Electric Power Systems Research 229 (2024) 110129

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

An optimized deep learning approach for forecasting day-ahead electricity prices

Çağatay Berke Bozlak *, Claudia Fernanda Yaşar

Department of Control and Automation Engineering, Yildiz Technical University, Istanbul, Turkey

ARTICLE INFO

Keywords:
Electricity price forecasting
SARIMAX
LSTM model
CNN model
Prediction algorithms
TensorFlow

ABSTRACT

Electricity price forecasting is essential for reliable and cost-effective operations in the power industry. However, the complex and nonlinear structure of the electricity price series presents uncertainties and challenges for energy management. To address this, artificial intelligence models such as SARIMAX, LSTM, and CNN-LSTM have been developed to predict short-term electricity prices. These models were tested using Mean Absolute Error, Root-Mean Squared Error, Mean Absolute Percentage Error, and percentage accuracy to verify their accuracy and to compare forecasting methodologies. The study includes the Diebold–Mariano test to confirm the statistical significance of the difference between the forecast errors of the two models. Ensemble learning was used to optimize a CNN-LSTM model, which automatically selects the best model by using CNN to extract valuable characteristics and LSTM to recognize data dependency in time series. Historical data from the German electrical market were used to validate the models' prediction performance. The results showed that the LSTM and CNN-LSTM models outperformed the SARIMAX model in terms of accuracy and simplicity, with the CNN-LSTM technique having significant forecasting advantages. These methods can be used for intelligent optimization forecasting of electricity prices.

1. Introduction

Price forecasting involves predicting the price of a product or service based on various factors, such as demand, seasonal trends, and the inclusion of other commodities. In the case of electrical prices, variations are often caused by cyclical changes in supply and demand, influenced by various factors unique to each electricity market. Factors such as electricity consumption in industrialized nations, greater renewable output, and rising natural gas prices have all affected these patterns [1]. Several research studies focus on developing methodologies for keeping track of market movements, identifying broad trends to reduce generation uncertainty, assisting regulators in making decisions on production and demand offers, and determining the level of market competition for businesses and customers [2–4].

Various methods such as multi-agent, reduced-form, fundamental, equilibrium models, statistical time series approaches, Artificial Intelligence (AI), ensemble, and portfolio decision models are used to predict energy prices and load [4]. Electricity prices have distinct characteristics, such as nonlinear historical data and the impact of large data on forecasting algorithm accuracy [5]. They exhibit seasonality on a weekly and daily basis, including peak and off-peak hours, surges, and so on. However, energy consumption forecasting is challenging because it is a resource that must be consistently supplied and is not easily stored.

There is no single forecasting model superior to all others for predicting electricity prices. Each model has its strengths and limitations, and the most appropriate model will depend on factors such as the available data, the forecasting horizon, and the specific objectives of the forecast. This study evaluates the forecasting efficacy of the Seasonal Auto-Regressive Integrated Moving Average with Exogenous Variables (SARIMAX) in addition to deep learning methods such as Long Short-Term Memory (LSTM) and CNN-LSTM. CNN-LSTM is a combination of LSTM with a Convolutional Neural Network (CNN) model. The models are used to forecast day-ahead electricity prices using historical data from the German electricity market in 2019, 2020, and 2021 (through September 29th) [6]. In addition, to assess unanticipated variance, this analysis anticipates electrical prices for the years 2019–2021, taking into account the effects of the unstable data during the pandemic crises. The inclusion of exogenous variables in the model is necessary to identify price trends and spikes using the total hourly electricity consumption [7]. In this context, SARIMAX and CNN-LSTM were employed to add the impact of this external variable. To compare model accuracy, The Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and percentage

* Corresponding author.
E-mail address: berke.bozlak@std.yildiz.edu.tr (Ç.B. Bozlak).

https://doi.org/10.1016/j.epsr.2024.110129
Received 11 May 2022; Received in revised form 25 December 2023; Accepted 7 January 2024
Available online 16 January 2024

accuracy of the SARIMAX, LSTM, and CNN-LSTM techniques are used as metrics for error measurement. In addition, the Diebold–Mariano test was used as a statistical assessment to determine whether one forecast model is significantly better than another by comparing the forecast errors.

The accurate performance of the deep learning approaches LSTM and CNN-LSTM, as well as their relatively easy implementation for future forecasting, is demonstrated in this work. In terms of anticipating outcomes for the hourly behavior of historical data, the CNN-LSTM method outperforms all models. However, historical data revealed that large, unexpected price changes in the 2021 timeframe reduced the accuracy of the CNN-LSTM model due to the exogenous variable's effect. The LSTM and CNN-LSTM need less time to converge and allow for the detection of hidden patterns. The deep learning methods proved to be useful when dealing with complex, non-linear forecasting problems and were scalable to larger data sets.

The algorithms were written in Python using the packages Pandas, Scipy, and Numpy [8]. TensorFlow [9] and Keras [10] libraries were used to create LSTM and CNN-LSTM models, while the pmdarima library [11] was used to program the SARIMAX. This paper is structured as follows: Section 2 introduces the related works. Section 3 presents the mathematical framework for the programmed models. In Section 4, study cases based on real electricity market data are described. Results and discussion are written in Section 5, while the relevant conclusions are written in Section 6.

2. Related works

Accurate forecasting of price increases and unexpected changes is critical, and the inclusion of exogenous variables and underlying causes of price spikes is necessary in the model [7]. In this context, SARIMAX is employed [12]. In 2022, Hou et al. conducted a review of load forecasting methodologies and challenges based on AI [13]. AI has the potential to enhance forecasting accuracy and efficiency, but several gaps in the literature must be addressed to ensure the effective implementation of precise algorithms with rapid convergence. These gaps include interpretability, data availability, understanding of uncertainty, overfitting, and dataset diversity. By using genetic algorithms for bootstrap stacking ensemble learning, wind power forecasting accuracy was improved [14].

Artificial Neural Networks (ANNs), machine learning, and artificial intelligence-based technologies have proven to be effective [15]. Some papers have compared ANNs to ARIMA models in terms of forecasting abilities. The data generation process and linearity constraints limit the performance of ARIMAX models, while ANNs are simple, adaptable, and can account for nonlinear modeling. Hybrid approaches that combine ANNs and time series have also been explored [16]. ANNs-based forecasting systems are highly successful if sufficient data is available for training and substantial computing resources are available. Additionally, since ANNs are data-driven, they can approximate nonlinear functions and tackle problems where the input–output relationship is poorly defined or difficult to compute. RNNs are used in research involving sequential data, such as text, audio, and video. However, RNNs struggle to learn relevant information from the input data when the input gap is substantial.

