

Article

Error Compensation Enhanced Day-Ahead Electricity Price Forecasting

Dimitrios Kontogiannis ^{1,*}, Dimitrios Bargiotas ¹, Aspasia Daskalopulu ¹,
Athanasios Ioannis Arvanitidis ¹ and Lefteri H. Tsoukalas ²

¹ Department of Electrical and Computer Engineering, School of Engineering, University of Thessaly, 38334 Volos, Greece; bargiotas@uth.gr (D.B.); aspasia@uth.gr (A.D.); atarvanitidis@uth.gr (A.I.A.)

² Center for Intelligent Energy Systems (CiENS), School of Nuclear Engineering, Purdue University, West Lafayette, IN 47906, USA; tsoukala@purdue.edu

* Correspondence: dimkonto@uth.gr

Abstract: The evolution of electricity markets has led to increasingly complex energy trading dynamics and the integration of renewable energy sources as well as the influence of several external market factors contributed towards price volatility. Therefore, day-ahead electricity price forecasting models, typically using some kind of neural network, play a crucial role in the optimal behavior of market agents. The most prominent models and benchmarks rely on improving the accuracy of predictions and the time for convergence by some sort of *a priori* processing of the dataset that is used for the training of the neural network, such as hyperparameter tuning and feature selection techniques. What has been overlooked so far is the possible benefit of *a posteriori* processing, which would consider the effects of parameters that could refine the predictions once they have been made. Such a parameter is the estimation of the residual training error. In this study, we investigate the effect of residual training error estimation for the day-ahead price forecasting task and propose an error compensation deep neural network model (ERC–DNN) that focuses on the minimization of prediction error, while reinforcing error stability through the integration of an autoregression module. The experiments on the Nord Pool power market indicated that this approach yields improved error metrics when compared to the baseline deep learning structure in different training scenarios, and the refined predictions for each hourly sequence shared a more stable error profile. The proposed method contributes towards the development of more flexible hybrid neural network models and the potential integration of the error estimation module in future benchmarks, given a small and interpretable set of hyperparameters.



Citation: Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A.; Arvanitidis, A.I.; Tsoukalas, L.H. Error Compensation Enhanced Day-Ahead Electricity Price Forecasting. *Energies* **2022**, *15*, 1466. <https://doi.org/10.3390/en15041466>

Academic Editors: Hannu Laaksonen and Galih Bangga

Received: 17 January 2022

Accepted: 15 February 2022

Published: 17 February 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/>).

Keywords: electricity price forecasting; energy; machine learning; deep learning; neural networks; artificial intelligence; error estimation

1. Introduction

Modern energy markets follow increasingly complex processes in order to perform efficient electricity trading that balances supply and demand while reacting to the dynamics derived from the unique characteristics and challenges of each energy system. One of the main challenges that urge the development of more sophisticated techniques for the coordinated production and supply of electricity is price volatility [1]. The price of electricity can fluctuate due to several factors and the sudden peaks and valleys in the price curves could lead to suboptimal energy market agent behavior, hindering the ability of those entities to execute economic transactions in the electricity market to the best of their envisaged capacity. Some of the most notable factors that could cause price fluctuations to include seasonal trends [2], weather conditions [3], penetration of renewable energy sources [4], challenges involving economic growth and changes in fuel cost [5], supply availability [6] and neighboring market dynamics [7]. It can be easily observed that load and generation

dependencies on the time of day or year as well as seasonal trends coupled with extreme hot or cold temperatures and extreme conditions, such as hurricanes could have a noticeable impact on the electricity price. Furthermore, the infrastructural development of growing economies often leads to increased energy demand and electricity costs. Additionally, electricity price fluctuations could depend on the availability of fossil fuels for sufficient generation. Price spikes could occur if more expensive forms of electricity generation are used due to fossil fuel shortages. Since renewable energy sources constitute enticing alternatives for electricity generation, the stability of each energy source is related to the stability of electricity prices and the price fluctuation patterns vary for each region. Lastly, the impact of neighboring markets on price volatility could be attributed to the increased player participation and decentralization that increases the complexity of price formation.

Real-time energy markets could be negatively affected by price volatility since market participants could be unable to react proactively when price fluctuations and energy transactions occur throughout the operating day. However, price volatility can be tempered with the development of day-ahead energy markets that allow buyers and sellers to determine and secure energy prices before the operating day [8]. Therefore, short-term forecasting models that predict day-ahead prices are valuable for the successful monitoring of price trends and coordination of supply and demand. Price data and influential features are typically collected in the form of time series, following an hourly sampling rate. Statistical methods and machine learning models contribute greatly towards the development of accurate and robust day-ahead electricity price forecasting models that are capable of processing time series data efficiently and handling the complexity of those energy markets [9,10]. Forecasting models derived from statistical methods often utilize linear regression [11] in order to model the target variable as the linear combination of independent features. Additionally, autoregressive models [12–14] expanded on this concept by highlighting the importance of autocorrelation between values of the same variable from previous time steps. Machine learning models for day-ahead forecasting often rely on the development of neural networks that operate as function approximators and aim to detect the linear and nonlinear relationships between the input and output features. The primary neural network types utilized for this forecasting task include the multi-layer perceptron (MLP) [15] and the feed-forward deep neural network (DNN) [16], long short-term memory networks (LSTM) [17] and convolutional neural networks (CNN) [18]. The appeal of methods involving the MLP and DNN [19] can be justified due to the ease of use and the simplicity of structure since MLPs include fully connected layers of neurons that form a computation path from the input to the output, resulting in a network that is acyclic in nature. On the other hand, LSTMs are recurrent neural networks (RNN) [20] that follow a block structure consisting of gates that interact with the previous and next state of the network. Long short-term memory networks are more complex when compared to fully connected feed-forward network types, but their structure could handle temporal dependencies between time series lags of unknown length more efficiently. Lastly, CNN networks [21] use one-dimensional convolution to learn patterns within specific time windows and can inspect the data from a broader perspective through data shuffling. Neural networks constitute impactful short-term forecasting tools in the energy sector and a plethora of different standalone and combinatorial structures are studied for fast and accurate predictions [22].

Models derived from the aggregation of previously mentioned networks form the category of ensemble learning [23] and have substantial forecasting performance benefits. The combination of different types of models belonging either in the statistical method or the machine learning class with the integration of modules that contribute towards data decomposition, feature selection, clustering, or heuristic optimization, form the class of hybrid forecasting methods [24–26] that often succeed in the analysis of more complex dynamics and patterns. Benchmarks in the field of day-ahead price forecasting mainly utilize autoregressive and deep neural network models since these structures offer state-of-the-art performance and simplicity of implementation. The evaluation of new approaches and the

process of model selection through those benchmarks rely primarily on hyperparameter optimization, feature selection and regularization techniques [27].

Recent research projects and reviews highlighted interesting short-term electricity price forecasting approaches that utilize elements from statistical and machine learning methods. Alamaniotis et al. [28] proposed a multiple regression model based on relevance vector machines for day-ahead electricity price forecasting, contributing towards the development of optimal bidding strategies in electricity markets. Moreover, Alamaniotis et al. [29] developed a hybrid forecasting model featuring relevance vector machines in a linear regression ensemble method for efficient short-term price forecasting. Zhang et al. [30] presented a forecasting method that aggregates the combined predictions from CNN and RNN structures in a gradient boosting regressor yielding improved performance. Additionally, this study highlighted the importance of elastic net regularization for the stability and reliability of this combinatorial method. Alamaniotis et al. [31] developed a combinatorial approach that couples load and price forecasting and modifies forecasted load demand through the implementation of smart scheduling algorithms. Chinnathambi et al. [32] developed a multi-stage day-ahead forecasting model based on the autoregressive integrated moving average (ARIMA) statistical approach and the consequent residual error forecast that improves the performance of the initial predictions for different time periods. This research project provides some useful insights on the utilization of post-processing factors, such as the error for the improvement of statistical methods. Chang et al. [33] proposed a forecasting model that utilizes wavelet transform and an LSTM network featuring the stochastic gradient optimizer Adam, demonstrating that a well-optimized recurrent neural network could capture and process the nonlinear patterns in this task efficiently. Su et al. [34] utilized the least squares regression boosting algorithm to predict natural gas spot prices, outperforming existing approaches, such as linear regression. Atef and Eltawil [35] conducted a comparison between support vector regression (SVR) and LSTM electricity price forecasting models, concluding that while both methods could be suitable for this predictive task, the deep learning approach outperforms the regression model in terms of error metrics. Bissing et al. [36] investigated the different combinations of regression, namely the ARIMA and Holt-Winters models, for day-ahead forecasting and provided some interesting results regarding the performance benefits of hybrid implementations. Xu and Baldick [37] compared different neural network architectures and some state-of-the-art statistical methods, concluding that neural network models could perform better for price forecasting while yielding lower mean absolute error. Zhang et al. [38] studied the performance of deep recurrent neural networks for electricity price forecasts in a deregulated market, providing useful insights on the suitability of this neural network type as a multivariate time series model. Lago et al. [39] presented a review of state-of-the-art price forecasting models covering statistical, machine learning and hybrid approaches. Furthermore, this research work provided a useful open-access benchmark including a regression and a deep neural network model that utilize hyperparameter optimization for future model comparisons. Tao et al. [40] proposed a bias compensation LSTM network utilizing the LightBGM algorithm for feature selection. This work contributed significantly towards the development of hybrid short-term forecasting models since the introduction of residual error analysis for recurrent neural networks is a novel approach that could refine time series predictions. Vega-Márquez et al. [41] approached the electricity price forecasting task from a univariate time series perspective and tested well-known deep learning and statistical methods through hyperparameter optimization, distinguishing LSTM, CNN and regression tree methods as the most performant. Jiang et al. [42] utilized a decomposition-selection-ensemble forecasting system that adapts to different data characteristics and focuses on accurate and stable price predictions. Li et al. [43] presented a price forecasting model based on variational mode decomposition and sparse Bayesian learning of time series, showing that aggregate predictions derived from components featuring simple characteristics could outperform state-of-the-art models. Pourdaryaei et al. [44] investigated the impact of different optimization methods for day-ahead price forecasting.

