



Article Forecasting Short-Term Electricity Load Using Validated Ensemble Learning

Chatum Sankalpa ^{1,2}, Somsak Kittipiyakul ^{1,*} and Seksan Laitrakun ¹

- ¹ Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani 12000, Thailand
- ² Department of Electrical and Information Engineering, Faculty of Engineering, University of Ruhuna,
 - Galle 80000, Sri Lanka
- * Correspondence: somsak@siit.tu.ac.th

Abstract: As short-term load forecasting is essential for the day-to-day operation planning of power systems, we built an ensemble learning model to perform such forecasting for Thai data. The proposed model uses voting regression (VR), producing forecasts with weighted averages of forecasts from five individual models: three parametric multiple linear regressors and two non-parametric machine-learning models. The regressors are linear regression models with gradient-descent (LR), ordinary least-squares (OLS) estimators, and generalized least-squares auto-regression (GLSAR) models. In contrast, the machine-learning models are decision trees (DT) and random forests (RF). To select the best model variables and hyper-parameters, we used cross-validation (CV) performance instead of the test data performance, which yielded overly good test performance. We compared various validation schemes and found that the Blocked-CV scheme gives the validation error closest to the test error. Using Blocked-CV, the test results show that the VR model outperforms all its individual predictors.

Keywords: short-term load forecasting; time series forecasting model validation; ensemble learning; accuracy improvement; Thailand EGAT dataset

1. Introduction

The increased energy demand due to the rise in the population over the past few decades has been an eye-opener for the efficient use of energy all over the world. One of the critical objectives of energy forecasting is to allocate a sufficient and efficient energy supply to cater for the future demand. Many countries seek alternative energy resources to balance supply and demand [1]. Therefore, forecasting the required demand at least for a 1 day ahead has become a popular theme among energy providers to maintain that equilibrium.

Forecasting is divided into three subsections according to the prediction horizon: short-term load forecasting (STLF), medium-term load forecasting (MTLF), and long-term load forecasting (LTLF) [2]. Each plays a different role in the power system, benefiting supply and demand-side management. Accurate forecasting of a country's short-term electric energy demand is the key to making day-to-day decisions on hourly/day-ahead demand compared to the medium-term and long-term forecasts. Forecasting the electric energy demand is carried out by developing models using historical data, and many factors, such as climate conditions, calendar parameters, and some seasonal features [3]. Since Thailand is a tropical country, its electric demand is heavily influenced by climate and weather conditions, along with many public, religious, and long holidays. The limited research on Thai data found in the literature, which will be further elaborated in Section 2, suggests that the accuracy of short-term forecasting can be further improved with a better selection of features and using appropriate models.

Machine learning (ML) plays a leading role when predicting electricity demand compared to classical methods, such as statistical time series analysis and smoothing techniques.



Citation: Sankalpa, C.; Kittipiyakul, S.; Laitrakun, S. Forecasting Short-Term Electricity Load Using Validated Ensemble Learning. *Energies* **2022**, *15*, 8567. https:// doi.org/10.3390/en15228567

Academic Editors: Antonio Gabaldón, María Carmen Ruiz-Abellón and Luis Alfredo Fernández-Jiménez

Received: 6 October 2022 Accepted: 9 November 2022 Published: 16 November 2022

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



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). They have become popular among developers because of their convenient computer implementation and ability to produce both linear and nonlinear, parametric/non-parametric models suitable for the nature of the data. If sufficient historical data are available to train a model, supervised learning is used. Adequate feature engineering must also tackle a time series's seasonality, trend, and possible irregularities. Unarguably, there are many ML models already developed to forecast demand. For example, in [4], the authors applied regression and neural network (NN) to their dataset as ML methods. However, the prediction accuracy of such models is still questionable, since the model's variables and hyper-parameters were selected from the test performance on the test data.

Model evaluation is crucial in ML before launching the model in practice. Developers use various validation techniques to evaluate a model's performance to ensure that the model generalizes well for unseen data. After time series analysis, the most preferred technique is cross-validation (CV). However, the use of CV in the context of time series is questionable because of the potential non-stationarity and the serial correlation of data [5]. However, as proposed in [5], cross-validation can be used in time series if the series is stationary. Since it is hard to find studies on Thai electricity load data that have used validation techniques to select the best models, model variables, and parameters, we used cross-validation in this work. However, to use it, we needed to show that our data are stationary.

Many available supervised ML predictors are standalone learners. The accuracy of the predictions might be low when considered individually. However, the forecasts from multiple independent forecasters can be combined to form a better forecaster [6]. Regarding Thai data in our context, the use of EL is lacking in the literature. Therefore, this research focused on developing a cross-validated EL model to forecast the short-term electricity load in Thailand.

Main Contributions

The main contributions of this paper are summarized as follows:

- We extended our previous work in [4] using the same dataset and data grouping approach to use cross-validation. In our previous work, we used the test error to check the model's performance, which might have resulted in overly good test performance due to over-fitting the test data.
- We considered linear regression models with decision tree and random forest methods. We combined the forecasting models via a simple fixed-weight ensemble learning with the weights selected manually to improve the forecasting performance. Although simple, this ensemble learner outperforms all other individual forecasting models.
- We show via the augmented Dickey–Fuller (ADF) test that our data are stationary and have high confidence levels. This means we can use the cross-validation schemes for our time-series data.
- For each forecasting model, we evaluated the performances of three validation schemes and found that most of the time, the Blocked-CV scheme gives the best performance, and these results are also the closest to those of the ensemble model.
- Using this validation scheme, the test results show that the ensemble learning model outperforms all its individual predictors.

The remainder of the paper is structured as follows. Section 2 presents a detailed discussion of the related works. Section 3 presents the methods, including the data preparation, selection of the prediction horizon, and overall model design procedure, which includes the details of all the estimators used and their model evaluation schemes. Section 4 presents the results, extensively comparing the forecasting accuracies of the individual predictors based on the validation and the test errors. Finally, Section 5 concludes on the findings and highlights some limitations and possible future directions.

2. Related Works

2.1. Categories of Load Forecasting Based on the Prediction Horizon

As previously mentioned, STLF, MTLF, and LTLF are the three main categories of load forecasting based on different prediction horizons [2]. Some researchers introduce another category called very short-term load forecasting (VSTLF), which runs from only a few minutes to hours [7]. A comparison of forecasting in different time scales and examples of their applications are given in Table 1.

Forecasting	Prediction Horizon	Applications	References
VSTLF	Few minutes to hours	Predict instant electric demand, power consumption monitoring	[7,8]
STLF	Hours to weeks	Day-to-day energy management, economic dispatch (ED), unit commit- ment (UC)	[2,4,9]
MTLF	Weeks to months	Fuel allocation, system maintenance schedules, energy trading	[2,10]
LTLF	Months to years	Planning generation expansion, energy policy reforms	[8,11]

Table 1. Forecasting in different time scales and examples of their applications.

In common, there are several advantages of forecasting the electric energy demand, both economic and environmental [12]. One of the most straightforward benefits, if predictions are accurate, is that there will be no unintentional over-generation of energy so that it would reduce the costs associated with any possible wastage [13]. The same would reduce environmental impacts, such as the emission of CO_2 and global warming. On the other hand, any underestimation of demand can result in severe power outages, risking the power system's stability, security, and reliability.

The focus of this research is on SLTF, whose horizon usually runs from hours to a week. Energy providers use it to plan their day-to-day operations. Effective forecasting helps them to decide on when and which power plants to operate and/or shut down to cater to the required demand. That process is called unit commitment (UC), which ensures the optimum utilization of resources. Furthermore, the energy providers also aim to meet the required demand at a minimum cost. That is called economic dispatch (ED), which is beneficial not only for the providers but also for the consumers. ED and UC are considered direct benefits of STLF [2].

The works on SLTF for Thailand seem to have appeared only recently. Using the same dataset that we used in this work (i.e., the data from the Electricity Generating Authority of Thailand (EGAT) from 2009 to 2013), several works have been published [4,8,9,14–16]. The STLF model implemented in [4] implemented 1-day ahead forecasting by setting the prediction horizon as 10 to 34 h.

2.2. Factors Affecting the Electric Energy Demand

Due to the auto-regressive and periodic nature of electricity consumption, the historical or lagged load is a crucial factor to determine the electricity load. Huang and Shih [17] developed a univariate auto-regressive moving average (ARMA) model capable of handling both Gaussian and non-Gaussian processes. However, consideration of other factors which affect the electric energy demand is essential. As explained previously, the demand is also affected by weather/climate conditions, calendar parameters, and seasonal features. A time-varying periodic spline model with temperature as the exogenous variable was used in [18] and achieved the mean absolute percentage error (MAPE) value of about three. Using the same dataset for this study, Chapagain and Kittipiyakul [8] developed two MLR models and showed that the model with added temperature variables resulted in an at least 20% accuracy improvement.

In [4], calendar parameters such as year, month, day of the week, and hour; and seasonal features such as holidays and special days, were taken into account as driving factors for the demand and used, along with other essential factors, to develop many pre-

dictive models. In general, these factors are called deterministic variables. The demand on weekdays has significantly different patterns compared to the demand on weekends [19]. Specifically, weekday demand is considerably higher and more stable compared to weekend demand due to the presence of industrial sector operations on weekdays. Holidays, whether on weekdays or weekends, see much lower demand. Therefore, we needed to assign dummy variables to represent the seven days of the week as weekdays, weekends, and holidays so that the different day types would be clearly specified.

Since we used the same dataset as in [4], we used a similar set of dummy variables to represent the deterministic variables. Holidays/special days, such as bridging working days between weekends and a major holiday or days before and after major holidays, also affect the electric demand, and ignorance of their effects in a model can result in a severe dip in accuracy. Non-stationarity in the data series due to special days can be overcome by either introducing dummy variables or creating separate models for each hour or half-hour. With the same dataset we used, Chapagain et al. [4] selected both options and obtained better results. Since it is a Thai electric demand dataset, the special day factors considered were Newyear, Songkran, etc. To make the predictions more accurate, they introduced a series of dummy variables representing different holiday types and the days before/after special days, as in [20].

Introducing interaction variables helps to increase forecasting accuracy. For example, since the weekday demand is more sensitive to the weather parameters compared to the weekend demand, implementing interaction terms between the demand and the meteorological parameters, such as the temperature, after separating the weekdays and weekends, has resulted in a significant accuracy improvement [21]. Furthermore, since historical demand is used to predict the next day's demand, there is a possibility of an error in predicting the days around the margins between weekdays and weekends, since the weekend demands are significantly less than on weekdays. Therefore, interaction terms between the day-of-the-week dummy variables and the lagged load up to two days were introduced in [4]. They also included interaction terms between the day-of-week/month dummy variables and temperature to improve accuracy further.

2.3. Importance of Grouping the Dataset

Grouping a dataset into categories with similar demand patterns helps to improve the accuracy of predictions. For example, working days/weekdays, weekends, and holidays have their own demand patterns. Using the same EGAT dataset and grouping, Chapagain et al. [4] built separate models and achieved improved results compared to having a single model trained with the entire dataset. They achieved 1.81%, 1.74%, and 16.63% MAPE accuracy for weekdays, weekends, and holidays, respectively. Since the holiday's prediction accuracy became worst due to limited observational data, they went with a model trained with the entire dataset to predict the holiday demand and achieved an overall MAPE of 2.95%. A similar kind of grouping was carried out in [3] but using an energy dataset from a cold region in Japan. They achieved 0.9%, 1.81%, 2.51%, and 1.72% MAPE for weekdays, weekends, holidays, and overall demand, respectively.

Another effective two-fold grouping was also found in the literature as working days and holidays [22]. Srinivasan et al. [23] considered holiday demand to resemble the demand on Sundays and divided the dataset into working days, holidays and Sundays, and Saturdays. They achieved impressive performance with 1% MAPE for all three categories. Su and Chawalit [24], followed by [25], introduced seven groups of training data—Monday, Tuesday, etc.—considering the days of the week. The authors of the latter research eliminated the holiday and bridge-holiday effects by replacing them with the weighted average load of the same day of the week from the previous two weeks.