Recent advances in deep learning have led to the increased use of LSTM models in time series forecasting and natural language processing [16,17]. LSTMs can handle the issue of long-term dependencies by using gate functions to selectively keep, forget, or ignore data points based on a probabilistic model that includes a short-memory component. Studies have shown that LSTMs outperform standard ANNs and ARIMA models in wind speed prediction [18,19]. LSTMs are also effective in addressing gradient problems during the training phase. Combining CNNs with LSTMs (CNN-LSTM) improves price forecasting accuracy by detecting data interdependence in time series [20]. Additionally, CNN-LSTMs are more efficient than several other established approaches for short-term forecasting of residential electricity consumption [21].

Up-to-date literature on electricity price forecasting indicates that Neural Networks (NN) models outperform non-Neural Networks algorithms [22,23]. Within NNs, LSTM models demonstrate superiority over Gated Recurrent Units (GRU) [24]. Other models, like the Adaptive Neuro-Fuzzy Inference System (ANFIS) or the Genetic Algorithm in ANN (ANN-GA), are also considered for forecasting [25]. In this study, the CNN-LSTM model exhibits significant superiority over other approaches. On the other hand, non-Neural Networks models such as XGBoost have been used for the same purpose [26]. However, the SARIMA model notably outperforms XGBoost. Over longer forecast periods, ARIMA models also outperform regression models like those that Support Vector Regression (SVR) [27]. This research aims to provide a fair comparison among the most successful methods. Thus, it investigates LSTM along with an enhanced CNN-LSTM variant, comparing it with its equivalent SARIMAX method.

3. Dataset preliminaries

In the following Subsections, a short description of the dataset preliminary analysis, mathematical models and parameters, and programmed methods are summarized.

3.1. Data features and analysis

The dataset used in this study comprises price and demand features spanning the years 2019, 2020, and 2021. This timeframe allows for the analysis of the pre-pandemic, pandemic, and post-pandemic periods in Germany. While the primary focus is on price data, the impact of electricity demand as an exogenous variable on the accuracy of the results is also explored. A flowchart in Fig. 1 outlines the necessary preprocessing steps for analyzing the dataset. The process involves identifying missing data and filling it, followed by testing for normality and performing power transformations as needed. The correlation test is used to determine interdependence among variables as a preliminary measure of prediction success. Standardization of the dataset is required for many machine-learning approaches, and in this study, a minimum-maximum scaling strategy was employed.

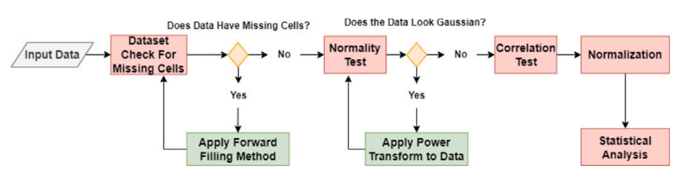

Fig. 1. Flowchart: steps to analyze electricity price dataset.

The dataset consists of historical data from the SMARD application on German electricity prices and total electrical consumption. Section 4 provides a statistical summary of the case studies as a preliminary analysis before using the dataset to train prediction models.

3.2. SARIMAX model

The ARMA model is the summation of the AR and MA models. The ARIMA model is obtained when an integration operator is added to the ARMA model. A SARIMAX model takes the exogenous variables measured at time t that affect the value of input data at time t and integer multiples of seasonality [28]. Table 1 includes a list of the main parameters used to define the SARIMAX model.

Table 1
SARIMAX, List of symbols and model parameters.

Symbol	Remark
p	Number of time lags to regress on.
β	Constant (Measured as deviations from its mean).
θ_i	Parameters of Moving Average.
L	Lag operator
d	Order of differencing used.
$\Theta(L)^p$	An order p polynomial function of L.
$\phi(L)^q$	An order q polynomial function of L.
y_t	Prediction value
n	Number of exogenous variables.
Δ^d	Integration operator.
β_n	Coefficients of exogenous variables.
x_t^i	Exogenous variables defined at each time step t.
Δ_s^D	Differencing operator.
q	Number of time lags of the error term to regress on.
$\phi(L^s)^Q$	An order Q polynomial function with seasonality of L.
ϵ_t	Gaussian white noise at time t. (Zero mean)

3.3. Input data of the LSTM and CNN-LSTM model

The Autoregressive Model AR(p) can be written as:

$$y_t = \Theta(L)^p * y_t + \epsilon_t \tag{1}$$

while the Moving Average Model MA(d) is expressed as follows

$$y_t = \phi(L)^q * \epsilon_t + \epsilon_t \tag{2}$$

Autoregressive Moving Average Model ARMA(p,q) is stated as

$$y_t = \Theta(L)^p * y_t + \phi(L)^q * \epsilon_t + \epsilon_t \tag{3}$$

and the Autoregressive Integrated Moving Average Model ARIMA(p,d,q) is

$$y_t^{[d]} = \Delta^d * y_t = y_t^{[d-1]} - y_{t-1}^{[d-1]} \tag{4}$$

$$\Delta^d * y_t = \Theta(L)^p * \Delta^d * y_t + \phi(L)^q * \Delta^d * \epsilon_t + \Delta^d * \epsilon_t \tag{5}$$

Seasonal Autoregressive Integrated Moving Average Model with Exogenous Variable SARIMAX $(p,d,q) * (P,D,Q)_s$ can be written as follows:

$$\Theta(L)^p * \theta(L^s)^p * \Delta_s^D * \Delta^d * y_t = \phi(L)^q * \phi(L^s)^Q * \Delta^d * \Delta_s^D * \epsilon_t + \sum_{i=1}^{n} \beta_i * x_t^i \tag{6}$$

3.4. LSTM and CNN-LSTM

LSTM and CNN-LSTM networks were programmed, and the list of main parameters and functions are written in Table 2 [29]. An LSTM network is a type of RNN that processes input sequences using three gates: the input gate, the forget gate, and the output gate. The forget gate determines which input should be rejected from the cell state, and the input gate defines the new value. A candidate value vector is produced by a selected activation function and added to the state. The output is obtained based on both the cell and a filtered version of it, and a designated activation function is used to determine which part of the cell state goes to the output [29,30]. The LSTM network can be described by Eqs. (7)–(11).

