by the linear combination of the mapping ϕ, and the weights are the Lagrangian multipliers, i.e.,

$$\omega = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\phi(x_i) = \begin{pmatrix} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)z_i \\ \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\varphi(x_i) \end{pmatrix}. \tag{20}$$

Then the partially linear function can be written as

$$\begin{aligned} \omega^T\phi(x) &= \left(\sum_{i=1}^{N}(\alpha_i - \alpha_i^*)z_i^T, \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\varphi^T(x_i)\right) \cdot \begin{pmatrix} z_j \\ \varphi(x_j) \end{pmatrix} \\ &= \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)z_i^T z_j + \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\varphi^T(x_i)\varphi(x_j) \end{aligned}. \tag{21}$$

Recalling the definition of ω in (16), it is easy to notice that

$$\beta = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)z_i. \tag{22}$$

Thus the partially linear function can be rewritten as

$$\omega^T\phi(x) = \beta^T z_j + \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\varphi^T(x_i)\varphi(x_j). \tag{23}$$

According to the kernel trick, the inner product of a feature mapping can be expressed by a kernel function that satisfies Mercer's condition, i.e.,

$$\varphi^T(x_i)\varphi(x_j) = k(x_i, x_j). \tag{24}$$

Noticing that the nonlinear mapping ϕ contains a linear and nonlinear part according to its definition (15), the inner products should be written as

$$\begin{aligned} \phi^T(x_i)\phi(x_j) &= \left(z_i^T, \varphi^T(x_i)\right)\begin{pmatrix} z_j \\ \varphi(x_j) \end{pmatrix} \\ &= z_i^T z_j + \varphi^T(x_i)\varphi(x_j) \\ &= z_i^T z_j + k(x_i, x_j) \end{aligned}. \tag{25}$$

Finally, the partially linear model can now be written as

$$\begin{aligned} y &= \omega^T\phi(x) + b \\ &= (\beta^T, w^T)\begin{pmatrix} z \\ \varphi(x) \end{pmatrix} + b \\ &= \beta^T z + w^T\varphi(x) + b \\ &= \beta^T z + \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)k(x_i, x) + b \end{aligned}. \tag{26}$$

The Gaussian kernel (also known as the radial basis function kernel) is often used

$$k(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right), \tag{27}$$

where γ is known as the reciprocal of the squares of the kernel width σ. The dual problem that can be used for computation can now be expressed within the inner product (25) as

$$\begin{aligned} \max J(\alpha, \alpha^*) = & -\frac{1}{2}\left(\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(Z^T Z + K)\right) \\ & -\varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \\ s.t. & \begin{cases} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}, \end{aligned} \tag{28}$$

where $K = (k(x_i, x_j))_{N\times N}$.

Remark 2. *The Gram matrix $Z^T Z$ is a positive semi-definite symmetric matrix, while K is also known as a positive semi-definite symmetric matrix; thus, their addition $Z^T Z + K$ is also a positive semi-definite symmetric. Thus, the dual problem (28) satisfies the condition of the typical quadratic programming (QR), and it can be solved using sequential minimum optimization (SMO) with global convergence as proven by Takahashi et al. [71].*

Within the above procedures and analysis, the overall computational steps of the proposed PLC-SVM are now clear, and a summarization is presented in the pseudo-code in Algorithm 1. The main computational steps can be roughly divided into four parts: the first part is preparing the data set and initializing the key settings; the second part utilizes the PCA to extract the principal linear components while building the kernel matrix used in (28); the third part solves the dual problem using the SMO algorithm, which has the same implementation as LibSM [72]; the last part makes predictions using the trained PLC-SVM model.

Algorithm 1: Algorithm of PLC-SVM (training and predicting).

Input: Training sample $\mathcal{S} = \left\{(x_i, y_i)\middle| i = 1, 2, \ldots, N\right\}$
Set: $\varepsilon = 10^{-6}$, C *in* (28), γ *for Gaussian kernel* (27), *threshold* $r_p = 0.95$
$A, \lambda_k \leftarrow$ eigen decomposition in (9) A defined in (8) $r_k \leftarrow$ Equation (10)
$n_{\text{pc}} \leftarrow \arg_k\left\{\left(\sum_{i=1}^{k} r_i\right) \geq r_p\right\}$ number of principal components **for** i = *1 to* N **do**
  $z_i \leftarrow$ first n_{pc} elements of $A(x_i - u)$ as defined in (7)
end
for i = *1 to* N **do**
  for j = *1 to* N **do**
    $K_{i,j} \leftarrow$ Equation (27) the kernel gram matrix
  end
end
α, b in (26) $\leftarrow$ solve the dual problem (28) SMO in LibSVM [72]
for i = *1 to* N_{pred} **do**
  $y_i^{\text{pred}} \leftarrow$ output function in Equation (26) with x_i
end
output values of the PLC-SVM **Output:** $Y = \left\{y_i \mid i = N+1, ..., N_{\text{pred}}\right\}$

Remark 3. *The complexity of the proposed PLC-SVM model is mainly contributed by the PCA and the cost for solving the dual problem (28). The complexity of the PCA is known to be $\mathcal{O}(d^2 \cdot N + d^3)$ for the worse cases. The complexity of the SMO in LibSVM is between $\mathcal{O}(N^2)$ and $\mathcal{O}(N^3)$.*

In general, the sample size is much larger than the dimension of the input vector, i.e., $N \gg d$; therefore, the total complexity of the PLC-SVM model is generally slightly larger than the SVM model with the same hyperparameters.

4.3. Forecasting Scheme for Univariate Time Series

The proposed model presented above essentially estimates a static model describing the relationship between the input and the output. But for time series forecasting, the model should estimate the correlation between the time series and the former points. One typical formulation is the auto-regressive model, which is represented by

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-\tau}). \tag{29}$$

In other words, the former series with τ points constructs a vector $x_t = [y_{t-1}, y_{t-2}, \ldots, y_{t-\tau}]^T$, which play a role as the input of the regression models. When the function $f(\cdot)$ is nonlinear, the Equation (29) is known as the nonlinear auto-regressive model (NAR). In this regard, it is easy to use the PLC-SVM model to build such an auto-regressive model; the main difference is that PLC-SVM considers the principal linear components of the input. Thus, the final model used in this work can be written as

$$y_t = w^T z_t + g(y_{t-1}, y_{t-2}, \ldots, y_{t-\tau}). \tag{30}$$

where z_t is the vector of which the elements are the linear components that are transformed by the PCA.

A complete partially linear auto-regression forecasting scheme is presented in Algorithm 2.

When executing the forecasting procedures, the newly predicted values of y_t would be added into the input at the next time step; thus the future points can be estimated using the models based on such recurrent scheme. It should be noticed that such a procedure is different to the n-step ahead forecasting, while all the future values would be forecasted only based on the in-sample data.

Algorithm 2: Algorithmof partially linear auto-regression based on PLC-SVM.

Input: Time series $y_t, t = 1, 2, \ldots, N$
Set: time lag τ, prediction horizon
stage I: reconstruct the sample data **for** $t = 1$ *to* $N - \tau$ **do**
  $x_t \leftarrow (y_{t+\tau-1}, y_{t+\tau-2}, \ldots, y_t)^T$
end
$\mathcal{S} = \left\{ (x_t, y_t) \middle| t = 1, 2, \ldots, N - \tau \right\}$
stage II: train the base model PLC-SVM ← Train PLC-SVM using Algorithm 1 with sample $\mathcal{S}$
stage III: forecasting by trained model $x_{\text{in}} = (y_N, y_{N-1}, \ldots, y_{N-\tau+1})^T$
for $t = 1$ *to* N_{pred} **do**
  $y_t^{\text{pred}} \leftarrow$ PLC-SVM(x_{in})
  update the input values remove the last element of x_{in}
  append y_t^{pred} as the first element of x_{in}
end
Output: $\left\{ y_t^{\text{pred}} \mid t = N + 1, 2, \ldots, N_{\text{pred}} \right\}$

5. Case Study

In this section, a real-world case study of forecasting the monthly primary energy consumption of the electric power sector in the US will be presented with three cases. The background information, evaluation metrics, and models for comparison will be introduced first, and then the results along with a discussion of the results will be presented.

The general framework illustrating the overall procedures of this case study is presented in Figure 2.

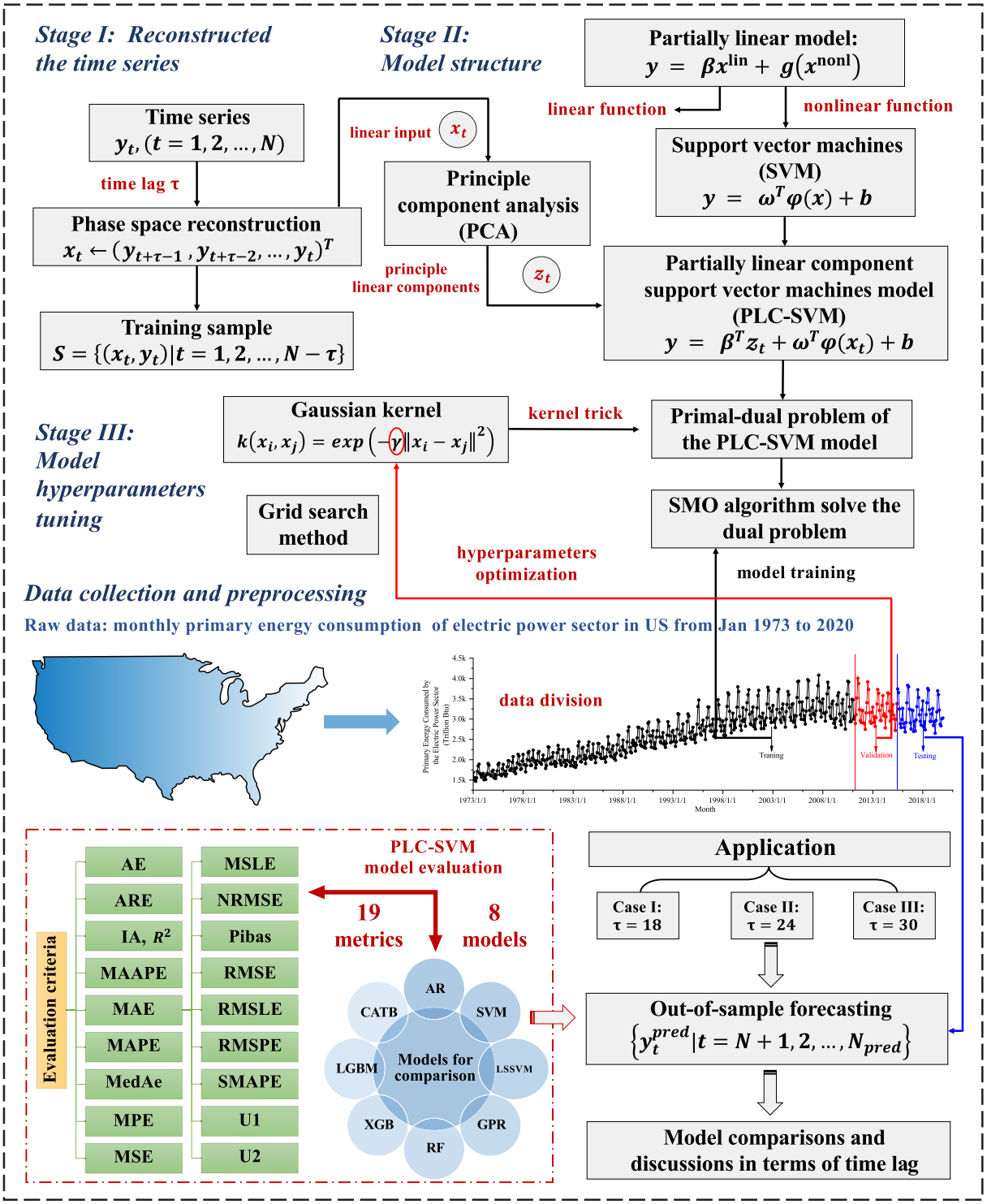

Figure 2. The general framework of the proposed PLC-SVM model structure and its application in US primary energy consumption in the electric power sector forecasting.

