

## Article

# Predicting the Gap in the Day-Ahead and Real-Time Market Prices Leveraging Exogenous Weather Data

Nika Nizharadze <sup>1,†</sup> , Arash Farokhi Soofi <sup>1,2,†</sup>  and Saeed Manshadi <sup>1,\*</sup> 

<sup>1</sup> Department of Electrical and Computer Engineering, Diego State University, San Diego, CA 92182, USA; nnizharadze@sdsu.edu (N.N.); afarokhisooofi7161@sdsu.edu (A.F.S.)

<sup>2</sup> Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA

\* Correspondence: smanshadi@sdsu.edu

† These authors contributed equally to this work.

**Abstract:** Predicting the price gap between the day-ahead Market (DAM) and the real-time Market (RTM) plays a vital role in the convergence bidding mechanism of Independent System Operators (ISOs) in wholesale electricity markets. This paper presents a model to predict the values of the price gap between the DAM and RTM using statistical machine learning algorithms and deep neural networks. In this paper, we seek to answer these questions: What will be the impact of predicting the DAM and RTM price gap directly on the prediction performance of learning methods? How can exogenous weather data affect the price gap prediction? In this paper, several exogenous features are collected, and the impacts of these features are examined to capture the best relations between the features and the target variable. An ensemble learning algorithm, namely the Random Forest (RF), is used to select the most important features. A Long Short-Term Memory (LSTM) network is used to capture long-term dependencies in predicting direct gap values between the markets stated. Moreover, the advantages of directly predicting the gap price rather than subtracting the price predictions of the DAM and RTM are shown. The presented results are based on the California Independent System Operator (CAISO)'s electricity market data for two years. The results show that direct gap prediction using exogenous weather features decreases the error of learning methods by 46%. Therefore, the presented method mitigates the prediction error of the price gap between the DAM and RTM. Thus, the convergence bidders can increase their profit, and the ISOs can tune their mechanism accordingly.

**Keywords:** electricity market; real-time market; day-ahead market; locational marginal pricing; long short-term memory (LSTM); multivariate time series forecasting



**Citation:** Nizharadze, N.; Farokhi Soofi, A.; Manshadi, S. Predicting the Gap in the Day-Ahead and Real-Time Market Prices Leveraging Exogenous Weather Data. *Algorithms* **2023**, *16*, 508. <https://doi.org/10.3390/a16110508>

Academic Editors: Gloria Cerasela Crisan, Ha Duy Long and Elena Nechita

Received: 19 September 2023

Revised: 27 October 2023

Accepted: 29 October 2023

Published: 4 November 2023



**Copyright:** © 2023 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/>).

## 1. Introduction

One major concern in the design of a two-settlement electricity market is the gap in the clearing prices across the DAM and RTM. The DAM is a financial market that schedules the supply and demand before the operating day, while the RTM is a physical market that settles based on the served demand and provided supply. Based on the concept of locational marginal pricing (LMP) [1], the ISOs determine DAM and RTM prices daily using generation units' offers and locational demands. The difference between the locational marginal pricing (LMP) values of the DAM and RTM is an indicator of the surplus or shortage of electricity in the electric grid compared to the predicted values. Wholesale electricity market prices are volatile due to fuel-cost alterations; weather-sensitive generation units, such as solar generation [2] and Distributed Energy Resources (DERs) [3]; weather-related demands [4]; and planned and forced outages [5]. Multiple settlements create more stable prices and lessen the RTM's vulnerability to price surges [6]. It is shown in [7] that with two-settlement electricity markets, generation units have incentives to enter into real-time contracts, which will reduce real-time electricity prices, which in turn will increase social

welfare. Consequently, all market participants will benefit from such a settlement. Nevertheless, there will be a gap between the day-ahead and real-time settlement. The increase in the penetration level of renewable energy resources exacerbated the volatility of energy supply and prices within the RTM [8]. Thus, predicting the price gap between the DAM and RTM has become more complicated. Predicting such a gap plays an integral role in establishing the operating schedules and adjusting the bidding strategies of the market participants, i.e., physical and virtual market participants within the market [9]. This is particularly important for convergence bidders who are interested in a hedge against the price gap across the two markets [10,11]. The day-ahead market (DAM) is a financial market where participants submit their bids for the following 24 h, whereas the real-time market (RTM) is a physical market in which buyers and sellers submit their bids during the day, usually for a 5-min interval. The RTM balances out the differences between DAM purchases and the actual real-time demand and generation of electricity. In this paper, the focus is on predicting the gap between the cleared prices within the two markets. The gap value can provide significant insights about the supply and demand of the electricity market, which could be valuable information for ISOs, market designers, and physical and virtual market participants to help them enhance the efficiency of the market and reduce their risks.

The prediction of day-ahead hourly electricity prices by leveraging an integrated machine learning model is proposed in [12]. In this article, the authors employed Bayesian clustering by dynamics to cluster the data set into several subsets, and Support Vector Machines (SVM) were used to fit the training data into each subset. The error metrics of the integrated model are significantly improved compared to that of the single SVM network. In [13], the authors proposed Auto-Regressive Integrated Moving Average (ARIMA) models to predict next-day prices for Spanish and Californian markets. In [14], a Random Forest regression is leveraged to predict DAM prices. The proposed approach outperformed the ARIMA model. However, this paper does not consider the impacts of exogenous features such as temperature and solar irradiance to predict prices. Moreover, the price gap between the DAM and RTM is not predicted in the literature.

Even though statistical models perform well at identifying patterns and indicators that will influence the price of electricity, they struggle to predict prices accurately in the presence of spikes, which is particularly important for predicting the gap price across a two-settlement market [15]. The electricity market will be cleared based on the net demand, which in turn depends on many characteristics such as weather, temperature, wind speed, and precipitation. Thus, the LMP tends to fluctuate over an operational horizon. In [16], the authors compared the ARIMA model with Artificial Neural Networks (ANNs) to forecast an electricity price. To handle the complexity of the electricity market, ANNs are used in [17]. The increase in the number of computation layers increases the feature abstraction capability of the networks, which makes them better at identifying non-linear trends [18]. An ensemble of CNN-LSTM and an ARMA model is utilized for financial time series data in [19]. In [20], a Temporal Convolutional Neural Network (TCNN) model is utilized for the analysis of financial time series data, specifically focusing on applications in Forex markets. This approach is contrasted with Recurrent Neural Networks and other deep learning models, as well as some of the top-performing Machine Learning methods, to demonstrate its effectiveness in handling financial data. The ARMA model captures the linear dependencies of features and target variables, while CNN-LSTM models the nonlinear spatial connections in data features between adjacent time intervals and also accounts for long-term time-based patterns in the data. The ensemble of CNN-LSTM and the ARMA model achieved a 0.8837 MAE score for The European Union Emission Trading System (EU ETS) dataset. In [21], the LSTM network and a variation of the deep Recurrent Neural Network (RNN) are used to forecast electricity load, and the outputs of the models are compared to those of statistical models. The electricity consumption of the past 10 days is used to predict the electricity consumption of the next day. The LSTM-based network significantly outperformed the Seasonal-ARIMA and Support Vector Regression (SVR)

models. A similar model is used in [22] to predict electricity load, but in this case, in addition to the historical load data, weather datasets are also utilized. However, no change in the model performance is observed when weather-related features are removed and only the time lags are used as inputs. In [23], the LSTM network is used to predict the next 24 h of electricity prices for Australian and Singaporean markets. The mean absolute percentage error (MAPE) was used to evaluate the model, and up to a 47.3% improvement was observed compared to a multi-layer ANN.