This research work focuses mostly on the pre-processing and learning steps, while the impact of post-processing optimization techniques remains unexplored.

After a thorough overview of the literature, it is important to note that while a plethora of forecasting models exist and deep neural networks have been some of the most frequently used models, the effect of error compensation for the state-of-the-art feed-forward DNN is not sufficiently covered. We can observe that benchmarks and relevant studies utilize hyperparameter optimization as well as feature selection to tune the models and achieve lower error metrics, but fewer studies have applied post-processing techniques in order to refine and improve the predictions. Therefore, while there are recent studies that utilize error residuals for this short-term forecasting task, the application of this technique on the simple yet highly performant DNN is not thoroughly explored. As a result, the potential utilization of an error estimation module for benchmarks utilizing the DNN model as an additional tuning tool remains an open question. In this study, we identified these research gaps and developed a hybrid error compensation deep neural network model, the ERC-DNN, which utilizes a feed-forward deep neural network for day-ahead electricity price predictions, as well as an autoregression module, which operates on the hourly residual error sequences and performs a step-by-step error estimation to refine the predicted prices. The main goals of this research project are: (i) to showcase the improvement of price predictions in terms of error metrics; (ii) to investigate the stability of hourly predicted sequences after the error refinement; and (iii) to provide insights into the suitability of error estimation modules in modern benchmarks for future integration, when the appropriate parameters are defined. This hybrid approach was evaluated on the dataset of the Nord Pool market following the guidelines of the benchmark presented in [39], and through different training scenarios that highlight the positive impact of error refinement. Moreover, the resulting error metrics of this approach are compared to a baseline DNN structure developed using well-known configuration and training practices in order to achieve a similar score to the DNN benchmark with a static set of hyperparameters that does not alter the tests and produces consistent results during recalibration. Additionally, the error metrics of ERC-DNN are compared to the benchmark scores despite the differences in training epochs and hyperparameter optimization in order to highlight the overall effect of the error estimation module.

Section 2 presents the main methods utilized in the implementation of the proposed forecasting approach with references to the core components of the network, as well as information regarding the dataset and the configuration of the experiments. Furthermore, this section defines the error metrics used to evaluate the performance of ERC-DNN. Section 3 discusses the results of the experiments and compares performance metrics to the baseline and benchmark models. Finally, in Section 4, the advantages, as well as the challenges of this hybrid model, are outlined. Additionally, comments regarding the impact of this model as a standalone project, the potential expansion of the proposed architecture, and the integration of this model to more complex forecasting structures and open-access benchmarks in the future are included, in the hope that they contribute to the intelligence gathered in this area of research.

2. Materials and Methods

2.1. Feedforward Deep Neural Network

The feedforward deep neural network is an acyclic artificial neural network [45] that follows a simple layer structure and extends the MLP architecture for the purposes of function approximation. The base unit of the feedforward DNN is the neuron which is a node designed to receive a specified number of inputs, perform computations and pass the output to connected nodes found deeper in the network. The value of the output at each node is determined by activation functions, such as the rectified linear unit and hyperbolic tangent [46]. The neurons of the DNN are organized into layers and the connections of those layers denote the computation path from the input to the output. The simplest and most frequently used DNN structure contains the input layer, where input features are passed to

the first set of neurons, several hidden layers that perform additional computations and tune the learnable parameters of the network, and the output layer where one or more output values are generated at each node. For the purposes of this study, we consider the role of the feedforward DNN for the supervised learning task of regression [47] since we focus on the prediction of the electricity price for the next day. Based on this task, the goal of the DNN is to learn the mapping function that describes the complex relationship between the input variables and the output variables. As a general example, we consider the fully connected DNN presented in Figure 1. The DNN features an input layer i containing k inputs, a variable number f of hidden layers h , where each one contains a variable number of neurons z and, finally, an output layer o containing j neurons for the predictions of j outputs.

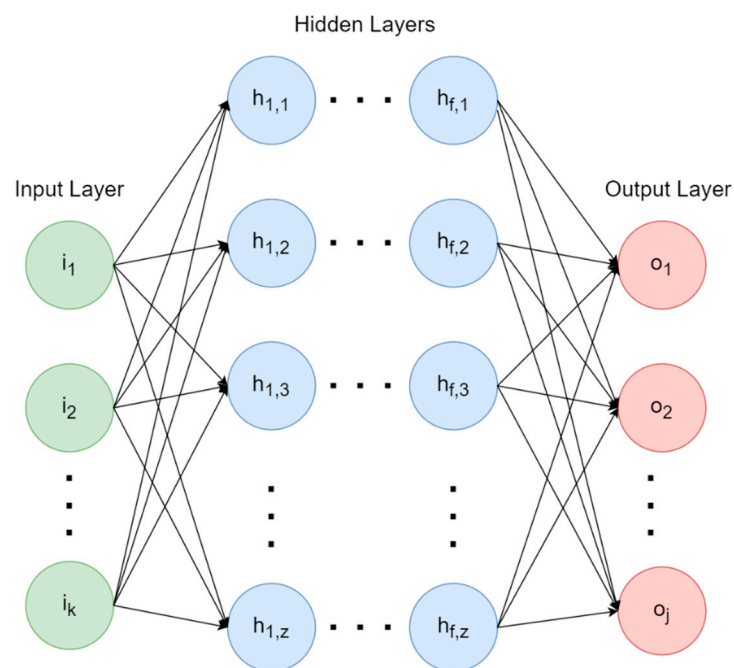


Figure 1. General structure of the fully connected feedforward deep neural network.

The main learnable parameters of the DNN are the weights and biases [48]. Those parameters are initially randomized and iteratively refined through the training process since the network will be able to predict the output after several passes of the training dataset, called epochs. Weights quantify the influential strength that a change in the input could have on the output and biases denote the difference between the generated output and the desired output, essentially quantifying the extent to which the network assumes that the output should have specific values. The training process of the DNN mainly follows the back-propagation algorithm [49] where the generated output values are compared to the desired output and the value of error, which is calculated by a plethora of pre-specified loss functions [50], is fed back to the network, in order to adjust the weights. Since the goal of this training process is to minimize the error function and consequently discover the best weights, optimization methods, such as gradient descent need to be specified for the training process.

The DNN architecture shows an impressive performance in time series forecasting tasks and it is widely used in the energy sector as a standalone network or as a member of hybrid and ensemble learning methods. However, the default configuration of this structure may not always be sufficient for the generation of accurate predictions due to several training scenarios that need to be avoided, such as the existence of local minima [51] of the error function that could hinder the convergence of the network and the occurrence of overfitting or underfitting that are connected to the relative complexity of the model

and the dataset structure. Most deep learning models achieve optimal performance either by following a set of best practices or by exhaustively searching for the best training configuration through hyperparameter optimization [52]. Some of the most important hyperparameters include the number of neurons and layers, the choice of activation function, the choice of optimizer and the associated learning rate [53], the number of training epochs, regularization [54] and the application of early stopping [55]. The search space of those hyperparameters could be large and the total training time needed for the derivation of the best set of hyperparameters could be restrictive for models aimed at short-term and real time forecasts. Therefore, while we often see meticulous and time consuming hyperparameter optimization approaches being suitable for benchmarks, many deep learning approaches rely on the results of experiments with different combinations of best practices complemented by feature selection techniques, in order to derive their baseline models and conduct comparisons. The interpretation of those results, given a specified set of parameters, requires considerable effort towards the practical evaluation of a network and the overall demystification of the black-box structure that provides added value to research work.

2.2. Autoregressive Forecasting Model and Model Selection

Autoregressive models constitute a class of simple time series models used to forecast future values of the target variable based on previous observations of the same variable, called lags [56]. The target variable is linearly dependent on the lags and this relationship occurs due to some degree of correlation between lags of adjacent time steps. The number of lags utilized in the construction of an autoregressive model determines the order of the model and it is usually derived from the inspection of partial autocorrelations. The maximum lag at time step $t - n$ beyond which all other partial autocorrelations are close to zero is often used as an indicator of the order, and the model is expected to perform adequately when including lags up to that time step. The definition of the autoregressive model is made complete by the estimation of the coefficients φ_i that are multiplied by each lag, the constant term c as well as the error term ε_t . The estimation of those parameters is usually achieved with the use of the ordinary least squares method [57]. In order to present a general example, we consider the autoregressive model of order p for the prediction of the value y_t on the next time step of the sequence formed by the variable y with time lags ranging from y_{t-1} to y_{t-p} . The formula that defines this autoregressive model given the previously mentioned parameters is the following:

$$y_t = c + \sum_{i=1}^p \varphi_i * y_{t-i} + \varepsilon_t \quad (1)$$

Since autoregressive models are widely used forecasting tools with several applications in the energy sector, a few core elements need to be explored for optimal performance and the fairness of the model selection process. First, the stationarity of the data needs to be investigated since statistical models often perform better when no trend or seasonality is present. Different implementations of the autoregressive model take into consideration constant and time-dependent trends but the potential inaccurate detection of the trends and their effects on the time series forecast could sometimes lead to larger error terms. In this situation, the augmented Dickey–Fuller test [58] is utilized to determine the stationarity of a time series. According to this method, the null hypothesis assumes that a unit root exists in a time series sample and the alternate hypothesis rejects the previous assumption and considers that the time series is stationary. The p -value of the statistic results in the rejection of the null hypothesis when it is lower than 0.05. Alternatively, the comparison is between the values of the statistic and the critical values of the Dickey–Fuller t -distribution, where the value of the statistic must be more negative than the critical values to confirm stationarity. The stationarity criterion imposes restrictions to the autoregressive model

that could often be seen as necessary countermeasures towards the overall reduction of uncertainty.