5.1. Data Collection and Preprocessing

As discussed in Section 1, the primary energy consumption is of great importance for industrial economics. In this section, the real-world case of the primary energy consumption of the electric power sector in the US was considered.

The raw data of the monthly primary energy consumption from January 1973 to January 2020 were collected from the US Energy Information Administration (EIA) website (https://www.eia.gov/totalenergy/data/monthly/ Monthly Energy Review of the US, accessed on 1 March 2020). As shown in Figure 3, the data set contains 565 points of monthly primary energy consumption of the electric power sector in the US (unit: trillion Btu). The time series data were firstly reconstructed using the steps presented in line 2 to line 5 in Algorithm 2. Then, the first 90% of points are used as in-sample data, and the remaining 10% of points are used as out-of-sample data; furthermore, the first 90% as in-sample data are finally used for training the models, and the remaining 10% of in-sample data are used for validating the performance of the models. In order to make it easier to train the machine learning models, the raw data were divided by the largest value in the in-sample data before training, and the final predicted values were multiplied by the same largest value.

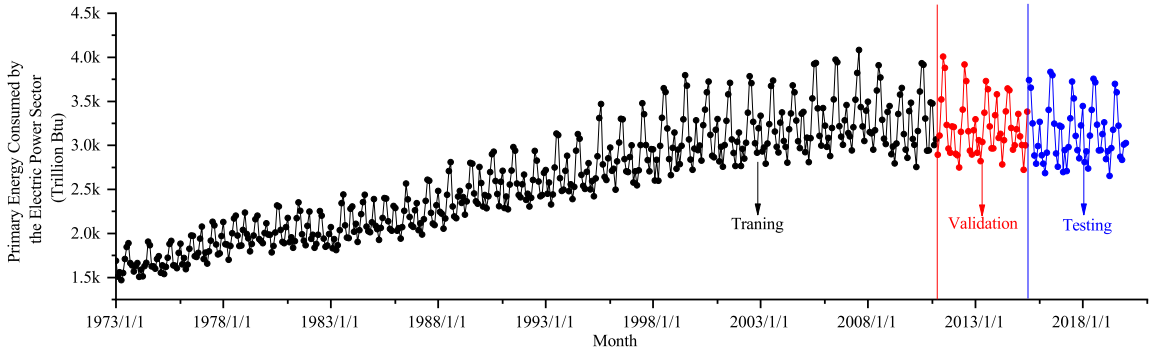

Figure 3. Raw data of monthly primary energy consumption of the electric power sector in the US from January 1973 to January 2020.

5.2. Models for Comparison and Evaluation Metrics

Nine models were selected for comparison with the proposed PLC-SVM, of which their information is summarized in Table 1 with descriptions of the corresponding hyperparameters. As described above, the PLC-SVM model is essentially based on the methodology of the SVM model; thus, the most closely related models are chosen for comparison. For convenience, the Gaussian kernel (27) is selected for SVM and LSSVM, and the rational quadratic kernel is selected for GPR, as suggested in [73]. On the other hand, the PLC-SVM model has a partially linear structure, and the linear auto-regressive model is also used as the baseline model for comparison.

- **AR:** The linear auto-regressive model (AR) used in this work is formulated as $y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_\tau y_{t-\tau}$, which can be regarded as a simplified version of PLC-SVM (without the kernel-based term and $C \to \infty$), and the parameters are estimated using the ordinary least squares method. With no hyperparameters, the AR does not need to be further optimized by the grid search cross-validation like in the other machine learning models.
- **SVM:** The ε-insensitive support vector machine (SVM) model for regression is selected in this work, of which the modelling details are described in [70,72]. It shares the most similar regularization formulation to PLC-SVM but has no partially linear part.
- **LSSVM:** The least squares support vector machine (LSSVM) model presented by Suykens in 1999 [74] is another version of SVM that uses quality constraints. The regression version of LSSVM is based on the LSSVM model for function estimation described in [75].
- **GPR:** The Gaussian process regression (GPR) model also uses the kernel combinations developed from the SVM model as described in [73]; the main difference is that the GPR approach is mainly based on the Bayesian theory.

Decision tree-based models are another kind of cutting-edge method, and they are widely used in the energy forecasting fields, such as in carbon energy [53], building

energy [54], solar energy [55], and hydro-energy applications [56], among others. These models all use the regression tree and ensemble learning method, such as boosting and bagging, and often have high performance in time series forecasting with high accuracy and stability and very low time cost. Thus, it will be very interesting to see whether the proposed PLC-SVM can outperform these emerging models in this case. Information on these models is listed below:

- **RF:** The random forest (RF) model is one of the most classical tree-based models, which mainly ensembles the weak regressors using bagging. The general method was first proposed by Ho in 1995 [76], and a complete work was first presented by Breiman in 2001 [77].
- **XGB:** The extreme gradient boosting (XGB) model was proposed by Chen in 2015, and the complete work was published in 2016 [78]. It was famous for its high performance in dealing with complex features and its extremely fast speed [79].
- **LGBM:** The light gradient boosting model was proposed by Ke in 2017 [80], who has won the one million bonus from Alibaba Ltd. using this model with his partners. The LGBM model uses multiple technologies to boost the original gradient boosting models, and it can even be more stable and faster than XGB in some tasks.
- **CATB:** Gradient boosting with categorical features support (CATB) was proposed by Prokhorenkova et al. in 2018 [81]. It has a very good performance in dealing with categorical features and has very good robustness.

The recurrent neural networks are widely used in time series forecasting and in related works in recent years. In this work, a state-of-the-art gated recurrent unit is used for comparison. The detailed information of this model is described as follows.

- **GRU:** The gated recurrent unit (GRU) model was introduced by Cho et al. [82] in 2014 as a simplified version of the long short-term memory (LSTM) model by Hochreiter and Schmidhuber [83] in 1997. In time series forecasting, the GRU model is often combined with other layers to capture more complex data patterns or shapes. In this study, a three-layer neural network was used, consisting of a GRU layer directly connected to the input data, an activation layer using a sigmoid function, and an output layer with a linear full connection.

Table 1. Models for comparison and their hyperparameters.

Model	Abbreviation	References	Hyperparameters
Auto-Regressive	AR	[21]	
Support Vector Machine	SVM	[70,72]	Kernel parameter, regularization parameter
Least Squares Support Vector Machine	LSSVM	[74]	Kernel parameter, regularization parameter
Gaussian Process Regression	GPR	[73]	Kernel type
Random Forest	RF	[76]	Bootstrap (whether bootstrap samples are used when building trees), maximum tree depth, number of features for the best split, minimum samples at a leaf node, minimum samples for splitting an internal node, number of trees
Extreme Gradient Boosting	XGB	[78]	Minimum loss reduction, learning rate, maximum tree depth, minimum weight for new node, L1 regularization parameter
Light Gradient Boosting	LGBM	[80]	Maximum tree depth, maximum tree leaves, minimum number of data needed in a child, L1 regularization parameter, L2 regularization parameter
Gradient Boosting with Categorical Features Support	CATB	[81]	Maximum number of trees, tree depth, L2 regularization parameter
Gated Recurrent Unit	GRU	[82]	Hidden size

To ensure a fair comparison, all machine learning models were utilized as nonlinear auto-regressive models, similar to PLC-SVM in Algorithm 2 (one can use these models in line-6 to implement the overall workflow). The models were implemented using Python 3.7, and their forecasting performances were evaluated using the multiple criteria listed in Table 2. The scikit-learn [84] library's built-in grid search method was used for tuning the hyperparameters of the models except for the AR model. Detailed information on the hyperparameters and original references are summarized in Table 1. In order to make the grid search process executable, we only choose the most important hyperparameters for

each model, following the engineering experience or suggestions made by the original references. As time series require forward series validation to determine the model's performance, it is more reasonable to use 90% of the in-sample data for training and the remaining 10% for validation, as in [85].

Table 2. Metrics used in this paper.