According to [24], the prediction of real-time LMP is even more challenging, and most of the approaches adopted from previous studies generate an MAPE of around 10–20%. In [25], a homogeneous Markov chain representation of the RTM LMP is used to predict the RTM LMP for the prediction horizon of 6–8 h. Future prices are computed based on state transition matrices using the Monte Carlo method. Although the mean average error (MAE) metric of the model was 11.75 USD/MWh, it has a huge computation burden. In [26], the authors proposed a deep LSTM (D-LSTM) network to estimate short- and medium-term demand as well as the LMP. The D-LSTM network turned out to have a flat trend without the validation set. However, once the network was tuned, it outperformed the nonlinear auto-regressive network with exogenous variables (NARX) and Extreme Learning Machine (ELM) models in terms of accuracy. In [27], the researchers used the Generative Adversarial Network (GAN)-based video prediction approach on market data from ISO NE to predict RTM LMPs. The market data images are created from the historical data, and by the concatenation of these images, a video stream is created. Consequently, the prediction of the next frame is used to predict the next-hour RTM LMP. The proposed method achieved approximately an 11% MAPE score. However, weather data sets are not utilized to enhance the prediction model of the price spikes. The enhanced convolutional neural networks are also used in [28] to predict electricity load and prices. Here, feature selection is carried out using the Random Forest model, and the extracted features are passed to the convolutional layer, which later is filtered using the max pooling layer. The showcased work resulted in smaller error measurements than the SVR using NYISO market data.

Leveraging the LSTM network to predict the gap between RTM and DAM prices using weather features brings the following question to mind: Can we improve the prediction of the price gap across the DAM and RTM by leveraging exogenous information (e.g., weather data, including solar irradiance)?

The contributions of this paper are summarized as follows:

1. Syncing the exogenous information on weather data with the electricity market information, i.e., prices and demand data, to create an extensive dataset. The significance of leveraging the external dataset is illustrated, and the importance of features is also demonstrated.
2. Both the DAM and RTM are analyzed for price prediction. A realistic set of assumptions is made regarding the availability of features for both the RTM and DAM once the prices are predicted 24–36 h in advance for the following market operation day upon the clearing of the market.
3. The ensemble learning method, namely the Random Forest (RF), is used to calculate the probability distribution of the predicted market prices for the DAM and RTM, as well as the gap.
4. An LSTM architecture is deployed to enhance predictions given the complexity of predicting values for the time series dataset. The proposed model is compared with other statistical machine learning methods, which demonstrate significant improvements.

The rest of the paper is organized as follows. The learning methods used and the LSTM network are discussed in Section 2: Learning Methods. The metrics deployed to evaluate the performance of learning methods are presented in Section 3. The data collection and data cleansing procedures are detailed in Section 4: Data. The performance of the proposed method is evaluated in Section 5. The paper is concluded in Section 6.

## 2. Learning Algorithms and Methodologies

In this section, the methods that are leveraged to examine the direct price gap values are introduced. The described learning algorithms are utilized to predict price gaps between the RTM and DAM as well as to rank features based on their importance. In addition, these methods are leveraged to construct probability distributions for the DAM and RTM price predictions.

### 2.1. Least Absolute Shrinkage and Selection Operator (LASSO)

The objective of the linear regression model is to find a relationship between two variables by fitting a linear equation to observed data points. The most common way to find a fitted line is to use the least-squares method, in which the model finds a fitted line by minimizing the sum of squared residuals; however, shrinking or setting some coefficients to 0 can increase the accuracy of the mentioned model. In the LASSO model, an  $L_1$  regularization term is added to the cost function to address the above-mentioned issue.  $L_1$  regularization is a technique that modifies the objective function of a model by adding a penalty based on the absolute values of the coefficients, leading to simpler and sparser models [29]. The penalty term,  $\lambda$ , controls the amount of regularization. LASSO is a good method to eliminate irrelevant variables and only consider related variables to compute the output of the model. The cost function,  $J$ , of the LASSO method is presented in (1). Here,  $m$  represents the size of the dataset, while  $g$  denotes the model.

$$J_{\text{LASSO}}(\theta) = \frac{1}{2} \sum_{i=1}^m (g_{\text{LASSO}}(x_i) - y_i)^2 + \lambda \|\theta\|_1 \tag{1}$$

In this paper, the LASSO method is utilized to predict the DAM price, the RTM price, and the price gap between the DAM and RTM directly. LASSO can set some coefficients to zero, so it can perform variable selection. On the other hand, LASSO has difficulties handling correlated features. One of the correlated features will have a high coefficient, while the rest will be nearly zero. However, this one feature is selected randomly. In addition, the LASSO algorithm can only learn linear mappings; thus, due to the nature of non-linearity in the existing dataset, it may not be the best family of functions in the hypothesis space.

### 2.2. Support Vector Regression (SVR)

The SVR method is a non-linear learning algorithm. One of the most common versions of SVR regression is  $\epsilon$ -SV regression. The goal of  $\epsilon$ -SV regression is to find a function that has the most  $\epsilon$  divergence for all the data points. The algorithm accepts errors only within the range of  $\epsilon$ , as presented in (2)–(5).

$$\min_{w, b, \zeta, \zeta^*} \frac{1}{2} w_2^2 + C \sum_{i=1}^m (\zeta_i + \zeta_i^*) \tag{2}$$

$$\text{subject to} \quad w^T \phi(x_i) + b - y_i \leq \epsilon + \zeta_i \tag{3}$$

$$y_i - w^T \phi(x_i) - b \leq \epsilon + \zeta_i^* \tag{4}$$

$$\zeta_i, \zeta_i^* \geq 0, i = 1, \dots, m \tag{5}$$

Here, the constant,  $C > 0$ , balances the flatness of a function and the amount up to which deviations larger than  $\epsilon$  are tolerated.  $\phi(x_i)$  maps  $x_i$  into a higher-dimensional space, where  $w$  and  $b$  are coefficients.  $\zeta$  and  $\zeta^*$  represent the distance from the actual values to the margin of the  $\epsilon$ -tube with support vectors. Errors outside the margin are penalized linearly.

A predictor,  $g$ , of the SVR with  $m$ -training examples is presented in (6).

$$g_{\text{SVR}}(x) = \sum_{i=1}^m (-\alpha_i + \alpha_i^*)K(x_i, x) + b. \tag{6}$$

The SVR with an RBF kernel is a non-linear algorithm, and it enables choosing the acceptable error of the model. The hyper-parameter  $\epsilon$  controls the maximum acceptable error for the model. Thus, it is expected that the SVR with an RBF kernel predicts prices better than the LASSO algorithm.