Second, the selection of the best autoregressive model plays a crucial role towards the minimization of forecasting error and several information criteria could be considered for the statistical evaluation of fitness to the data, such as the Akaike Information Criterion (AIC) [59], the Bayesian Information Criterion (BIC) [60] and the Hannan–Quinn Information Criterion (HQIC) [61]. The Akaike information criterion provides an estimation of information loss given the number of estimated model parameters k and the maximum value \hat{L} of the likelihood function for the model with the following formula:

$$\text{AIC} = 2k - 2\ln(\hat{L}) \quad (2)$$

Furthermore, the Bayesian Information Criterion follows a similar formula with a slightly altered first term that features the sample size n of the observed data:

$$\text{BIC} = k\ln(n) - 2\ln(\hat{L}) \quad (3)$$

Lastly, the Hannan–Quinn Information Criterion utilizes the previously mentioned parameters in order to derive a more consistent fitness evaluation metric when compared to the AIC and follows the formula:

$$\text{HQIC} = 2k\ln(\ln(n)) - 2\ln(\hat{L}) \quad (4)$$

The selection of models with the lowest values of information criteria and the search for lags that have high autocorrelation values could result in a more accurate estimation of the target variable.

2.3. Proposed Model Structure

This research project focused on the design and implementation of a hybrid day-ahead electricity price forecasting model based on the well-known feedforward deep neural network architecture, with an additional error compensation module that estimates the prediction error and contributes towards the refinement of the final prediction. At the first step, the dataset of the model is constructed, and market data is processed in order to derive the input features, consisting of electricity price lags and exogenous variables relevant to the price time series, as well as the output features of the targeted electricity price sequences for the next day. The dataset is split into training and validation sets, undergoes normalization and is fed to the input layer of the feedforward deep neural network. At the second step, the deep neural network is trained for m epochs featuring an early-stopping mechanism that monitors the decrease of the loss function for the avoidance of overfitting with a specified patience interval, proportional to the number of epochs. Consequently, after m epochs or after the loss function stops decreasing in that patience interval, 24 sequences are generated at the output layer, each one denoting the electricity price prediction for the i_{th} hour of the next day.

At the third step, the sequences are inverted back to their original values and the residual forecasting error for each hourly sequence is calculated from the training set. The definition of the residual training error at every hour h for the price p of the day of interest d given the known values of the training dataset and the predicted output is defined by the formula:

$$p_{d,h}^{residual} = p_{d,h}^{expected} - p_{d,h}^{predicted} \quad (5)$$

Following this step, the residual error sequences are fed to an autoregressive model for their step-by-step estimation, resulting in the derivation of coefficients that are used to predict the error value of the next hour based on historical error data. The final price prediction is derived from the addition of the estimated error and the price forecast of the feedforward DNN. The structure of this model is presented in Figure 2 and this forecasting approach is used in our case study featuring several experiments on different training

scenarios for the interpretation and analysis of the error compensation process. We refer to this model as ERC–DNN in the remainder of this paper.

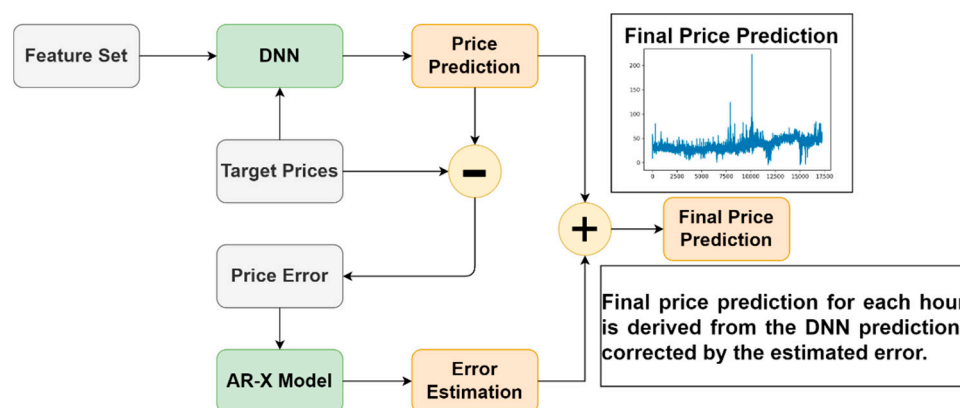


Figure 2. The structure of the ERC–DNN model featuring a feedforward deep neural network and an autoregressive model for error compensation.

2.4. Case Study and Experiments

In this section, we present a case study consisting of several experiments used to test the forecasting performance of the proposed ERC–DNN model and investigate the impact of error compensation in the stability of error profiles for each hour in the day-ahead electricity price prediction task. The dataset used for our experiments contains hourly observations of day-ahead electricity prices, as well as the exogenous sequences that represent the day-ahead forecast of load and the day-ahead forecast of wind generation for the Nord Pool energy market during the time period between 1 January 2013 and 24 December 2018. The dataset is freely available in [62] and was used by the open access benchmark of [39] to evaluate the performance of the standard feedforward deep neural network. The data is organized according to the feature formation proposed by the benchmark. The input features include historical day-ahead prices from the previous three days as well as the prices from one week ago labeled as $p_{d-1,h}$, $p_{d-2,h}$, $p_{d-3,h}$ and $p_{d-7,h}$, respectively, where d denotes the day of interest and h denotes the hour ranging from 1 to 24. Additionally, the day-ahead forecasts of the two exogenous variables are included for the day of prediction, made available on the previous day and labeled as $x_{d,h}^1$ and $x_{d,h}^2$, essentially defining a set of 48 features. Furthermore, historical values of each exogenous variable for the previous day and one week ago, labeled as $x_{d-1,h}^1$, $x_{d-7,h}^1$, $x_{d-1,h}^2$ and $x_{d-7,h}^2$. Lastly, a feature representing the day of the week as a binary vector with 7 elements is included, resulting in a total of 241 input features. The output features consist of the 24 h of day-ahead electricity prices. The dataset is split into a training set of the first 3 years, including the hourly observations from 2013 through 2016 and a validation set of the last 2 years, including years 2017 and 2018 similar to the benchmark model. According to the review and benchmark of [39], the recommended minimum testing period for the evaluation of electricity price forecasting models includes one year of observations since the common practice of including a total of four weeks, one for each season, could be unsuitable due to inadequate representation of the average model performance, the potential exclusion of extreme events that could have an impact on dataset values and the possibility of selecting only the weeks where the model shows improved performance. Therefore, following these recommendations and acknowledging the two-year period used in the benchmark, we believe that the selection of testing period in this review is a suitable evaluation practice and utilize it for the evaluation of our model. Moreover, we acknowledge that the training period varies between price forecasting models and select the maximum available historical data in the remainder of this dataset for our case study in order to have a sufficient number of observations for the convergence of the deep neural network.

Following the guidelines of the open access benchmark, we first constructed a baseline feedforward deep neural network of 4 layers for this multivariate time series forecasting task. The base DNN implements a set of best practices and consists of a fixed set of hyperparameters in order to exclude the performance benefits of hyperparameter optimization and isolate the effects of error compensation in our comparison. The exclusion of hyperparameter tuning at the preprocessing and training steps highlights the role of error estimation as an additional computational layer that reinforces the interpretability of the performance improvement through a smaller and simpler set of parameters. It is evident that the best set of hyperparameters for a forecasting model designed to perform well on a specific machine learning task is dependent on several factors including the dataset, the forecasting horizon, system or application constraints and the intended architecture. The search space for those optimal parameters is large and the resulting optimal set is often chosen based on the improvement of error metrics without having a direct and easily interpretable association to the architecture of the model. On the other hand, error estimation presents the simple concept of error refinement through the discovery of the coefficients that define the polynomial which best fits to the residual error sequences, providing a prediction of the error value that could correct the final prediction of the network by bringing the initial forecast to a value closer to the target. Therefore, error estimation operates independently from the computational structure of the deep neural network and the search goal shifts towards the selection of parameters that could prevent the values of error from exhibiting large variations and irregular patterns instead of proposing a set of parameters that attempt to configure a black-box approach.

The baseline model achieves comparable performance to the open-access benchmark in terms of error metrics as we will analyze in the following sections. The DNN structure contains an input layer of 241 neurons, two fully connected hidden layers with 100 and 52 neurons, respectively, and an output layer with 24 neurons for the prediction of the 24 hourly sequences of the day-ahead prices. The activation function is the rectified linear unit (ReLU) [63] and the optimizer is based on stochastic gradient descent [64] with a learning rate of 0.0005 for the avoidance of local minima. The dataset is normalized using min-max normalization and the neural network features an early-stopping mechanism with a patience interval that is equal to 10% of the total number of epochs in order to ensure the stability of predictions and the avoidance of overfitting. Figure 3 presents the structure of the baseline DNN, which is used to derive the day-ahead price predictions as a core component of the ERC–DNN model.