Metrics	Abbreviation	Formula
Average Error	AE	$\frac{1}{n}\sum_{k=1}^{n}\left(x^{(0)}(k)-\hat{x}^{(0)}(k)\right)$
Average Relative Error	ARE	$\frac{1}{n}\sum_{k=1}^{n}\left\|\frac{x^{(0)}(k)-\hat{x}^{(0)}(k)}{x(k)}\right\|$
Index of Agreement	IA	$1-\frac{\sum_{k=1}^{n}\left(x^{(0)}(k)-\hat{x}^{(0)}(k)\right)^2}{\sum_{k=1}^{n}\left(\left\|x^{(0)}(k)-\bar{x}^{(0)}\right\|+\left\|\hat{x}^{(0)}(k)-\bar{\hat{x}}^{(0)}\right\|\right)^2}$
Mean Arctangent Absolute Percentage Error	MAAPE	$\frac{1}{n}\sum_{k=1}^{n}\arctan\left(\left\|\frac{x^{(0)}(k)-\hat{x}^{(0)}(k)}{x(k)}\right\|\right)$
Mean Absolute Error	MAE	$\frac{1}{n}\sum_{k=1}^{n}\left\|x^{(0)}(k)-\hat{x}^{(0)}(k)\right\|$
Mean Absolute Percentage Error	MAPE	$\frac{1}{n}\sum_{k=1}^{n}\left\|\frac{x^{(0)}(k)-\hat{x}^{(0)}(k)}{x(k)}\right\|\times 100\%$
Median Absolute Error	MedAe	$\frac{1}{n}\sum_{k=1}^{n}\arctan\left(\left\|\frac{x^{(0)}(k)-\hat{x}^{(0)}(k)}{x(k)}\right\|\right)$
Mean Percentage Error	MPE	$\frac{1}{n}\sum_{k=1}^{n}\frac{x^{(0)}(k)-\hat{x}^{(0)}(k)}{x(k)}\times 100\%$
Mean Squared Error	MSE	$\frac{1}{n}\sum_{k=1}^{n}\left(x^{(0)}(k)-\hat{x}^{(0)}(k)\right)^2$
Mean Squared Logarithmic Error	MSLE	$\frac{1}{n}\sum_{k=1}^{n}\left\|\log\left(x^{(0)}(k)+1\right)-\log\left(\hat{x}^{(0)}(k)+1\right)\right\|^2$
Normalized Root Mean Square Error	NRMSE	$\frac{\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(x^{(0)}(k)-\hat{x}^{(0)}(k)\right)^2}}{x^{(0)}(k)_{\max}-x^{(0)}(k)_{\min}}$
Percent Bias	Pibas	$\frac{\sum_{k=1}^{n}\left(x^{(0)}(k)-\hat{x}^{(0)}(k)\right)}{\sum_{k=1}^{n}\hat{x}^{(0)}(k)}$
Coefficient of Determination	R^2	$1-\frac{\sum_{k=1}^{n}\left(x^{(0)}(k)-\hat{x}^{(0)}(k)\right)^2}{\sum_{k=1}^{n}\left(x^{(0)}(k)-\bar{x}^{(0)}\right)^2}$
Root Mean Square Error	RMSE	$\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(x^{(0)}(k)-\hat{x}^{(0)}(k)\right)^2}$
Root Mean Square Logarithmic Error	RMSLE	$\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left\|\log\left(x^{(0)}(k)+1\right)-\log\left(\hat{x}^{(0)}(k)+1\right)\right\|^2}$
Root Mean Square Percentage Error	RMSPE	$\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left\|\frac{x^{(0)}(k)-\hat{x}^{(0)}(k)}{x(k)}\right\|^2}$
Symmetric Mean Absolute Percentage Error	SMAPE	$\frac{1}{n}\sum_{k=1}^{n}\left\|\frac{x^{(0)}(k)-\hat{x}^{(0)}(k)}{0.5x^{(0)}(k)+0.5\hat{x}^{(0)}(k)}\right\|\times 100\%$
Theil U Statistic 1	U1	$\frac{\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(x^{(0)}(k)-\hat{x}^{(0)}(k)\right)^2}}{\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(x^{(0)}(k)\right)^2}+\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(\hat{x}^{(0)}(k)\right)^2}}$
Theil U Statistic 2	U2	$\frac{\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(x^{(0)}(k)-\hat{x}^{(0)}(k)\right)^2}}{\sqrt{\sum_{k=1}^{n}\left(x^{(0)}(k)\right)^2}}$

5.3. Results

In order to make a comprehensive comparison between the PLC-SVM model and the other models, three sub-cases based on the same data sets with different lags were carried out.

5.3.1. Case I: $\tau = 18$

In this case, the time lag is set as $\tau = 18$, i.e., every point will be predicted based on the former 18 points in the way presented in Algorithm 2. Four linear principal components are transformed by the PCA ($r = 0.95$) from eighteen dimensions, which are presented in

Equation (A1) in Appendix A. Then, the semi-analytical output function of PLC-SVM can be written as

$$\begin{aligned} y &= \beta^T z + w^T \varphi(x) + b \\ &= -0.1643 z_1 - 0.0156 z_2 - 0.2404 z_3 - 0.0692 z_4 + w^T \varphi(x) + 0.6824. \end{aligned} \tag{31}$$

The testing metrics of all models are listed in Table 3. It is clear that the overall performance of the PLC-SVM model is the best among all models as all of its metrics are the best. It is very interesting to see that the SVM model has the closest performance to PLC-SVM in this case, and this is easy to explain as they share similar methodologies (kernel method and ε-insensitive loss function). In the kernel-based models, the SVM model has the best performance aside from the PLC-SVM model, while the GPR model has the worst performance. The RF model performs the best and CATB performs the worst in the tree-based models. The GRU model only outperforms the worst tree-based models, which is a performance that is even worse than the linear AR model.

Table 3. Results of the metrics of the ten models with time lag $\tau = 18$.

	PLC-SVM	SVM	LSSVM	GPR	RF	LGBM	XGB	CATB	GRU	AR
AE	**−30.4026**	−31.9230	−131.3443	−144.5654	−105.5290	−135.2636	−86.1456	251.4290	161.4449	−79.9688
ARE	**0.0372**	0.0379	0.0489	0.0511	0.0387	0.0478	0.0398	0.0866	0.0622	0.0423
IA	0.9371	0.9347	0.9114	0.9224	**0.9492**	0.9194	0.9354	0.5436	0.8856	0.9202
MAAPE	**0.0372**	0.0379	0.0488	0.0510	0.0386	0.0477	0.0397	0.0859	0.0621	0.0422
MAE	**115.2908**	117.3400	145.8265	157.6530	116.6060	144.3825	119.7081	290.9737	193.6625	127.7884
MAPE	**3.7240**	3.7935	4.8940	5.1123	3.8695	4.7775	3.9777	8.6588	6.2247	4.2286
MedAe	94.8200	97.0436	123.2384	135.7107	106.0792	119.4479	**88.2682**	205.1306	178.9841	95.8115
MPE	**−1.3489**	−1.4058	−4.5055	−4.7546	−3.5528	−4.5172	−3.0012	7.2134	5.0866	−2.9021
MSE	**19,396.6580**	19,970.1730	32,269.8433	33,986.8550	20,322.6371	33,104.5516	24,215.1642	146,391.5235	51,779.6932	26,994.5302
MSLE	**0.0020**	0.0021	0.0034	0.0033	0.0022	0.0034	0.0026	0.0142	0.0060	0.0028
NRMSE	**0.1181**	0.1199	0.1524	0.1564	0.1209	0.1543	0.1320	0.3245	0.1930	0.1394
Pibas	**−0.0096**	−0.0101	−0.0402	−0.0440	−0.0325	−0.0413	−0.0267	0.0871	0.0543	−0.0249
R2	**0.8228**	0.8175	0.7051	0.6895	0.8143	0.6975	0.7787	−0.3376	0.5269	0.7533
RMSE	**139.2719**	141.3159	179.6381	184.3552	142.5575	181.9466	155.6122	382.6115	227.5515	164.3001
RMSLE	**0.0446**	0.0453	0.0587	0.0575	0.0464	0.0582	0.0506	0.1192	0.0777	0.0534
RMSPE	**0.0457**	0.0464	0.0615	0.0599	0.0481	0.0610	0.0529	0.1087	0.0739	0.0559
SMAPE	**3.6733**	3.7405	4.7180	4.9439	3.7609	4.6019	3.8561	9.2711	6.4815	4.1009
U1	**0.0220**	0.0223	0.0279	0.0286	0.0222	0.0282	0.0244	0.0633	0.0370	0.0257
U2	**0.0441**	0.0448	0.0569	0.0584	0.0452	0.0577	0.0493	0.1213	0.0721	0.0521

The predicted values of all 10 models, along with the percentage errors at each point, are plotted in Figure 4. It is very interesting to see that the values predicted by PLC-SVM and SVM are very close, which is coincident with the results in the metrics described above. It is also very clear that the predicted series of the other models except for CATB appear to be larger than the raw data, which are less stable than PLC-SVM and SVM, whereas the predicted values of CATB tend to be approximately constant in the last steps. It is interesting to see that the predicted values of GRU in the first few steps are actually acceptable, but most predicted values become smaller than the raw data with longer steps. The predicted values of AR are very close to the average value, which is coincident with its properties.

From another point of view, the PEs of PLC-SVM and SVM are approximately distributed around zero, as shown in Figure 4. However, more PEs of LSSVM, GPR, RF, LGBM, XGB, and AR are larger than zero; this indicates that these models overestimated future consumption. In contrast, more PEs of CATB and GRU are smaller than zero; this indicates that these models underestimated the future trend of consumption. Overall, the PLC-SVM model has the best performance in primary energy consumption in this case.

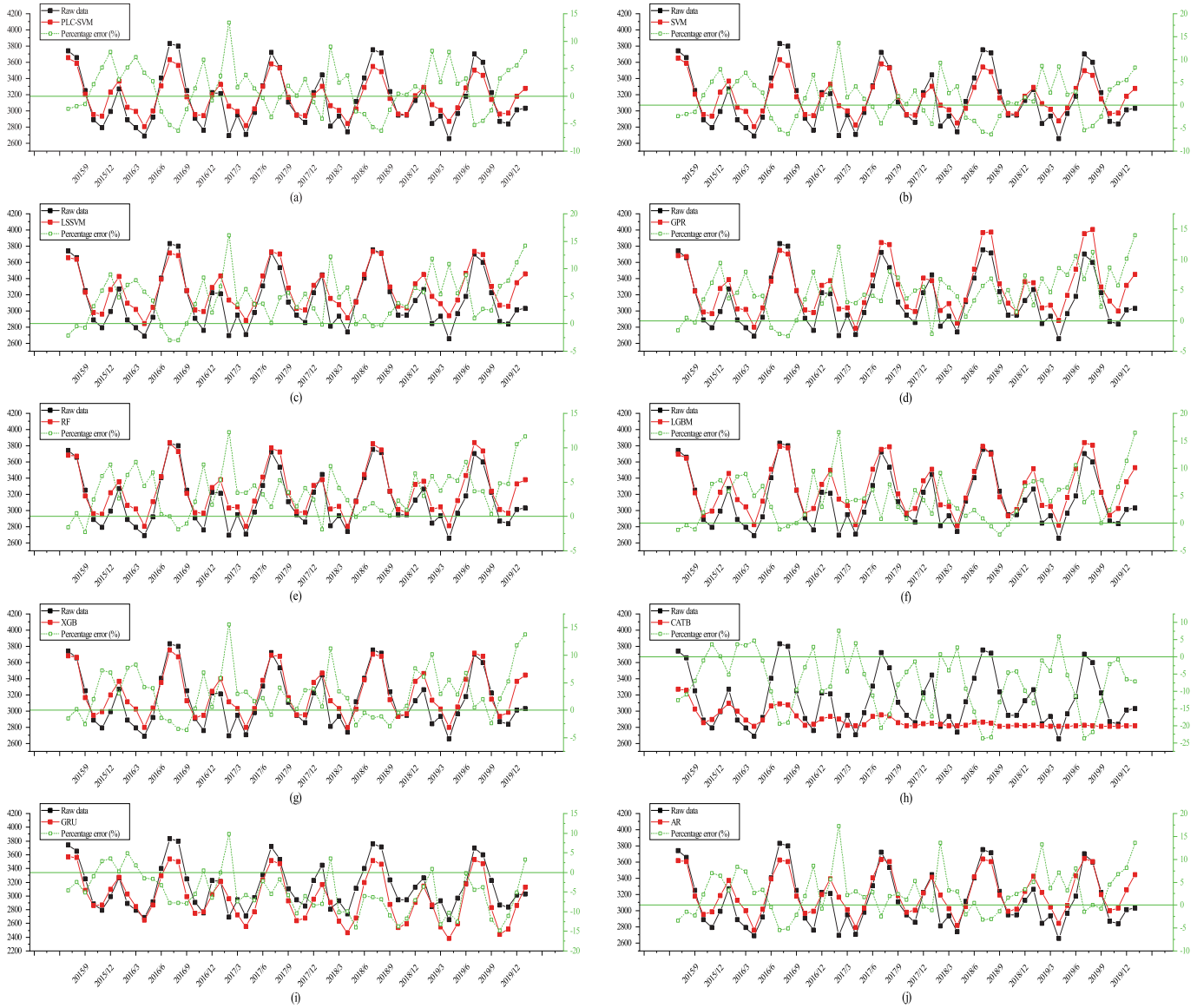

Figure 4. Predicted values using (**a**) PLC-SVM, (**b**) SVM, (**c**) LSSVM, (**d**) GPR, (**e**) RF, (**f**) LGBM, (**g**) XGB, (**h**) CATB, (**i**) GRU, and (**j**) AR with $\tau = 18$.