2.4. Predictive Methods

Methods used to build forecasting models can be classified into two main categories: classical methods and machine-learning methods. A few examples of classical methods

used in the literature are statistical time series analysis; and the ARMA model and its extensions, such as auto-regressive integrated moving average (ARIMA), seasonal auto-regressive integrated moving average (SARIMA), auto-regressive moving average with exogenous variable (ARMAX), and regression methods. However, their applications are limited to supporting only linear problems or limited forms of nonlinearity. A comparison between linear and nonlinear models for forecasting short-term electricity demand in Czech was performed in [22]. They concluded that forecasting the Czech electricity demand is almost a linear problem. The selected linear ARIMA model outperformed the nonlinear NN model in univariate and multivariate cases. Taylor et al. [26] compared six different univariate short-term demand forecasting models with an hourly dataset in Brazil and a half-hourly dataset in England. The double seasonal exponential smoothing model and the ARMA model outperformed all other models in the accuracy of the predictions.

An ARMAX model was developed using the same dataset we used for our study. By including temperature as the exogenous variable and another set of dummy variables, the short-term electric demand in Thailand was forecast [4]. The estimation methods used in the ARMAX model were MLR:OLS for non-serially correlated error cases and GLSAR for serially correlated error cases. These two estimation methods of the ARMAX model were then compared with a NN, and the results showed that the ARMAX model outperformed the NN. Another study was conducted in [27]. Three different ARMA(2,6) models were built to predict short-term electricity demand in Hokkaido, Japan. One of the models included some meteorological parameters—temperature, wind speed, relative humidity, solar irradiation, etc.—as the exogenous variables to make it an ARMAX model. The results showed that the ARMAX model had a performance improvement of at least 0.015% compared to the other two models.

Machine learning is now more often used for energy forecasting applications compared to classical methods. The main reason for this is their applicability in linear and nonlinear problems so that the seasonality issues can be quickly taken care of. The models developed in ML also can be parametric or non-parametric. ML models such as artificial neural networks (ANN), fuzzy logic, and support vector machines (SVM) are good examples in recent literature. Three machine learning models were developed in [28]—SVM, nonlinear auto-regressive (NAR) recurrent ANN, and long short-term memory (LSTM) ANN—to do multi-step ahead forecasting in residential microgrids. Those ML models outperformed an ARMA model. One of the problems of ML approaches such as SVM and ANN is their vulnerability to getting stuck at a local optimum during the training. Furthermore, an improved version of an LSTM model with the empirical wavelet transform (EWT) developed in [29] outperformed several benchmark ML models, resulting in MAPE values below 6% for three real-life cases.

ML models used for STLF with Thai data are limited in the literature. For example, a combined particle swarm optimization (PSO) algorithm that used the ANN technique to forecast the short-term electric demand in Thailand was developed in [15]. They achieved the overall training MAPE accuracy of 3.44%. Furthermore, they extended their work by introducing a hybrid PSO with a genetic algorithm (GA) and improved their accuracy to 2.86% [25]. Su and Chawalit [24] used more recent data than us to develop a deep neural network (DNN), SVM, and NN to forecast the short-term energy demand in Thailand. They found that the best test MAPE result among all the considered methods was 4.2% using one of the DNN methods.

Not only in electric energy management but in many other fields is time-series forecasting being used. A classical example is electricity price forecasting, as in [30,31]. They have used functional auto-regressive (FAR) models to optimally forecast the short-term electricity prices and demand in different electricity markets. Furthermore, they have proved that their component (deterministic and stochastic) estimation technique is highly effective at forecasting electricity prices based on the short-term [32] and the medium-term demand [33].

2.5. Model Validation Techniques

There is an apparent scarcity of time series research that uses model validation schemes to evaluate models and select the best hyper-parameters. The main reason for that gap is that many developers believe validation is meaningless in time series data, since they are prone to being serially correlated [5]. For independent and identically distributed (i.i.d) data series, CV is used extensively to randomly divide the whole training dataset into several validation sets for evaluation. Nevertheless, with time-series data, developers omit that, since the random selections could create holes in the dataset destroying the auto-correlative nature and could easily leak future information to the model [34].

Recently, several works [5,34,35] have started to look at when CV can be applied to time series and which CV techniques are most appropriate. Validation is divided into two categories: CV schemes and forward validation (FV) schemes. Each of them has its extensions. Since Random CV causes problems with time series data, Nielsen [34] suggested two FV schemes, EWFW and rolling window forward validation (RWFV), for model validation. However, they used the whole dataset, including the test set, to combine the full training, validation, and testing. In addition, Schnaubelt [35] compared eight different validation schemes with three different ML models—LR, RF, and NN—including both CV and FV extensions on non-stationary time-series data produced by a synthetic data-generating process (DGP). Six validation schemes were 5-fold, and the other two were derivations of a test-set-evaluation-like approach called last-block validation (LBV). The author compared the validation error of each scheme with the test error and selected the scheme, which yielded the minimum difference between them. The author concluded that the FV schemes perform well when the perturbation strength of the stationarity gets higher. However, with a closer look at the results, the author suggested that CV and FV schemes perform similarly for small perturbation strengths and prefer LBV if the perturbation strength is very high.

Validity of k-fold CV in time series forecasting under a few conditions was proved in [5]. The conditions allowing the use of k-fold CV are:

- 1. the AR process should be stationary,
- 2. the model should be purely auto-regressive,
- 3. the model should be nonlinear and non-parametric (preferably an ML model), and
- 4. the fitted model should have uncorrelated errors/residuals.

They compared three different CV schemes: 5-fold CV, 5-fold non-dependent CV (nonDepCV), and leave-one-out CV (LOOCV), with an LBV-like approach, called out-of-sample (OOS) evaluation, in a linear and a nonlinear model. They used standard multi-layer perceptron (MLP) NN as the nonlinear model. The results showed that the CV schemes outperformed the OOS evaluation. They applied the Ljung box test to check the serial correlation of residuals. They used an AR(5) model as the linear model, and the results reflected the same. The authors indirectly suggested that if the series is stationary and the model is linear, then CV can be applied without any problems.

It is not easy to find research on Thai electricity load data that has used validation techniques to evaluate the forecasting models. The ARMAX model used in [4] followed a similar RWFV-like training and testing pattern, as explained in [34], but real validation of the model could not be seen there.

2.6. Ensemble Methods

The method of producing a robust model by combining a diverse set of individual learners is called ensemble learning (EL) [6]. The simplest form of an ensemble in classification is the voting classifier (VC) which aggregates the predictions of each model. A similar form called the voting regressor (VR) is available for regression tasks. It takes the majority voted class in the case of classification, but it is the simple average or the weighted average for regression. In this work, we developed a voting regressor with weighted averaging as our EL model.

In addition to the voting EL models, there are three main methods of building more complex EL models: bagging, boosting, and stacking [6]. Unlike VR, bagging uses a single base estimator trained with different random samples of the entire dataset. An ensemble of multiple DTs called RF uses the bagging method to train its model. Boosting is quite different, as it can use different learners and tries to evolve by correcting the errors of each predecessor. In contrast, stacking uses at least two stages of predictions. The first stage can have multiple learners while using all of their predictions to train the second stage, and so on.

Many researchers recently adopted the ensemble approach due to the proven accuracy improvement in the predictions. A survey of load forecasting models for power system management was conducted [2]. They included information about linear models such as MLR, exponential smoothing, and nonlinear models such as NN. Finally, they preferred the ensemble approach to get an optimum forecast by combining those multiple predictive models with a probabilistic nature. Divina et al. [12] developed a stacking ensemble-learning model to forecast short-term electric energy demand using a dataset in Spain collected over nine years. Their model included two stages. In the first stage, they used three regression-based machine-learning models: evolutionary decision tree (EvTree), RF, and NN. Their predictions were then input to the second stage generalized boosted model (GBM) to have the ultimate predictions, which were finally found to be more accurate than all three first stage models.

EL methods have also been used to forecast electric demand in Thailand, particularly for medium-term forecasting. For example, using Thai electric peak demand data from 2002 to 2017, reference [10] developed and compared several medium-term forecasting models: ANN, SVM, deep belief network (DBN), and their ensembles. The EL models used a simple averaging method similar to VR to aggregate the predictions. The results showed that the ensemble model built by combining the ANN and DBN models outperformed other models for 1-month-ahead forecasting with a MAPE accuracy of 1.44%.

EL is used in other fields, such as electricity price forecasting, as in [36]. They modeled their stochastic component using an ensemble of ARMA, neural network auto-regressive (NNAR), RF, support vector regression (SVR), and GBM. The results suggest that the proposed EL method efficiently predicts electricity prices in the Italian electricity market (IPEX).

A summary of recently published state-of-the-art related work that used our full dataset or a part of the dataset to perform SLTF in Thailand is given in Table 2.

Dataset	Methods Used	Major Results	Work
2009–2013	OLS, GLSAR, FF-ANN	OLS and GLSAR models showed a better forecasting accuracy than FF-ANN.	[4]
2009–2013	OLS	Addition of interaction variables to the model has improved the prediction accuracy.	[14]
2009–2013	OLS and Bayesian esti- mation	Addition of a temperature variable to the model has improved the prediction accuracy by 20%.	[8]
2009–2013	MLR with AR(2)	Bayesian estimation provides better and more consistent performance than that of OLS estimation	[16]
2013	PSO with ANN	PSO outperforms the backpropagation training algorithm to train the ANN for STLF.	[15]
2013	PSO + GA with ANN	PSO + GA outperforms the backpropagation and PSO training algorithms.	[25]
2013	ANN	Addition of a temperature variable to the model has improved the prediction accuracy.	[9]

Table 2. A state-of-the-art comparison of recently published related work.

3. Methods

3.1. Data Preparation

The dataset used to train and test the models throughout this study was provided by EGAT, the leading player in the Thai energy market possessing almost 50% of the generation and 100% of transmission system. EGAT divides the country into five regions and keeps recordings of the electric loads separately. The central region, MCC, which includes Bangkok and nearby provinces with the highest electricity demand, is the focus of this research. The dataset includes 84,816 instances of half-hourly electricity load (in MW) and the corresponding half-hourly temperature measurements (in °C) for almost about five years (from 1 March 2009 to 31 December 2013). This dataset has been used in many studies to develop different forecasting models in recent years [4,8,14,16]. The final dataset provided for this study was a preprocessed version of [4], after filling in some missing load values and some temperature adjustments. The preprocessed data also include deterministic, meteorological, historical load, and interaction terms.

As in [4], the half-hourly data from 29 March 2009 to 31 December 2012 were selected as the training set, which included almost four years (65,952 instances); we kept aside the year 2013 for the test data (17,520 instances). We started training from 29 March 2009, since we used a 28-day lag load as one of the features in our models.

As grouping the dataset proved to simplify modeling and showed a significant accuracy improvement in the literature, as discussed in Section 2.3, the dataset was divided into four different groups, as illustrated in Figure 1, similarly to [4].



Figure 1. Data grouping.

Since the dataset included half-hourly (HH) data, each group was further divided into N = 48 subsets (of similar half-hours). Therefore, the training and test instances for each HH in each group were as follows.

- Group 1: train = 896, test = 239
- Group 2: train = 336, test = 87
- Group 3: train = 142, test = 39
- Group 4: train = 1374, test = 365

The corresponding training sets were used to train and validate the models. The model selection in our study was based on the validation error. We did not use a rolling window approach like in [4] to train and test the models. Therefore, the test sets were only used to check the performances of the models trained with all the training data.

3.2. Prediction Horizon

Since for each working day, EGAT collects load data up to 2 pm and makes forecasts for the next day's demand, the HH forecasts are generally 10 to 34 h ahead. Our study is also limited to predicting 10 to 34 leading hours of the next day, considering the data up

to 2 pm are available for the current day. However, the "next day" has quite a different meaning here, since the dataset is divided into groups. For example, it is 1-day ahead forecasting in Group 4. However, in the case of Group 2, where it uses only weekends to train the model, the next day for Saturdays is Sunday (1-day ahead), but for Sundays, it is the following Saturday (6-days ahead). Similarly, Groups 1 and 3 also have their unique interpretations for the "next day".

3.3. Model Design

The general workflow of this work is shown in Figure 2. It had five major steps: data preparation, model design, linear/nonlinear/ensemble estimation, model evaluation, and performance analysis. Details of the data preparation have already been discussed in Section 3.1.



Figure 2. Proposed methodology.

The overall model consists of five different individual predictors and their ensemble. Those individual predictors are a set of classical and ML regression-based methods. Classical methods include three MLR-based parametric ARMAX models, which use LR, OLS, and GLSAR as estimators; and ML methods include two nonlinear nonparametric estimators: DT and RF. The LR and OLS models are generally used for cases assuming uncorrelated errors, and the GLSAR is for correlated errors. DT and RF are selected due to their well-known adaptability to nonlinear nature of data.