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

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

$$\widetilde{C}_t = tanh(W_i * [h_{t-1}, x_t] + b_c) \tag{9}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{10}$$

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

The 1D forward propagation of each CNN layer can be written as [31].

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} conv1D(w_{ik}^{l-1}, s_i^{l-1}) \tag{12}$$

$$y_k^l = f(x_k^l) \tag{13}$$

$$s_k^l = y_k^l \downarrow ss \tag{14}$$

Table 2
LSTM and CNN-LSTM, List of symbols and model parameters.

Symbol	Remark
	LSTM model
f_t	Forget gate.
i_t	Input gate.
o_t	Output gate.
W_f	Weight for forget gate neurons.
σ	Sigmoid function.
W_i	Weight for input gate neurons.
$tanh$	Hyperbolic tangent function.
W_o	Weight for output gate neurons.
h_{t-1}	Output for the previous LSTM block. (At time step t − 1.)
x_t	Input at current time step.
b_f	Biases for forget gate.
b_i	Biases for input gate.
b_o	Biases for output gate.
$\widetilde{C}_t$	Candidate for cell state at time step(t).
	CNN model
x_k^l	Input.
b_k^l	Biases of the kth neuron at layer l.
s_i^{l-1}	Output of the ith neuron at layer $l-1$
w_{ik}^{l-1}	Kernel from the ith neuron at layer $l-1$ to the kth neuron at layer l.
$conv1D$	1D convolution.
y_k^l	Intermediate output.
$f(.)$	Activation function.
s_k^l	Output of the kth neuron at layer l.
$\downarrow ss$	Down-sampling operation with a scalar factor, ss.

4. Case study

The data utilized in this study were obtained from the SMARD application, which provides information on electricity prices and total electrical consumption in Germany. Numerous other research papers [23,26] have also focused on the German market. This region was chosen for this work due to the availability of consistent, reliable, and hourly data. The data were collected for the years 2019, 2020, and 2021, with the latest available data being until September 29th, 2021, as reported in [6]. Summary statistics of the historical data are presented in Table 3. The analysis was conducted using the Pandas, Scipy, and Numpy libraries in Python. Analysis of the mean values revealed a continuous growth in electricity prices over the years, with 2021 reporting values almost twice as high as those in 2019. However, the high variance observed in 2021 may be attributed to noise, particularly reported in that year. Fig. 2 visualizes historical data; Fig. 2(a) shows electricity prices, and Fig. 2(b) represents total electricity consumption.

Table 3
Summary statistics of the Germany electricity prices (€/MWh) of 2019–2021.

Statistics	2019	2020	2021
Data points	8760	8760	6527
Minimum	−90.01	−83.94	−69.0
25% percentile	31.06	21.75	48.89
50% percentile	38.06	30.99	64.50
75% percentile	46.27	40.24	85.99
Mean	37.66	30.46	69.02
Maximum	121.46	200.40	237.01
Interquartile range	15.21	18.49	37.09
Standard deviation	15.51	17.50	34.35
Variance	240.81	306.29	1180.51
Mean absolute deviation	10.43	12.47	25.30
Median absolute deviation	11.26	13.71	26.52
Skewness	−1.42	−0.28	0.60

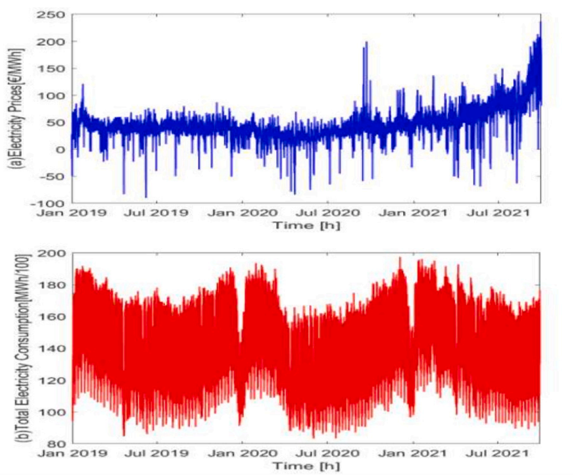

Fig. 2. Historical data of electricity prices and consumption in Germany, from 2019 to 29 September 2021: (a) Electricity prices, (b) Total electricity consumption divided by 100.

In this study, the input dataset is split into two sub-datasets as training data and test data. Overfitting is an important issue to consider when testing prediction models. To overcome this problem, the dataset was trained and repeatedly increased until it reached 80% of the total data points, as shown in Table 3. It should be noted that as the dataset grew larger than 80%, the validation of the training data did not improve. As a result, the remaining 20% was used for testing. The term prediction has been used for estimating outcomes from unseen data. The models in this work were trained on the training dataset, and an estimator predicts new samples for the test dataset. Forecasting is a sub-discipline of prediction that has been used for future prediction where the model cannot be fed with input/actual data [32]. For example, predictions for October 1, 2021, were counted as forecasts since input data of 2021 models ends on September 21, 2021, at 11 P.M., and the predictions on October 1, 2021, cannot be fed with input data and they are the future predictions.

4.1. Input data of the SARIMAX model

In Fig. 3, a flowchart that describes the programmed algorithm and input data for the SARIMAX model is illustrated. The Augmented Dickey–Fuller (ADF) test is a unit root test to check stationarity. ADF is used in this study because it includes a high-order regressive process, and the unit-roots can cause unpredictable results to given input data. The ADF is described as follows [33]:

$$\Delta Y_t = \alpha_{ADF1} + \beta_{ADF} * t_{ADF} + \gamma Y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta Y_{t-i} + \epsilon_t \tag{15}$$

where ΔY_t is the current period asset difference, ΔY_{t-i} is lagged values of dependent variables, and t_{ADF} stands for linear trend variable. ϵ_t is a Gaussian noise, γ is the coefficient of lagged Y_{t-1}, p is the number of lags included within the test, α_{ADF1} is the regression constant, β_{ADF} is the coefficient of time trend, and δ_i are regression coefficients. If the p-value is less than 0.05 and $\gamma = 1$, the null hypothesis can be rejected and then the series is said to be stationary.

The forecasting process is performed by predicting one step ahead of the last actual data ($h_{tlast+1}$) and appending actual input data to this predicted data (iteratively 63 times), as shown in Fig. 3. To have 64 forecast data, one step ahead was predicted starting from $h_{tlast+1}$, then taking this predicted value as actual input data and predicting $h_{tlast+2}$ until predicting $h_{tlast+64}$. More information about the process can be found in Table 4.