5.3.2. Case II: $\tau = 24$

In this case, the time lag is set as $\tau = 24$, i.e., every point will be predicted based on the former 24 points as described in Algorithm 2. The PCA ($r = 0.95$) transforms the 24 dimensions into 5 principal components, which are presented in Equation (A1) in Appendix A. Then, the output function of the PLC-SVM model can be written as:

$$\begin{aligned} y &= \beta^T z + w^T \varphi(x) + b \\ &= -0.074z_1 - 0.0036z_2 - 0.2802z_3 + 0.0157z_4 - 0.2419z_5 + w^T \varphi(x) + 0.6149 \end{aligned} \tag{32}$$

The testing metrics of all models are listed in Table 4. In this case, the performance of PLC-SVM is also the best among these models, and the errors are smaller than the other models on a more significant scale; SVM still has the closest performance to PLC-SVM. RF performs best among the tree-based models, while GRP and CATB perform the worst in the kernel-based and the tree-based models, respectively. In this case, GRU has the worst performance of all of the models. For the AR model, although it outperforms several other models, its metrics are still significantly worse than PLC-SVM.

Table 4. Results of the metrics of the ten models with time lag $\tau = 24$.

	PLC-SVM	SVM	LSSVM	GPR	RF	LGBM	XGB	CATB	GRU	AR
AE	**−69.5776**	−88.3138	−146.5563	−158.9098	−122.4937	−129.8408	−140.6233	198.5871	−290.2731	−126.2264
ARE	**0.0396**	0.0417	0.0504	0.0532	0.0439	0.0504	0.0505	0.0742	0.1009	0.0488
IA	0.9309	0.9262	0.9178	0.9200	**0.9365**	0.9044	0.9078	0.6071	0.7168	0.9054
MAAPE	**0.0395**	0.0416	0.0503	0.0531	0.0438	0.0502	0.0503	0.0738	0.1001	0.0486
MAE	**120.1745**	125.5630	152.0696	163.6975	132.1869	152.4988	151.0061	248.7770	298.6850	145.9707
MAPE	**3.9617**	4.1726	5.0412	5.3196	4.3856	5.0356	5.0492	7.4202	10.0853	4.8787
MedAe	**94.6035**	104.8577	123.1539	143.7897	119.0864	116.1281	125.8849	169.8898	294.6203	123.9808
MPE	**−2.5357**	−3.1289	−4.8879	−5.1840	−4.1159	−4.3672	−4.7381	5.5953	−9.8633	−4.3340
MSE	**23,899.4588**	26,147.9685	33,695.2393	36,364.6930	25,312.3053	39,167.0882	37,253.4219	106,653.0431	119,973.9310	35,560.0139
MSLE	**0.0025**	0.0028	0.0035	0.0035	0.0027	0.0040	0.0039	0.0100	0.0122	0.0037
NRMSE	**0.1311**	0.1372	0.1557	0.1617	0.1349	0.1679	0.1637	0.2770	0.2938	0.1599
Pibas	**−0.0217**	−0.0274	−0.0446	−0.0482	−0.0376	−0.0397	−0.0429	0.0676	−0.0847	−0.0387
R2	**0.7816**	0.7611	0.6921	0.6677	0.7687	0.6421	0.6596	0.0255	-0.0962	0.6751
RMSE	**154.5945**	161.7033	183.5626	190.6953	159.0984	197.9068	193.0115	326.5778	346.3725	188.5736
RMSLE	**0.0502**	0.0527	0.0591	0.0594	0.0516	0.0629	0.0625	0.1002	0.1106	0.0611
RMSPE	**0.0524**	0.0552	0.0620	0.0620	0.0536	0.0663	0.0659	0.0929	0.1193	0.0645
SMAPE	**3.8536**	4.0426	4.8589	5.1372	4.2496	4.8406	4.8452	7.8382	9.4289	4.6886
U1	**0.0243**	0.0253	0.0285	0.0295	0.0248	0.0307	0.0299	0.0536	0.0526	0.0293
U2	**0.0490**	0.0513	0.0582	0.0604	0.0504	0.0627	0.0612	0.1035	0.1098	0.0598

The predicted values and PEs of all 10 models are plotted in Figure 5. The values predicted by PLC-SVM and SVM still seem to be close. But in this case, it is more obvious that LSSVM, GPR, RF, LGBM, XGB, and AR all overestimate the observations, and it is clear that the overall trends reflected by these models are less stable and appear to be increasing. The values predicted by CATB still appear to decay, of which the peak values are too far away from the observations. Only some of the first predicted values by GRU are close to the raw data, but most of the following predicted values are larger than the average value of the corresponding raw data.

By analyzing the PEs shown in Figure 5, it is very clear that most PEs of LSSVM, GPR, RF, LGBM, XGB, GRU, and AR are larger than zero. This presents a clearer picture that these models all overestimate the future trend of real consumption. Meanwhile, most PEs of CATB are smaller than zero, and most of them are too large, indicating that the results of this model are not acceptable at all. The positive and negative PEs of PLC-SVM and SVM appear to be approximately equivalent, and the MPE (defined in Table 2) is closest to zero. Overall, the advantage of PLC-SVM over the other models is still significant in this case.

5.3.3. Case III: $\tau = 30$

In this case, the time lag is set as $\tau = 30$, i.e., every point will be predicted based on the former 30 points. There are five principal components that are transformed by the PCA ($r = 0.95$), which are presented in Equation (A1) in Appendix A. The output function of the PLC-SVM model is obtained as:

$$\begin{aligned} y &= \beta^T z + w^T \varphi(x) + b \\ &= -0.1029z_1 - 0.1242z_2 - 0.2137z_3 - 0.0626z_4 - 0.104z_5 + w^T \varphi(x) + 0.9909 \end{aligned} \tag{33}$$

The testing metrics of all models are listed in Table 5. PLC-SVM is still the best model in this case, and it is very interesting to see that all of its metrics are generally better than the previous two cases. GRU has the second-best performance in this case, and its MedAe is even closer to zero than PLC-SVM. The performance of SVM is significantly worse than PLC-SVM in this case. XGB performs the best among the tree-based models, which has the closest performance to PLC-SVM. Meanwhile, GPR and CATB still have the worst performance in this case.

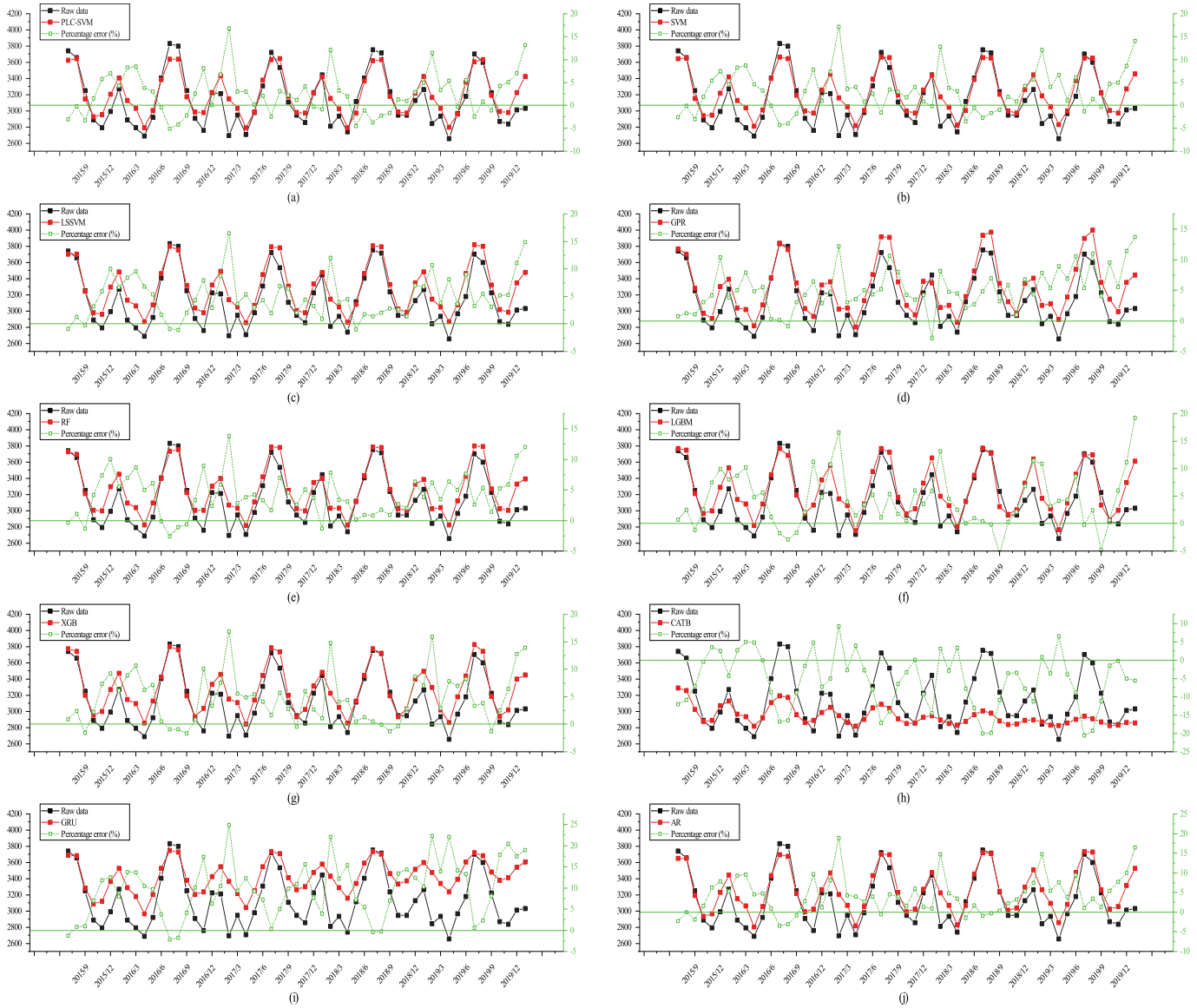

Figure 5. Predicted values using (**a**) PLC-SVM, (**b**) SVM, (**c**) LSSVM, (**d**) GPR, (**e**) RF, (**f**) LGBM, (**g**) XGB, (**h**) CATB, (**i**) GRU, (**j**) AR with $\tau = 24$.