For each group, these linear and nonlinear models were the training and test data as explained in Section 3.1. They commonly used the lag loads (1-day, 7-day, 14-day, 21-day, and 28-day), temperature as the exogenous variable, and some other deterministic and interaction variables to tackle the seasonal effects. Corresponding feature identifications for each group are presented in Appendix A.1. Since we did not use the rolling window approach [32] to estimate the models, the parameters remained constant for a given half-hourly model in a given group, irrespective of the day we predicted. Furthermore, a major difference experienced between linear and nonlinear models was that the model parameters of the MLR process could be known for the linear models but not the nonlinear models.

3.3.1. Parametric Models

The MLR-based ARMAX models used in this study were parametric models which used a fixed number of parameters/features. Therefore, we built an equation representing the relationship between the target variable (load) and the features. The features included deterministic variables, temperature, lag loads, and interaction terms. Since each group's datasets was divided into 48 subsets, each model included 48 individual equations to perform a 1-day forecast. The demand at day *d* and half-hour *h* was modeled as

$$D_{h,d} = \sum_{i=1}^{n} \alpha_i \cdot (Dt_{h,d})_i + \sum_{i=1}^{m} \beta_i \cdot (Tmp_{h,d})_i + \sum_{i=1}^{p} \theta_i \cdot (LL_{h,d})_i + \sum_{i=1}^{q} \gamma_i \cdot (It_{h,d})_i + \mu_{h,d}$$
(1)

where $Dt_{h,d}$, $Tmp_{h,d}$, $LL_{h,d}$, and $It_{h,d}$ are groups of deterministic, temperature, lag loads, and interaction terms, respectively. Similarly, α , β , θ , and γ are groups of their corresponding coefficients. The error term $\mu_{h,d} \sim N(0, \Sigma)$ is vital to address, since it tends to be serially correlated with errors in previous days. Depending on the properties of Σ , the model is classified as [4]:

- 1. OLS/LR: for i.i.d errors $\Sigma = I$
- 2. GLSAR: for AR(p) errors $\Sigma = \Sigma(\rho)$

Therefore, we used OLS and LR models with the assumption of uncorrelated errors and the GLSAR with AR(1) structure for the correlated errors in our study. The mathematical forms for OLS and LR models were quite similar and followed the exact representation of the OLS model given in [4]. However, during the parameter estimation process, OLS used the matrix inversion to minimize the loss function, and LR used the iterative gradient-descent algorithm. GLSAR also used the same mathematical form given in [4] with correlated errors, but we selected the AR(1) structure. The Durbin–Watson (DW) test on the errors/residuals of the models was conducted to check for any serial correlation. Its test statistic usually ranges from 0 to 4. The values between 1.5 and 2.5 suggest no significant serial correlation of errors.

3.3.2. Nonparametric Models

The DT and the RF were the nonlinear nonparametric models used in this study. Compared to parametric models, DT and RF do not have a fixed set of features in a predetermined form but can be adjusted according to the information derived from the data. Therefore, it was impossible to formulate a relationship between the features and the target variable as we did with the parametric models.

DT is a robust ML algorithm that forms a tree-like structure to predict the target variable based on the decision rules induced by the features [6]. Each internal node, branch, and leaf node of the tree consist of a test conducted on a feature, the outcome of the test, and the final decision which predicts the target variable, respectively. Usually, DT models are prone to over-fitting, since they have high flexibility. The over-fitting could be overcome by fine-tuning their hyper-parameters. By comparing the CV MAPE of a DT model with different random sets of hyper-parameters, as shown in Table A3 of Appendix A.2, we selected a good hyper-parameter combination to proceed (random_state = 42, max_depth = None, min_samples_split = 10, min_samples_leaf = 5, max_features = None, max_leaf_nodes = None, etc.).

RF is an ensemble of several DTs trained on random subsets of instances and the dataset's features. It uses averaging to improve the accuracy of the prediction and controls the over-fitting [6]. It usually uses bootstrap sampling with the max_samples set to the size of the train set. The hyper-parameters for the RF model were also selected in a similar approach to DT (n_jobs = -1, random_state = 42, n_estimators = 100, max_depth = None, min_samples_split = 4, min_samples_leaf = 2, max_features = 1.0, etc.), as shown in Table A4 of Appendix A.2. One issue is the time it takes to train several DTs, which was compensated for by allocating all available cores on the machine.

3.3.3. Ensemble Model

The EL model was developed by optimally combining the five aforementioned individual predictors' outputs. This is a VR model which uses the weighted-averaging method to predict the demand. For simplicity, the corresponding weights (w_1, \ldots, w_n) were realized by a trial-and-error method giving higher weights for individual models with higher accuracies. The predicted demand using the VR model at day *d* and half-hour *h* can be modeled as

$$(\hat{D}_{h,d})_{VR} = \sum_{i=1}^{n} w_i . (\hat{D}_{h,d})_i,$$
 (2)

where *n* and $(\hat{D}_{h,d})_i$ represent the number of individual predictors (n = 5 in our case) and the predicted demand with the *i*th individual model at day *d* and half-hour *h*, respectively. The method of identifying the best set of weights for the VR model is presented in Appendix A.3. It is expected that VR has the best prediction accuracy compared to all other individual models. Although there are various performance metrics, we use the mean absolute percentage error (MAPE), defined as

$$MAPE = \frac{1}{N|D|} \sum_{k=1}^{N} \sum_{\alpha \in D} \left| \frac{D_{h,d} - \hat{D}_{h,d}}{D_{h,d}} \right| \times 100\%,$$
(3)

where N = 48; D represents the set of test days; and $D_{h,d}$ and $\hat{D}_{h,d}$ represent the actual demand and predicted demand, respectively, at day d and half-hour h [4].

3.3.4. Model Evaluation

Our work used the validation error to select the best model, model features, and model parameters. Validation was conducted with the training dataset, which provided an additional measure of the model's accuracy before testing with unseen data. We used three different 4-fold validation schemes, Random CV, Blocked-CV, and EWFV, on the training set (2009–2012). They were set to 4-fold validation, since the training set spanned four years. Figure 3 illustrates the three validation schemes.



Figure 3. Illustration of validation schemes. Each split consisted of the corresponding validation fold (orange) and the training fold (blue). Since there were four data splits, we used four validation folds for each scheme.

If the *i*th data split of a validation scheme *x* has the validation error of $e_{i,x}$, the overall validation error $(E_{v,x})$ for scheme *x* is given as the average across all splits:

$$E_{v,x} = \frac{1}{s} \sum_{i=1}^{s} e_{i,x}, \tag{4}$$

where *s* represents the number of splits (s = 4 in our case).

We used the standard *k*-fold CV as the Random CV here with k = 4. The issue with this scheme for time series data is evident, since it randomly selects the validation folds without considering any serial correlations/dependencies of the data. Furthermore, this randomness could cause a validation fold to become a weaker representative of the full-

time series dataset. However, it is still applicable if the time series is found to be almost stationary [35], and that is one primary reason for us to find the stationarity of the data series at the beginning before putting them into the models.

Blocked-CV seems the most attractive scheme, since it uses each year from 2009 to 2012 as its four validation folds, making each fold uniform and a clear representation of the entire dataset. For example, if any annual seasonality persists on the entire dataset, a one-year fold can represent that. However, the Blocked-CV still has a dependency issue, but at a lower level compared to Random CV, since the selection of the validation fold is no longer random, and the validation fold for each split breaks precisely at the beginning/end of the year.

The previous two schemes are categorized as CV schemes, and the last, EWFV, as an FV scheme. The main difference between CV and FV is that the latter never uses future data to predict the past data so that it can adapt well to deal with possible non-stationarity in time series data [35]. This difference also benefits validation with time series data, providing a minimum dependency issue. One major characteristic of EWFV is that it uses validation folds with a fixed size while increasing the size of the training folds with every split. For example, we used data from March 2009 to December 2010 to create the training fold of the first split and selected only the following six months (from January 2011 to June 2011) as the validation fold. Therefore, we had the disadvantage of inefficient data usage, since the initial splits did not cover all available data.

3.3.5. Performance Analysis

As stated in Equation (3), the performance index used to measure the accuracy of the predictions in this study was the MAPE value. The usual practice is to use the test accuracy (or the test MAPE value) to decide the final model. However, the overall model developed here is capable of determining all the training, validation (for all three schemes discussed in Section 3.3.4), and test MAPE values denoted by E_{train} , $E_{v,x}$, and E_{test} , respectively, for each model, including the EL model in each group.

For each model in each group, the validation scheme with the minimum difference between the $E_{v,x}$ and E_{test} values in the majority was chosen as the final validation scheme. Furthermore, the model with the minimum validation MAPE under that chosen validation scheme denoted by E_{v,x^*} in each group was selected as the final model. It was expected that the minimum validation error $E_{v,x}$ would lead the final model to reflect the minimum test error E_{test} too, which is proved with experimental results in Section 4.

4. Results and Discussion

We first show the results of the stationarity test of our data to show that it is stationary and has high confidence levels, and hence validation schemes can be applied. Then, we performed the training and validation of different models. We then evaluated the models with their best hyper-parameters and/or features on the test data to compare the test performance with the validation performance.

4.1. Checking for the Stationarity of Time-Series Data

It is required to analyze the given time series dataset beforehand to check for stationarity. If any non-stationarity persists, it may result in spurious regressions. Furthermore, prior identification of the stationarity/non-stationarity of the time series is essential when selecting an appropriate validation scheme [35]. For example, CV uses random samples at the risk of not being a clear representation of the entire data series due to possible non-stationarity.

The load profile for the whole EGAT electricity data used in this paper is plotted in Figure 4. From a visual inspection of the plot, it might seem that some parts of the data, especially during the transition of 2011 and 2012 when a heavy flood occurred, may be non-stationary or trend away from being stationary. Monthly, weekly, and daily electricity load profiles shown in Figure 5 also seem to have seasonal effects. However, since we

divided our dataset into groups and further into half-hours, we only need to focus on the stationarity of each half-hourly series. To test for stationarity, we employed the augmented Dickey–Fuller (ADF) test [37], which tests the null hypothesis that a unit root is present in the time series data. The ADF test statistic was computed, and if it was less than the relevant critical value of the Dickey–Fuller (DF) test, the null hypothesis could be rejected. There are three versions of the ADF and DF tests [38]: test for a unit root with no constant and trend terms, test with only a constant term, and test with both constant and trend terms. For our dataset, we used the version with only a constant term, since it means stricter stationarity than the trend of stationarity. The ADF tests for the whole dataset and individual half-hourly series show that we can reject the null hypothesis at high confidence levels (more than 90%).



Figure 4. Half-hourly electricity load profile from 1 March 2009 to 31 December 2013.

For the whole dataset, the ADF test results shown in Table 3 show that we can reject the null hypothesis at almost 100% confidence. The test statistic is -18.18, much less than the critical value of -3.43 for 99% confidence. Hence, the test result shows that the whole dataset is stationary, even without the trend term. Among all the half-hourly data series (in all four groups) used in our study, the ADF test results show that about 80% of the series are stationary at a 95% confidence level. In contrast, the other series are stationary at slightly less confidence levels (around 90% to 95%). Therefore, the ADF test results confirm that our dataset, either as the whole or the sub-scale of each group, is stationary at high confidence levels. This result assisted in conveniently implementing the validation schemes discussed in Section 3.3.4.

Table 3. ADF test result for the full dataset.

Test Statistic	-18.18469
<i>p</i> -value	$2.427935 imes 10^{-30}$
#lags used	64
number of observations used	84751
critical value (1%)	-3.430427
critical value (5%)	-2.861574
critical value (10%)	-2.566788



Figure 5. (a) Monthly variation in electric load for the year 2010. (b) Weekly variation in electric load for May 2011. (c) Daily variation in electric load for the period 22–28 May 2011. (d) Daily load curves (half-hourly variation) for the period 22–28 May 2011.

4.2. Model Selection

The key quantitative task of this study was to train, validate, and test the models. There were six models, including the VR, for each group to proceed with. Corresponding MAPE values determined using Equation (3) for each group are presented in Table 4.