### 2.3. Random Forest Algorithm

The Random Forest is an ensemble learning algorithm. It combines multiple weak models to build a strong predictor by taking advantage of methods called bagging and decision trees. The goal of the decision tree algorithm is to build a tree-like structure from the existing data points, where each leaf will only contain labels from the same class. The algorithm will split the dataset into roughly two halves until the leaves are pure. To find the best split that will keep the tree compact, the impurity function is minimized. In the case of regression tasks, usually the squared loss, as given in (7), is used as an impurity function, while classification problems employ the Gini impurity, as presented in (8).

$$L(D) = \frac{1}{|D|} \sum_{(x,y) \in D} (y - \bar{y}_D)^2 \tag{7a}$$

$$\text{where } \bar{y}_D = \frac{1}{|D|} \sum_{(x,y) \in D} y \tag{7b}$$

Given a dataset,  $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , with  $c$  distinct categories, where  $D_k$  is all the inputs with the label  $k$ , the squared loss impurity outputs the average squared difference of the actual value and the average prediction, while the Gini impurity measures the homogeneity of the classes.

$$G(D) = \sum_{k=1}^c p_k(1 - p_k) \tag{8a}$$

$$\text{where } p_k = \frac{|D_k|}{|D|} \tag{8b}$$

Decision trees learn the exact patterns in the training set, so they do not generalize well enough, so they are prone to overfitting. The Random Forest uses bagging to decrease the high variance caused by decision trees. Bagging generates datasets  $D_1, \dots, D_m$  from the existing data points,  $D$ . The created datasets are the replicated datasets, each consisting of  $k$  features drawn at random but with replacements from the original dataset [30]. The new datasets are equal in size to the original dataset and have approximately the same probability distribution.

The Random Forest consists of large number of decision trees,  $h(x, D_m)$ , from  $D_1, \dots, D_m$ , where  $D_m$  is an independent, identically distributed vector [31]. In the case of classification tasks, the majority vote acquired from all the decision trees will be the prediction, and for regression purposes, the average of all the predictors will be the output. Moreover, the RF algorithm has only two hyper-parameters,  $m$  and  $k$ . Based on empirical evidence, a good choice for  $k$  is  $k = \sqrt{d}$ , where  $d$  is the total number of features in the dataset, and increasing the size of  $m$  will only benefit the model.

The RF algorithm performs feature selection and generates uncorrelated decision trees by choosing a random set of features to build each decision tree. In addition, by averaging the results from each decision tree that the model builds, it also averages variance as well. Consequently, the RF can balance the bias-variance trade-off well. Thus, in this paper, the RF algorithm is utilized to select the most important features.

#### 2.4. Long Short-Term Memory (LSTM) Networks

Neural networks try to model the behavior of the human brain. They consist of artificially created neurons and a set of edges that connect those neurons. Furthermore, each neuron has its associated activation function, which models neuron impulses. The RNN is a special type of neural network, where the input is a sequence. An RNN is very powerful because it not only uses the input to predict the output but it also utilizes the information from previously observed timestamps. All RNNs form a sequence of connected units that represent the state of the network at a timestamp,  $t$ . A single module takes data from the previous unit,  $h_{t-1}$ , and input for that timestamp,  $x_t$ , then uses the  $\tanh$  function to compute the output for a timestamp,  $t$ . According to [32], a finite-sized RNN can compute any function that exists. However, RNNs suffer either from exploding or vanishing gradients when back-propagating through time. To update the weights, the neural network computes partial derivatives of the loss function of the current layer at each timestamp. Consequently, when the gradients are very small, either learning happens at a very slow rate or not at all. To overcome this issue with RNNs, the LSTM network is proposed, as suggested in [33]. The LSTM network is a special kind of RNN architecture. Instead of only using the  $\tanh$  function in a unit, the LSTM network utilizes three gate units: a forget gate, an input gate, and an output gate. The forget gate is responsible for keeping only the relevant information, as given in (9a). It takes an input at timestamp  $x_t$  and the data from the previous hidden layer,  $h_{t-1}$ . Then, the sigmoid function is applied to those inputs, and as a result, the output of the forget gate is somewhere between 0 and 1. The output closer to 0 will be forgotten, and the output with a numeric value of 1 will be kept for further calculations. Furthermore, the input gate decides how the memory cell will be updated, as shown in (9b). First, the candidate value is computed using (9c), then the result is scaled by the output of the input gate to decide by how much the cell state will be updated, as shown in (9d). Finally, the LSTM network employs an output gate, which is a filtered version of the cell state. First, the cell state is normalized using the  $\tanh$  function, then the sigmoid layer that is presented in (9e) is utilized to decide which parts of the memory will be output, as presented in (9f). The outputs of the hidden state,  $h_t$ , and the prediction,  $y_t$ , is the same; however, the notation  $h_t$  are used as a hidden state input at timestamp  $t + 1$ . The structure of the LSTM network for a single unit is given in Figure 1.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (9a)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (9b)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (9c)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad (9d)$$

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

$$h_t = \tanh(c_t) \otimes o_t \quad (9f)$$

The LSTM architecture is better suited for time series problems compared to the other mentioned algorithms. The LSTM model will learn the previously observed sequence before predicting the output, whereas the mentioned models treat each row in the dataset as an individual training sample.

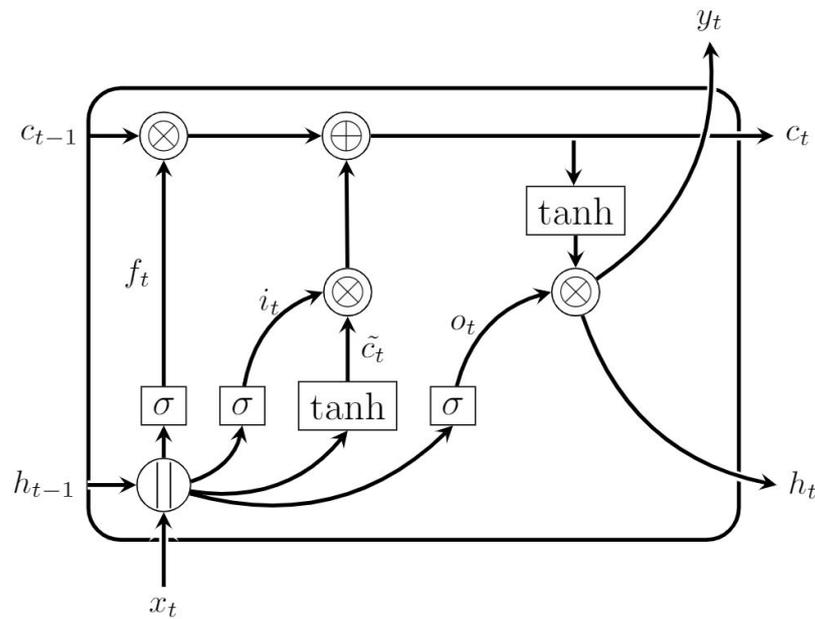


Figure 1. The LSTM architecture of a single unit.