The DNN is trained and the sequences for price prediction are generated at the output. The experiments presented in this work consider three training scenarios, with 10, 100 and 1000 epochs, respectively, for the investigation of error compensation in a scenario where the values of error are large and the network is not near convergence, a moderate scenario where the error has improved but there is still room for further training and a training scenario where the error of the network could marginally improve after a large number of epochs. In all three experiments, the residual error sequences for each hour are calculated and their stationarity is verified by the augmented Dickey–Fuller test. Additionally, the inspection of the partial autocorrelation function for each error sequence reveals that after the first 24 lags the partial autocorrelations decay to values near zero. The results of the stationarity test as well as the observation of the partial autocorrelation function encourage the integration of an autoregressive model for the estimation of each error sequence. Therefore, the residual error sequences are passed to an AR model utilizing a window of 24 lags for the prediction of the next value of error in each sequence. After the fitting of the model to the data, the autoregression coefficients are computed and the estimated hourly error sequences are added to the electricity price forecasts for the refinement of the final prediction. Furthermore, the information criteria of AIC, BIC and HQIC were examined for the suitability of the 24-lag autoregressive model and the potential refinement of the model selection process when a threshold for feature autocorrelation is set at 0.2, 0.3 and 0.4. This additional experiment could contribute towards the appropriate

selection of hyperparameters that could be included in future benchmarks adopting this technique for post-prediction processing of the model. Since hyperparameter optimization for this type of forecasting task already considers a sizable set of hyperparameters, the choice between the window length and the more complex threshold inspection based on information criteria could often be an important decision that could determine the size of the search space and the overall computational burden for the recalibration of a model or benchmark, given that short-term and real-time forecasting models need to recalibrate relatively fast. Figure 4 presents the diagram for the autoregressive model of the ERC–DNN used in the experiments.

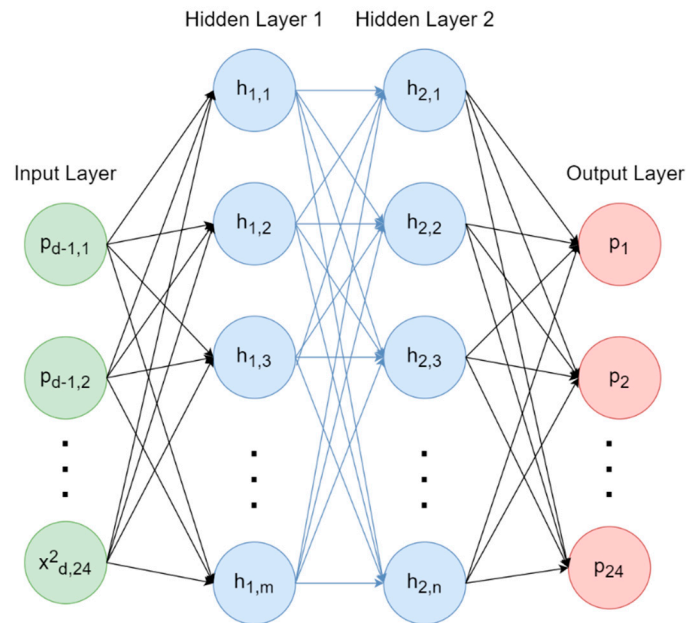


Figure 3. Baseline deep neural network structure integrated in the ERC–DNN model with $m = 100$ and $n = 52$.

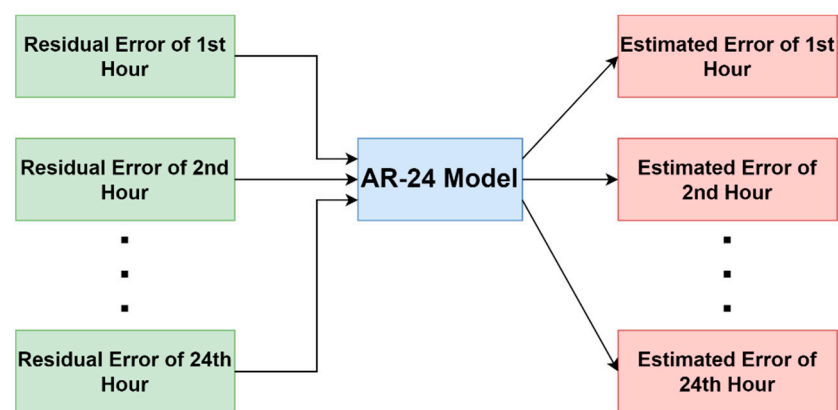


Figure 4. Diagram of the autoregressive error estimation model.

The ERC–DNN model and the experiments analyzed in this research project were developed in Python 3.8.8, using pandas 1.2.3, numpy 1.19.2 and scikit-learn 0.24.1 for data analysis, tensorflow 2.3.0 and keras 2.4.3 for the implementation of the deep neural network model, statsmodels 0.12.2 for the implementation and evaluation of the autoregressive error estimation model and matplotlib 3.3.4 for the visualization of results. The project was executed on a desktop computer with an AMD Ryzen 1700X processor, 8 gigabytes

of RAM, and a NVIDIA 1080Ti graphics processor. The code of this day-ahead electricity price forecasting model is publicly available on GitHub (Supplementary Materials) [65].

2.5. Performance Metrics

In this section, we define the performance metrics utilized in our experiments for the comparison of forecasting error and the examination of the error refinement on the stability of error metrics for each hourly sequence of this day-ahead forecasting task. For the purposes of this study, four error metrics were used to cover different characteristics of the performance evaluation process. Mean absolute error (MAE) [66] provides an easily interpretable and natural error metric that is indifferent to the direction of errors. This performance metric was used as a loss function for the training of the deep neural network, the configuration of the early-stopping mechanism, as well as the evaluation of the ERC–DNN approach. Given the predicted values y_i and real values x_i in a set of n samples, the mean absolute error is computed by the formula:

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (6)$$

Furthermore, mean absolute percentage error (MAPE) [67] was used as a scale independent performance metric since it is a widely used measure for time series regression tasks and could provide a generalized percentage score for forecasting models. Given the same parameters for the calculation of MAE, MAPE is computed by the formula:

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - x_i}{x_i} \right| \quad (7)$$

Moreover, the error metrics of mean squared error (MSE) [68] and root mean squared error (RMSE) [69] is included in the performance evaluation of the experiments since they provide quadratic loss functions that measure the forecasting uncertainty while focusing on the impact of large errors. The values of MSE could express the sum of the variance and square value of bias, further contributing to the performance analysis of a model. Additionally, the values of RMSE increase with the variance of the frequency distribution of error magnitudes, resulting in larger values when large error values are present [70]. Given the same parameters used for the computation of the previously described error functions, the formulae for MSE and RMSE are the following:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2 \quad (8)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} \quad (9)$$

3. Results

In this section, we present the results of the experiments with the inclusion of figures featuring a comparison of error metrics between the ERC–DNN and the baseline DNN for each training scenario. This comparison provides an overview of the stability and performance refinement that occurred in each hourly price sequence after the autoregressive error compensation module is added to the DNN architecture. Additionally, the overall performance of the model for each scenario is presented based on aggregated error metrics, in order to examine the generalized improvement in prediction accuracy stemming from the error estimation process. Furthermore, the exploration of information criteria for the selection of a refined autoregressive model is investigated and the value of implementing a threshold method instead of the window of lagged error observations for error estimation is discussed. Since the performance metrics did not fluctuate greatly after consecutive

executions, the results presented in this section constitute averages from 10 executions for each experiment. It is worth noting that the baseline DNN structure presented in this work performs similarly to the DNN model of the open access benchmark [39] since it achieves a MAE of 1.987, a MAPE of 6.895 and an RMSE score of 3.877 after 4000 training epochs, while the DNN benchmark configuration with the lowest error metrics achieved a MAE of 1.797, a MAPE of 5.738 and an RMSE of 3.474 after hyperparameter optimization. Therefore, the resulting ERC–DNN model is utilizing a highly performant neural network component for the experiments.

First, we consider the training scenario of 10 epochs. The main purpose of this experiment is to present the effect of error compensation on the DNN forecast when the error has larger values that fluctuate greatly from sequence to sequence. In the simple univariate case, we could assume that this scenario refers to a network that has not reached convergence and could be unstable or not properly trained, while in the multivariate case we could observe that each output sequence differs greatly from the desired values and error magnitudes vary for each hour. Error compensation has the greatest impact on this scenario, as the accurate error estimation leads to a larger prediction refinement. In the subplots of Figure 5, we can observe that after the implementation of error compensation, large errors are no longer present, and this greatly improves the MSE and RMSE scores of the model. Moreover, the error profile for each hourly sequence is stabilized, resulting in an average model performance that is close to the model performance for each hourly predicted sequence.

The second experiment considers the training scenario of 100 epochs. In this task, the neural network reaches a more acceptable forecasting performance with each hourly sequence having similar error metrics. As can be observed from the subplots of Figure 6, there are slight error variations between the hourly sequences showing that the network is still unable to predict every hour of the day-ahead prediction equally well. The effect of error compensation in the ERC–DNN improves the forecasting performance and the error metrics are lower than those presented in the open-access benchmark. Since neural network models on sufficiently large datasets do not typically converge after 100 epochs and the values of error are not distinctly high, the slight error variations observed in the baseline evaluation are passed down to the ERC–DNN. Therefore, when compared to the 10-epoch scenario, the performance of the model improved in a similar way but the stability improvement of error among hourly sequences was not as drastic.

The third scenario considers 1000 training epochs and refers to models that are near finalization, where the model converges to predicted values close to the target output and the error metrics remain relatively low. Through this experiment, we can observe that the error metrics could follow more consistent patterns, in this case denoting that the first hourly sequences of the day-ahead forecasting task are predicted more accurately when compared to the last few hours. This phenomenon could be a cause of concern when the model is deployed for real-world applications since the model could generate substantially divergent values for the last few hours of each day. The error compensation improves the performance of this model and flattens the previously described effect, resulting in more consistently accurate predictions. However, it is worth noting that as the neural network is close to reaching convergence, the error values are considerably lower, and the overall refinement of predictions is smaller for larger numbers of epochs. The subplots of Figure 7 visualize this scenario.