The "Error type" column in Table 4 includes the training MAPE E_{train} ; the validation MAPE $E_{v,x}$, where x denotes the corresponding validation scheme (Random CV, Blocked-CV, and EWFV as *Random CV*, *Blocked* – *CV*, and *EWFV*, respectively); and the test MAPE E_{test} . Note that a bold $E_{v,x}$ value is the minimum among all three validation schemes for the corresponding model in each group. Blocked-CV dominates, with 75% of $E_{v,x}$ values among all groups. Surprisingly, in groups 1 and 4, we see some unrealistic error figures for Random CV and EWFV with LR and VR models. After an analysis, we found that some predictions of LR and VR models at the half-hour HH = 28 (i.e., at 2:00 p.m. where the usual weekday peak occurs) are extraordinarily large with those particular validation schemes. The issue primarily appeared in the LR model, which uses the iterative gradient-descent algorithm to minimize the loss function. The VR model is also affected, since it uses the LR model as one of the individual predictors, even through it is given a small weight. Nevertheless, the OLS model, which uses the matrix inversion technique for coefficient estimation, generated no such large errors.

Group #	Error Type	LR	OLS	GLSAR (rho = 1)	DT	RF	VR
	E _{train}	1.7929	1.7929	1.8809	1.6775	1.1043	1.6816
	E _{v.Random CV}	$2.0111 imes 10^6$	2.6740	2.6754	4.2787	3.4503	2.6257
1	$E_{v,Blocked-CV}$	2.6113	2.6110	2.6142	4.2785	3.4557	2.5636
	$E_{v,EWFV}$	$3.4955 imes10^6$	3.1891	3.1891	4.2930	3.7216	$6.9911 imes 10^5$
	E_{test}	1.9055	1.9053	1.9661	3.6142	3.0299	1.8817
	E _{train}	1.6163	1.6163	1.6274	1.6718	1.1103	1.5171
	E _{v,Random CV}	2.7604	2.7598	2.7814	4.2924	3.4702	2.6976
2	$E_{v,Blocked-CV}$	2.7506	2.7505	2.7535	4.2450	3.4046	2.6665
	$E_{v,EWFV}$	3.0149	3.0132	3.0074	4.2301	3.3950	2.8311
	Etest	2.0161	2.0161	2.0201	3.6391	3.0927	1.9458
	E _{train}	4.4025	4.4025	4.6923	6.8680	4.6377	4.1866
	E _{v,Random} CV	9.2514	9.2524	9.3811	15.4341	12.6259	8.8199
3	$E_{v,Blocked-CV}$	8.2515	8.2530	8.2161	15.4770	12.3826	7.9559
	$E_{v,EWFV}$	10.8780	10.8790	10.9010	14.8387	12.4338	10.3365
	E _{test}	5.5077	5.5084	5.4306	13.7666	11.7341	5.1858
	E _{train}	2.3238	2.3225	2.4048	2.3436	1.5115	2.1789
	E _{v,Random CV}	$1.8061 imes 10^6$	3.2325	3.2329	5.3151	4.1512	$8.1916 imes10^3$
4	$E_{v,Blocked-CV}$	3.1880	3.1817	3.1822	5.3634	4.2053	3.0910
	$E_{v,EWFV}$	$3.7727 imes 10^6$	3.7141	3.7106	5.2222	4.3171	$7.5455 imes10^5$
	E_{test}	2.4577	2.4564	2.5760	4.8106	4.0102	2.4549

Table 4. Training, validation, and test MAPE values (in %) for all groups.

The findings from Table 4 are not strong enough to justify how good the Blocked-CV scheme is compared to the other two schemes. One suggestion is to compare the validation performance with respect to the test performance [35]. The difference between these two errors ($E_{v,x} - E_{test}$) for each model in each group is listed in Table 5.

Group #	Validation Scheme	LR	OLS	GLSAR (rho = 1)	DT	RF	VR
1	Random CV Blocked-CV EWFV	$\begin{array}{c} 2.0111 \times 10^{6} \\ \textbf{0.7058} \\ 3.4955 \times 10^{6} \end{array}$	0.7687 0.7057 1.2838	0.7093 0.6481 1.223	0.6645 0.6643 0.6788	0.4204 0.4258 0.6917	$\begin{array}{c} 0.7440 \\ \textbf{0.6819} \\ 6.9911 \times 10^5 \end{array}$
2	Random CV Blocked-CV EWFV	0.7443 0.7345 0.9988	0.7437 0.7344 0.9971	0.7613 0.7334 0.9873	0.6533 0.6059 0.5640	0.3775 0.3119 0.3023	0.7518 0.7207 0.8853
3	Random CV Blocked-CV EWFV	3.7437 2.7438 5.3703	3.744 2.7446 5.3706	3.9505 2.7855 5.4704	1.6675 1.7104 1.0700	0.8918 0.6485 0.6997	3.6341 2.7701 5.1507
4	Random CV Blocked-CV EWFV	$\begin{array}{c} 1.8061 \times 10^{6} \\ \textbf{0.7303} \\ 3.7727 \times 10^{6} \end{array}$	0.7761 0.7253 1.2577	0.6569 0.6062 1.1346	0.5045 0.5528 0.4116	0.1410 0.1951 0.3069	$\begin{array}{c} 8.1892 \times 10^{3} \\ \textbf{0.6361} \\ 7.5455 \times 10^{5} \end{array}$

Table 5. Differences between validation MAPE and test MAPE for all groups.

According to [35], the best validation scheme has its validation MAPE close to the test MAPE. For example, for data group 1 and the OLS model, the Blocked-CV has the minimum MAPE difference of 0.7057%; compare that to 0.7687% for Random CV and 1.2838% for EWFV. Furthermore, for all the linear models (LR, OLS, and GLSAR) and the VR model, the Blocked-CV was the best validation scheme for all data groups.

However, for the nonlinear models (DT and RF), there was no dominating validation scheme among all data groups. Nevertheless, note that for DT, the EWFV performed significantly better than other schemes for data groups 2 to 4. One of the reasons for such

varying performance among validation schemes may be that the hyper-parameters of these nonlinear models may have stuck in some local optima. Furthermore, the splits of DT and RF were optimized based on the default mean-squared error (MSE), not the MAPE.

If we consider all the bold minimum values in Table 5, 75% (majority) of that values resulted from the Blocked-CV scheme. Therefore, we selected it as the best validation scheme to proceed, as per our suggestion in Section 3.3.5. However, although the Blocked-CV dominates, Random CV performed very similarly to it. EWFV was significantly worse than the other two schemes.

Note that a 75% majority is the same figure we found in Table 4 too. This implies that a minimum $E_{v,x}$ results in a minimum $E_{v,x} - E_{test}$ for each model with each group of our study. For a given model with a given group, the corresponding E_{test} value is always less than all three $E_{v,x}$ values.

Since we planned to select the best model based on the minimum validation error of the selected scheme (Blocked-CV in our case), we can see from Table 4 that the VR performs better than all other individual models, resulting in minimum $E_{v,Blocked-CV}$ values of 2.5636%, 2.6665%, 7.9559%, and 3.0910% for Groups 1, 2, 3, and 4, respectively. Further, as per our expectation in Section 3.3.5, the minimum $E_{v,Blocked-CV}$ also led the final VR models to reflect a minimum test MAPE, resulting in E_{test} values of 1.8817%, 1.9458%, 5.1858%, and 2.4549% for Groups 1, 2,3, and 4, respectively.

In addition, another important result when looking at the Blocked-CV error and the test error in Table 4 is that the parametric MLR models always performed better than nonparametric ML models on our dataset. For example, the $E_{v,Blocked-CV}$ values for LR, OLS, and GLSAR models in Group 1 are 2.6113%, 2.6110%, and 2.6142%, respectively. However, those for DT and RF models are comparatively higher, at 4.2785% and 3.4557%, respectively. E_{test} values for Group 1 also convey a similar pattern.

Note that the MAPE values shown in Table 4 are the average values across all half hours. To see what the MAPE is in each half-hour, we plotted the validation and test MAPE values for each half-hour in Figure 6 and Figure 7, respectively.



Figure 6. Cont.



Figure 6. Comparison of the Blocked-CV MAPE values obtained by each half-hourly model in each group; (a) Group 1. (b) Group 2. (c) Group 3. (d) Group 4.

The errors of DT and RF models are comparatively higher than those of the other models for each half-hour in each group, and they are worse during off-peak times. The use of MSE instead of MAPE to measure the quality of the split in default for these two nonparametric models could be one reason for this. The other three parametric models always produced lower error values than DT and RF. Overall, the results suggest that the VR model (green color curve) in each group excels in predicting with lower values for both $E_{v,Blocked-CV}$ and E_{test} .



Figure 7. Cont.



Figure 7. Comparison of the test MAPE values obtained by each half-hourly model in each group. (a) Group 1. (b) Group 2. (c) Group 3. (d) Group 4.

In Appendix A, we show additional results on the feature identification for the models in each group, hyper-parameter tuning of the nonlinear models, weight identification of the individual predictors in the VR model, $E_{v,Blocked-CV}$ and E_{test} values obtained by each half-hourly model in each group, the graphical representation of the forecasting performance of the VR model in each group, and the auto-correlation function (ACF) and partial auto-correlation function (PACF) plots of the VR model residuals (HH = 28) in each group.

5. Conclusions and Future Work

This paper presented a trending ML approach using EL to perform SLTF in Thailand accurately. The individual predictors in the EL model consisted of LR, OLS, and GLSAR—parametric predictors; and DT and RF as nonparametric predictors. The dataset provided by EGAT was divided into four groups and into half-hours for the convenience of modeling.

Three different model validation schemes were implemented to evaluate the models, and the Blocked-CV was then selected as the best scheme based on the results. The ADF test results greatly supported the validity of applying model validation on time series data, as they show that all sub-data series used in this study are stationary at high confidence levels.

The proposed EL model resulted in the minimum Blocked-CV MAPE values of 2.5636%, 2.665%, 7.9559%, and 3.0910% for Groups 1, 2, 3, and 4, respectively; and the minimum test MAPE values of 1.8817%, 1.9458%, 5.1858%, and 2.4549% for Groups 1, 2, 3, and 4, respectively. MAPE values for forecasting holiday loads (Group 3) are quite high because of the limited amount of training data available. However, they are still competitive compared to the holiday models in previous research [4]. Grouping the dataset into weekdays (Group 1) and weekends (Group 2) proved its worth, since this resulted in lower values in both validation MAPE and test MAPE compared to the full dataset (Group 4). Furthermore, for our data, when considering the forecasting accuracy of the individual predictors, all MLR-based parametric regressive models outperformed the nonparametric DF and RF models.

Let us discuss some limitations and the future directions of our work. Since the dataset used in this study was somewhat old (2009–2013), the developed models may not be efficiently used with more recent data. Therefore, an essential extension for this study

might be to use more recent data to build models. The dataset size could also be increased based on the availability to build robust models and achieve better prediction accuracy.

The EL model in our study only consists of classical and ML models. Although [4] showed that a simple feed-forward neural network performed worse than linear regression models with our dataset, it would be interesting to see how our EL model would perform by adding some DL models such as neural networks to its aggregation. At the same time, it is essential to see how more modern techniques would perform with respect to the regression techniques within the cross-validation sets. In addition, our EL model was developed as a voting regressor with a weighted average. However, stronger and more complex EL models could be developed using bagging, boosting, or stacking methods, which could also be a possible extension for this study to make the EL model more accurate.

Furthermore, we found some important statistical tests used in the literature to verify the superiority of the models: Diebold and Mariano (DM) test [33] and Wilcoxon signed-rank (WSR) test [29]. Those tests will be used in our future work to validate the results.

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

Funding: This research received no external funding.

Data Availability Statement: The dataset is not publicly available.

Acknowledgments: We want to extend our sincere gratitude to Chawalit Jeenanunta and the EGAT for providing the original dataset used in this research, and the Sirindhorn International Institute of Technology (SIIT), Thammasat University for providing the necessary facilities for this work.

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

Abbreviations

The following abbreviations are frequently used in this manuscript:

STLF	Short-Term Load Forecasting
ML	Machine Learning
EL	Ensemble Learning
VR	Voting Regression
MLR	Multiple Linear Regression
LR	Linear Regression
OLS	Ordinary Least Square
GLSAR	Generalized Least Square Auto Regression
DT	Decision tree
RF	Random Forest
TSF	Time Series Forecasting
CV	Cross-validation
EWFV	Expanding Window Forward Validation
EGAT	Electricity Generating Authority of Thailand
MAPE	Mean Absolute Percentage Error
ARMAX	Auto-regressive Moving Average with Exogenous Variable
FV	Forward Validation
HH	Half-hour
ADF	Augmented Dickey-Fuller
DW	Durbin-Watson

Appendix A. Extended Results

Appendix A.1. Feature Identification

Since our dataset is a preprocessed version with many deterministic features and meteorological parameters such as temperature, lag loads, and interaction terms, we selectively utilized those available features for each group. Note that, to preserve uniformity, we kept the selected set of features common to all the models in a given group. However, selecting the best set of features among many available features for any model is challenging. We have used two strategies: the correlation test and the CV. We selected the initial set of features based on the standard correlation coefficient (also called Pearson's r) [6], which analyzes the correlation between each feature and the load. Since the Blocked-CV was expected to be promising due to its structure, we used it to fine-tune and filter the best set. The resulting set of features allocated to the corresponding group/s are presented in Table A1.