### 3. Prediction Performance Evaluation

The metrics introduced in Table 1 are presented to measure the performance of the learning algorithms presented in the previous section. Here,  $y_t$  is the actual value at time  $t$ , while  $\hat{y}_t$  is the predicted value for the same timestamp. The maximum and minimum values of all the actual values are represented by  $y_{\max}$  and  $y_{\min}$ . In addition,  $s$  is the number of samples in the testing dataset. The mean absolute error (MAE) measures the average magnitude of the errors between the predictions and actual values. Similarly, the RMSE also expresses the average model prediction error. However, it is measured by taking the square root of the average of the squared differences between the actual and predicted values. Both of these metrics measure prediction errors, and they can range from 0 to  $\infty$ . Consequently, lower values characterize a better-performing model. The error metric nRMSE outlined represents the normalized RMSE value. In this case, the normalization is carried out by dividing the RMSE score by the difference between the maximum and minimum values of the actual values. Furthermore, the metric max error represents the evaluation of the worst-case scenario and measures the maximum error in the predicted value of the samples.

Table 1. Prediction errors for Station A while using temporal data for various learning algorithms.

MAE	$\left[ \frac{1}{s} \sum_{t=1}^s  y_t - \hat{y}_t  \right]$
RMSE	$\sqrt{\frac{1}{s} \sum_{t=1}^s (y_t - \hat{y}_t)^2}$
nRMSE	$\sqrt{\frac{1}{s} \sum_{t=1}^s (y_t - \hat{y}_t)^2} / [y_{\max} - y_{\min}]$
Max Error	$\max( y_t - \hat{y}_t ) \forall t \in \{0, s\}$

## 4. Data Preparation

### 4.1. Data Collection

Three distinct datasets are collected and merged to form an extensive dataset for studying the price gap across the DAM and RTM. The first dataset is the one with information from the electricity market. The California Independent System Operator (CAISO) provides an Open-Access Same-time Information System (OASIS) Application Programming Interface (API), which produces reports for the energy market and power grid information

in real-time. To demo the results of this paper, the *MURRAY6N015* node, located in San Diego, CA, is chosen, and its reports for the energy market and power grid information in real-time are leveraged. The LMP, the LMP congestion component, the LMP energy component, and the LMP loss component are collected for the DAM and RTM. Furthermore, a seven-day-ahead load forecast as well as load forecasts for the next two days are acquired using the CAISO API. The period of collection for the dataset is two years, starting from the 1 January 2017.

The second one has meteorological data. For the historical hourly weather dataset, the Meteostat API is used. Meteostat collects hourly weather measurements from more than 5000 weather stations around the world. In addition, it offers comprehensive historical datasets that combine their measurements with the NOAA's Global Historical Climatology Network's dataset. The weather data are obtained from the San Diego International Airport weather station, which is the closest weather station to the node of interest. The collected data include information about temperature, dew point, humidity, wind speed, wind direction, weather condition, sea level pressure, wind gust, cloud layers, and weather forecasts for the next 3 and 6 days. Weather conditions directly and indirectly influence both the demand and supply of electricity, which in turn affects the price. Solar power generation is directly influenced by the amount of sunlight, while the production of electricity from wind turbines depends on wind speeds. If wind speeds are predicted to be low, wind-generated electricity might be reduced, potentially leading to higher prices. On the other hand, weather conditions may influence the demand for electricity. Extreme temperatures, both hot and cold, increase the demand for electricity.

The third dataset concerns renewable energy availability. The mesonet API is utilized to acquire a dataset for solar irradiation. This API offers quality-controlled, surface-based environmental data such as Global Horizontal Irradiance (GHI), Direct Normal Irradiance (DNI), Diffuse Horizontal Irradiance (DHI), solar zenith angle, cloud type, and precipitable water. The GHI is the total amount of terrestrial irradiance received from above by a surface horizontal to the ground. The DNI means the radiation that comes in a straight line directly from the sun and is absorbed by a unit perpendicular to the rays. Furthermore, the DHI is the radiation that does not arrive on a direct path from the sun, and it is equally absorbed by the particles in the atmosphere. It should be noted that the historical weather data (i.e., the second dataset), the forecast weather data (i.e., the third dataset), and the forecast demand data (i.e., the first dataset) are utilized.

#### 4.2. Data Cleansing and Pre-Processing

The collected datasets are merged based on the date and the hour of the day. Only the data from the time span of 1 January 2017, 00:00 to 30 December 2018, 23:00 are utilized. Data cleansing techniques are applied to ensure the quality of the data. Duplicate rows are dropped, categorical variables are converted to numerical representations, and every measurement is converted to a floating-point value. In addition, the missing values are substituted with a global constant. Mean imputation is performed to address cases in which the measurements from the weather dataset are not available at a particular timestamp. Mean imputation replaces all missing values with a mean value calculated across the whole dataset. After data cleansing, since the data for some hours are missing from the official API, 16,566 h worth of data are available. The dataset is arbitrarily split into two parts. The earlier 90% of the data are used for training, and the later 10% are utilized for testing purposes. Then, the input data are normalized using a Min–Max scaler. As a result, each feature is converted into a  $\{0, 1\}$  range. The Random Forest model is employed to select features. The data collected are extensive and combine three different datasets. Consequently, it is important to showcase which features contribute to prediction and which are insignificant. Moreover, feature selection ensures that features that do not affect the prediction are removed and do not introduce extra noise into the system.

## 5. Simulation Results

### 5.1. Feature Importance

LASSO, SVR, and Random Forest algorithms can not inherently capture temporal dependencies for sequential data; that is why day-ahead prices for a previous 48 h time horizon are added as features. Therefore, 48 new columns are created that contain the delayed values of the DAM LMP. Similarly, lagged real-time and gap values are added to the existing dataset; however, for the RTM, the most recent prices that are available are at  $t - 12$  h. Consequently, only those features that are realistically available for the RTM are taken into consideration.

In this section, the RF algorithm is procured to select the most important features. Since the RF method employs decision trees, it can be leveraged for feature selection. The RF naturally ranks by how well each decision tree improves the purity of the node. The Gini index of decision tree algorithms is leveraged to procure feature importance values. For example, the greatest decrease in impurity happens at the root of the tree, while the least decrease in impurity happens at the leaves of the tree. Consequently, pruning the tree below a particular node creates a subset of the most relevant features. In comparison to PCA, the above-described algorithm captures the non-linear dependencies of the features, while PCA only captures linear relationships between features. Table 2 presents the selected features, ranked by their importance for predictions of the price gap between the two markets. The RF algorithm renders 204 features useful for gap prediction. Note that the 13 most relevant features are shown in Table 2. The right column of Table 2 represents the importance coefficient. The importance coefficient is scaled so that the sum of all the importance coefficients is 100. It is interesting that the external features that are collected demonstrated a significant effect on predicting the price gap between the two markets. For example, the solar zenith angle has an importance coefficient of 1.0, while the DHI contributes to the prediction with a 0.39 importance score.