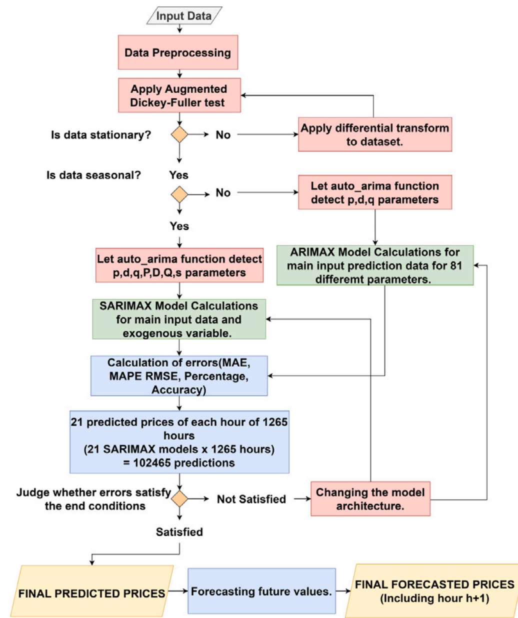

Fig. 3. Diagram of the input data SARIMAX model.

4.2. Input data of the LSTM and CNN-LSTM model

Fig. 4 shows a flowchart that describes the programmed algorithm and input data for the CNN-LSTM model. The LSTM model's working principle is the same as the CNN-LSTM model but without the CNN layer calculations. The forecasting for LSTM and CNN-LSTM models was performed using a similar procedure as developed for the SARIMAX model. More information about LSTM and CNN-LSTM models and the programmed algorithms can be found in Tables 5 and 6.

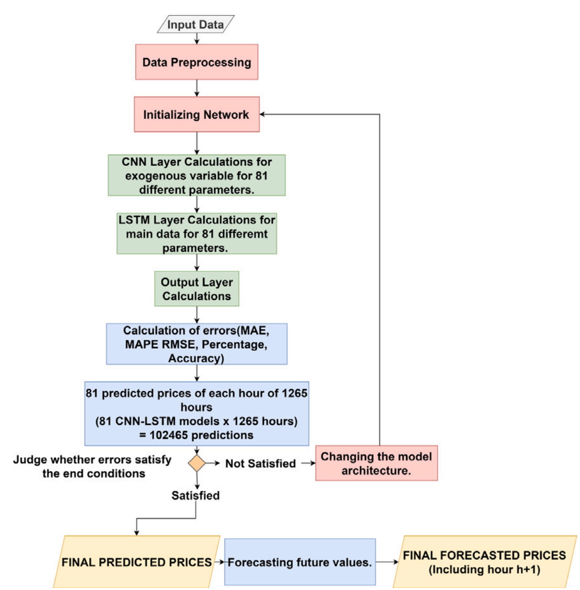

Fig. 4. Diagram of the input data CNN-LSTM model.

4.3. Setting model parameters

Table 4 depicts the main parameters of SARIMAX models programmed for the studied period 2019-Sep2021. The architecture of the SARIMAX model was built using the pmdarima library of Python auto.arima function. This auto.arima takes the minimum and maximum order values of the function and searches for an optimal architecture with the least Akaike Information Criterion (AIC). AIC is an estimator of the prediction error [34], which can be calculated as follows:

$$AIC = 2k - 2ln(\hat{L}) \tag{16}$$

where k represents the number of estimated parameters in the model, and $\hat{L}$ represents the maximum value of the likelihood function of the model. Seasonal parameters such as time lag or forecast window are depicted in Table 4.

Table 4
Architecture of SARIMAX Models.

Models	Seasonal Parameters $(P, D, Q)_s$	AR Parameter (p)	I Parameter (d)	MA Parameter (q)	Exogenous Variable (X)
SARIMAX 2019	$(1, 1, 2)_{24}$	7	0	2	Total hourly electrical consumption (2019)
SARIMAX 2020	$(1, 1, 2)_{24}$	3	0	1	Total hourly electrical consumption (2020)
SARIMAX Sep2021	$(1, 1, 2)_{24}$	7	1	1	Total hourly electrical consumption (2021)

In Table 5, the LSTM architecture parameters are depicted for the studied period 2019-Sep2021. These parameters are the hyperparameters that control the learning process of the model and are applied to each of the models. The learning rate was 0.0001, and it regulates how much the model changes when weights vary in response to the predicted error [35]. Batch size is the number of sub-samples provided before updating parameters, and it was picked at 32 to prevent long execution times. Epoch size controls the number of times the learning algorithm runs over the full training dataset [36]. An optimal epoch size was selected as 300, minimizing model errors with each iteration. To detect hyperparameters other than these three, two algorithms have been used. The first algorithm was programmed to choose the hidden layer 1's neuron numbers. A k-fold cross-validation was used for the models [37], as follows:

$$N_h = \frac{N_s}{(\alpha(N_i + N_o))} \tag{17}$$

where, N_h is hidden layer 1's neuron number, N_s is the number of samples in the training data, N_i is the number of input neurons, N_o is the number of output neurons, and α is a scaling factor from 2 to 10. The first algorithm incremented α by 0.1 from 2 to 10, which generated 81 models. The model with the most successful trends is, therefore, picked and used in the second algorithm. The hidden layer 2's neuron numbers are chosen five times greater than the hidden layer 1's number. On the other hand, the second algorithm uses the optimal neuron number and changes recurrent activation and activation functions with sigmoid, hyperbolic tangent, and rectified linear unit (ReLu). These three functions are the most commonly used in neural networks. This algorithm generates the results with the most successful trends. The model shows a time step limit of sl = 65. After this prediction window limit, the forecasting starts to lose accuracy. The accurate forecast is in the range of sl-1. The hidden layer2's activation function was a hyperbolic tangent; the output gate's activation function was a hyperbolic tangent with ReLu recurrent activation function for all the models. Kernel initializers, recurrent initializers, and early stopping were also used to overcome the overfitting problem in all the models. By running multiple tests, the 64-h forecast has the highest accuracy.

Table 5
Architecture of LSTM models.