Table 5. Results of the metrics of the ten models with time lag $\tau = 30$.

	PLC-SVM	SVM	LSSVM	GPR	RF	LGBM	XGB	CATB	GRU	AR
AE	**−85.5618**	−123.2492	−145.0989	−157.4847	−131.4612	−128.6614	−108.7032	173.3637	−95.0323	−138.1859
ARE	**0.0390**	0.0458	0.0492	0.0517	0.0447	0.0486	0.0428	0.0676	0.0402	0.0494
IA	0.9321	0.9161	0.9196	0.9215	**0.9345**	0.9086	0.9317	0.6547	0.9235	0.9063
MAAPE	**0.0389**	0.0457	0.0491	0.0516	0.0446	0.0485	0.0427	0.0673	0.0400	0.0492
MAE	**117.0587**	136.7596	148.3506	158.8722	135.0314	145.6343	128.2523	224.7215	120.2737	147.6295
MAPE	**3.8959**	4.5838	4.9192	5.1695	4.4729	4.8630	4.2765	6.7565	4.0169	4.9359
MedAe	94.3220	111.6924	114.1242	129.1694	121.9611	123.7311	108.4888	147.0317	**76.1974**	128.1547
MPE	**−3.0065**	−4.2051	−4.8233	−5.1293	−4.3722	−4.3847	−3.6771	4.8896	−3.2737	−4.6751
MSE	**23,675.5999**	30,820.1597	32,348.9584	34,821.7870	26,206.9815	33,890.0351	26,217.1006	85,262.6857	29,073.8466	35,635.1807
MSLE	**0.0025**	0.0033	0.0034	0.0034	0.0027	0.0036	0.0028	0.0080	0.0031	0.0037
NRMSE	**0.1305**	0.1489	0.1525	0.1583	0.1373	0.1561	0.1373	0.2477	0.1446	0.1601
Pibas	**−0.0266**	−0.0379	−0.0444	−0.0480	−0.0404	−0.0395	−0.0336	0.0587	−0.0295	−0.0423
R2	**0.7736**	0.7053	0.6907	0.6671	0.7494	0.6760	0.7494	0.1848	0.7220	0.6593
RMSE	**153.8688**	175.5567	179.8582	186.6060	161.8857	184.0925	161.9170	291.9977	170.5105	188.7728
RMSLE	**0.0502**	0.0571	0.0579	0.0583	0.0522	0.0597	0.0525	0.0894	0.0553	0.0611
RMSPE	**0.0525**	0.0600	0.0606	0.0608	0.0543	0.0628	0.0547	0.0837	0.0586	0.0644
SMAPE	**3.7758**	4.4170	4.7445	4.9923	4.3319	4.6801	4.1388	7.0836	3.8643	4.7417
U1	**0.0242**	0.0274	0.0280	0.0290	0.0252	0.0287	0.0253	0.0479	0.0267	0.0294
U2	**0.0490**	0.0559	0.0572	0.0594	0.0515	0.0586	0.0515	0.0929	0.0543	0.0601

The predicted values of all 10 models are plotted in Figure 6. The values predicted by PLC-SVM appear to be closer to the observations in this case than they were in the previous two cases. Having the closest performance to PLC-SVM, the predicted values of GRU are very close to most peak values, which appear to be closer to the raw data than the tree-based model XGB. The values predicted by CATB still decay with more steps. It is very interesting to see that only the predicted values by PLC-SVM and CATB all fall within the range of the observations, while there are several points by the other models that are larger than the nearby peak values.

By looking at the results of PEs plotted in Figure 6, most values of PEs of CATB are negative, and most PEs of the other models are positive; this indicates that most models over-estimated the raw data in this case. Moreover, it is very clear that the distributions of PEs of PLC-SVM and GRU appear to be more uniform than others. However, it is clear that the PEs of GRU with larger steps become larger than PLC-SVM; this is the reason why the overall metrics for GRU are not the best. Overall, although GRU presents a highly competitive performance, the PLC-SVM model still performs the best in this case.

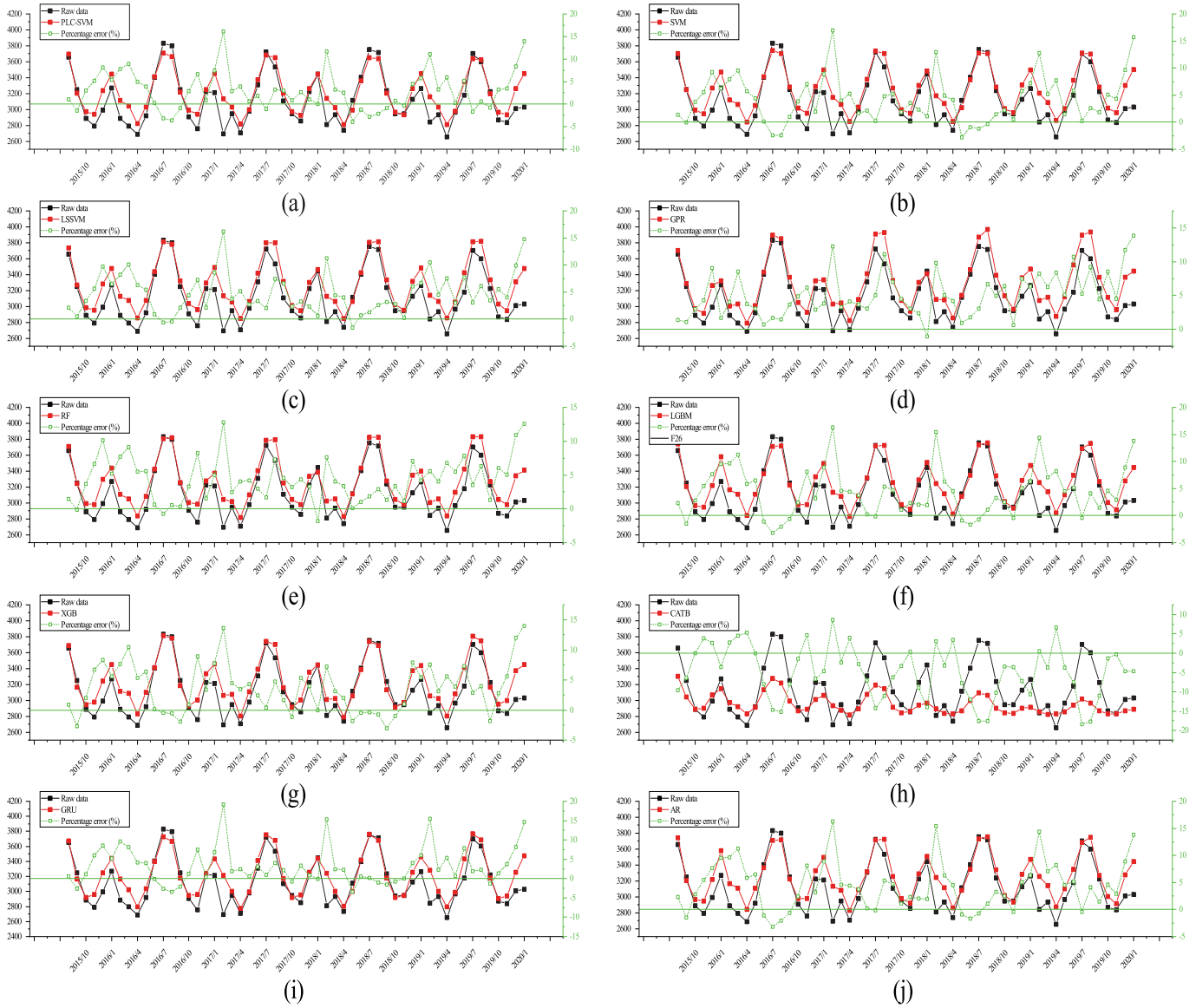

Figure 6. Predicted values using (**a**) PLC-SVM, (**b**) SVM, (**c**) LSSVM, (**d**) GPR, (**e**) RF, (**f**) LGBM, (**g**) XGB, (**h**) CATB, (**i**) LSTM, (**j**) AR with $\tau = 30$.

5.4. Discussion

It is clear that the PLC-SVM model has the best performance in all cases. One significant finding is that the PLC-SVM model indeed improved the accuracy of the SVM model. Having a similar structure and training algorithm, the SVM model can approach PLC-SVM with a smaller τ, as it was shown when $\tau = 18, 24$. But it is interesting to note that the difference between PLC-SVM and SVM becomes larger with longer lag, as it is shown that the related metrics of PLC-SVM are significantly better than SVM when $\tau = 30$. This indicates that the PLC-SVM model has a better performance in higher dimensional problems than the SVM model. It is very interesting to note that although the performance of the AR model is not the best, it generally presents a moderate performance in all cases. This indicates that there indeed exists a linear relationship between the current primary energy consumption and the former ones. Having a partially linear structure, the PLC-SVM model has taken advantage of such linear features. It is easy to see that such improvements are from its structure of a partially linear formulation, which takes most advantages of the linear features of the original series. At this stage, it can be confirmed that such linear features make the predicted series using the PLC-SVM model more accurate and stable than the SVM model, and this is also reflected in the Figures 4–6.

It should also be noted that the tree-based models are also very competitive compared with the PLC-SVM model. The best tree-based model in each case often presents a very close performance to the PLC-SVM model and is even much better than the other kernel-based models in some cases. Moreover, it is very interesting to see that the XGB model performs the second best when $\tau = 30$ and is much better than the other kernel-based models. This greatly coincides with a well-recognized result that tree-based models have very good performance in high-dimensional problems.

Although neural networks using the GRU model often perform much worse than other models with shorter lags, it is also very interesting to see that it performs quite well when $\tau = 30$, of which the metrics are the closest to the best model in this case, and even the MedAe model is better than that of the PLC-SVM model. This implies that the GRU model is very competitive with larger lags. However, even in such conditions, the overall performance of GRU is still slightly worse than PLC-SVM.

However, the advantages of PLC-SVM over the tree-based models and GRU is still significant. One of the most significant advantages of PLC-SVM is that it only has some hyperparameters to tune. In the above cases, only the regularization parameter C and kernel parameter γ are tuned, while the ε is set as a determined value (this is reasonable because it uses the ε-insensitive cost function). However, all tree-based models and GRU (also the other neural networks) have a lot of hyperparameters to tune, such as the maximum depth of trees, the number of estimators, and even other parameters that need fine-tuning. This is very important because less hyperparameter often means that the model is easier to tune, less time-consuming, and further makes it easier to design an optimal prediction scheme in real-world applications. Another advantage of PLC-SVM is its global convergence. And as mentioned in Section 4.2, the dual formulation is essentially a convex optimization; thus, the PLC-SVM model can be trained with global convergence. However, the algorithms used for the tree-based models and GRU (e.g., bagging for RF and gradient-based algorithms for other tree-based models and GRU) do not have global convergence; thus, they generally need more trials to obtain well-trained models.