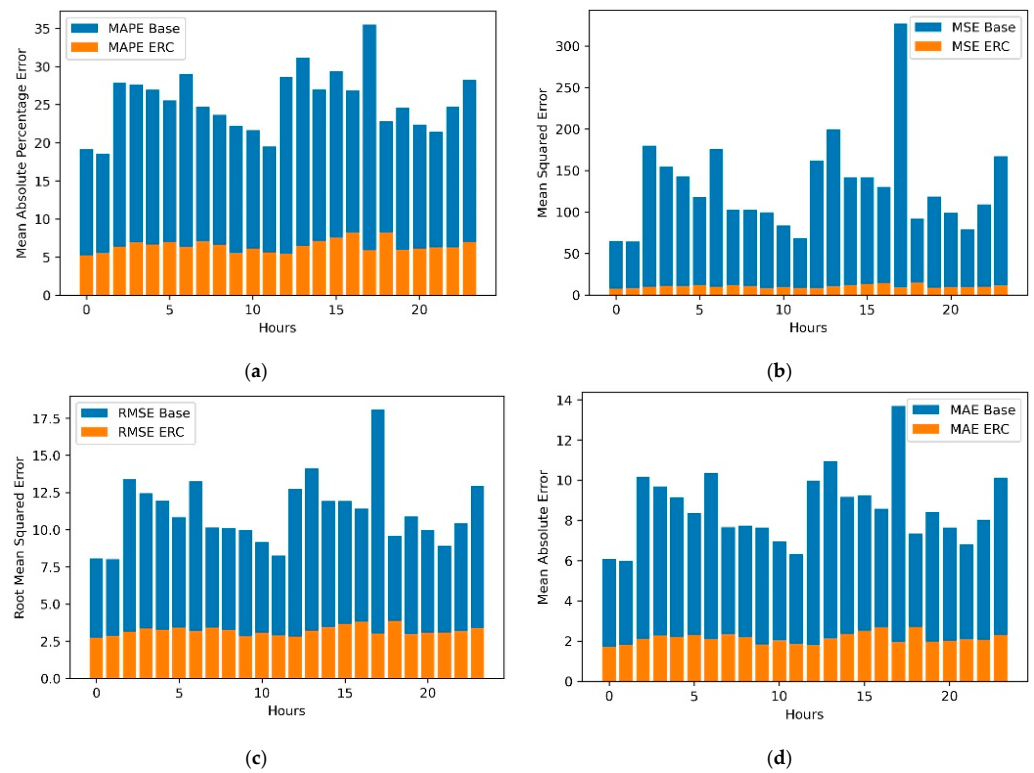


Figure 5. Hourly error metric comparison between the baseline DNN and the ERC–DNN model for 10 training epochs including: (a) MAPE; (b) MSE; (c) RMSE; (d) MAE.

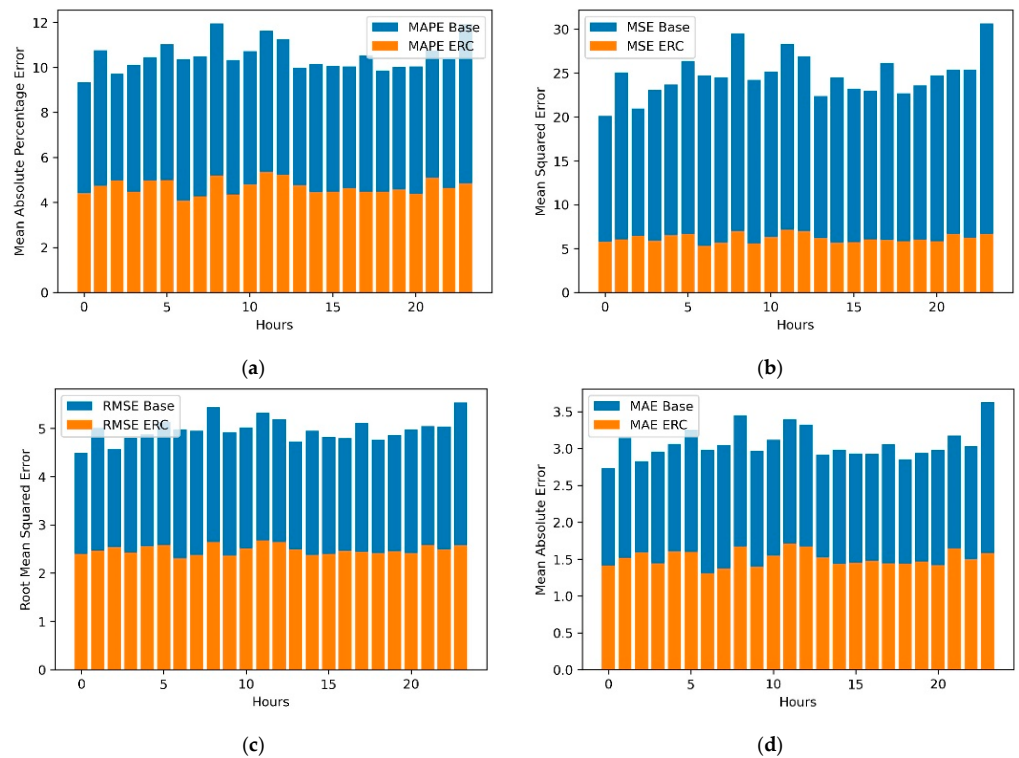


Figure 6. Hourly error metric comparison between the baseline DNN and the ERC–DNN model for 100 training epochs including: (a) MAPE; (b) MSE; (c) RMSE; (d) MAE.

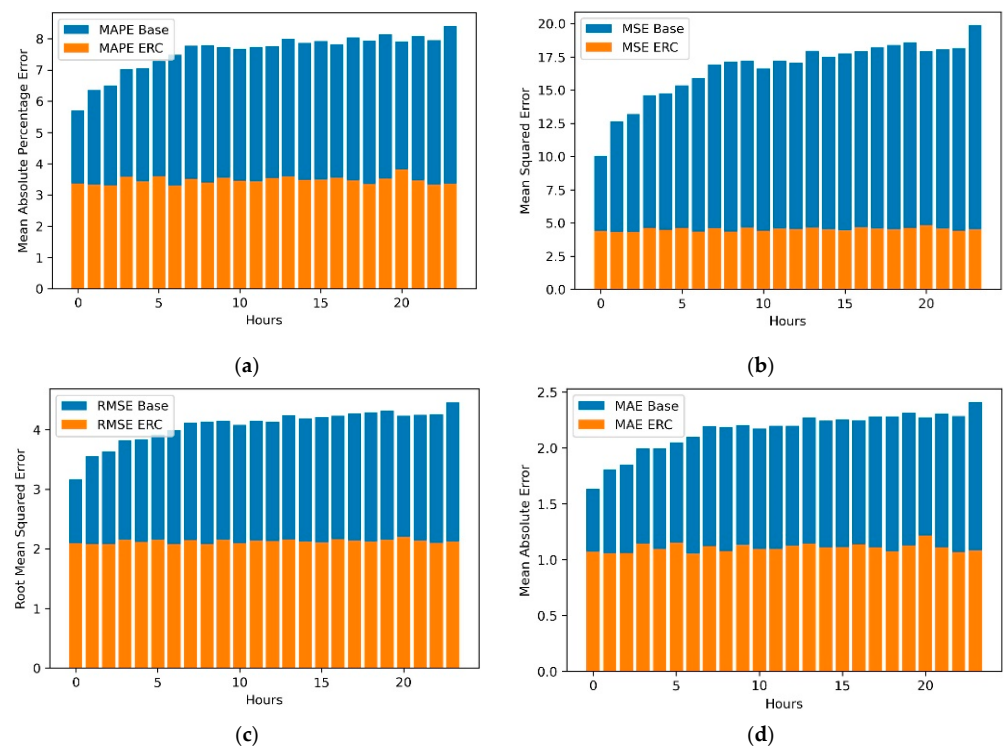


Figure 7. Hourly error metric comparison between the baseline DNN and the ERC–DNN model for 1000 training epochs including: (a) MAPE; (b) MSE; (c) RMSE; (d) MAE.

Overall, we can observe that across all four performance metrics, the integration of the error compensation module refined the predictions and resulted in improved performance in every training scenario, denoting that better and substantially more stable error metrics can be derived even in situations where the neural network is not close to convergence. Table 1 presents the overall error metric comparison that cohesively depicts the impact of this post-processing error estimation model.

Table 1. Error metrics for the performance evaluation of the baseline DNN and the proposed ERC–DNN.

Model	Scenario	MAPE	MSE	RMSE	MAE
Base DNN	10 Epochs	25.375	130.332	11.194	8.581
ERC–DNN	10 Epochs	6.456	10.367	3.206	2.137
Base–DNN	100 Epochs	10.492	24.761	4.970	3.068
ERC–DNN	100 Epochs	4.688	6.165	2.481	1.507
Base–DNN	1000 Epochs	7.583	16.625	4.067	2.156
ERC–DNN	1000 Epochs	3.464	4.510	2.123	1.105

Hyperparameter optimization considers a large space of training parameters in search of a combination that produces optimal error metrics after training. These parameters are specified before the training process starts and affect the error of the model during the training iterations. After the inspection of the results presented in this work, the argument for the inclusion of parameters that regulate error estimation and affect the error after the initial training is complete, such as the window of lagged observations for the definition of an autoregressive model, or the choice of error estimation method could be valid as future benchmarks could consider the full spectrum of error optimization, in an attempt at setting the new standard for model comparisons, where prediction refinement becomes one of the core final steps. However, expanding the search space and introducing additional hyperparameters is not always a viable option, especially when we consider the poten-

tial lack of computing power or the time restrictions imposed by the short recalibration period of real-time models. In this study, the consideration of an autoregressive model utilizing a window of 24 lagged observations for error estimation was a reasonable and computationally inexpensive choice, since the total execution time of the experiments was not dramatically increased. Additionally, the execution of the experiments considered the parameters that could encourage the usage of an autoregressive model for this task, such as the augmented Dickey–Fuller test of stationarity, the computation of partial autocorrelations, and the computation of information criteria for the error estimator.