Models	Hidden layer 1 neuron numbers	Hidden layer 2 neuron numbers	Time steps (sl)	Activation function	Recurrent activation function
LSTM 2019	16	80	65	Hyperbolic tangent	ReLu
LSTM 2020	14	70	65	Hyperbolic tangent	Sigmoid
LSTM Sep2021	24	120	65	ReLu	Sigmoid

Table 6 shows the main hyperparameters of CNN to be used with the LSTM model. To make fair comparisons, the LSTM-based and CNN-LSTM model parameters were kept equal. The Kernel size determines sliding windows, and filters determine the size. The CNN-LSTM algorithms were coded with the same logic that has been used in the LSTM model. In this case, the programmed algorithm increased the kernel size from 1 to 10, and the filters increased from 2^0 to 2^{11}.

Table 6
Architecture of CNN models.

Models	Input data	Convolution dimension	Kernel size	Filters	Activation function
CNN-LSTM 2019	Electrical Consumption of Germany(2019)	1D	5	1024	ReLu
CNN-LSTM 2020	Electrical Consumption of Germany(2020)	1D	10	256	Hyperbolic tangent
CNN-LSTM Sep2021	Electrical Consumption of Germany(2021)	1D	5	218	Hyperbolic tangent

4.4. Predicted errors

It is remarkable that for the forecasted prices, the actual values are unknown. To have a successful forecast, test data that consists of the last 20% values of input/actual data were predicted with the SARIMAX, LSTM, and CNN-LSTM, and the architecture was given in Tables 4, 5, and 6, respectively. The predicted errors are the difference between the actual and predicted prices. Successful trends illustrate the same trends. In this work, successful trends are maximized while predicted errors are reduced. Low errors and high successful trends make the models reliable for future forecasts. Fig. 5 compares the prediction errors of the SARIMAX, LSTM, and CNN-LSTM models for the corresponding years. Fig. 5(a) shows predicted errors for the year 2019. Fig. 5(b) illustrates the predicted errors of 2020, and Fig. 5(c) depicts the predicted errors of 2021.

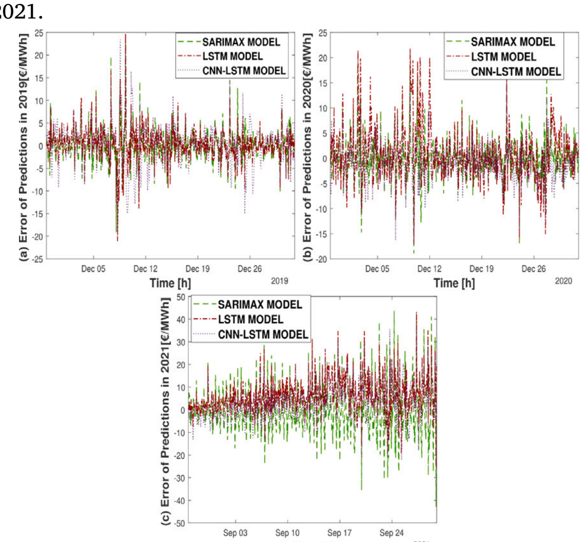

Fig. 5. SARIMAX, LSTM, and CNN-LSTM Predicted Errors: (a) 2019, (b) 2020, and (c) 2021-till Sep2021.

5. Results and discussion

Some sceneries have been used to exemplify the results: (1) Test data prediction prices are shown in Fig. 6. (2) Future data's forecast prices are illustrated in Fig. 7. (3) The yearly accuracy of predictions for SARIMAX, LSTM, and CNN-LSTM (2019, 2020, Sep2021) are depicted in Table 7. (4) The accuracy measures of each model for a 64-h forecast are depicted in Table 8. (5) The DM test results measure the 64-h forecast for SARIMAX, LSTM, and CNN-LSTM in Table 9.

5.1. Predicted prices

To validate model performances, the obtained values of the last 600 predictions are compared to actual data, as seen in Fig. 6. Fig. 6(a) are predictions for 2019. Fig. 6(b) depicts predictions for 2020, while Fig. 6(c) shows the year 2021.

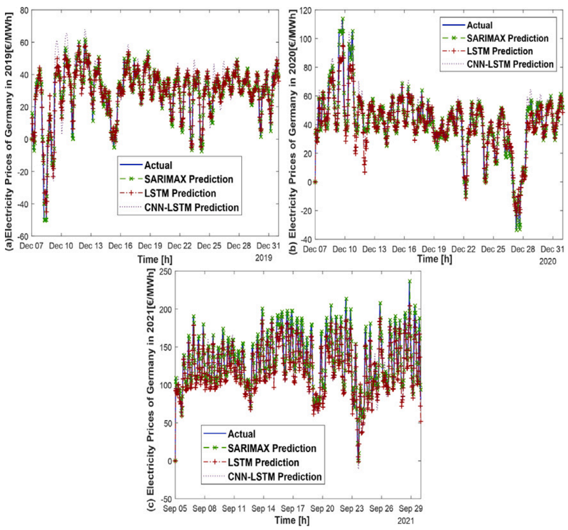

Fig. 6. Predicted prices of SARIMAX, LSTM, and CNN-LSTM models: (a) 2019, (b) 2020, and (c) until Sep2021.

5.2. Forecasted prices

Forecasts were performed using optimized models according to the aforementioned predictions. Each of the models was run on September 30, 2021, at 3:00 A.M. (GMT+2), and the forecast data until October 2, 2021, at 4:00 P.M. was obtained from the 2021 models. Fig. 7 shows data coming from October 2, 2021, at 5:00 P.M., waiting for actual data to be compared to forecast values. A similar procedure was followed for the previous models of 2019 and 2020. All models forecasted 64 steps ahead.

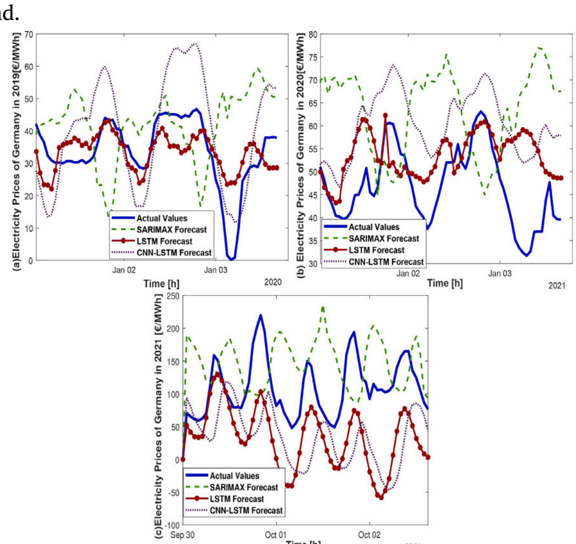

Fig. 7. SARIMAX, LSTM, and CNN-LSTM models forecast comparison with respect to actual prices: (a) year 2019, (b) 2020, and (c) 2021.