**Table 2.** Selected feature importance for gap predictions.

Feature	Importance Coefficient
GAP LMP price 24 h before	3.32
DAM LMP price 24 h before	1.9
RTM LMP price 24 h before	1.3
Solar Zenith Angle	1.0
Demand Forecast Day-Ahead	0.78
Relative humidity	0.67
Perceptible Water	0.61
Cloud Layer	0.54
Dew point	0.53
Wind Speed	0.47
Wind Direction	0.46
Demand Forecast 2 Days Ahead	0.46
DHI	0.39

### 5.2. Hyper-Parameter Tuning

To perform day-ahead, real-time gap predictions, the hyper-parameters of each learning method presented in Section 2, are optimized. Hyper-parameters control the learning process, and they have to be optimized so that the predefined loss function is minimized for a given dataset. A grid search with nested cross-validation is used to tune hyper-parameters. A grid search is a brute force algorithm that calculates the output for all subsets of predefined parameters and picks the best estimator. The performance of the estimators

is evaluated using time-series nested K-Fold cross-validation, where  $k = 5$ . The time-series nested cross-validation divides the existing dataset into  $k$  inner loops, and each inner loop is split into a training subset and a validation set. Then, the parameters that minimize the error on the validation set are chosen. The outer loop splits the dataset into multiple different training and test sets, and the error on each split is averaged to compute a robust estimate of the model's error. K-fold cross-validation helps mitigate the risk of overfitting and provides a more reliable assessment of how well the model is expected to perform on unseen data.

The Lasso algorithm presented in (1) has only one hyper-parameter:  $\lambda$ . To find the optimal value for  $\lambda$ , a set of arbitrarily chosen values,  $\{0.0001, 0.0002, 0.0003, 0.0004, 0.0005, 0.001, 0.002, 0.003, 0.004, 0.005, 0.01\}$ , is examined. The highest accuracy or minimal loss is acquired using a grid search when the hyper-parameter  $\lambda$  is 0.0003.

The SVR model is optimized for four different hyper-parameters, including  $C$ ;  $\epsilon$ ; the kernel function,  $K$ ; and the kernel coefficient,  $\gamma$ . The optimal value found using the grid search for the constant  $C$  is  $C = 1000$ , while the margin of the tube  $\epsilon = 0.001$  turned out to give the most accurate estimator. In addition, different kernel functions, including linear, sigmoid, and RBFs, are tested, and the most accurate results are obtained using an RBF with  $\gamma = 0.1$ .

Similarly, the Random Forest algorithm is also tuned for hyper-parameters. Generally, an increase in the number of trees in the forest can only benefit the algorithm. However, this increment also introduces significant overhead in computation time, so only forests with 50 and 100 trees were tested, and 100 trees turned out to give more accurate results.

In addition, the maximum number of features considered when looking for the best split turned out to be equal to the total number of features. Moreover, different maximum depths of the trees are passed to the grid search, and the optimal value is found when the nodes are expanded until all the leaves are pure. Finally, the algorithm is tuned for the methods of sampling the data points, and sampling with replacement turned out to be the optimal option.

The LSTM network is optimized for the number of units, loss function, optimizer, and lookback period, which represent several previous timestamps that are considered for a prediction at each time unit. The following set of values,  $\{10, 20, 50, 100\}$ , is examined for the number of units, while the MAE and MSE are tested for the loss function, and Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSProp), and Adaptive Moment Optimization (ADAM) are utilized for optimizer choices. Note that RMSProp is a gradient-based optimization technique that uses the moving average of squared gradients to normalize the gradient, while ADAM is a combination of RMSProp and SGD. Moreover, for lookback options, a day, a week, and a month are tested, and for epoch numbers, the following set of values are examined:  $\{10, 20, 50, 100\}$ . It turned out that 100 LSTM cells with a loss function of the MSE and with an ADAM optimizer resulted in the most accurate results. In addition, the optimal lookback period is a day, and the best number of epochs is 100. Note that here, the different learning methods presented in Section 2 are utilized to predict the electricity prices for the DAM and RTM and the direct gap between DAM and RTM prices.

### 5.3. Analysis of Probability Distributions

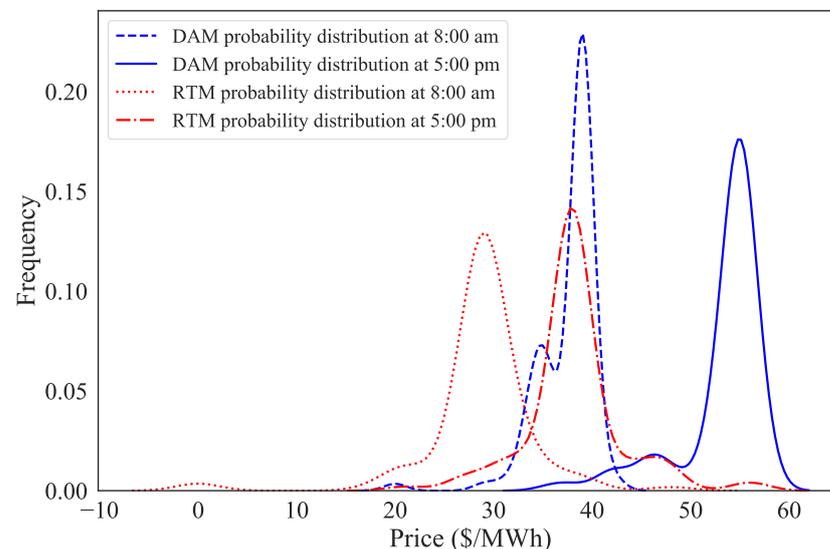
The Random Forest algorithm described in Section 2 is used to calculate the probability distribution of the predicted electricity prices for the DAM and RTM. The outputs of the 100 regression trees are used to approximate the probability distribution for both markets. The spread of predictions from the individual decision trees showcases the uncertainty and variance of the predictions. To represent the results, the 15 October 2018 is chosen for the test, and the hours of interest are 8 a.m. and 5 p.m.