While the search for the optimal window size based on partial autocorrelations could be regarded as an important step in model selection, and as a potential hyperparameter in more complex optimization problems, the investigation of different model selection criteria could introduce additional hyperparameters is equally necessary. This work explored the information criteria threshold selection method as an alternative to the simpler window selection. The information criteria threshold selection method iteratively fits the autoregressive model using lagged observations that surpass a specified autocorrelation function threshold (ACF). The three information criteria scores of AIC, BIC and HQIC are computed and the model that achieves the lowest score for each hourly error sequence is selected. After examining the scores extracted from this alternative model selection approach in Tables 2–4, we observed that in the scenario of 10 epochs, where the error compensation model achieves the greatest prediction refinement, not all error sequences led to improved information criteria when lagged observations over a certain autocorrelation threshold were selected since the values depend on the error sequences generated by the DNN. This also holds true for the 100 and 1000 epoch scenarios. Furthermore, the improvement of the information criteria is negligible when compared to the 24-lagged window method. Consequently, in the scenario where all hourly error sequences were able to benefit from the threshold method, the increase in forecasting performance would not be impactful enough to justify the computational burden of iteratively searching for the model that satisfies that criteria. Hence, the simplicity of the window method for autoregressive error estimation would be the preferred method for ERC–DNN and the window size would be an appropriate hyperparameter to tune that model.

Table 2. Comparison between the 24-lag window method and the threshold method based on AIC scores for the 10-epoch scenario of ERC–DNN. Cells colored in green denote an improvement in information criteria score while cells colored in blue denote worse overall scores when compared to the window method.

Criterion	24 Lag Window	ACF ≥ 0.2	ACF ≥ 0.3	ACF ≥ 0.4
AIC H0	1.9790	1.9292	1.9457	1.9593
AIC H1	2.3797	2.3523	2.3672	2.4724
AIC H2	2.2449	2.2150	2.2319	2.2567
AIC H3	1.9656	1.9224	1.9224	1.9353
AIC H4	1.8745	1.8592	1.8865	1.9663
AIC H5	1.9847	1.9182	1.9391	2.1244
AIC H6	1.8880	1.8673	1.8815	1.9277
AIC H7	2.1403	2.1229	2.1346	2.2355
AIC H8	2.1850	2.1650	2.2633	2.2739
AIC H9	1.7241	1.7407	1.8363	1.8498
AIC H10	1.9550	1.9456	1.9843	2.1282
AIC H11	1.9167	1.8956	1.9077	1.9281
AIC H12	2.1428	2.0839	2.0982	2.1227

Table 2. Cont.

Criterion	24 Lag Window	ACF ≥ 0.2	ACF ≥ 0.3	ACF ≥ 0.4
AIC H13	1.9478	1.9120	1.9276	1.9573
AIC H14	2.1011	2.0691	2.0691	2.1047
AIC H15	2.0097	1.9600	1.9786	1.9980
AIC H16	1.7016	1.6398	1.6466	1.6662
AIC H17	2.0564	2.0659	2.0754	2.1979
AIC H18	2.1026	2.0382	2.0585	2.0754
AIC H19	2.1297	2.0518	2.0838	2.0931
AIC H20	1.8115	1.7889	1.8569	1.8683
AIC H21	2.3006	2.2009	2.2022	2.2343
AIC H22	1.9503	1.8856	1.8926	1.9041
AIC H23	1.9740	1.9508	1.9687	1.9879

Table 3. Comparison between the 24-lag window method and the threshold method based on BIC scores for the 10-epoch scenario of ERC–DNN. Cells colored in green denote an improvement in information criteria score while cells colored in blue denote worse overall scores when compared to the window method.

Criterion	24 Lag Window	ACF ≥ 0.2	ACF ≥ 0.3	ACF ≥ 0.4
BIC H0	2.0017	1.9432	1.9736	2.0901
BIC H1	1.9672	1.9173	1.9173	1.9173
BIC H2	1.9315	1.8794	1.8915	1.9250
BIC H3	2.1973	2.1854	2.3153	2.3311
BIC H4	2.0103	1.9635	1.9782	2.0938
BIC H5	1.9218	1.8858	1.8889	1.9136
BIC H6	1.8862	1.8436	1.8553	1.8759
BIC H7	2.2287	2.1715	2.1715	2.1715
BIC H8	2.0533	1.9991	2.0230	2.2113
BIC H9	1.7851	1.7451	1.7534	1.7636
BIC H10	1.9882	1.9916	2.0489	2.0610
BIC H11	1.9267	1.8791	1.9633	2.0198
BIC H12	1.9583	1.8886	1.9166	2.0390
BIC H13	2.2644	2.2398	2.2596	2.2722
BIC H14	1.9082	1.8561	1.8827	1.9004
BIC H15	2.0200	1.9953	2.0252	2.0324
BIC H16	2.2123	2.1749	2.1958	2.2805
BIC H17	1.9061	1.8738	1.9013	1.9500
BIC H18	2.2660	2.2364	2.3039	2.3164
BIC H19	2.2894	2.2566	2.2606	2.2723
BIC H20	2.0338	2.0195	2.0519	2.1707
BIC H21	1.9222	1.8812	1.8894	1.8964
BIC H22	2.1217	2.1079	2.1177	2.1850
BIC H23	2.2551	2.1922	2.2201	2.2400

Table 4. Comparison between the 24-lag window method and the threshold method based on HQIC scores for the 10-epoch scenario of ERC–DNN. Cells colored in green denote an improvement in information criteria score while cells colored in blue denote worse overall scores when compared to the window method.

Criterion	24 Lag Window	ACF ≥ 0.2	ACF ≥ 0.3	ACF ≥ 0.4
HQIC H0	1.7015	1.6794	1.7168	1.7683
HQIC H1	1.7430	1.7622	1.9189	1.9189
HQIC H2	1.9680	1.8916	1.8972	1.9285
HQIC H3	2.1361	2.1003	2.1188	2.1306
HQIC H4	2.0807	2.0603	2.0747	2.1937
HQIC H5	2.1427	2.0985	2.1003	2.1074
HQIC H6	1.9908	1.9275	1.9384	2.0965
HQIC H7	2.2277	2.1472	2.1683	2.3292

Table 4. Cont.

Criterion	24 Lag Window	ACF \geq 0.2	ACF \geq 0.3	ACF \geq 0.4
HQIC H8	1.9853	1.9692	2.0512	2.0531
HQIC H9	1.7594	1.7209	1.7608	1.7660
HQIC H10	1.9160	1.9157	2.0202	2.0238
HQIC H11	1.7992	1.7590	1.7652	1.7720
HQIC H12	1.7352	1.6870	1.6892	1.6954
HQIC H13	2.0105	1.9698	1.9835	2.0081
HQIC H14	2.1561	2.1161	2.2623	2.2693
HQIC H15	2.2703	2.2803	2.3123	2.3182
HQIC H16	2.4097	2.3869	2.4141	2.5297
HQIC H17	1.9114	1.8745	1.8745	1.8745
HQIC H18	2.3749	2.4596	2.4653	2.4653
HQIC H19	1.8727	1.8393	1.8588	1.8701
HQIC H20	1.9275	1.8702	1.8853	2.0014
HQIC H21	1.9794	1.9832	2.0800	2.0936
HQIC H22	2.0047	1.9994	2.1103	2.1193
HQIC H23	2.1320	2.0913	2.0997	2.1247

4. Discussion

This paper presented an error compensation deep neural network for the task of day-ahead electricity price forecasting. The proposed model used an autoregressive module to estimate hourly residual error sequences and refine and improve the predictions of the neural network model. This approach was tested in three different training scenarios, where the values of the error were high, moderate, and low in order to cover several potential network behaviors, ranging from fairly unstable to nearly convergent. The ERC–DNN yielded impressive results, with improved error metrics in every training scenario when compared to the baseline model. In detail, the error compensation method stabilized the performance of the poorly trained network in the first scenario, decreasing the value of MAE from 8.581 to 2.137. Additionally, significant performance improvements were observed in the moderate and the longer training scenarios with the values of MAE decreasing from 3.068 to 1.507 in the 100-epoch experiment and from 2.156 to 1.105 in the 1000-epoch experiment. This forecasting approach resulted in improved error metrics when compared to the benchmark results presented in [39].

The improvement of forecasting performance is not the only benefit provided by this approach, since the error compensation method manages to create more consistent predictions, resulting in multivariate models that can predict each hourly sequence at a similar level of accuracy. The inclusion of an autoregressive module resulted in a clear and interpretable approach to error improvement since it operates on the output of neural networks. Therefore, error estimation and refinement through this approach could be easily associated with the analysis of hourly residual error sequences instead of searching for the optimal combination of structural parameters that configure complex deep neural networks in a black-box approach. The design, implementation and testing of this method provides some useful insights towards the development of more robust and stable hybrid models, as well as the integration of error compensation as an additional optimization option for benchmarks during post-processing. However, one potential disadvantage of this method is the dependence on the error sequences and their characteristics. In this project, we implemented several methods, such as stationarity and autocorrelation analysis to ensure that the autoregressive module would behave appropriately. In scenarios where those methods would yield inconsistent results, this approach may not result in substantial error improvements. As a result, we believe that the analysis of error sequences is a crucial part that precedes the integration of that data in post-processing techniques and should not be omitted. Since most hybrid models and benchmarks utilize hyperparameter optimization to search for the optimal combination of parameters that minimize the error metrics, the integration of error compensation could introduce a wide set of additional parameters that would increase the overall complexity of the models and potentially render that

refinement more computationally expensive. While the simple choice of the window size in an autoregressive error estimation model seems to be an appropriate hyperparameter for the configuration of this method, the consideration of more complex estimation methods could result in refinement techniques that greatly hinder the execution time of those models.