Fig. 7 shows the comparison of SARIMAX, LSTM, and CNN-LSTM forecast models with respect to actual prices. Fig. 7(a) illustrates the forecast of 2019 models used for forecasting January 1, 2020, to January 3, 2020. Fig. 7(b) shows the forecast of 2020 models used for forecasting January 1, 2021, to January 3, 2021. Fig. 7(c) Forecast of 2021 models includes forecasts from September 30, 2021, to October 2, 2021. The aforementioned periods consider 64 h of forecasting.

5.3. Predicted and forecasted accuracy measures

MAE, MAPE, RMSE, and percentage accuracy are used to compare models (18)–(22). Error measurements are also tested for forecasted results using the DM (Diebold–Mariano) test. The DM test is used to determine whether a forecasting result is more accurate than other results [38] (23)–(24). When actual and predicted data increase or decrease at the same time, this is considered a successful trend. Successful trend, percentage of accuracy, MAE, MAPE, RMSE, loss differential, and DM test statistic values are calculated in this study (18)–(24). Here, y_i is the predicted or forecasted value, x_i is the actual input data, n is the total predicted or forecasted data number (i.e., n is 1762 in test data calculations of 2019 and 2020), d_t is loss differentials, S_1 is DM test statistic value, $g(e_{it})$ and $g(e_{jt})$ are loss functions, $f_d(0)$ is the spectral density of the loss differential at the frequency at 0, while $\hat{f}_d(0)$ is a consistent estimate of $f_d(0)$, the spectral density of the loss differential at frequency 0 and $\tilde{d}$ is the sample mean loss differential:

$$MAE = \sum_{i=1}^{n} |y_i - x_i| \tag{18}$$

$$MAPE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{\sum_{i=1}^{n} y_i} * 100 \tag{19}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}} \tag{20}$$

$$Succesful\ Trend\ occurs\ when \frac{(Actu\dot{a}l\ Data)_t}{(Predic\dot{t}ed\ Data)_t} > 0 \tag{21}$$

$$Percentage\ Accuracy\ (\%) = \frac{Number\ of\ Succesful\ Trends}{All\ Predictions} * 100 \tag{22}$$

$$d_t = [g(e_{it}) - g(e_{jt})] \tag{23}$$

$$S_1 = \frac{\tilde{d}}{\sqrt{\frac{2*\pi*\hat{f}_d(0)}{T}}} \tag{24}$$

The sign of the first-order differencing of series indicates whether the data is rising or decreasing. The prediction's trend is accurate if the division of the difference between the actual and predicted data is greater than zero. When both actual and predicted data decrease and rise at the same time, the prediction trend is successful. The null hypothesis of the Diebold–Mariano test states that there is no difference in accuracy between two competing forecasts if the population mean of the differential loss is zero. The null hypothesis can be rejected when the probability value (p-value) is less than 0.05, implying that one model is more accurate than the other. The DM test, on the other hand, shows that there is no significant difference in accuracy between the competing forecasts.

Tables 7 and 8 show the results for each of the models applied in this work: SARIMAX, LSTM, and CNN-LSTM. Table 7 includes the accuracy measure of predictions. Table 8 depicts the accuracy measures of the 64-h forecast. Table 9 presents the Diebold–Mariano test results for all the tested models and the 64-h forecast.

Table 7
Test input data (last 1762 data of years 2019 and 2020, and last 1305 data of the year 2021) accuracy measure of predictions for SARIMAX, LSTM, and CNN-LSTM.

Years	Models	MAE (€/MWh)	RMSE (€/MWh)	MAPE (%)	Percentage accuracy(%)
2019	SARIMAX2019	2.49	5.05	5.68	86.77
	LSTM2019	2.62	5.22	5.98	81.56
	CNN-LSTM2019	2.56	6.17	5.84	86.97
2020	SARIMAX2020	3.08	5.39	4.86	75.15
	LSTM2020	4.24	7.40	6.77	76.36
	CNN-LSTM2020	3.02	5.03	4.82	72.21
2021	SARIMAX2021	11.77	16.04	5.96	91.98
	LSTM2021	12.62	15.66	6.39	96.59
	CNN-LSTMSep2021	10.03	12.96	5.08	98.59

In Table 7, MAE and RMSE values are around five for all models in 2019 and 2020. However, in 2021, these numbers enlarged to more than ten. The MAPE value is relatively low in these three years, hovering around 5%. The magnitude of prices increased in 2021, and in terms of accuracy, all models fared similarly. In 2019, the percentage accuracy ranged between 81% and 87%. While in 2020, it is between 72% and 76%. Nevertheless, in 2021, models performed better, and their percentage accuracy increased to 98%. It is because of the post-pandemic effect that generates noisy data. Due to the periodicity of the data, models were able to detect trends precisely. But, due to the high magnitude changes, algorithms were unable to recognize the correct magnitude. This explains the projected increase in MAEs and RMSEs in 2021. Since predictions were done for historical dates and algorithms were fed with actual values after predicting one step ahead, models corrected mistakes by comparing them with subsequent steps. CNN-LSTM was the most successful model in 2019, followed by LSTM in 2020 and CNN-LSTM in 2021, with percentage accuracy of 86.97%, 76.36%, and 98.59%, respectively. LSTM and CNN-LSTM share the same LSTM structure for electricity price analysis, and CNN is added for the evaluation of total hourly electrical consumption. However, the addition of the consumption input in the CNN-LSTM model had a negative influence in 2020.

Table 8
Accuracy measures of 64-h forecast for SARIMAX, LSTM, and CNN-LSTM.

Years	Models	MAE (€/MWh)	RMSE (€/MWh)	MAPE (%)	Percentage accuracy (%)
2019	SARIMAX2019	15.92	3.99	11.89	50.2
	LSTM2019	5.85	2.61	4.37	71.15
	CNN-LSTM2019	11.29	3.5	7.69	92.30
2020	SARIMAX2020	20.17	4.49	10.79	51.9
	LSTM2020	7.32	2.88	3.91	80.76
	CNN-LSTM2020	11.46	3.66	6.13	88.46
2021	SARIMAX2021	67.83	8.23	15.54	42.85
	LSTM2021	52.04	7.71	11.92	92.85
	CNN-LSTM2021	52.31	7.79	11.98	67.30

In this case, the forecast performance was evaluated using percentage accuracy, and the values show that the magnitude of the forecasted prices is close to the actual prices. The percentage accuracy indicated the increment and reduction of forecasted prices at the same time as actual prices. In Table 8, MAEs and RMSEs were raised in 2021, while MAPE remained stable, as in Table 7. It is due to an increase in the size of electricity prices in 2021. Forecasting models are expected to operate with lower accuracy when compared to predicted price models. Because forecasting models do not have feedback and are unable to rectify errors since actual prices are not accessible for the forecasted dates. Among the models, the LSTM and CNN-LSTM models were more accurate than the SARIMAX models.