As shown in Figure 2, the prices for the DAM at 8am range from 18 USD/MWh to 44 USD/MWh, while the prices for the same market at 5 p.m. range from 30 USD/MWh to 62 USD/MWh. For this case study, the RTM prices tend to be in a lower range. The price

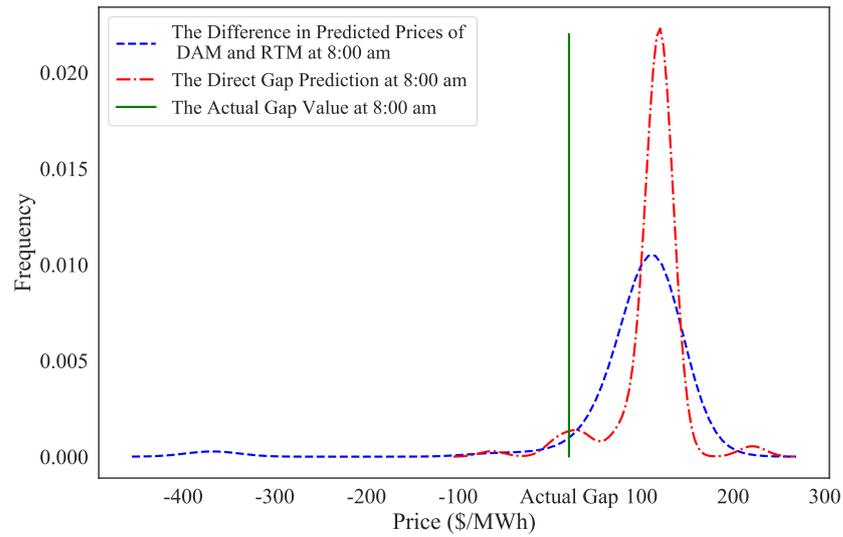
prediction for the RTM at 8 p.m. is in the range of  $-3$  USD/MWh to  $44$  USD/MWh, while the price prediction at 5 p.m. ranges from  $17$  USD/MWh to  $60$  USD/MWh. The electricity prices tend to be much higher at 5 p.m. compared to those at 8 a.m.

Direct predictions by leveraging the RF algorithm would render the most promising probability distribution on the price gap. Figures 3 and 4 present the importance of direct gap prediction in comparison to the difference in predicted prices of the DAM and RTM. In direct gap prediction, the target value for the model is the gap price between the DAM and the RTM. However, to calculate the difference between the predicted DAM and RTM, two distinct models are developed to predict DAM and RTM prices, and then the predictions are subtracted. The time and date are the same as in the case study described above. However, in this case, the actual electricity gap price is also displayed to underline the significance of direct gap prediction. The ground truth gap price for the mentioned date at 8 a.m. is  $17.4$  USD/MWh.

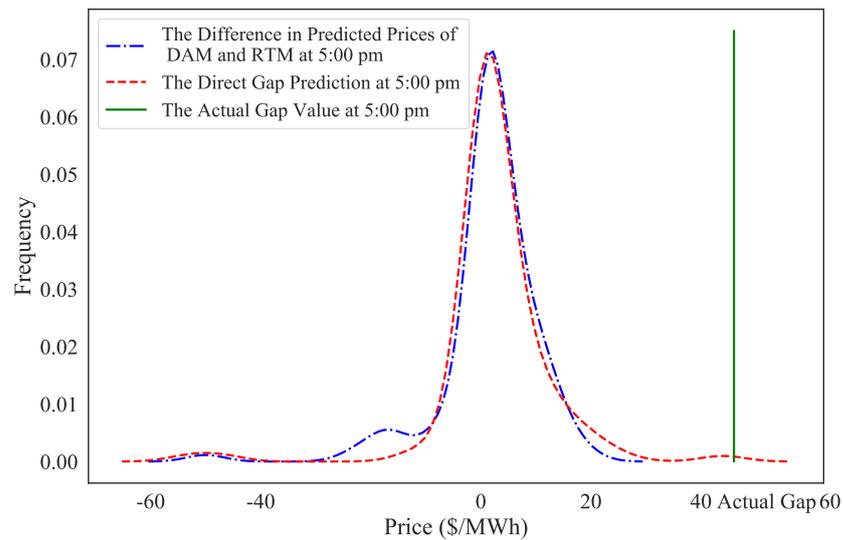
It can be observed from Figure 3 that the probability of the gap procured by the direct gap prediction case is  $17.4$  USD/MWh higher than the probability acquired by subtracting the day-ahead and real-time price predictions. Figure 4 shows the probability distributions for 5 p.m. The direct gap prediction has a higher chance of being more accurate. The actual gap price for 5 p.m. is  $49$  USD/MWh. On the one hand, the price range acquired by subtracting the DAM and RTM price predictions is from  $-60$  USD/MWh to  $30$  USD/MWh; in this case, it would be virtually impossible to correctly predict the actual gap by calculating the difference in predictions for the mentioned markets. On the other hand, the range for direct gap predictions includes the ground truth gap value. Even though, in this case, the probability of accurately predicting the actual gap using the direct gap prediction is not very high, it is still the better choice between those two methods. Furthermore, the direction of the gap difference is more solid toward a positive range in the direct gap prediction than in the difference calculation, which has a smaller expected value.



**Figure 2.** The probability distribution of the DAM and RTM price predictions for a specific date and time, procured by the RF algorithm.



**Figure 3.** A comparison between the probability distribution of direct gap predictions and the difference between separately predicted prices for the DAM and RTM at 8 a.m., procured by LSTM network.



**Figure 4.** A comparison between the probability distribution of direct gap predictions and the difference between separately predicted prices for the DAM and RTM at 5 pm, procured by LSTM network.

5.4. Performance Evaluation

To understand the quantitative insights for the DAM, RTM, and gap prices, the descriptive statistics of the entire data set are presented in Table 3, where all the values are in USD/MWh. It is worth pointing out that the standard deviation for the DAM was almost half of that for the RTM, which means that the values tend to be closer to the mean in the case of the DAM and prices do not fluctuate as much as the price fluctuations in the case of the RTM. The 25th percentile of all gap prices is less than  $-0.16$ , which means that almost a quarter of the direct gap prices are negative.

**Table 3.** Statistical analysis of DAM, RTM, and gap prices of the data set.

Statistics	Gap	DAM	RTM
Mean	3.1	39.1	36.0
Standard Deviation	87.7	45.6	84.5
Min	−1488.6	−61.6	−262.3
Max	2276.2	2374.4	1545.4
25th Percentile	−0.16	25.0	19.6
50th Percentile	6.6	33.4	27.1
75th Percentile	17.5	46.5	36.7

In Tables 4 and 5, an evaluation of the DAM prices and RTM prices procured by the learning methods presented in Section 2 is illustrated, where the unit for the MAE, RMSE, and max error is USD/MWh. All algorithms performed significantly better at predicting DAM prices than predicting RTM electricity prices. For instance, the LSTM network, which is the best-performing algorithm, had an MAE and RMSE of 4.9 and 7.1, respectively, while, for the RTM, the same algorithm resulted in a MAE of 21.2 and an RMSE of 48. Consequently, the complexity of predicting the gap between these markets is dependent on the accuracy of the prediction of the RTM.

**Table 4.** Prediction errors for the DAM.

Error Measure	LASSO	SVR	RF	LSTM
MAE	9.6	11.7	5.1	4.9
RMSE	13.3	33.7	7.9	7.1
nRMSE [%]	7.4	39.8	4.4	4.2
Max Error	95.2	122.7	62	40

**Table 5.** Prediction errors for the RTM.

Error Measure	LASSO	SVR	RF	LSTM
MAE	18.9	21.4	26.4	21.2
RMSE	59	54	71.9	48
nRMSE [%]	5.1	5.0	6.2	4.4
Max Error	1064	1060	1058	1040

LASSO failed to capture any spikes in the price change, and the maximum error between the prediction and the actual gap values was 1054.8 USD/MWh. The poor performance of the LASSO algorithm can be explained by the fact that LASSO is a linear algorithm and leverages a linear function for prediction, while gap prediction should be a non-linear mapping based on empirical evidence.