Table A1. List of selected input features in each group.	
--	--

Types	Features	Description	Group	
	WD	Week day dummy (Mon, Tue,, Sat, Sun)	3	
	MD	Month dummy (Jan, Feb,, Nov, Dec)	1,2,4	
	DayAfterHoliday	Binary 0 or 1	1,2,4	
Deterministic	DayAfterLongHoliday	Binary 0 or 1	1,4	
Deterministic	DayAfterSongkran	Binary 0 or 1	1,4	
	Flood	Binary 0 or 1	1,2,4	
	Year	Year (2009,, 2013)	1,2,3,4	
	HolidayType	Type of the holiday (Songkran, Newyear,, etc)	-	
	MaxTempYesterday	1-day ahead maximum temperature	1,2,3	
Temperature	MaxTemp	Maximum forecasted temperature	1,2	
	MA2pmTemp	Moving avearage of temperature at 2:00 p.m.	2	
	Temp	Forecasted temperature	-	
	Load2pmYesterday	1-day ahead load at 2:00 p.m.	1	
	load7d	7-days ahead load	1,2,3,4	
Lagged	load14d	14-days ahead load	1,2,3,4	
	load21d	21-days ahead load	1,2,3,4	
	load28d	28-days ahead load	1,2,3,4	
	load1d_cut2pm	1-day ahead untill 2:00 p.m. and 2-days ahead after 2:00 p.m. load	-	
	load2d_cut2pm	2-days ahead untill 2:00 p.m. and 3-days ahead after 2:00 p.m. load	-	
	WD:Temp	Interaction of week day dummy to temperature	2,3,4	
	MD:Temp	Interaction of month dummy to temperature	1,4	
	WD:load1d_cut2pm	Interaction of week day dummy to load1d_cut2pm	1,3,4	
Interaction	WD:load2d_cut2pm	Interaction of week day dummy to load2d_cut2pm		
	WD:Load2pmYesterday	Interaction of week day dummy to Load2pmYesterday	2,3,4	
	HolidayType:Load2pmYesterday	Interaction of HolidayType dummy to Load2pmYesterday	3,4	
	MD:load1d_cut2pm	Interaction of month dummy to load2d_cut2pm	2	

The correlation matrix of some features used in this research is presented in Table A2. Since our models use more than 115 features, we included only the first seven features with the highest correlations with the load.

	load	load7d	load14d	load21d	load28d	load1d _cut2pm	temp	load2d _cut2pm
load	1.0000	0.8227	0.8159	0.8019	0.7950	0.7578	0.6728	0.6598
load7d	0.8227	1.0000	0.8239	0.8216	0.8051	0.6545	0.5740	0.6057
load14d	0.8159	0.8239	1.0000	0.8250	0.8219	0.6369	0.5755	0.5703
load21d	0.8019	0.8216	0.8250	1.0000	0.8260	0.6306	0.5555	0.5717
load28d	0.7950	0.8051	0.8219	0.8260	1.0000	0.6208	0.5494	0.5543
load1d_cut2pm	0.7578	0.6545	0.6369	0.6306	0.6208	1.0000	0.6221	0.8103
temp	0.6728	0.5740	0.5755	0.5555	0.5494	0.6221	1.0000	0.5966
load2d_cut2pm	0.6598	0.6057	0.5703	0.5717	0.5543	0.8103	0.5966	1.0000

Table A2. Correlation matrix for the first seven features with the highest correlations with the load.

Appendix A.2. Hyper-Parameter Tuning of the Nonlinear Models

Since RF is made of several DTs, they have many hyper-parameters in common. Selecting the best set of hyper-parameters for a nonlinear model is always challenging, and there is a high chance that they will get stuck at a local optimum. The strategy used in our study was comparing the $E_{v,Blocked-CV}$ of nonlinear models with different random sets of hyper-parameters. We changed only the most significant hyper-parameters while leaving others at their default settings [6]. The sets that offer the minimum $E_{v,Blocked-CV}$ for each DT and RF model were then selected as the final hyper-parameter sets.

The hyper-parameter tuning process of the DT model for Group 1 is shown in Table A3. The hyper-parameters not used in the tuning process were left at their default settings.

Table A3. Performance of the DT model with different sets of hyper-parameters (Group 1).

random _state	max _depth	min_samples _split	min_samples _leaf	max _features	max_leaf _nodes	E _{train}	$E_{v,Blocked-CV}$	Etest
42	max	2	1	max	max	0	4.6386	4.0080
42	20	2	1	max	max	0.0225	4.6249	4.0110
42	5	2	1	max	max	2.7819	4.6158	3.8731
42	5	4	2	max	max	2.8021	4.6111	3.8433
42	5	4	2	15	10	3.7598	4.6004	4.8738
42	10	2	1	max	max	1.0252	4.5467	3.9288
42	max	4	2	max	max	0.7222	4.4841	3.8579
42	20	4	2	max	max	0.7278	4.4792	3.8521
42	max	10	5	max	max	1.6775	4.2785	3.6142

The set of hyper-parameters in the last row of Table A3, shown in bold, resulted in minimum $E_{v,Blocked-CV}$ and E_{test} values for the DT model. Not only did this occur with Group 1, but it also produced similar results with other groups. Therefore, that particular set of hyper-parameters and those left at their default settings were selected to build the DT model in each group as follows.

DecisionTreeRegressor(*, criterion = 'squared_error', splitter = 'best', max_depth = None, min_samples_split = 10, min_samples_leaf = 5, min_weight_fraction_leaf = 0.0, max_features = None, random_state = 42, max_leaf_nodes = None, min_impurity_decrease = 0.0, ccp_alpha = 0.0).

The hyper-parameter tuning process of the RF model for Group 1 is shown in Table A4. The hyper-parameters not used in the tuning process were also left at their default settings with this model.

Similarly, the set of hyper-parameters in the last row of Table A4 shown in bold and those left at their default settings were selected to build the RF model in each group as follows.

RandomForestRegressor(n_estimators = 100, *, criterion = 'squared_error', max_depth = None, min_samples_split = 4, min_samples_leaf = 2, min_weight_fraction_leaf = 0.0, max_

features = 1.0, max_leaf_nodes = None, min_impurity_decrease = 0.0, bootstrap = True, oob_score = False, n_jobs = -1, random_state = 42, verbose = 0, warm_start = False, ccp_alpha = 0.0, max_samples = None).

n_jobs	random _state	n _estimators	max _depth	min _samples _split	min _samples _leaf	max _features	E _{train}	E _{v,Blocked-CV}	E _{test}
max	42	100	5	4	1	max	2.3844	3.8418	3.3234
max	42	200	5	4	1	max	2.3757	3.8360	3.3168
max	42	100	10	2	2	15	1.4864	3.6119	3.2806
max	42	200	max	10	5	20	1.7615	3.5757	3.2076
max	42	100	max	10	5	max	1.6615	3.4842	3.0356
max	42	200	max	10	5	max	1.6493	3.4831	3.0311
max	42	500	max	10	5	max	1.6467	3.4827	3.0336
max	42	100	max	2	1	max	0.9444	3.4733	3.0753
max	42	500	max	2	1	max	0.9241	3.4707	3.052
max	42	200	max	2	1	max	0.9266	3.4674	3.0536
max	42	100	max	6	2	max	1.2059	3.4574	3.0324
max	42	100	max	4	2	max	1.1043	3.4557	3.0299

Table A4. Hyper-parameter tuning of the RF model (Group 1).

Appendix A.3. Weight Identification of the VR Model

Even though we implemented a VR model with the weighted-averaging method, it is hard to find a straightforward method to determine the corresponding weights associated with its predictors. However, giving more weights to the best performing individual models has proven its effectiveness by improving the performance of the final VR model, as shown in Table A5.

	Weig	Performance	e of VR			
OLS	GLSAR	LR	DT	RF	$E_{v,Blocked-CV}$	E _{test}
0.1	0.1	0.2	0.3	0.3	2.8821	2.3769
0.3	0.3	0	0.25	0.15	2.6714	2.1047
0.2	0.2	0.2	0.2	0.2	2.6438	2.0797
0.3	0.3	0	0.15	0.25	2.6220	2.0589
0.4	0.3	0.3	0	0	2.6120	1.9055
0.3	0.3	0.3	0.05	0.05	2.5643	1.8818
0.6	0.1	0.2	0.05	0.05	2.5636	1.8817

Table A5. Performance of VR based on different weights for the individual models (Group 1).

We found that the parametric linear models OLS, GLSAR, and LR performed better than DT and RF with our data. Therefore, we selected a few arbitrary sets of weights, but many of them gave more weight to these parametric models (especially the OLS). The last row which is in bold—0.6, 0.1, 0.2, 0.05, and 0.05 for OLS, GLSAR, LR, DT, and RF, respectively—resulted in minimum $E_{v,Blocked-CV}$ and E_{test} and values. Not only did this occur with Group 1, but it also produced similar results with other groups. Therefore, that particular set was selected to build the final VR model in each group.

Appendix A.4. Tables and Figures for More Detailed Information

The Blocked-CV and test MAPE values obtained by half-hourly models in each group are presented in Table A6 and Table A7, respectively. In addition, the forecasting performance of the VR models in each group are illustrated in Figure A1–A4. Finally, the ACF and PACF plots for the residuals of VR models (special case: HH = 28) in each group are shown in Figure A5.