In general, the SARIMAX model's percentage accuracy is lower. It is because of the basic structure of SARIMAX. The SARIMAX models' time steps were set to 7, so the models sought a pattern for the next step largely in the previous 7 data points. The LSTM and CNN-LSTM models performed better for forecasted prices since the time step for models was set to 65. The CNN-LSTM model was the most effective for the years 2019 and 2020, with a higher accuracy percentage of 92.30% and 88.46%, respectively. The exogenous variable remained consistent in 2019 and 2020, as shown in Fig. 2(b). However, the LSTM was the most successful model in 2021, with 92.85% percentage accuracy. In 2021, the CNN-LSTM model maintained 67.30% accuracy. Nevertheless, the accuracy of this model was reduced owing to the CNN model's exogenous variable effect. It should be emphasized that the pandemic electricity price was rising and falling abruptly in 2021, as shown in Fig. 2(a). The use of the exogenous variable, on the other hand, had a favorable influence in 2019 and 2020 for forecasting the next 64 h. The CNN models with the addition of electricity consumption as an exogenous variable were added to the model for better forecasting results when compared to the simple LSTM model; see years 2019 and 2020. However, the pandemic effect resulted in a change in the characteristics of the data in 2021. As a conclusion, electricity consumption became a poor exogenous variable for the year 2021, in which the simple LSTM model fared better.

Table 9
Diebold–Mariano test results of the 64-h forecast for SARIMAX, LSTM, and CNN-LSTM.

Years	Error loss functions	SARIMAX vs. LSTM	SARIMAX vs. CNN-LSTM	LSTM vs. CNN-LSTM
		DM test		
2019	MAE	6.9599	2.1378	6.1763
	RMSE	5.4456	2.8168	5.2168
	MAPE	1.3176	1.269	1.3961
2020	MAE	10.4057	4.83	7.7495
	RMSE	8.9698	6.2112	5.0367
	MAPE	9.8777	5.2687	6.7744
2021	MAE	1.5678	0.9857	2.1256
	RMSE	1.122	1.7672	−1.5077
	MAPE	1.5673	1.1284	−1.4532
		p-values		
2019	MAE	2.3456e−9	0.0364	5.3099e−8
	RMSE	9.0697e−7	0.0064	2.1549e−6
	MAPE	0.1923	0.2082	0.2582
2020	MAE	2.5971e−14	8.8902e−6	9.7845e−11
	RMSE	7.3031e−13	4.6223e−8	4.2356e−6
	MAPE	2.0198e−14	1.7967e−8	4.8720e−6
2021	MAE	0.4353	0.1897	0.3234
	RMSE	0.0269	0.0320	0.0136
	MAPE	0.04765	0.0023	0.0145

Table 9 compares error measurements and models using the DM test between two different forecasting models. When the null hypothesis can be rejected, the second forecasting model does better if the DM test result is positive. If the result is negative, the first forecasting model performed better. For the following error measurements and years, the null hypothesis can be rejected: MAE and RMSE in 2019, MAE, RMSE, and MAPE in 2020, and RMSE and MAPE in 2021. DM test results show that LSTM and CNN-LSTM are more accurate than the SARIMAX model in 2019 and 2020, but CNN-LSTM is more accurate than the LSTM model. The DM test, on the other hand, demonstrates that LSTM was the most accurate in 2021.

6. Conclusion

This study developed and compared three different models, SARIMAX, LSTM, and CNN-LSTM, using programming languages like Python and TensorFlow, to predict and forecast electricity prices in the German electricity market for the day ahead. The models' performances were evaluated using various metrics such as MAE, RMSE, MAPE, percentage accuracy, and the Diebold–Mariano test. The study utilized historical data from 2019 to 2021 to assess the efficiency of the models in anticipating electricity prices. The models were tested using multiple case studies and demonstrated remarkable results. However, prediction tasks were found to be better than forecasting tasks due to the models being fed with actual values after predicting one step ahead, enabling them to correct mistakes. LSTM and CNN-LSTM models performed better than SARIMAX, as they allowed for larger time steps, and CNN-LSTM was the most effective model for 2019 and 2020. In contrast, LSTM was the most successful model in 2021. The exogenous variable of total hourly electricity consumption used as an input affected the CNN-LSTM model's accuracy in 2021, but it had a positive influence on forecasting the next 64 h in 2019 and 2020. Lastly, the data preprocessing conditions for LSTM and CNN-LSTM were found to be simpler than those for SARIMAX, allowing for the operation of nonlinear input data.

CRediT authorship contribution statement

Çağatay Berke Bozlak: Data curation, Investigation, Resources, Software, Validation, Visualization, Writing – original draft, Writing –