The contribution of this work is not limited to the research and development of electricity price forecasting models since there are several ways this approach could benefit market participants and the grid. Firstly, this approach could reduce the price uncertainty of generators while assisting them indirectly in the maximization of profit. Since generators often need to select the highest price after inspecting offers from different markets in order to sell the production [71], this method could lead to more informed decisions due to the increased stability and forecasting performance. Secondly, trading companies could develop more robust short-term contracts due to the availability of more accurate price estimates. Lastly, the grid could benefit from more stable and accurate price predictions since the effect of price volatility could lead to more blackouts and the urgent usage of reserves.

This project attempted to cover several research gaps through the investigation of the error compensation effect on the well-known DNN structure used in open-access benchmarks and several forecasting applications. While recent studies shared a similar direction in the implementation of error compensation on the LSTM structure [40] as well as more traditional statistical methods for different forecasting tasks, this study considered the feedforward deep neural network as the building block for the development of performant forecasting models that include error estimation. The examination of the results in conjunction with recent research findings derived from statistical and machine learning models reinforces the concept that error estimation is a beneficial post-processing technique for deep learning models in the energy sector. There are several additional aspects regarding this method that could be explored in future work. First, a wide comparison of error estimation models ranging from simple statistical approaches to the increasingly complex neural network models could contribute towards the optimal model selection of the error refinement module post-training. Second, the ERC–DNN model could be tested on many electricity markets that display different price characteristics, such as different levels of price fluctuations in an attempt to study the effects of the unique price curve behavior on the training error. Additionally, the inspection of distinctly different error sequences could result in useful insights into the behavior of the model and the adaptability to different market dynamics. Lastly, the benefits of hyperparameter optimization could be studied in combination with error compensation, in an attempt to quantify the overall performance improvement and the computational tradeoff for short-term and real-time applications.

Supplementary Materials: The implementation of this research project in Python code is available online at <https://github.com/dimkonto/ERC-DNN>, referenced in [65] (accessed on 14 January 2022).

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available in a publicly accessible, general-purpose repository. The data used in this study are openly available in [Zenodo] at [10.5072/zenodo.715409], reference number [62]. The dataset was processed as the input for the design and performance assessment of the day-ahead electricity price forecasting approach described in this article. This dataset was chosen for this work due to the insightful comments presented in the open-access benchmark referenced

in [39] and as it is stated on the website hosting the files, the purpose of the data is to help researchers test their methods easily.

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

References

1. What Is Price Volatility, Eia.gov. 2021. Available online: https://www.eia.gov/naturalgas/weekly/archivenew_ngwu/2003/10_23/Volatility%2010-22-03.htm (accessed on 11 January 2022).
2. Fanelli, V.; Schmeck, M. On The Seasonality in the Implied Volatility of Electricity Options. *Quant. Financ.* **2019**, *19*, 1321–1337. [CrossRef]
3. NRG Editorial Voices. The Relationship between Weather and Energy Prices, NRG Energy. 2021. Available online: <https://www.nrg.com/insights/innovation/the-relationship-between-weather-and-energy-prices.html> (accessed on 11 January 2022).
4. Pereira da Silva, P.; Horta, P. The Effect Of Variable Renewable Energy Sources On Electricity Price Volatility: The Case Of The Iberian Market. *Int. J. Sustain. Energy* **2019**, *38*, 794–813. [CrossRef]
5. Prices and Affordability—World Energy Outlook 2021—Analysis—IEA, IEA. 2021. Available online: <https://www.iea.org/reports/world-energy-outlook-2021/prices-and-affordability> (accessed on 11 January 2022).
6. Van Cleef, H. Electricity Monitor—Power Prices Are Skyrocketing! ABN AMRO Group Economics. 2021. Available online: <https://www.abnamro.com/research/en/our-research/electricity-monitor-power-prices-are-skyrocketing1> (accessed on 11 January 2022).
7. Bjarghov, S.; Loschenbrand, M.; Ibn Saif, A.; Alonso Pedrero, R.; Pfeiffer, C.; Khadem, S.; Rabelhofer, M.; Revheim, F.; Farahmand, H. Developments And Challenges In Local Electricity Markets: A Comprehensive Review. *IEEE Access* **2021**, *9*, 58910–58943. [CrossRef]
8. Day-Ahead and Real-Time Energy Markets, Iso-ne.com. 2021. Available online: <https://www.iso-ne.com/markets-operations/markets/da-rt-energy-markets/> (accessed on 11 January 2022).
9. Vivas, E.; Allende-Cid, H.; Salas, R. A Systematic Review of Statistical and Machine Learning Methods for Electrical Power Forecasting with Reported MAPE Score. *Entropy* **2020**, *22*, 1412. [CrossRef] [PubMed]
10. Cincotti, S.; Gallo, G.; Ponta, L.; Raberto, M. Modeling and forecasting of electricity spot-prices: Computational intelligence vs classical econometrics. *AI Commun.* **2014**, *27*, 301–314. [CrossRef]
11. Ferreira, Â.; Ramos, J.; Fernandes, P. A Linear Regression Pattern For Electricity Price Forecasting In The Iberian Electricity Market. *Rev. Fac. De Ing. Univ. De Antioq.* **2019**, *93*, 117–127. [CrossRef]
12. Zhang, J.; Han, J.; Wang, R.; Hou, G. Day-Ahead Electricity Price Forecasting Based On Rolling Time Series And Least Square-Support Vector Machine Model. In Proceedings of the 2011 Chinese Control and Decision Conference (CCDC), Mianyang, China, 23–25 May 2011. [CrossRef]
13. Bikcora, C.; Verheijen, L.; Weiland, S. Density forecasting of daily electricity demand with ARMA-GARCH, CAViaR, and CARE econometric models. *Sustain. Energy Grids Netw.* **2018**, *13*, 148–156. [CrossRef]
14. Marín, J.; Villada, F. Regionalized discount rate to evaluate renewable energy projects in Colombia. *Int. J. Energy Econ. Policy* **2020**, *10*, 332–336. [CrossRef]
15. Cerjan, M.; Matijaš, M.; Delimar, M. Dynamic Hybrid Model for Short-Term Electricity Price Forecasting. *Energies* **2014**, *7*, 3304–3318. [CrossRef]
16. Marcjasz, G.; Lago, J.; Weron, R. Neural Networks In Day-Ahead Electricity Price Forecasting: Single Vs. Multiple Outputs. *arXiv* **2020**, arXiv:2008.08006. Available online: <https://arxiv.org/abs/2008.08006> (accessed on 11 January 2022).
17. Jiang, L.; Hu, G. Day-Ahead Price Forecasting For Electricity Market Using Long-Short Term Memory Recurrent Neural Network. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018. [CrossRef]
18. Khan, Z.; Fareed, S.; Anwar, M.; Naeem, A.; Gul, H.; Arif, A.; Javaid, N. Short Term Electricity Price Forecasting Through Convolutional Neural Network (CNN). In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2020; pp. 1181–1188. [CrossRef]
19. Honkela, A. Multilayer Perceptrons. 2001. Available online: <https://users.ics.aalto.fi/ahonkela/dippa/node41.html> (accessed on 14 January 2022).
20. Srivastava, P. Long Short Term Memory | Architecture of LSTM, Analytics Vidhya. 2017. Available online: <https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/> (accessed on 14 January 2022).
21. Smeda, K. Understand the Architecture of CNN, Medium. 2021. Available online: <https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7> (accessed on 14 January 2022).
22. Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A. Minutely Active Power Forecasting Models Using Neural Networks. *Sustainability* **2020**, *12*, 3177. [CrossRef]
23. Brownlee, J. A Gentle Introduction to Ensemble Learning Algorithms, Machine Learning Mastery. 2021. Available online: <https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/> (accessed on 14 January 2022).
24. Yajnik, A. The Emergence of Hybrid Models in Time Series Forecasting, Medium. 2021. Available online: <https://medium.com/@ayushyajnik2/the-emergence-of-hybrid-models-in-time-series-forecasting-31e0341bb538> (accessed on 14 January 2022).