			Grou	ıp 1					Grou	ıp 2					Gro	up 3			Group 4						
HH	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	
0	2.4310	2.4299	2.4337	4.1803	3.3336	2.3592	2.3608	2.3607	2.3631	5.1452	4.2357	2.3766	5.0767	5.0768	5.0343	14.5507	13.0661	4.9545	2.5302	2.5303	2.5321	5.3576	4.0958	2.4239	
1	2.3847	2.3839	2.3875	4.1356	3.3681	2.3039	2.4382	2.4392	2.4462	5.1728	4.3240	2.4689	4.9403	4.9401	4.8714	14.7321	12.4802	4.8272	2.5563	2.5561	2.5588	5.3745	4.0761	2.4488	
2	2.4754	2.4729	2.4766	4.3037	3.3953	2.4159	2.5159	2.5159	2.5133	4.7368	4.2390	2.5095	4.8118	4.8180	4.7540	14.9982	12.4134	4.7558	2.6295	2.6296	2.6306	5.4003	4.1432	2.5282	
3	2.5367	2.5382	2.5431	4.2700	3.4242	2.4779	2.7899	2.7888	2.7757	5.0234	4.2097	2.7596	4.8839	4.8784	4.8793	14.4652	12.6275	4.8979	2.7224	2.7223	2.7223	5.3164	4.1148	2.6072	
4	2.6573	2.6588	2.6646	4.3606	3.4047	2.6006	2.9581	2.9590	2.9474	4.8961	4.2375	2.8844	4.8992	4.8978	4.9006	13.9103	12.5296	4.8601	2.8637	2.8642	2.8653	5.2812	4.1902	2.7505	
5	2.6780	2.6793	2.6839	4.2868	3.3695	2.6093	2.9689	2.9660	2.9549	4.9948	4.0980	2.9036	4.9312	4.9281	4.9286	14.8110	12.4283	4.9341	2.8520	2.8522	2.8536	5.2874	4.1909	2.7326	
6	2.7262	2.7258	2.7310	4.4684	3.3811	2.6439	2.9260	2.9268	2.9202	4.6655	4.0699	2.8786	5.0590	5.0588	5.0727	14.4617	12.6898	4.9504	2.8612	2.8606	2.8629	5.3842	4.1731	2.7535	
7	2.7183	2.7188	2.7247	4.2299	3.3399	2.6326	2.8419	2.8419	2.8413	4.9227	4.1071	2.7927	4.9746	4.9716	4.9888	15.4364	12.5801	4.9189	2.8309	2.8299	2.8333	5.4080	4.0853	2.7267	
8	2.7640	2.7636	2.7691	4.2383	3.3699	2.6778	2.8957	2.8920	2.8743	4.9668	3.9018	2.8535	5.0222	5.0250	5.0440	15.9058	12.4580	4.9319	2.8640	2.8643	2.8665	5.3119	4.1192	2.7678	
9	2.7015	2.7023	2.7071	4.2491	3.2892	2.6027	2.7930	2.7912	2.7781	4.8386	3.7447	2.7680	5.2220	5.2236	5.2528	13.9452	12.6082	5.0923	2.8160	2.8154	2.8176	5.0545	4.0188	2.7143	
10	2.6154	2.6131	2.6182	4.0768	3.2658	2.5341	2.7143	2.7130	2.7036	4.0346	3.4320	2.6326	4.9934	4.9956	5.0156	14.0617	12.1876	4.8153	2.7305	2.7304	2.7328	4.9439	3.9075	2.6501	
11	2.6489	2.6483	2.6534	4.1290	3.2771	2.5706	2.8387	2.8382	2.8289	3.9373	3.3386	2.7100	5.3051	5.3026	5.3141	16.8615	12.2315	5.1550	2.7734	2.7729	2.7748	5.0152	3.8752	2.7102	
12	2.5487	2.5481	2.5522	4.2710	3.6706	2.5260	2.8805	2.8819	2.8976	3.7443	3.3776	2.7490	5.9025	5.9050	5.9160	14.8501	11.4190	5.7377	2.7857	2.7859	2.7893	4.9289	3.9853	2.7237	
13	2.6199	2.6209	2.6267	4.7805	4.2020	2.6234	3.0707	3.0710	3.0713	4.1452	3.4535	2.9094	6.9625	6.9665	6.9501	14.8089	10.9926	6.7743	2.9199	2.9203	2.9229	5.2390	4.0780	2.8668	
14	2.4680	2.4678	2.4734	4.3759	3.8379	2.4628	3.1698	3.1713	3.1560	4.0283	3.3103	3.0208	7.6285	7.6417	7.5528	13.0677	10.7485	7.2710	2.9345	2.9343	2.9337	4.9286	3.7233	2.8373	
15	2.3192	2.3192	2.3245	3.8180	3.0699	2.2535	2.9338	2.9336	2.9354	3.5930	3.1931	2.7774	9.1358	9.1324	9.1473	14.7006	11.5698	8.7516	2.9968	2.9968	2.9985	4.8726	3.6422	2.8903	
16	2.2770	2.2780	2.2807	3.5099	2.8977	2.1981	2.8742	2.8737	2.8834	3.5018	2.9611	2.6975	9.5672	9.5684	9.3278	15.2502	11.9663	9.0885	3.1494	3.1497	3.1529	4.9302	3.7474	3.0233	
17	2.2949	2.2920	2.2939	3.6023	2.8936	2.2207	2.8523	2.8528	2.8582	3.6948	2.8584	2.6911	10.1200	10.1222	9.9444	17.8848	13.8165	9.8507	3.3877	3.3878	3.3914	5.5057	4.1115	3.2874	
18	2.3298	2.3295	2.3310	3.5937	2.9724	2.2817	2.7523	2.7509	2.7594	3.6490	2.7941	2.6278	10.2235	10.2238	10.1268	18.4897	13.5299	9.9867	3.4666	3.4659	3.4710	5.3508	4.2055	3.3545	
19	2.4463	2.4455	2.4486	3.7609	3.0535	2.4077	2.7545	2.7540	2.7625	3.6/42	2.8081	2.6412	10.3383	10.3301	10.3237	18.1362	13.5328	10.0900	3.4802	3.4817	3.4853	5.3353 E 47E0	4.2748	3.3880	
20	2.3249	2.3249	2.3200	3.304Z	2.9751	2.2000	2.0421	2.0421	2.0011	3.7607	2.0449	2.5515	10.3026	10.3132	10.2000	16.3223	12.3937	0.7625	3.3371 2.2971	3.3337	2 2804	5.4750	4.2942	3.2909	
21	2.3733	2.3734	2.3779	3.3243 2.6041	2.9570	2.3378	2.7240	2.7233	2.7310	3.3330 2.6570	2.0973	2.6071	10.1516	10.1076	10.1015	16.7778	12 2651	9.7635	2.36/1	2 4427	2 1126	5.2987 5.6606	4.3717	3.3223	
22	2.4000	2.4000	2.4000	2 6740	2,0099	2.4147	2.0003	2.0003	2.0900	3.0370	2.7732	2.3770	10.0012	10.0009	10.0140	17 5554	12 2557	10.1759	3.44ZZ 2.4019	3.4427 2.4021	2 4022	5.0000	4.2904	2 2207	
23	2.3030	2.3629	2.3633	3.0749	2.9099	2.3317	2.7290	2.7209	2.7401	4.0742	2.0009	2.0340	0 5108	0 5205	0 1110	17.5554	13.2337	0.0714	3.4010	3.4031	3.4025	5 8001	4.3300	3.3397	
24	2.4414	2.4403	2.4455	3.7372	2.9973	2.4024	2.4904	2.4900	2.4958	3.8511	2 9307	2.4127	9.3190	9.5205	9.4444	1/ 8980	12.1110	9.0714 8.8157	3 1091	3.1084	3 1071	5 3/12	4.3001	3.0365	
25	2 2700	2.2799	2.2020	3 7980	3.0146	2.2505	2.3303	2.5501	2.3320	3 7410	2.9507	2.4007	10.0324	10.0382	9.9816	15 8001	12 3047	9 5261	3 2374	3 2382	3 2385	6 0164	4 3722	3 1798	
20	2 3078	2 3080	2 3111	3 5546	3.0050	2 2893	2.4027	2.4021	2.4010	4 0488	2.0011	2.6317	10.0024	10.0002	10 5198	17 5354	13 3845	9.8853	3 4246	3 4250	3 4232	5 6049	4 3240	3 3531	
28	2 3061	2.3024	2.3051	3 4559	2 9577	2.2090	2 7188	2 7199	2 7302	3 9525	2 9552	2.6464	9 6605	9 6674	10.0588	18 1763	13 4573	9.3535	3 7014	3 3914	3 3916	5 7529	4 3940	3 3453	
29	2.2945	2.2940	2.2954	3.6082	3.0618	2.2670	2.8088	2.8101	2.8267	3.9818	3.0073	2.7288	10,1999	10.2077	10.4019	19.5539	13.5246	9.9131	3.4451	3.4453	3 4450	5.6108	4.3841	3.3787	
30	2.2628	2.2619	2.2634	3.6372	3.1594	2.2406	2.7414	2.7424	2.7516	4.1532	2.9923	2.6777	10.8234	10.8347	10.9037	18,7007	13.3191	10.3823	3.3823	3.3830	3.3831	5.1741	4.3662	3.2969	
31	2.2842	2.2839	2.2859	3.8607	3.2394	2.2879	2.7868	2.7860	2.7960	4.0731	3.1156	2.7275	10.9546	10.9515	10.8792	18.5521	13.5019	10.3829	3.4121	3.4112	3.4109	5.6027	4.4686	3.3430	
32	2.2711	2.2693	2.2713	3.8630	3.1900	2.2627	2.8520	2.8519	2.8605	3.9364	3.0828	2.7570	10.5977	10.6088	10.6220	16.4207	13.5917	9.9075	3.3707	3.3703	3.3702	5.4776	4.4276	3.3079	
33	2.2898	2.2884	2.2904	3.9588	3.2651	2.2765	2.8206	2.8224	2.8251	4.0672	3.2204	2.7431	10.3175	10.3146	10.2315	21.1591	13.2674	9.7591	3.3268	3.3282	3.3271	5.6921	4.4517	3.2554	
34	2.5191	2.5188	2.5213	4.0084	3.4432	2.4947	2.6407	2.6399	2.6507	3.9064	3.0917	2.5732	9.5576	9.5711	9.7515	16.6259	12.3087	9.0861	3.3523	3.3520	3.3495	5.4837	4.3731	3.2843	
35	2.7271	2.7267	2.7299	4.5934	3.6465	2.6975	2.5716	2.5711	2.5834	3.7845	3.0696	2.4757	8.8375	8.8404	8.9474	13.3173	11.2307	8.3728	3.4163	3.4169	3.4142	5.3478	4.3906	3.3441	
36	3.0137	3.0132	3.0162	4.2830	3.8438	2.9219	2.4448	2.4439	2.4634	3.9042	3.2980	2.3588	8.6497	8.6406	8.5422	14.2890	10.9366	8.2917	3.5467	3.5456	3.5433	5.2865	4.3017	3.4337	
37	2.9683	2.9708	2.9736	4.6466	4.0361	2.9010	2.5074	2.5080	2.5062	3.8764	3.2969	2.4096	8.3557	8.3437	8.4651	12.8407	10.2389	8.0618	3.3961	3.3956	3.3952	5.3841	4.2240	3.2938	
38	2.7927	2.7928	2.7953	4.4861	3.8681	2.7598	2.2789	2.2793	2.2758	3.8734	3.0221	2.1839	8.7827	8.7853	8.6908	12.5255	10.6675	8.4454	3.2535	3.2534	3.2512	5.0492	4.1021	3.1333	

Table A6. Blocked-CV MAPE value (in %) obtained by each half-hourly model in each group.

Table	A6.	Cont.

			Grou	ıp 1					Grou	ip 2					Gro	up 3				Group 4					
HH	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	
39	2.8005	2.8008	2.8035	4.7368	3.7627	2.7701	2.2517	2.2517	2.2668	3.7167	3.0445	2.1798	8.7165	8.7213	8.5595	11.7371	10.7161	8.3071	3.2247	3.2243	3.2230	5.1346	3.9667	3.1279	
40	2.8227	2.8226	2.8255	5.2978	3.7520	2.8116	2.3243	2.3235	2.3175	3.9169	3.1175	2.2502	8.6953	8.6932	8.5612	13.4336	11.0261	8.3184	3.1801	3.1804	3.1794	4.8376	3.9637	3.0695	
41	2.8740	2.8744	2.8770	5.3450	3.9833	2.8551	2.4151	2.4154	2.4137	4.0154	3.2747	2.3485	8.3157	8.3150	8.1973	10.8769	10.7859	7.9863	3.1614	3.1615	3.1596	5.0513	3.9854	3.0622	
42	2.9742	2.9749	2.9766	5.1515	4.0987	2.9452	2.6318	2.6328	2.6350	4.3298	3.4965	2.5497	8.4364	8.4322	8.2487	13.5478	11.1704	8.1240	3.2441	3.2442	3.2430	5.0109	4.0892	3.1414	
43	3.0585	3.0580	3.0598	6.1914	4.1821	3.0422	2.7106	2.7117	2.7197	4.7270	3.6933	2.6344	8.3101	8.3106	8.1429	13.5850	11.5459	8.0354	3.2874	3.2855	3.2849	5.1292	4.1768	3.1860	
44	3.1740	3.1747	3.1757	6.2450	4.3388	3.1361	2.8910	2.8902	2.8926	5.0446	3.9648	2.8144	8.3835	8.3744	8.2210	15.6902	12.5735	8.2382	3.4036	3.4010	3.3996	5.4993	4.4079	3.3121	
45	3.3814	3.3807	3.3819	5.1909	4.2938	3.2771	3.2774	3.2764	3.2859	5.6486	4.2257	3.1977	8.5802	8.5824	8.3227	14.1206	12.6343	8.3030	3.6111	3.6120	3.6119	5.8413	4.5127	3.5006	
46	3.5135	3.5135	3.5154	5.5094	4.4850	3.4153	3.4986	3.5001	3.5104	5.4973	4.2966	3.3805	9.2393	9.2346	9.0166	14.4375	13.1417	9.0219	3.7508	3.7518	3.7514	5.9089	4.7459	3.6242	
47	3.5365	3.5370	3.5376	5.4243	4.5555	3.4188	3.5786	3.5790	3.5872	5.5784	4.3191	3.4559	9.5150	9.5216	9.2795	14.7160	13.6729	9.2377	3.8003	3.8017	3.8018	6.0642	4.8822	3.6697	
avg	2.6113	2.6110	2.6142	4.2786	3.4557	2.5636	2.7506	2.7505	2.7534	4.2450	3.4046	2.6665	8.2515	8.2530	8.2161	15.4770	12.3826	7.9559	3.1880	3.1817	3.1822	5.3634	4.2054	3.0910	
stdev	0.3294	0.3298	0.3296	0.6812	0.4642	0.3164	0.2732	0.2733	0.2728	0.5933	0.5173	0.2588	2.1118	2.1132	2.1120	2.1049	0.9900	1.9788	0.3189	0.3116	0.3109	0.3017	0.2372	0.3140	

Table A7. Test MAPE values (in %) obtained by each half-hourly model in each group.