review & editing. **Claudia Fernanda Yaşar:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Supervision, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Electricity market report - december 2020 – analysis, 2021, https://www.iea.org/reports/electricity-market-report-december-2020, [Online; accessed 11-September-2021].
[2] P.D. Necoechea-Porras, A. López, J.C. Salazar-Elena, Deregulation in the energy sector and its economic effects on the Power Sector: A Literature Review, Sustainability 13 (6) (2021) 3429, http://dx.doi.org/10.3390/su13063429.
[3] V. Guerrero-Mestre, A.A. Sanchez de la Nieta, J. Contreras, J.P. Catalao, Optimal bidding of a group of wind farms in day-ahead markets through an external agent, IEEE Trans. Power Syst. 31 (4) (2016) 2688–2700, http://dx.doi.org/10.1109/tpwrs.2015.2477466.
[4] J. Contreras, Forecasting models of electricity prices, Energies 10 (2) (2017) 160, http://dx.doi.org/10.3390/en10020160.
[5] C.-Y. Lee, C.-E. Wu, Short-term electricity price forecasting based on similar day-based Neural Network, Energies 13 (17) (2020) 4408, http://dx.doi.org/10.3390/en13174408.
[6] Strommarktdaten, stromhandel und stromerzeugung in deutschland, 2021, https://www.smard.de/, [Online; accessed 12- September-2021].
[7] C. Bennett, R. Stewart, J. Lu, Autoregressive with exogenous variables and neural network short-term load forecast models for residential low voltage distribution networks, Energies 7 (5) (2014) 2938–2960, http://dx.doi.org/10.3390/en7052938.
[8] G. Van Rossum, F.L. Drake, Python 3 Reference Manual, CreateSpace, Scotts Valley, CA, 2009.
[9] M. Abadi, et al., TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org.
[10] F. Chollet, et al., Keras, 2015, https://keras.io.
[11] T.G. Smith, et al., Pmdarima: ARIMA estimators for Python, 2017, http://www.alkaline-ml.com/pmdarima.
[12] C. McHugh, S. Coleman, D. Kerr, D. McGlynn, Forecasting day-ahead electricity prices with a SARIMAX model, in: 2019 IEEE Symposium Series on Computational Intelligence (SSCI), 2019.
[13] H. Hou, C. Liu, Q. Wang, X. Wu, J. Tang, Y. Shi, C. Xie, Review of load forecasting based on artificial intelligence methodologies, models, and challenges. http://dx.doi.org/10.1016/j.epsr.2022.108067.
[14] M.H.D.M. Ribeiro, R.G. da Silva, S.R. Moreno, V.C. Mariani, L. dos Santos Coelho, Efficient bootstrap stacking ensemble learning model applied to wind power generation forecasting. http://dx.doi.org/10.1016/j.ijepes.2021.107712.
[15] C. Monteiro, I. Ramirez-Rosado, L. Fernandez-Jimenez, P. Conde, Short-term price forecasting models based on artificial neural networks for intraday sessions in the Iberian Electricity Market, Energies 9 (9) (2016) 721, http://dx.doi.org/10.3390/en9090721.
[16] J. Catalão, S. Mariano, V. Mendes, L. Ferreira, Short-term electricity prices forecasting in a competitive market: A neural network approach, Electr. Power Syst. Res. 77 (10) (2007) 1297–1304, http://dx.doi.org/10.1016/j.epsr.2006.09.022.
[17] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: LSTM cells and network architectures. http://dx.doi.org/10.1162/neco_a_01199.
[18] D. Kumar, H.D. Mathur, S. Bhanot, R.C. Bansal, Forecasting of solar and wind power using LSTM RNN for load frequency control in isolated microgrid, Int. J. Model. Simul. 41 (4) (2020) 311–323, http://dx.doi.org/10.1080/02286203.2020.1767840.
[19] A. Ghaderi, B.M. Sanandaji, F. Ghaderi, Deep Forecast: Deep Learning-based Spatio-Temporal Forecasting, Cornell University, Ithaca, NY, USA, 2017.
[20] W. Lu, J. Li, Y. Li, A. Sun, J. Wang, A CNN-LSTM-based model to forecast stock prices, Complexity 2020 (2020) 1–10, http://dx.doi.org/10.1155/2020/6622927.
[21] A. Lotfipoor, S. Patidar, D. Jenkins, Short-term forecasting of residential electricity demand using CNN-LSTM, 2020.
[22] A. Jędrzejewski, J. Lago, G. Marcjasz, R. Weron, Electricity price forecasting: The dawn of machine learning, IEEE Power Energy Mag. 20 (3) (2022) 24–31, http://dx.doi.org/10.1109/MPE.2022.3150809.
[23] G. Marcjasz, M. Narajewski, R. Weron, F. Ziel, Distributional neural networks for electricity price forecasting, 2023, http://dx.doi.org/10.1016/j.eneco.2023.106843.
[24] F. Zhang, H. Fleyeh, C. Bales, A hybrid model based on bidirectional long short-term memory neural network and Catboost for short-term electricity spot price forecasting, 2022, URL https://ideas.repec.org/a/taf/tjorxx/v73y2022i2p301-325.html.
[25] M. Heidarpanah, F. Hooshyaripor, M. Fazeli, Daily electricity price forecasting using artificial intelligence models in the Iranian electricity market, 2023, https://doi.org/10.1016/j.energy.2022.126011.
[26] A. Poggi, L.D. Persio, M. Ehrhardt, Electricity price forecasting via statistical and deep learning approaches: The german case. http://dx.doi.org/10.3390/appliedmath3020018.
[27] H. Su, X. Peng, H. Liu, H. Quan, K. Wu, Z. Chen, Multi-step-ahead electricity price forecasting based on temporal graph convolutional network, 2022, https://doi.org/10.3390/math10142366.
[28] From AR to SARIMAX: Mathematical definitions of time series models. https://phosgene89.github.io/sarima.html. journal= phosgene89.github.io.
[29] S. Alam, Recurrent neural networks in electricity load forecasting, 2018.
[30] M.J.W. Lysfjord, Modeling and forecasting the nord pool day-ahead power market through deep-learning, 2017.
[31] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, 1D convolutional neural networks and applications: A survey, Mech. Syst. Signal Process. 151 (2021) 107398, http://dx.doi.org/10.1016/j.ymssp.2020.107398.
[32] Data science blog: Understand. Implement. Succed, 2021, https://www.datascienceblog.net/post/machine-learning/forecasting_vs_prediction/, [Online; accessed 15- September-2021].
[33] Augmented dickey-fuller (ADF) test, 2021, https://rtmath.net/assets/docs/finmath/html/93a7b7b9-e3c3-4f19-8a57-49c3938d607d.htm, [Online; accessed 17- September-2021].
[34] Wikimedia foundation. Akaike information criterion. Wikipedia, 2021, https://en.wikipedia.org/wiki/Akaike_information_criterion, [Online; accessed 17-September-2021].
[35] J. Brownlee, Understand the impact of learning rate on neural network performance, Mach. Learn. Mastery (2020) https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/, [Online; accessed 18- September-2021].
[36] J. Brownlee, Difference between a batch and an epoch in a neural network, 2019, https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/, [Online; accessed 18- September-2021].
[37] K. Eckhardt, Choosing the right hyperparameters for a simple LSTM using keras, Medium (2018).
[38] F.A. Mohammed, M.A. Mousa, Applying diebold–mariano test for performance evaluation between individual and hybrid time-series models for modeling bivariate time-series data and forecasting the unemployment rate in the USA, 2017.