For application implications, it is first suggested to use larger time lags as the PLC-SVM model presents a better performance with such settings, and this implies that more features may make the performance of PLC-SVM better. Another point is that the forecasting terms considered in this work are not short. In the above cases, the forecasting steps are all 55, which means that the monthly primary energy consumptions in 55 months (almost 5 years) are predicted. Considering the performance of stability and accuracy, it is reasonable to say that PLC-SVM is eligible to be used for primary energy consumption forecasting in the electric power sector for the mid-/long-term. Such performance may make it a potential tool for decision-making and marketing planning in the future.

6. Conclusions

A partially linear component support vector machine, named PLC-SVM, was proposed in this work. By using the PCA algorithm, the linear part of PLC-SVM has fewer linear dimensions, reducing the risk of multicollinearity and computational complexity. The methodology of SVM was used to construct the partially linear framework, and the use of the primal-dual trick causes the PLC-SVM model to have global optimality and easy implementation. The case study focused on the primary energy consumption forecasting of the electric power sector in the US by using the univariate time series data from January 1973 to January 2020, which contains 565 points of monthly primary energy consumption. The results of three sub-cases showed that the PLC-SVM model presents more accurate and stable forecasting results than the other three kinds of typical machine learning models and the linear AR model with different lags; larger lags might improve the performance of the PLC-SVM model. Within the above discussions, the PLC-SVM model is eligible to make mid-/long-term forecasting for primary energy forecasting of electric sectors in the US. Considering its general formulation, it can be expected to be used for forecasting more kinds of energies in future works.

The possible limitations of this work are twofold. The first issue is that this model might not be suitable for cases with too small of data sets. In such conditions, the available lags would be very small, which means that the original dimension of the linear part is already small; thus, obviously, the PCA will not work well. Another limitation is that this work only considered the most commonly used Gaussian kernel in the applications. More kernels can be designed based on achieving a very good performance if proper knowledge is used. In this regard, future works can also be extended by using more advanced kernels or new kernels that are designed for specific cases, as is suggested in the kernel cookbook by David [86].

Author Contributions: Conceptualization, methodology, writing—original draft preparation, funding acquisition, X.M.; software, X.M. and Y.C.; writing—review and editing, H.Y. and Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Humanities and Social Science Fund of the Ministry of Education of China (19YJCZH119), the Scientific and Technological Achievements Transformation Project of the Sichuan Scientific Research Institute (2022JDZH0035), and the National College Students Innovation and Entrepreneurship Training Program of China (S202210619106).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Publicly available datasets were analyzed in this study. The data can be found at https://www.eia.gov/totalenergy/data/monthly/, accessed on 1 March 2020.

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

Appendix A

The expressions of the principal components obtained in the case studies are presented in this section. In all of the following formulae, z_t^i represents the ith element in the vector z_t.

The four principal components in Case I:

$$
\begin{aligned}
z_t^1 = {} & -0.2298y_{t-1} - 0.2321y_{t-2} - 0.2343y_{t-3} - 0.2353y_{t-4} - 0.2355y_{t-5} \\
& - 0.2346y_{t-6} - 0.2331y_{t-7} - 0.2326y_{t-8} - 0.2326y_{t-9} - 0.2332y_{t-10} \\
& - 0.2337y_{t-11} - 0.2348y_{t-12} - 0.2371y_{t-13} - 0.2391y_{t-14} - 0.2410y_{t-15} \\
& - 0.2417y_{t-16} - 0.2415y_{t-17} - 0.2401y_{t-18} - 2.7449 \\
z_t^2 = {} & 0.2538y_{t-1} + 0.3692y_{t-2} + 0.1697y_{t-3} - 0.1758y_{t-4} - 0.3613y_{t-5} \\
& - 0.2335y_{t-6} + 0.0537y_{t-7} + 0.2129y_{t-8} + 0.1098y_{t-9} - 0.1205y_{t-10} \\
& - 0.2098y_{t-11} - 0.0388y_{t-12} + 0.2465y_{t-13} + 0.3589y_{t-14} + 0.1618y_{t-15} \\
& - 0.1788y_{t-16} - 0.3623y_{t-17} - 0.2350y_{t-18} + 0.0178 \\
z_t^3 = {} & -0.2789y_{t-1} + 0.0345y_{t-2} + 0.3278y_{t-3} + 0.3192y_{t-4} + 0.0167y_{t-5} \\
& - 0.2898y_{t-6} - 0.2963y_{t-7} - 0.0460y_{t-8} + 0.2106y_{t-9} + 0.2058y_{t-10} \\
& - 0.0534y_{t-11} - 0.2939y_{t-12} - 0.2745y_{t-13} + 0.0343y_{t-14} + 0.3226y_{t-15} \\
& + 0.3152y_{t-16} + 0.0162y_{t-17} - 0.2853y_{t-18} - 0.0094 \\
z_t^4 = {} & -0.1046y_{t-1} - 0.1707y_{t-2} - 0.1736y_{t-3} - 0.1765y_{t-4} - 0.1749y_{t-5} \\
& - 0.1029y_{t-6} + 0.0856y_{t-7} + 0.3274y_{t-8} + 0.5002y_{t-9} + 0.4970y_{t-10} \\
& + 0.3211y_{t-11} + 0.0822y_{t-12} - 0.1002y_{t-13} - 0.1688y_{t-14} - 0.1716y_{t-15} \\
& - 0.1726y_{t-16} - 0.1686y_{t-17} - 0.0982y_{t-18} + 0.0197,
\end{aligned}
\tag{A1}
$$

The five principal components in Case II:

$$
\begin{aligned}
z_t^1 = {} & -0.1974y_{t-1} - 0.1983y_{t-2} - 0.1994y_{t-3} - 0.2003y_{t-4} - 0.2006y_{t-5} \\
& - 0.2007y_{t-6} - 0.2013y_{t-7} - 0.2021y_{t-8} - 0.2031y_{t-9} - 0.2037y_{t-10} \\
& - 0.2041y_{t-11} - 0.2042y_{t-12} - 0.2045y_{t-13} - 0.2052y_{t-14} - 0.2060y_{t-15} \\
& - 0.2064y_{t-16} - 0.2064y_{t-17} - 0.2066y_{t-18} - 0.2066y_{t-19} - 0.2073y_{t-20} \\
& - 0.2082y_{t-21} - 0.2086y_{t-22} - 0.2088y_{t-23} - 0.2085y_{t-24} - 3.1723 \\
z_t^2 = {} & 0.2448y_{t-1} + 0.2681y_{t-2} + 0.0273y_{t-3} - 0.2352y_{t-4} - 0.2593y_{t-5} \\
& - 0.0202y_{t-6} + 0.2412y_{t-7} + 0.2631y_{t-8} + 0.0233y_{t-9} - 0.2385y_{t-10} \\
& - 0.2610y_{t-11} - 0.0239y_{t-12} + 0.2365y_{t-13} + 0.2601y_{t-14} + 0.0241y_{t-15} \\
& - 0.2340y_{t-16} - 0.2580y_{t-17} - 0.0217y_{t-18} + 0.2346y_{t-19} + 0.2550y_{t-20} \\
& + 0.0197y_{t-21} - 0.2375y_{t-22} - 0.2598y_{t-23} - 0.0274y_{t-24} + 0.0187 \\
z_t^3 = {} & -0.1634y_{t-1} + 0.1274y_{t-2} + 0.2912y_{t-3} + 0.1644y_{t-4} - 0.1281y_{t-5} \\
& - 0.2937y_{t-6} - 0.1675y_{t-7} + 0.1244y_{t-8} + 0.2910y_{t-9} + 0.1672y_{t-10} \\
& - 0.1206y_{t-11} - 0.2850y_{t-12} - 0.1628y_{t-13} + 0.1231y_{t-14} + 0.2846y_{t-15} \\
& + 0.1610y_{t-16} - 0.1273y_{t-17} - 0.2893y_{t-18} - 0.1642y_{t-19} + 0.1217y_{t-20} \\
& + 0.2844y_{t-21} + 0.1630y_{t-22} - 0.1195y_{t-23} - 0.2806y_{t-24} + 0.0035 \\
z_t^4 = {} & +0.1268y_{t-1} + 0.2401y_{t-2} + 0.2935y_{t-3} + 0.2712y_{t-4} + 0.1755y_{t-5} \\
& + 0.0331y_{t-6} - 0.1173y_{t-7} - 0.2321y_{t-8} - 0.2813y_{t-9} - 0.2537y_{t-10} \\
& - 0.1586y_{t-11} - 0.0226y_{t-12} + 0.1203y_{t-13} + 0.2325y_{t-14} + 0.2837y_{t-15} \\
& + 0.2588y_{t-16} + 0.1609y_{t-17} + 0.0186y_{t-18} - 0.1304y_{t-19} - 0.2435y_{t-20} \\
& - 0.2899y_{t-21} - 0.2600y_{t-22} - 0.1640y_{t-23} - 0.0287y_{t-24} + 0.0157 \\
z_t^5 = {} & -0.2626y_{t-1} - 0.1649y_{t-2} - 0.0228y_{t-3} + 0.1231y_{t-4} + 0.2348y_{t-5} \\
& + 0.2849y_{t-6} + 0.2609y_{t-7} + 0.1695y_{t-8} + 0.0315y_{t-9} - 0.1180y_{t-10} \\
& - 0.2375y_{t-11} - 0.2914y_{t-12} - 0.2640y_{t-13} - 0.1631y_{t-14} - 0.0192y_{t-15} \\
& + 0.1277y_{t-16} + 0.2367y_{t-17} + 0.2839y_{t-18} + 0.2578y_{t-19} + 0.1635y_{t-20} \\
& + 0.0244y_{t-21} - 0.1229y_{t-22} - 0.2389y_{t-23} - 0.2886y_{t-24} + 0.0057
\end{aligned}
\tag{A2}
$$