			Grou	ıp 1					Grou	p 2					Gro	up 3				Group 4						
HH	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR		
0	1.6367	1.6361	1.6585	3.8487	3.0361	1.6798	1.5067	1.5080	1.5074	4.7329	3.7762	1.5605	5.6291	5.6286	5.7646	11.9988	11.7283	4.4163	1.8075	1.8087	1.8165	4.9079	4.1017	1.7786		
1	1.7090	1.7087	1.7178	3.9278	2.9936	1.7324	1.6345	1.6338	1.6149	4.1714	3.8416	1.6473	4.3676	4.3702	4.7070	12.8359	11.9655	3.4100	1.8417	1.8401	1.8755	5.1429	4.1813	1.8206		
2	1.6873	1.6869	1.7052	4.0163	3.0758	1.7159	1.7313	1.7302	1.7078	4.7978	3.7795	1.7379	4.1892	4.1820	4.3853	12.2196	12.6604	3.5080	1.8131	1.8133	1.8602	5.0090	4.2415	1.8013		
3	1.7573	1.7576	1.7702	4.0782	3.0425	1.7689	1.6785	1.6789	1.6525	4.0380	3.8914	1.6729	4.2924	4.2971	4.3590	12.2234	12.4522	3.5620	1.8887	1.8887	1.9209	4.9116	4.2535	1.8974		
4	1.8142	1.8138	1.8436	3.8964	3.0276	1.8526	1.7655	1.7651	1.7279	4.4654	3.8314	1.7674	4.6843	4.6848	4.7217	11.2544	11.7404	3.8847	1.9417	1.9416	1.9745	4.9659	4.2764	1.9315		
5	1.8797	1.8799	1.9034	3.8721	3.1085	1.9065	1.8163	1.8164	1.7993	4.0471	3.6953	1.8308	4.3780	4.3809	4.4250	13.0530	12.0709	3.6945	1.9840	1.9840	2.0379	4.7122	4.1702	1.9925		
6	1.9133	1.9122	1.9527	3.9877	3.1096	1.9667	1.8562	1.8563	1.8226	3.7828	3.7183	1.8301	4.6323	4.6358	4.6616	11.5247	10.8906	3.9251	2.0164	2.0169	2.0837	4.9342	4.1356	2.0407		
7	2.0101	2.0090	2.0492	4.5166	3.2271	2.0659	1.8784	1.8783	1.8583	5.5931	3.7931	1.8810	4.4459	4.4457	4.4550	10.9201	10.9253	3.6281	2.0743	2.0747	2.1967	5.2218	4.1787	2.1504		
8	1.9201	1.9198	1.9632	4.3884	3.0976	1.9757	1.8643	1.8641	1.8567	4.0883	3.6567	1.8571	4.6663	4.6652	4.6433	9.9826	10.4138	3.9403	2.0180	2.0188	2.1500	4.9877	4.0030	2.0776		
9	1.8929	1.8940	1.9301	4.1726	3.0299	1.9369	1.7663	1.7665	1.7552	3.9210	3.5847	1.7571	4.3969	4.4032	4.3971	9.3415	10.2251	3.7622	1.9853	1.9854	2.1084	5.2605	4.0177	2.0574		
10	1.7993	1.8005	1.8382	3.6361	2.9246	1.8295	1.8059	1.8060	1.7940	3.7150	3.1701	1.7818	4.3415	4.3386	4.2674	12.3501	10.0841	3.6820	1.9002	1.9004	2.0053	5.2082	3.9547	2.0039		
11	1.7079	1.7078	1.7405	3.5043	2.7747	1.7388	1.7110	1.7096	1.6916	3.0716	3.2750	1.6583	4.9343	4.9350	4.9137	13.9500	10.4617	4.3503	1.8078	1.8074	1.8772	4.7536	3.8269	1.9113		
12	1.6149	1.6157	1.6419	3.3080	2.8904	1.6297	1.7307	1.7311	1.7171	3.1682	3.0185	1.6949	4.7098	4.7100	4.7209	12.8405	9.3243	4.0224	1.7963	1.7970	1.8422	4.3946	3.6452	1.8840		
13	1.5999	1.5998	1.6494	3.5240	2.9851	1.5624	1.5635	1.5660	1.5348	3.5772	2.9075	1.5625	4.5434	4.5442	4.6086	12.5053	8.7819	4.4405	1.9098	1.9093	1.9561	4.1432	3.5129	1.9441		
14	1.5981	1.5981	1.6280	3.3478	2.8135	1.5492	1.5868	1.5866	1.5388	3.3429	2.7710	1.5193	4.6398	4.6364	4.6364	10.1939	9.1903	4.6103	1.9381	1.9372	1.9714	3.7320	3.3390	1.9397		
15	1.6871	1.6871	1.7268	3.0921	2.6594	1.6456	1.8848	1.8859	1.8666	3.3138	2.7956	1.8103	6.1097	6.1102	5.8105	13.3971	10.2593	6.0091	2.1382	2.1394	2.1613	3.7353	3.4160	2.1178		
16	1.7447	1.7452	1.7886	3.7680	2.8206	1.7287	1.8050	1.8049	1.8176	3.4590	2.5555	1.7256	6.8598	6.8730	6.4473	12.2772	11.0759	6.6563	2.2850	2.2854	2.3070	5.0796	3.7683	2.3454		
17	1.7324	1.7324	1.7801	3.4065	2.8912	1.7539	2.0269	2.0273	2.0601	2.6043	2.4779	1.9137	6.7337	6.7292	6.3091	13.2202	12.3810	6.7267	2.5190	2.5185	2.5421	4.5503	3.6690	2.5422		
18	1.7318	1.7321	1.7863	3.5546	2.8993	1.7520	2.1813	2.1810	2.2258	3.0049	2.4579	2.0463	6.8644	6.8517	6.6157	13.2933	12.8977	6.9302	2.6147	2.6140	2.6307	4.4570	3.6682	2.6135		
19	1.7335	1.7331	1.8046	3.1733	2.9719	1.7544	2.1962	2.1969	2.2532	3.1087	2.5132	2.0800	6.6093	6.6081	6.6867	14.4944	13.3143	6.4417	2.6408	2.6413	2.6691	4.3263	3.8538	2.6375		
20	1.7611	1.7611	1.8439	3.4976	2.9870	1.7666	2.1591	2.1592	2.1959	2.9587	2.5326	2.0015	6.6658	6.6457	6.2722	14.6045	13.1461	6.7571	2.6105	2.6091	2.6400	4.1426	3.8740	2.5860		
21	1.8325	1.8326	1.9352	3.4019	3.0901	1.8456	2.1599	2.1600	2.1837	3.1736	2.5457	2.0478	6.6145	6.6105	5.8697	19.9865	13.1463	6.7895	2.6523	2.6526	2.6827	4.2094	3.8363	2.6311		
22	1.9306	1.9300	2.0283	3.3023	2.9146	1.9285	2.2163	2.2160	2.2403	3.0524	2.7686	2.1059	6.2122	6.2248	5.9416	13.6551	12.9035	6.2240	2.7196	2.7190	2.7549	4.2051	3.7889	2.6920		

Table A7. Cont.

			Grou	up 1					Grou	ıp 2					Gro	oup 3				Group 4						
HH	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR	LR	OLS	GLSAR	DT	RF	VR		
23	1.9081	1.9081	2.0180	3.5875	2.8123	1.9169	2.1875	2.1871	2.2100	2.9604	2.5927	2.0702	6.8548	6.8433	6.6324	12.9296	13.2213	6.5592	2.6962	2.6965	2.7376	4.1782	3.7667	2.6702		
24	1.8508	1.8510	1.9880	2.9187	2.6553	1.7913	2.1481	2.1485	2.1241	3.1980	2.5408	2.0395	5.9710	5.9584	6.0463	12.9834	12.1305	5.8795	2.5942	2.5976	2.6421	3.9819	3.7987	2.5728		
25	1.7835	1.7828	1.8714	3.1748	2.6232	1.7292	2.0919	2.0920	2.0709	3.0854	2.6058	1.9582	5.6360	5.6597	5.9139	13.4519	11.6664	5.5132	2.4915	2.4905	2.5223	5.3190	3.8630	2.5128		
26	1.7962	1.7952	1.8626	3.0945	2.6468	1.7858	2.2956	2.2935	2.2803	3.0970	2.4413	2.1597	7.3781	7.3823	7.4255	14.7204	13.0661	6.7933	2.6680	2.6680	2.6950	4.4811	3.9217	2.6486		
27	1.9972	1.9967	2.0636	3.3686	2.8610	1.9656	2.3185	2.3176	2.3293	3.0128	2.6944	2.2508	8.4802	8.4751	8.1987	14.9524	14.0359	7.9946	2.9433	2.9434	2.9704	4.7949	4.1568	2.9190		
28	2.1108	2.1107	2.1538	3.0343	2.8674	2.0766	2.3895	2.3918	2.4082	3.0882	2.6322	2.3122	6.5325	6.5156	6.3160	15.5854	13.9909	6.5177	3.1055	3.0530	3.0690	4.5689	4.1671	3.0140		
29	1.9152	1.9152	1.9598	3.0687	2.7613	1.8865	2.3397	2.3395	2.3463	3.1281	2.7131	2.2520	6.4698	6.5033	6.3301	14.7285	13.9816	6.4056	2.9336	2.9348	2.9827	4.6562	3.9694	2.9090		
30	1.9293	1.9293	1.9535	3.2359	2.6917	1.8815	2.2512	2.2512	2.2718	3.0524	2.7030	2.1671	6.5025	6.5104	6.3932	15.3273	14.3000	6.4288	2.9679	2.9664	3.0119	4.8461	4.0491	2.9335		
31	1.9155	1.9154	1.9665	3.1895	2.7471	1.8770	2.4261	2.4261	2.4474	3.3497	2.7834	2.3360	6.5862	6.5499	6.4458	18.9477	14.1629	6.3601	3.0281	3.0291	3.0843	4.7192	4.0481	2.9974		
32	1.9320	1.9320	2.0227	2.9762	2.6672	1.8804	2.5064	2.5066	2.5258	3.1862	2.7402	2.4174	6.1470	6.1470	6.2769	22.5275	13.4003	6.3989	3.0207	3.0203	3.0875	4.3495	4.0039	2.9419		
33	1.9776	1.9775	2.0804	2.8687	2.6626	1.9005	2.5740	2.5738	2.6109	3.5626	2.6591	2.4501	5.8889	5.8828	6.1075	16.4594	12.9743	5.8477	3.0362	3.0361	3.1352	4.9549	3.9206	2.9912		
34	2.2148	2.2152	2.3214	2.9426	2.5662	2.1100	2.4254	2.4254	2.4569	3.9142	2.7658	2.3056	5.4776	5.4584	5.6478	16.1051	12.1423	5.3629	3.0231	3.0235	3.1645	4.9954	3.9200	2.9757		
35	2.4356	2.4360	2.4722	3.2967	2.5142	2.2747	2.4039	2.4037	2.3993	3.5119	2.7404	2.2724	5.3207	5.3470	5.2972	11.1930	11.0185	5.2752	3.1171	3.1154	3.2348	4.9084	3.7610	3.0318		
36	2.4718	2.4718	2.5148	3.1825	2.5080	2.2837	2.3640	2.3636	2.3693	3.1140	2.5571	2.2546	5.4205	5.4655	5.4019	12.4659	10.1291	5.4197	3.0019	3.0000	3.1635	4.9833	3.6132	2.9209		
37	2.4166	2.4164	2.3820	3.2724	2.9239	2.2316	2.0302	2.0301	2.0359	3.5874	2.6618	1.9783	5.3484	5.3473	5.1592	14.2433	10.1949	5.4401	2.7159	2.7172	2.9048	4.7924	3.7090	2.6561		
38	2.1496	2.1494	2.1473	3.1581	2.9911	2.0065	1.7730	1.7728	1.7947	2.9492	2.6962	1.6923	4.7065	4.7122	4.7101	13.1695	10.0878	4.8308	2.4779	2.4784	2.7116	4.6642	3.6428	2.4327		
39	2.0988	2.0989	2.0935	3.2778	3.0187	1.9675	1.7133	1.7131	1.7563	2.9783	2.8938	1.6264	4.6079	4.5998	4.6976	12.6184	10.3253	4.6022	2.4122	2.4077	2.6515	4.2397	3.5762	2.3620		
40	2.0218	2.0215	2.0487	3.5231	3.1678	1.9188	1.6979	1.6978	1.7017	3.3371	3.1308	1.6100	4.4625	4.4631	4.6587	14.5196	10.1323	4.4438	2.3730	2.3728	2.6656	4.5839	3.8152	2.3629		
41	1.9955	1.9954	2.0542	3.9092	3.3902	1.9116	1.7766	1.7767	1.7821	3.8420	3.1656	1.6858	4.4382	4.4432	4.5411	14.4521	10.2084	4.3087	2.4245	2.4242	2.7310	4.7599	4.0107	2.4278		
42	1.9173	1.9173	2.0173	3.6457	3.6099	1.8530	1.9073	1.9072	1.9092	4.1288	3.4838	1.7953	4.5725	4.5710	4.5610	13.5601	10.8684	4.2087	2.4651	2.4648	2.8231	4.7594	4.3021	2.4469		
43	1.8691	1.8688	2.0136	4.3373	3.7218	1.8215	1.9699	1.9693	1.9820	3.8866	3.7234	1.8968	4.9545	4.9569	4.4934	15.4543	10.8141	4.4234	2.4896	2.4912	2.8950	5.4263	4.4711	2.5055		
44	1.9571	1.9558	2.1006	4.7005	3.8959	1.9164	2.0320	2.0320	2.0444	4.6403	3.7243	1.9594	5.2391	5.2444	4.4896	14.3310	11.7746	4.5557	2.6096	2.6091	3.0196	6.1245	4.7135	2.6394		
45	1.9823	1.9813	2.1174	4.9546	4.1335	2.0284	2.1564	2.1563	2.1888	5.5487	4.0188	2.0612	5.1455	5.1339	4.6784	14.9741	12.0457	4.5503	2.7342	2.7334	3.1286	6.1233	5.0097	2.7527		
46	2.0228	2.0215	2.1910	4.9381	4.3341	2.0727	2.2272	2.2272	2.2413	4.1861	4.2146	2.1790	5.3204	5.3213	5.1635	14.6814	12.2625	4.6947	2.8441	2.8441	3.2520	6.5123	5.2278	2.8897		
47	2.0994	2.0999	2.2738	4.6027	4.2919	2.1488	2.2398	2.2403	2.2546	5.1449	4.2108	2.1774	5.3899	5.4018	5.4611	14.2930	12.2648	4.7329	2.8921	2.8891	3.2993	6.1935	5.3407	2.9244		
avg	1.9055	1.9053	1.9661	3.6142	3.0299	1.8817	2.0161	2.0161	2.0201	3.6391	3.0927	1.9458	5.5077	5.5084	5.4305	13.7666	11.7341	5.1858	2.4577	2.4564	2.5760	4.8106	4.0102	2.4549		
stdev	0.2010	0.2010	0.2079	0.5372	0.4214	0.1671	0.2800	0.2799	0.2939	0.6966	0.5477	0.2497	1.0041	1.0024	0.9277	2.3560	1.4314	1.1870	0.4248	0.4230	0.4660	0.5825	0.4042	0.3978		