The most promising results were procured using the RF algorithm and the LSTM network. The Random Forest algorithm had an MAE score of 24.5 USD/MWh and an RMSE score of 67.5 USD/MWh when predicting direct gap prices. Even though these metrics are slightly worse than the above-described algorithms, it can be observed from Figure 5 that the RF algorithm is not as good as the LSTM network. While the RF does a good job of predicting correct values when predicting positive gap prices, it suffers from a notable error in capturing a big negative price spike in gap values. The LSTM network had the best performance in terms of error metrics as well as an empirical evaluation based on the plot provided in Figure 5. Furthermore, Figure 6 illustrates the relative error of the price gap predicted by the LSTM and RF methods. Table 6 shows that the LSTM network

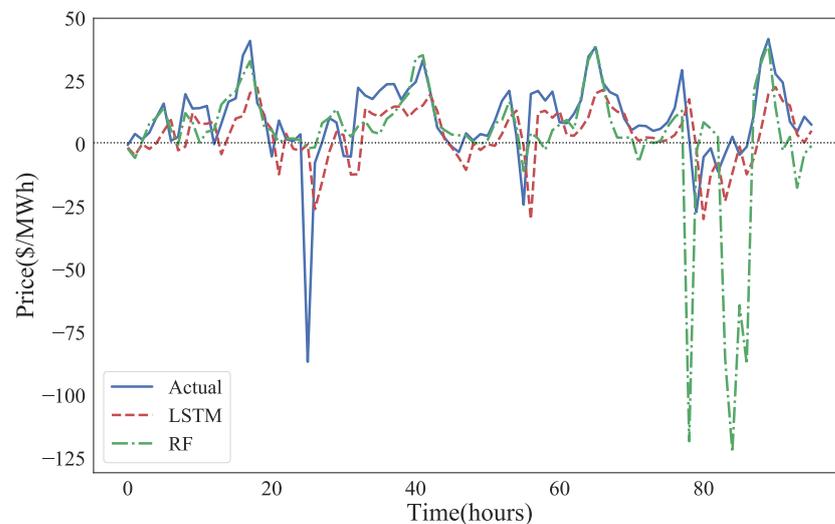
renders the lowest MAE, nRMSE, and Max Error values among all the learning methods used to predict the gap prices. In addition, the LSTM network also outperformed all the methods in predicting the DAM and RTM electricity prices individually.

**Table 6.** Prediction errors for the gap.

Error Measure	LASSO	SVR	RF	LSTM
MAE	19.6	28.2	24.5	17.1
RMSE	58.9	80.4	67.5	56.9
nRMSE [%]	4.98	6.1	5.7	4.8
Max Error	1054.8	1051	1048	1046

### 5.5. Importance of Exogenous Weather Information

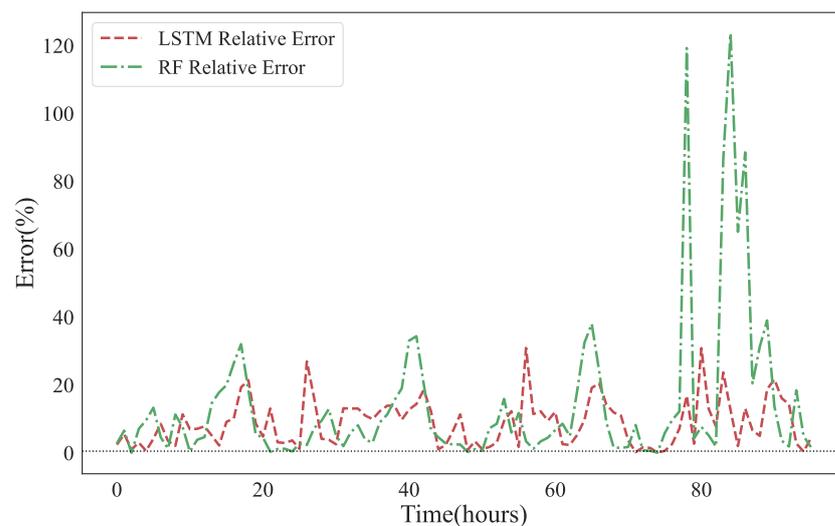
To illustrate the importance of the collected exogenous features, the learning methods described in Section 2 are leveraged without exogenous weather features to predict gap prices, and the results are compared to the predicted gap prices procured by those methods using collected features. Adding exogenous features such as weather conditions and solar irradiance significantly improved the accuracy of the price gap prediction for all the learning algorithms. The error metrics of all the algorithms without exogenous weather features are presented in Table 7. All the algorithms without exogenous weather features had worse error metrics than those with exogenous weather features, as illustrated by comparing Tables 6 and 7.



**Figure 5.** A comparison of predicting the gap using LSTM and RF algorithms for the next 96 h to the actual values of the gap.

**Table 7.** Prediction errors for the gap without exogenous features.

Error Measure	LASSO	SVR	RF	LSTM
MAE	19.7	44.8	24.7	31.8
RMSE	64	81.6	75	62.15
nRMSE [%]	5.4	6.9	6.3	5.2
Max Error	1053.7	1069	1058	1052



**Figure 6.** A comparison of the relative error of direct gap predictions procured by LSTM and RF algorithms for the next 96 h.

## 6. Conclusions

This paper proposed a model to predict the price gap between the RTM and the DAM. To this end, several machine learning algorithms and neural networks are leveraged to obtain the price gap across the DAM and the RTM. To improve the accuracy of the price gap prediction, exogenous weather data, e.g., solar irradiance, is added to the training data of the learning methods and the LSTM. To enable the integration of exogenous weather information, three distinct datasets are collected, matched, and synchronized. It is shown that consideration of related exogenous weather information will outweigh the importance of algorithm selection. Furthermore, this paper investigates the benefits of learning algorithms for direct gap prediction compared to the subtraction of price predictions. To fully achieve this goal, several learning methods are tested to evaluate the performance of the learning algorithms for direct gap prediction compared to the subtraction of price predictions, and it is shown that the prediction error will be lower with a direction price gap prediction. There is no single algorithm that will deliver the best performance all the time. The Random Forest algorithm did a better job of predicting positive gaps as well as the probability distribution of the price gap, while the overall prediction error for the LSTM network was lower. Thus, for future work, it is recommended to consider a combination of the Random Forest algorithm and the LSTM network to predict the price gap. While the former does predict the sign of the gap relatively well, the latter will be able to determine the value of the gap given the sign.

**Author Contributions:** Conceptualization, S.M. and N.N.; methodology, N.N. and A.F.S.; software, N.N. and A.F.S.; validation, S.M., N.N. and A.F.S.; formal analysis, N.N. and A.F.S.; data curation, N.N.; writing—original draft preparation, N.N. and A.F.S.; writing—review and editing, A.F.S. and S.M.; supervision, S.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to access permission restrictions.