The five principal components in Case III:

$$
\begin{aligned}
z_t^1 = {} & -0.1751y_{t-1} - 0.1763y_{t-2} - 0.1776y_{t-3} - 0.1786y_{t-4} - 0.1791y_{t-5} \\
& - 0.1789y_{t-6} - 0.1783y_{t-7} - 0.1783y_{t-8} - 0.1786y_{t-9} - 0.1794y_{t-10} \\
& - 0.1800y_{t-11} - 0.1805y_{t-12} - 0.1816y_{t-13} - 0.1826y_{t-14} - 0.1838y_{t-15} \\
& - 0.1846y_{t-16} - 0.1850y_{t-17} - 0.1848y_{t-18} - 0.1839y_{t-19} - 0.1836y_{t-20} \\
& - 0.1838y_{t-21} - 0.1844y_{t-22} - 0.1848y_{t-23} - 0.1852y_{t-24} - 0.1860y_{t-25} \\
& - 0.1870y_{t-26} - 0.1880y_{t-27} - 0.1888y_{t-28} - 0.1890y_{t-27} - 0.1887y_{t-30} - 3.5511 \\
z_t^2 = {} & 0.1790y_{t-1} + 0.2921y_{t-2} + 0.1452y_{t-3} - 0.1319y_{t-4} - 0.2838y_{t-5} \\
& - 0.1779y_{t-6} + 0.0656y_{t-7} + 0.2018y_{t-8} + 0.1083y_{t-9} - 0.1034y_{t-10} \\
& - 0.2005y_{t-11} - 0.0685y_{t-12} + 0.1748y_{t-13} + 0.2863y_{t-14} + 0.1417y_{t-15} \\
& - 0.1319y_{t-16} - 0.2826y_{t-17} - 0.1787y_{t-18} + 0.0610y_{t-19} + 0.1952y_{t-20} \\
& + 0.1042y_{t-21} - 0.1026y_{t-22} - 0.1974y_{t-23} - 0.0679y_{t-24} + 0.1700y_{t-25} \\
& + 0.2787y_{t-26} + 0.1371y_{t-27} - 0.1307y_{t-28} - 0.2784y_{t-29} - 0.1765y_{t-30} + 0.0241 \\
z_t^3 = {} & -0.2259y_{t-1} + 0.0134y_{t-2} + 0.2487y_{t-3} + 0.2509y_{t-4} + 0.0205y_{t-5} \\
& - 0.2200y_{t-6} - 0.2301y_{t-7} - 0.0257y_{t-8} + 0.1870y_{t-9} + 0.1886y_{t-10} \\
& - 0.0226y_{t-11} - 0.2285y_{t-12} - 0.2235y_{t-13} + 0.0123y_{t-14} + 0.2446y_{t-15} \\
& + 0.2475y_{t-16} + 0.0209y_{t-17} - 0.2171y_{t-18} - 0.2266y_{t-19} - 0.0258y_{t-20} \\
& + 0.1822y_{t-21} + 0.1840y_{t-22} - 0.0227y_{t-23} - 0.2241y_{t-24} - 0.2184y_{t-25} \\
& + 0.0123y_{t-26} + 0.2393y_{t-27} + 0.2427y_{t-28} + 0.0206y_{t-29} - 0.2127y_{t-30} - 0.0056 \\
z_t^4 = {} & -0.0758y_{t-1} - 0.1498y_{t-2} - 0.1754y_{t-3} - 0.1783y_{t-4} - 0.1588y_{t-5} \\
& - 0.0878y_{t-6} + 0.0486y_{t-7} + 0.2115y_{t-8} + 0.3274y_{t-9} + 0.3331y_{t-10} \\
& + 0.2270y_{t-11} + 0.0685y_{t-12} - 0.0694y_{t-13} - 0.1451y_{t-14} - 0.1714y_{t-15} \\
& - 0.1734y_{t-16} - 0.1522y_{t-17} - 0.0802y_{t-18} + 0.0554y_{t-19} + 0.2159y_{t-20} \\
& + 0.3289y_{t-21} + 0.3320y_{t-22} + 0.2245y_{t-23} + 0.0658y_{t-24} - 0.0718y_{t-25} \\
& - 0.1473y_{t-26} - 0.1729y_{t-27} - 0.1743y_{t-28} - 0.1526y_{t-29} - 0.0818y_{t-30} + 0.0134 \\
z_t^5 = {} & -0.1951y_{t-1} - 0.0796y_{t-2} - 0.0195y_{t-3} + 0.0073y_{t-4} + 0.0727y_{t-5} \\
& + 0.1963y_{t-6} + 0.3028y_{t-7} + 0.2909y_{t-8} + 0.1277y_{t-9} - 0.1064y_{t-10} \\
& - 0.2759y_{t-11} - 0.2963y_{t-12} - 0.1997y_{t-13} - 0.0826y_{t-14} - 0.0198y_{t-15} \\
& + 0.0100y_{t-16} + 0.0759y_{t-17} + 0.1962y_{t-18} + 0.2996y_{t-19} + 0.2851y_{t-20} \\
& + 0.1207y_{t-21} - 0.1114y_{t-22} - 0.2777y_{t-23} - 0.2967y_{t-24} - 0.1998y_{t-25} \\
& - 0.0824y_{t-26} - 0.0190y_{t-27} + 0.0107y_{t-28} + 0.0745y_{t-29} + 0.1899y_{t-30} - 0.00001.
\end{aligned}
\tag{A3}
$$

References

1. Statt, N. Google and DeepMind Are Using AI to Predict the Energy Output of Wind Farms. *The Verge*, p. 1. Available online: https://www.theverge.com/2019/2/26/18241632/google-deepmind-wind-farm-ai-machine-learning-green-energy-efficiency (accessed on 26 February 2019).
2. Ma, M.; Ma, X.; Cai, W.; Cai, W. Low carbon roadmap of residential building sector in China: Historical mitigation and prospective peak. *Appl. Energy* **2020**, *273*, 115247. [CrossRef]
3. Lu, H.; Ma, X.; Azimi, M. Us natural gas consumption prediction using an improved kernel-based nonlinear extension of the arps decline model. *Energy* **2020**, *194*, 116905. [CrossRef]
4. Zeng, B.; Zhou, M.; Liu, X.; Zhang, Z. Application of a new grey prediction model and grey average weakening buffer operator to forecast China's shale gas output. *Energy Rep.* **2020**, *6*, 1608–1618. [CrossRef]
5. Niu, T.; Wang, J.; Lu, H.y.; Yang, W.; Du, P. A learning system integrating temporal convolution and deep learning for predictive modeling of crude oil price. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4602–4612. [CrossRef]
6. Yang, J.; Cai, W.; Ma, M.; Li, L.; Liu, C.; Ma, X.; Li, L.; Chen, X. Driving forces of China's CO2 emissions from energy consumption based on kaya-lmdi methods. *Sci. Total Environ.* **2020**, *711*, 134569. [CrossRef]

7. Engle, R.F.; Granger, C.W.J.; Rice, J.; Weiss, A. Semiparametric estimates of the relation between weather and electricity sales. *J. Am. Stat. Assoc.* **1986**, *81*, 310–320. [CrossRef]
8. Smola, A.J.; Frieß, T.; Schölkopf, B. Semiparametric support vector and linear programming machines. In Proceedings of the Advances in Neural Information Processing Systems 11, NIPS Conference, Denver, CO, USA, 30 November–5 December 1998; pp. 585–591.
9. Espinoza, M.; Suykens, J.A.K.; De Moor, B. Kernel based partially linear models and nonlinear identification. *IEEE Trans. Autom. Control* **2005**, *50*, 1602–1606. [CrossRef]
10. Goethals, I.; Pelckmans, K.; Suykens, J.A.K.; De Moor, B. Identification of mimo hammerstein models using least squares support vector machines. *Automatica* **2005**, *41*, 1263–1272. [CrossRef]
11. Varoquaux, G. Cross-validation failure: Small sample sizes lead to large error bars. *Neuroimage* **2018**, *180*, 68–77. [CrossRef]
12. Castro-Garcia, R.; Agudelo, O.M.; Suykens, J.A.K. Impulse response constrained ls-svm modelling for mimo hammerstein system identification. *Int. J. Control* **2019**, *92*, 908–925. [CrossRef]
13. Ma, X.; Liu, Z. Predicting the oil production using the novel multivariate nonlinear model based on Arps decline model and kernel method. *Neural Comput. Appl.* **2018**, *29*, 579–591. [CrossRef]
14. Ma, X.; Liu, Z. The kernel-based nonlinear multivariate grey model. *Appl. Math. Model.* **2018**, *56*, 217–238. [CrossRef]
15. Ma, X. A brief introduction to the grey machine learning. *J. Grey Syst.* **2019**, *31*, 1–12.
16. Matías, J.M.; Taboada, J.; Ordóñez, C.; González-Manteiga, W. Partially linear support vector machines applied to the prediction of mine slope movements. *Math. Comput. Model.* **2010**, *51*, 206–215. [CrossRef]
17. Xu, Y.; Chen, D.R. Partially-linear least-squares regularized regression for system identification. *IEEE Trans. Autom. Control* **2009**, *54*, 2637–2641.
18. Fan, J.; Wu, L.; Zhang, F.; Cai, H.; Zeng, W.; Wang, X.; Zou, H. Empirical and machine learning models for predicting daily global solar radiation from sunshine duration: A review and case study in China. *Renew. Sustain. Energy Rev.* **2019**, *100*, 186–212. [CrossRef]
19. Chang, Y.; Choi, Y.; Kim, C.S.; Miller, J.I.; Park, J.Y. Forecasting regional long-run energy demand: A functional coefficient panel approach. *Energy Econ.* **2021**, *96*, 105117. [CrossRef]
20. Johannesen, N.J.; Kolhe, M.; Goodwin, M. Relative evaluation of regression tools for urban area electrical energy demand forecasting. *J. Clean. Prod.* **2019**, *218*, 555–564. [CrossRef]
21. Akdi, Y.; Gölveren, E.; Okkaoğlu, Y. Daily electrical energy consumption: Periodicity, harmonic regression method and forecasting. *Energy* **2020**, *191*, 116524. [CrossRef]
22. Khalifa, A.; Caporin, M.; Di Fonzo, T. Scenario-based forecast for the electricity demand in qatar and the role of energy efficiency improvements. *Energy Policy* **2019**, *127*, 155–164. [CrossRef]
23. Nafil, A.; Bouzi, M.; Anoune, K.; Ettalabi, N. Comparative study of forecasting methods for energy demand in morocco. *Energy Rep.* **2020**, *6*, 523–536. [CrossRef]
24. Dumitru, C.-D.; Gligor, A. Wind energy forecasting: A comparative study between a stochastic model (arima) and a model based on neural network (ffann). *Procedia Manuf.* **2019**, *32*, 410–417. [CrossRef]
25. Rakpho, P.; Yamaka, W. The forecasting power of economic policy uncertainty for energy demand and supply. *Energy Rep.* **2021**, *7*, 338–343. [CrossRef]
26. Karia, A.A.; Bujang, I.; Ahmad, I. Fractionally integrated arma for crude palm oil prices prediction: Case of potentially overdifference. *J. Appl. Stat.* **2013**, *40*, 2735–2748. [CrossRef]
27. Wang, Z.-X.; Jv, T.-Q. A non-linear systematic grey model for forecasting the industrial economy-energy-environment system. *Technol. Forecast. Soc. Chang.* **2021**, *167*, 120707. [CrossRef]
28. Ma, X.; Lu, H.; Ma, M.; Wu, L.; Cai Y. Urban natural gas consumption forecasting by novel wavelet-kernelized grey system model. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105773. [CrossRef]
29. Qian, W.; Sui, A. A novel structural adaptive discrete grey prediction model and its application in forecasting renewable energy generation. *Expert Syst. Appl.* **2021**, *186*, 115761. [CrossRef]
30. Wang, Y.; Nie, R.; Ma, X.; Liu, Z.; Chi, P.; Wu, W.; Guo, B.; Yang, X.; Zhang, L. A novel hausdorff fractional ngmc (p, n) grey prediction model with grey wolf optimizer and its applications in forecasting energy production and conversion of China. *Appl. Math. Model.* **2021**, *97*, 381–397. [CrossRef]
31. Wang, Z.-X.; He, L.-Y.; Zheng, H.-H. Forecasting the residential solar energy consumption of the united states. *Energy* **2019**, *178*, 610–623. [CrossRef]
32. Moonchai, S.; Chutsagulprom, N. Short-term forecasting of renewable energy consumption: Augmentation of a modified grey model with a kalman filter. *Appl. Soft Comput.* **2020**, *87*, 105994. [CrossRef]
33. Xie, N.; Yuan, C.; Yang, Y. Forecasting China's energy demand and self-sufficiency rate by grey forecasting model and markov model. *Int. J. Electr. Power Energy Syst.* **2015**, *66*, 1–8. [CrossRef]
34. Piazza, A.D.; Piazza, M.C.D.; Tona, G.L.; Luna, M. An artificial neural network-based forecasting model of energy-related time series for electrical grid management. *Math. Comput. Simul.* **2021**, *184*, 294–305. [CrossRef]
35. Kobylinski, P.; Wierzbowski, M.; Piotrowski, K. High-resolution net load forecasting for micro-neighbourhoods with high penetration of renewable energy sources. *Int. J. Electr. Power Energy Syst.* **2020**, *117*, 105635. [CrossRef]