Figure A1. Forecasting performance of the VR model (Group 1).



Figure A2. Forecasting performance of the VR model (Group 2).



Figure A3. Forecasting performance of the VR model (Group 3).



Figure A4. Forecasting performance of the VR model (Group 4).



Figure A5. ACF and PACF plots for the residuals of VR models (special case: HH = 28) in each group; (a) Group 1. (b) Group 2. (c) Group 3. (d) Group 4.

References

- Dobschinski, J.; Bessa, R.; Du, P.; Geisler, K.; Haupt, S.E.; Lange, M.; Mohrlen, C.; Nakafuji, D.; De La Torre Rodriguez, M. Uncertainty Forecasting in a Nutshell: Prediction Models Designed to Prevent Significant Errors. *IEEE Power Energy Mag.* 2017, 15, 40–49. [CrossRef]
- 2. Phuangpornpitak, N.; Prommee, W. A Study of Load Demand Forecasting Models in Electric Power System Operation and Planning. *GMSARN Int. J.* 2016, *10*, 19–24.
- 3. Chapagain, K.; Kittipiyakul, S. Performance analysis of short-term electricity demand with atmospheric variables. *Energies* **2018**, *11*, 2015–2018. [CrossRef]
- 4. Chapagain, K.; Kittipiyakul, S.; Kulthanavit, P. Short-term electricity demand forecasting: Impact analysis of temperature for Thailand. *Energies* **2020**, *13*, 2–5. [CrossRef]
- 5. Bergmeir, C.; Hyndman, R.J.; Koo, B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Comput. Stat. Data Anal.* **2018**, *120*, 70–83. [CrossRef]
- 6. Géron, A. Book Review: Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, 2nd ed.; Géron, A., Ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2020; Volume 43; pp. 1135–1136. [CrossRef]
- 7. Hong, T.; Fan, S. Probabilistic electric load forecasting: A tutorial review. Int. J. Forecast. 2016, 32, 914–938. [CrossRef]
- Chapagain, K.; Kittipiyakul, S. Short-term electricity load forecasting for Thailand. In Proceedings of the ECTI-CON 2018—15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Chiang Rai, Thailand, 18–21 July 2018; pp. 521–524. [CrossRef]
- Dilhani, M.H.; Jeenanunta, C. Daily electric load forecasting: Case of Thailand. In Proceedings of the 7th International Conference on Information Communication Technology for Embedded Systems 2016 (IC-ICTES 2016), Bangkok, Thailand, 20–22 March 2016; pp. 25–29. [CrossRef]
- 10. Pannakkong, W.; Aswanuwath, L.; Buddhakulsomsiri, J.; Jeenanunta, C.; Parthanadee, P. Forecasting medium-term electricity demand in Thailand: Comparison of ANN, SVM, DBN, and their ensembles. In Proceedings of the International Conference on ICT and Knowledge Engineering, Bangkok, Thailand, 20–22 November 2019. [CrossRef]
- 11. Parkpoom, S.; Harrison, G.P. Analyzing the impact of climate change on future electricity demand in Thailand. *IEEE Trans. Power Syst.* **2008**, 23, 1441–1448. [CrossRef]
- 12. Divina, F.; Gilson, A.; Goméz-Vela, F.; Torres, M.G.; Torres, J.F. Stacking ensemble learning for short-term electricity consumption forecasting. *Energies* **2018**, *11*, 949. [CrossRef]
- 13. Sharma, E. Energy forecasting based on predictive data mining techniques in smart energy grids. *Energy Inform.* **2018**, *1*, 44. [CrossRef]
- Chapagain, K.; Kittipiyakul, S. Short-Term Electricity Demand Forecasting with Seasonal and Interactions of Variables for Thailand. In Proceedings of the iEECON 2018—6th International Electrical Engineering Congress, Krabi, Thailand, 7–9 March 2018. [CrossRef]
- 15. Jeenanunta, C.; Abeyrathna, D. *Combine Particle Swarm Optimization with Artificial Neural Networks for Short-Term Load Forecasting*; Technical Report 1; SIIT, Thammasat University: Bangkok, Thailand, 2017.
- 16. Chapagain, K.; Kittipiyakul, S. Short-term Electricity Load Forecasting Model and Bayesian Estimation for Thailand Data. In *MATEC Web of Conferences*; EDP Sciences: Les Ulis, France, 2016; Volume 55. [CrossRef]
- 17. Huang, S.J.; Shih, K.R. Short-term load forecasting via ARMA model identification including non-Gaussian process considerations. *IEEE Trans. Power Syst.* 2003, 18, 673–679. [CrossRef]
- Harvey, A.; Koopman, S.J. Forecasting Hourly Electricity Demand Using Time-Varying Splines. J. Am. Stat. Assoc. 1993, 88, 1228. [CrossRef]
- 19. Chapagain, K.; Soto, T.; Kittipiyakul, S. Improvement of performance of short term electricity demand model with meteorological parameters. *Kathford J. Eng. Manag.* **2018**, *1*, 15–22. [CrossRef]
- 20. Ramanathan, R.; Engle, R.; Granger, C.W.; Vahid-Araghi, F.; Brace, C. Short-run forecasts of electricity loads and peaks. *Int. J. Forecast.* **1997**, *13*, 161–174. [CrossRef]
- 21. Li, B.; Lu, M.; Zhang, Y.; Huang, J. A Weekend Load Forecasting Model Based on Semi-Parametric Regression Analysis Considering Weather and Load Interaction. *Energies* **2019**, *12*, 3820. [CrossRef]
- 22. Darbellay, G.A.; Slama, M. Forecasting the short-term demand for electricity: Do neural networks stand a better chance? *Int. J. Forecast.* 2000, *16*, 71–83. [CrossRef]
- 23. Srinivasan, D.; Chang, C.S.; Liew, A.C. Demand Forecasting Using Fuzzy Neural Computation, With Special Emphasis On Weekend Additionally, Public Holiday Forecasting. *IEEE Trans. Power Syst.* **1995**, *10*, 1897–1903. [CrossRef]
- 24. Su, W.H.; Chawalit, J. Short-term Electricity Load Forecasting in Thailand: An Analysis on Different Input Variables. In *IOP Conference Series: Earth and Environmental Science;* IOP Publishing: Bristol, UK, 2018; Volume 192. [CrossRef]
- 25. Abeyrathna, K.D.; Jeenanunta, C. Hybrid particle swarm optimization with genetic algorithm to train artificial neural networks for short-term load forecasting. *Int. J. Swarm Intell. Res.* **2019**, *10*, 1–14. [CrossRef]
- 26. Taylor, J.W.; de Menezes, L.M.; McSharry, P.E. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *Int. J. Forecast.* 2006, 22, 1–16. [CrossRef]

- Chapagain, K.; Sato, T.; Kittipiyakul, S. Performance analysis of short-term electricity demand with meteorological parameters. In Proceedings of the ECTI-CON 2017—2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Phuket, Thailand, 27–30 June 2017; pp. 330–333. [CrossRef]
- 28. Bonetto, R.; Rossi, M. Machine learning approaches to energy consumption forecasting in households. *arXiv* 2017, arXiv:1706.09648.
- 29. Peng, L.; Wang, L.; Xia, D.; Gao, Q. Effective energy consumption forecasting using empirical wavelet transform and long short-term memory. *Energy* **2022**, 238, 121756. [CrossRef]
- 30. Lisi, F.; Shah, I. *Forecasting Next-Day Electricity Demand and Prices Based on Functional Models*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 11, pp. 947–979. [CrossRef]
- Jan, F.; Shah, I.; Ali, S. Short-Term Electricity Prices Forecasting Using Functional Time Series Analysis. *Energies* 2022, 15, 3423. [CrossRef]
- 32. Shah, I.; Iftikhar, H.; Ali, S.; Wang, D. Short-term electricity demand forecasting using components estimation technique. *Energies* **2019**, *12*, 2532. [CrossRef]
- Shah, I.; Iftikhar, H.; Ali, S. Modeling and Forecasting Medium-Term Electricity Consumption Using Component Estimation Technique. *Forecasting* 2020, 2, 163–179. [CrossRef]
- 34. Nielsen, A. Practical Time Series Analysis Preview Edition; O'Reilly: Sebastopol, CA, USA, 2019; p. 45.
- Schnaubelt, M. A Comparison of Machine Learning Model Validation Schemes for Non-Stationary Time Series Data; No. 11/2019. FAU Discussion Papers in Economics; Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute for Economics: Nürnberg, Germany, 2019; pp. 1–42. [CrossRef]
- Bibi, N.; Shah, I.; Alsubie, A.; Ali, S.; Lone, S.A. Electricity Spot Prices Forecasting Based on Ensemble Learning. *IEEE Access* 2021, 9, 150984–150992. [CrossRef]
- Dickey, D.A.; Fuller, W.A. Distribution of the Estimators for Autoregressive Time Series With a Unit Root. J. Am. Stat. Assoc. 1979, 74, 427. [CrossRef]
- 38. Fuller, W.A. Introduction to Statistical Time Series; Wiley: Hoboken, NJ, USA, 1996; p. 698.