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

## References

1. Stoft, S. *Power System Economics: Designing Markets for Electricity*; IEEE Press: Piscataway, NJ, USA, 2002; Volume 468.
2. Woo, C.K.; Moore, J.; Schneiderman, B.; Ho, T.; Olson, A.; Alagappan, L.; Chawla, K.; Toyama, N.; Zarnikau, J. Merit-order effects of renewable energy and price divergence in California’s day-ahead and real-time electricity markets. *Energy Policy* **2016**, *92*, 299–312. [[CrossRef](#)]

3. Khoshjahan, M.; Baembitov, R.; Kezunovic, M. Impacts of weather-related outages on DER participation in the wholesale market energy and ancillary services. In Proceedings of the CIGRE Grid of the Future Symposium, Providence, RI, USA, 17–20 October 2021.
4. Li, Y.; Flynn, P. Electricity deregulation, spot price patterns and demand-side management. *Energy* **2006**, *31*, 908–922. [[CrossRef](#)]
5. Benini, M.; Marracci, M.; Pelacchi, P.; Venturini, A. Day-ahead market price volatility analysis in deregulated electricity markets. In Proceedings of the IEEE Power Engineering Society Summer Meeting, Chicago, IL, USA, 21–25 July 2002; Volume 3, pp. 1354–1359.
6. Veit, D.J.; Weidlich, A.; Yao, J.; Oren, S.S. Simulating the dynamics in two-settlement electricity markets via an agent-based approach. *Int. J. Manag. Sci. Eng. Manag.* **2006**, *1*, 83–97. [[CrossRef](#)]
7. Yao, J.; Adler, I.; Oren, S.S. Modeling and computing two-settlement oligopolistic equilibrium in a congested electricity network. *Oper. Res.* **2008**, *56*, 34–47. [[CrossRef](#)]
8. Bayani, R.; Soofi, A.F.; Manshadi, S.D. Strategic Competition of Electric Vehicle Charging Stations in a Regulated Retail Electricity Market. *arXiv* **2021**, arXiv:2111.11592.
9. Mehdi-pourpicha, H.; Bo, R. Optimal Bidding Strategy for Physical Market Participants With Virtual Bidding Capability in Day-Ahead Electricity Markets. *IEEE Access* **2021**, *9*, 85392–85402. [[CrossRef](#)]
10. Kohansal, M.; Sadeghi-Mobarakeh, A.; Manshadi, S.D.; Mohsenian-Rad, H. Strategic Convergence Bidding in Nodal Electricity Markets: Optimal Bid Selection and Market Implications. *IEEE Trans. Power Syst.* **2020**, *36*, 891–901. [[CrossRef](#)]
11. Kohansal, M.; Samani, E.; Mohsenian-Rad, H. Understanding the Structural Characteristics of Convergence Bidding in Nodal Electricity Markets. *IEEE Trans. Ind. Informatics* **2020**, *17*, 124–134. [[CrossRef](#)]
12. Fan, S.; Liao, J.R.; Kaneko, K.; Chen, L. An integrated machine learning model for day-ahead electricity price forecasting. In Proceedings of the 2006 IEEE PES Power Systems Conference and Exposition, PSCE 2006, Atlanta, GA, USA, 29 October–1 November 2006; pp. 1643–1649. [[CrossRef](#)]
13. Contreras, J.; Espínola, R.; Nogales, F.J.; Conejo, A.J. ARIMA models to predict next-day electricity prices. *IEEE Trans. Power Syst.* **2003**, *18*, 1014–1020. [[CrossRef](#)]
14. Sadeghi-Mobarakeh, A.; Kohansal, M.; Papalexakis, E.E.; Mohsenian-Rad, H. Data mining based on random forest model to predict the California ISO day-ahead market prices. In Proceedings of the 2017 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2017, Washington, DC, USA, 23–26 April 2017. [[CrossRef](#)]
15. Weron, R. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *Int. J. Forecast.* **2014**, *30*, 1030–1081. [[CrossRef](#)]
16. Gao, G.; Lo, K.; Fan, F. Comparison of ARIMA and ANN models used in electricity price forecasting for power market. *Energy Power Eng.* **2017**, *9*, 120–126. [[CrossRef](#)]
17. Adebisi, A.A.; Adewumi, A.O.; Ayo, C.K. Comparison of ARIMA and artificial neural networks models for stock price prediction. *J. Appl. Math.* **2014**, *2014*, 614342. [[CrossRef](#)]
18. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [[CrossRef](#)]
19. He, K.; Yang, Q.; Ji, L.; Pan, J.; Zou, Y. Financial Time Series Forecasting with the Deep Learning Ensemble Model. *Mathematics* **2023**, *11*, 1054. [[CrossRef](#)]
20. Baldo, A.; Cuzzocrea, A.; Fadda, E.; Bringas, P.G. Financial forecasting via deep-learning and machine-learning tools over two-dimensional objects transformed from time series. In Proceedings of the Hybrid Artificial Intelligent Systems: 16th International Conference, HAIS 2021, Bilbao, Spain, 22–24 September 2021; pp. 550–563.
21. Zheng, J.; Xu, C.; Zhang, Z.; Li, X. Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network. In Proceedings of the 2017 51st Annual Conference on Information Sciences and Systems, CISS 2017, Baltimore, MD, USA, 22–24 March 2017; pp. 1–6. [[CrossRef](#)]
22. Bouktif, S.; Fiaz, A.; Ouni, A.; Serhani, M.A. Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies* **2018**, *11*, 1636. [[CrossRef](#)]
23. 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 2018, Singapore, 18–21 November 2018; pp. 949–954. [[CrossRef](#)]
24. Aggarwal, S.K.; Saini, L.M.; Kumar, A. Electricity price forecasting in deregulated markets: A review and evaluation. *Int. J. Electr. Power Energy Syst.* **2009**, *31*, 13–22. [[CrossRef](#)]
25. Ji, Y.; Kim, J.; Thomas, R.J.; Tong, L. Forecasting real-time locational marginal price: A state space approach. In Proceedings of the Conference Record—2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–6 November 2013; pp. 379–383. [[CrossRef](#)]
26. Mujeeb, S.; Javaid, N.; Ilahi, M.; Wadud, Z.; Ishmanov, F.; Afzal, M.K. Deep long short-term memory: A new price and load forecasting scheme for big data in smart cities. *Sustainability* **2019**, *11*, 987. [[CrossRef](#)]
27. Zhang, Z.; Wu, M. Predicting real-time locational marginal prices: A GAN-based video prediction approach. *arXiv* **2020**, arXiv:2003.09527.

28. Zahid, M.; Ahmed, F.; Javaid, N.; Abbasi, R.A.; Kazmi, H.S.Z.; Javaid, A.; Bilal, M.; Akbar, M.; Ilahi, M. Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids. *Electronics* **2019**, *8*, 122. [[CrossRef](#)]
29. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Methodol.* **1996**, *58*, 267–288. [[CrossRef](#)]
30. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [[CrossRef](#)]
31. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
32. Siegelmann, H.T.; Sontag, E.D. Turing computability with neural nets. *Appl. Math. Lett.* **1991**, *4*, 77–80. [[CrossRef](#)]
33. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.