25. de Marcos, R.; Bello, A.; Reneses, J. Electricity price forecasting in the short term hybridising fundamental and econometric modelling. *Electr. Power Syst. Res.* **2019**, *167*, 240–251. [[CrossRef](#)]
26. Bhagat, M.; Alamaniotis, M.; Fevgas, A. Extreme Interval Electricity Price Forecasting of Wholesale Markets Integrating ELM and Fuzzy Inference. In Proceedings of the 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Greece, 15–17 July 2019. [[CrossRef](#)]
27. Marcjasz, G. Forecasting Electricity Prices Using Deep Neural Networks: A Robust Hyper-Parameter Selection Scheme. *Energies* **2020**, *13*, 4605. [[CrossRef](#)]
28. Alamaniotis, M.; Ikononopoulos, A.; Alamaniotis, A.; Bargiotas, D.; Tsoukalas, L.H. Day-ahead electricity price forecasting using optimized multiple-regression of relevance vector machines. In Proceedings of the 8th Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2012), Cagliari, Italy, 1–3 October 2012; pp. 1–5. [[CrossRef](#)]
29. Alamaniotis, M.; Bargiotas, D.; Bourbakis, N.G.; Tsoukalas, L.H. Genetic Optimal Regression of Relevance Vector Machines for Electricity Pricing Signal Forecasting in Smart Grids. *IEEE Trans. Smart Grid* **2015**, *6*, 2997–3005. [[CrossRef](#)]
30. Zhang, W.; Cheema, F.; Srinivasan, D. Forecasting of Electricity Prices Using Deep Learning Networks. In Proceedings of the 2018 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Kota Kinabalu, Malaysia, 7–10 October 2018. [[CrossRef](#)]
31. Alamaniotis, M.; Gatsis, N.; Tsoukalas, L. Virtual Budget: Integration of electricity load and price anticipation for load morphing in price-directed energy utilization. *Electr. Power Syst. Res.* **2018**, *158*, 284–296. [[CrossRef](#)]
32. Angamuthu Chinnathambi, R.; Mukherjee, A.; Campion, M.; Salehfar, H.; Hansen, T.M.; Lin, J.; Ranganathan, P. A Multi-Stage Price Forecasting Model for Day-Ahead Electricity Markets. *Forecasting* **2019**, *1*, 3. [[CrossRef](#)]
33. Chang, Z.; Zhang, Y.; Chen, W. Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. *Energy* **2019**, *187*, 115804. [[CrossRef](#)]
34. Su, M.; Zhang, Z.; Zhu, Y.; Zha, D. Data-Driven Natural Gas Spot Price Forecasting with Least Squares Regression Boosting Algorithm. *Energies* **2019**, *12*, 1094. [[CrossRef](#)]
35. Atef, S.; Eltawil, A. A Comparative Study Using Deep Learning and Support Vector Regression for Electricity Price Forecasting in Smart Grids. In Proceedings of the 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), Tokyo, Japan, 12–15 April 2019. [[CrossRef](#)]
36. Bissing, D.; Klein, M.; Chinnathambi, R.; Selvaraj, D.; Ranganathan, P. A Hybrid Regression Model for Day-Ahead Energy Price Forecasting. *IEEE Access* **2019**, *7*, 36833–36842. [[CrossRef](#)]
37. Xu, J.; Baldick, R. Day-Ahead Price Forecasting in ERCOT Market Using Neural Network Approaches. In Proceedings of the Tenth ACM International Conference on Future Energy Systems, Phoenix, AZ, USA, 25–28 June 2019. [[CrossRef](#)]
38. Zhang, C.; Li, R.; Shi, H.; Li, F. Deep learning for day-ahead electricity price forecasting. *IET Smart Grid* **2020**, *3*, 462–469. [[CrossRef](#)]
39. Lago, J.; Marcjasz, G.; De Schutter, B.; Weron, R. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Appl. Energy* **2021**, *293*, 116983. [[CrossRef](#)]
40. Tao, C.; Lu, J.; Lang, J.; Peng, X.; Cheng, K.; Duan, S. Short-Term Forecasting of Photovoltaic Power Generation Based on Feature Selection and Bias Compensation–LSTM Network. *Energies* **2021**, *14*, 3086. [[CrossRef](#)]
41. Vega-Márquez, B.; Rubio-Escudero, C.; Nepomuceno-Chamorro, I.A.; Arcos-Vargas, Á. Use of Deep Learning Architectures for Day-Ahead Electricity Price Forecasting over Different Time Periods in the Spanish Electricity Market. *Appl. Sci.* **2021**, *11*, 6097. [[CrossRef](#)]
42. Jiang, P.; Liu, Z.; Wang, J.; Zhang, L. Decomposition-selection-ensemble forecasting system for energy futures price forecasting based on multi-objective version of chaos game optimization algorithm. *Resour. Policy* **2021**, *73*, 102234. [[CrossRef](#)]
43. Li, T.; Qian, Z.; Deng, W.; Zhang, D.; Lu, H.; Wang, S. Forecasting crude oil prices based on variational mode decomposition and random sparse Bayesian learning. *Appl. Soft Comput.* **2021**, *113*, 108032. [[CrossRef](#)]
44. Pourdaryaei, A.; Mohammadi, M.; Karimi, M.; Mokhlis, H.; Illias, H.A.; Kaboli, S.H.A.; Ahmad, S. Recent Development in Electricity Price Forecasting Based on Computational Intelligence Techniques in Deregulated Power Market. *Energies* **2021**, *14*, 6104. [[CrossRef](#)]
45. Subarna, D. Structure of Neural Network | Artificial Intelligence, Engineering Notes India. 2021. Available online: <https://www.engineeringenotes.com/artificial-intelligence-2/neural-network-artificial-intelligence-2/structure-of-neural-network-artificial-intelligence/35410> (accessed on 14 January 2022).
46. Ronaghan, S. Deep Learning: Overview of Neurons and Activation Functions, Medium. 2018. Available online: <https://srngn.medium.com/deep-learning-overview-of-neurons-and-activation-functions-1d98286cfe4> (accessed on 14 January 2022).
47. Kurama, V. Regression in Machine Learning: What It Is and Examples of Different Models, Built in. 2019. Available online: <https://builtin.com/data-science/regression-machine-learning> (accessed on 14 January 2022).
48. Weight (Artificial Neural Network), DeepAI. 2021. Available online: <https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network> (accessed on 14 January 2022).
49. Kostadinov, S. Understanding Backpropagation Algorithm, Medium. 2021. Available online: <https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd> (accessed on 14 January 2022).

50. Brownlee, J. Loss and Loss Functions for Training Deep Learning Neural Networks, Machine Learning Mastery. 2019. Available online: <https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/> (accessed on 14 January 2022).
51. Keim, R. Understanding Local Minima in Neural-Network Training, All About Circuits. 2020. Available online: <https://www.allaboutcircuits.com/technical-articles/understanding-local-minima-in-neural-network-training/> (accessed on 14 January 2022).
52. Ippolito, P. Hyperparameters Optimization, Medium. 2019. Available online: <https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d> (accessed on 14 January 2022).
53. Mack, D. How to Pick the Best Learning Rate for Your Machine Learning Project. 2018. Available online: <https://www.freecodecamp.org/news/how-to-pick-the-best-learning-rate-for-your-machine-learning-project-9c28865039a8/> (accessed on 14 January 2022).
54. Goyal, C. Regularization | Regularization Techniques in Machine Learning, Analytics Vidhya. 2021. Available online: <https://www.analyticsvidhya.com/blog/2021/05/complete-guide-to-regularization-techniques-in-machine-learning/> (accessed on 14 January 2022).
55. Vijay, U. Early Stopping to Avoid Overfitting in Neural Network- Keras, Medium. 2019. Available online: <https://medium.com/zero-equals-false/early-stopping-to-avoid-overfitting-in-neural-network-keras-b68c96ed05d9> (accessed on 14 January 2022).
56. Radečić, D. Time Series From Scratch—AutoRegression Theory and Implementation, Medium. 2021. Available online: <https://towardsdatascience.com/time-series-from-scratch-autoregression-theory-and-implementation-f8c614f738f2> (accessed on 14 January 2022).
57. Jantana, P.; Sudasna-na-Ayudhya, P. Least Squares and Discounted Least Squares in Autoregressive Process. Available online: <https://www.thaiscience.info/journals/Article/SUIJ/10559421.pdf> (accessed on 14 January 2022).
58. Verma, Y. Complete Guide to Dickey-Fuller Test in Time-Series Analysis, Analytics India Magazine. 2021. Available online: <https://analyticsindiamag.com/complete-guide-to-dickey-fuller-test-in-time-series-analysis/> (accessed on 14 January 2022).
59. Bevans, R. An introduction to the Akaike Information Criterion, Scribbr. 2020. Available online: <https://www.scribbr.com/statistics/akaike-information-criterion/> (accessed on 14 January 2022).
60. Analyttica Datalab. What is Bayesian Information Criterion (BIC)? *Medium*. 2019. Available online: <https://medium.com/@analyttica/what-is-bayesian-information-criterion-bic-b3396a894be6> (accessed on 14 January 2022).
61. van der Pas, S.; Gruenwald, P. Almost the Best of Three Worlds: Risk, Consistency and Optional Stopping for the Switch Criterion in Nested Model Selection. *Stat. Sin.* **2018**, *28*, 229–253. [CrossRef]
62. Datasets for Day-Ahead Electricity Prices (Version 3). 2020. Available online: <https://sandbox.zenodo.org/record/715409#.YgzS9uhByUk> (accessed on 14 January 2022). [CrossRef]
63. Bharath, K. Understanding ReLU: The Most Popular Activation Function in 5 Minutes! *Medium*. 2020. Available online: <https://towardsdatascience.com/understanding-relu-the-most-popular-activation-function-in-5-minutes-459e3a2124f> (accessed on 14 January 2022).
64. Stojiljković, M. Stochastic Gradient Descent Algorithm With Python and NumPy—Real Python. 2021. Available online: <https://realpython.com/gradient-descent-algorithm-python/> (accessed on 14 January 2022).
65. GitHub—Dimkonto/ERC-DNN: Error Compensation On Deep Neural Network Model for Day-Ahead Electricity Price Forecasting, GitHub. 2022. Available online: <https://github.com/dimkonto/ERC-DNN> (accessed on 14 January 2022).
66. Fürnkranz, J.; Chan, P.; Craw, S.; Sammut, C.; Uther, W.; Ratnaparkhi, A.; Jin, X.; Han, J.; Yang, Y.; Morik, K.; et al. Mean Absolute Error. In *Encyclopedia of Machine Learning*; Springer: Boston, MA, USA, 2011; p. 652. [CrossRef]
67. de Myttenaere, A.; Golden, B.; Le Grand, B.; Rossi, F. Mean Absolute Percentage Error for regression models. *Neurocomputing* **2016**, *192*, 38–48. [CrossRef]
68. Chaudhary, M. Python | Mean Squared Error—GeeksforGeeks, GeeksforGeeks. 2019. Available online: <https://www.geeksforgeeks.org/python-mean-squared-error/> (accessed on 14 January 2022).
69. Hyndman, R.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [CrossRef]
70. Hale, J. Which Evaluation Metric Should You Use in Machine Learning Regression Problems? *Medium*. 2021. Available online: <https://towardsdatascience.com/which-evaluation-metric-should-you-use-in-machine-learning-regression-problems-20cdaef258e> (accessed on 14 January 2022).
71. Maciejowska, K.; Nitka, W.; Weron, T. Day-Ahead vs. Intraday—Forecasting the Price Spread to Maximize Economic Benefits. *Energies* **2019**, *12*, 631. [CrossRef]