36. Al-Gabalawy, M.; Hosny, N.S.; Adly, A.R. Probabilistic forecasting for energy time series considering uncertainties based on deep learning algorithms. *Electr. Power Syst. Res.* **2021**, *196*, 107216. [CrossRef]
37. Katsatos, A.L.; Moustris, K.P. Application of artificial neuron networks as energy consumption forecasting tool in the building of regulatory authority of energy, athens, greece. *Energy Procedia* **2019**, *157*, 851–861. [CrossRef]
38. Bento, P.M.R.; Pombo, J.A.N.; Mendes, R.P.G.; Calado, M.R.A.; Mariano, S.J.P.S. Ocean wave energy forecasting using optimised deep learning neural networks. *Ocean. Eng.* **2021**, *219*, 108372. [CrossRef]
39. Abu-Salih, B.; Wongthongtham, P.; Morrison, G.; Coutinho, K.; Al-Okaily, M.; Huneiti, A. Short-term renewable energy consumption and generation forecasting: A case study of western australia. *Heliyon* **2022**, *8*, e09152. [CrossRef]
40. Somu, N.; Gauthama Raman, M.R.; Ramamritham, K. A hybrid model for building energy consumption forecasting using long short term memory networks. *Appl. Energy* **2020**, *261*, 114131. [CrossRef]
41. 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]
42. Etxegarai, G.; López, A.; Aginako, N.; Rodríguez, F. An analysis of different deep learning neural networks for intra-hour solar irradiation forecasting to compute solar photovoltaic generators' energy production. *Energy Sustain. Dev.* **2022**, *68*, 1–17. [CrossRef]
43. Gao, Y.; Ruan, Y.; Fang, C.; Yin, S. Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. *Energy Build.* **2020**, *223*, 110156. [CrossRef]
44. Hu, H.; Wang, L.; Peng, L.; Zeng, Y. Effective energy consumption forecasting using enhanced bagged echo state network. *Energy* **2020**, *193*, 116778. [CrossRef]
45. Hu, H.; Wang, L.; Lv, S. Forecasting energy consumption and wind power generation using deep echo state network. *Renew. Energy* **2020**, *154*, 598–613. [CrossRef]
46. Natarajan, Y.; Kannan, S.; Selvaraj, C.; Mohanty, S.N. Forecasting energy generation in large photovoltaic plants using radial belief neural network. *Sustain. Comput. Inform. Syst.* **2021**, *31*, 100578. [CrossRef]
47. Cui, Y.; Jia, L.; Fan, W. Estimation of actual evapotranspiration and its components in an irrigated area by integrating the shuttleworth-wallace and surface temperature-vegetation index schemes using the particle swarm optimization algorithm. *Agric. For. Meteorol.* **2021**, *307*, 108488. [CrossRef]
48. Zhang, F.; Deb, C.; Lee, S.E.; Yang, J.; Shah, K.W. Time series forecasting for building energy consumption using weighted support vector regression with differential evolution optimization technique. *Energy Build.* **2016**, *126*, 94–103. [CrossRef]
49. Wen, L.; Cao, Y. Influencing factors analysis and forecasting of residential energy-related CO2 emissions utilizing optimized support vector machine. *J. Clean. Prod.* **2020**, *250*, 119492. [CrossRef]
50. Mason, K.; Duggan, J.; Howley, E. Forecasting energy demand, wind generation and carbon dioxide emissions in ireland using evolutionary neural networks. *Energy* **2018**, *155*, 705–720. [CrossRef]
51. Hu, G.; Xu, Z.; Wang, G.; Zeng, B.; Liu, Y.; Lei, Y. Forecasting energy consumption of long-distance oil products pipeline based on improved fruit fly optimization algorithm and support vector regression. *Energy* **2021**, *224*, 120153. [CrossRef]
52. Abba, S.I.; Rotimi, A.; Musa, B.; Yimen, N.; Kawu, S.J.; Lawan, S.M.; Dagbasi, M. Emerging harris hawks optimization based load demand forecasting and optimal sizing of stand-alone hybrid renewable energy systems—A case study of Kano and Abuja, Nigeria. *Results Eng.* **2021**, *12*, 100260. [CrossRef]
53. Lu, H.; Ma, X.; Huang, K.; Azimi, M. Carbon trading volume and price forecasting in China using multiple machine learning models. *J. Clean. Prod.* **2020**, *249*, 119386. [CrossRef]
54. Lu, H.; Cheng, F.; Ma, X.; Hu, G. Short-term prediction of building energy consumption employing an improved extreme gradient boosting model: A case study of an intake tower. *Energy* **2020**, 117756. [CrossRef]
55. Fan, J.; Wang, X.; Zhang, F.; Ma, X.; Wu, L. Predicting daily diffuse horizontal solar radiation in various climatic regions of China using support vector machine and tree-based soft computing models with local and extrinsic climatic data. *J. Clean. Prod.* **2020**, *248*, 119264. [CrossRef]
56. Huang, G.; Wu, L.; Ma, X.; Zhang, W.; Fan, J.; Yu, X.; Zeng, W.; Zhou, H. Evaluation of catboost method for prediction of reference evapotranspiration in humid regions. *J. Hydrol.* **2019**, *574*, 1029–1041. [CrossRef]
57. Hong, T.; Xie, J.; Black, J. Global energy forecasting competition 2017: Hierarchical probabilistic load forecasting. *Int. J. Forecast.* **2019**, *35*, 1389–1399. [CrossRef]
58. Bedi, J.; Toshniwal, D. Energy load time-series forecast using decomposition and autoencoder integrated memory network. *Appl. Soft Comput.* **2020**, *93*, 106390. [CrossRef]
59. Adedeji, P.A.; Akinlabi, S.; Ajayi, O.; Madushele, N. Non-linear autoregressive neural network (narnet) with ssa filtering for a university energy consumption forecast. *Procedia Manuf.* **2019**, *33*, 176–183. [CrossRef]
60. Tayab, U.B.; Lu, J.; Yang, F.; AlGarni, T.S.; Kashif, M. Energy management system for microgrids using weighted salp swarm algorithm and hybrid forecasting approach. *Renew. Energy* **2021**, *180*, 467–481. [CrossRef]
61. Zhang, G.; Tian, C.; Li, C.; Zhang, J.J.; Zuo, W. Accurate forecasting of building energy consumption via a novel ensembled deep learning method considering the cyclic feature. *Energy* **2020**, *201*, 117531. [CrossRef]
62. Xiao, J.; Li, Y.; Xie, L.; Liu, D.; Huang, J. A hybrid model based on selective ensemble for energy consumption forecasting in China. *Energy* **2018**, *159*, 534–546. [CrossRef]

63. Khan, W.; Walker, S.; Zeiler, W. Improved solar photovoltaic energy generation forecast using deep learning-based ensemble stacking approach. *Energy* **2022**, *240*, 122812. [CrossRef]
64. Kazemzadeh, M.; Amjadian, A.; Amraee, T. A hybrid data mining driven algorithm for long term electric peak load and energy demand forecasting. *Energy* **2020**, *204*, 117948. [CrossRef]
65. Tran, D.; Luong, D.; Chou, J. Nature-inspired metaheuristic ensemble model for forecasting energy consumption in residential buildings. *Energy* **2020**, *191*, 116552. [CrossRef]
66. Liu, Z.; Wang, X.; Zhang, Q.; Huang, C. Empirical mode decomposition based hybrid ensemble model for electrical energy consumption forecasting of the cement grinding process. *Measurement* **2019**, *138*, 314–324. [CrossRef]
67. da Silva, R.G.; Dal Molin Ribeiro, M.H.; Moreno, S.R.; Mariani, V.C.; Leandro dos Santos Coelho, L. A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting. *Energy* **2021**, *216*, 119174. [CrossRef]
68. Härdle, W.; Liang, H.; Gao, J. *Partially Linear Models*; Springer: Berlin/Heidelberg, Germany, 2012.
69. Rudin, W. *Principles of Mathematical Analysis*; McGraw-Hill: New York, NY, USA, 1976; Volume 3.
70. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
71. Takahashi, N.; Guo, J.; Nishi, T. Global convergence of smo algorithm for support vector regression. *IEEE Trans. Neural Netw.* **2008**, *19*, 971–982. [CrossRef] [PubMed]
72. Chang, C.; Lin, C. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. TIST* **2011**, 2, 1–27. [CrossRef]
73. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Process Regression for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
74. Suykens, J.A.K.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [CrossRef]
75. De Brabanter, J.; De Moor, B.; Suykens, J.A.K.; Van Gestel, T.; Vandewalle, J.P.L. *Least Squares Support Vector Machines*; World Scientific: Singapore, 2002.
76. Ho, T.K. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832–844.
77. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
78. 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.
79. Dong, J.; Zeng, W.; Wu, L.; Huang, J.; Gaiser, T.; Srivastava, A.K. Enhancing short-term forecasting of daily precipitation using numerical weather prediction bias correcting with XGBoost in different regions of China. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105579. [CrossRef]
80. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T. Lightgbm: A highly efficient gradient boosting decision tree. In Proceedings of the NIPS'17: 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 3146–3154.
81. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. Catboost: Unbiased boosting with categorical features. In Proceedings of the NIPS'18: 32st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 3–8 December 2018; pp. 6638–6648.
82. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder–decoder approaches. In Proceedings of the SSST-8, 8th Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014; pp. 103–111.
83. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
84. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
85. Miranian, A.; Abdollahzade, M. Developing a local least-squares support vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *24*, 207–218. [CrossRef]
86. Duvenaud, D. Automatic Model Construction with GAUSSIAN Processes. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2014.