

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Optimal Energy Production Planning in Power Systems with Wind Energy Sources

Cristian-Dragos Dumitru and Adrian Gligor

Abstract

Due to its intermittent nature, the optimal production of electricity from wind energy represents a real challenge for nowadays power systems. Whether isolated or grid-connected systems are considered, wind power sources can be profitable, but their intermittent output may lead to problems in terms of power quality and increased costs related to the operation of the grid and to the production of energy. This chapter discusses the choice of the most appropriate solutions for planning the electricity production from wind energy based on different algorithms for obtaining models based on principles used in artificial intelligence techniques such as recurrent neural networks (RNNs) and long short-term memory (LSTM) neural networks. We discuss the situation of obtaining the optimal model for estimating energy production based on a criterion or on multiple criteria: energy production history or energy production history correlated with different parameters describing the weather conditions.

Keywords: energy planning, wind energy, optimal wind energy integration, forecasting, artificial intelligence, RNN, LSTM

1. Introduction

Due to the last years' awareness of climate changes and traditional energy resource depletion, renewable energies started to play a key role in the nowadays electricity market. However, transition to these types of primary resources leads to many challenges to be solved. From the power grid optimal operating point of view, one of the most important issues to be solved is the intermittent and, in some cases, unpredictable availability of the primary energy resources.

According to Ref. [1], in 2017, wind represented the renewable resource with the highest impact on European Union electricity production. The mentioned source reports wind energy with the most important contribution to EU-28 gross electricity consumption with a 30.7% ratio. As reported in Ref. [2], worldwide not including Europe, installed wind power capacities have followed also an ascending trend, being recorded at the beginning of 2017 about 378 GW operational installed power generation capacities, which is rated to about 78% from the total installed capacity. Meanwhile, the wind power is concentrated in few regions. Over 60% is allocated to tree states: China, the USA, and Germany, while most of the countries shares 16%.

The mentioned wind energy development was mainly possible not only due to fiscal facilities developed by many countries as a measure to reduce pollution but also due to some advantages of the wind power such as the lowest priced from the nowadays renewable energy technologies and reduced impact on the land on the installed sites.

The economic advantage on the electricity sale side could be in some situations overcome by the higher initial installation costs compared to the conventional power generation [2]. However, this aspect in future may suffer changes, as the market indicates a decrease in terms of required initial investments. Other features of the wind energy constitute major challenges in its adoption at a higher penetration level. The first one that has to be mentioned is its intermittent availability that is not correlated with the electricity demand and aspect that cannot be overcome simply by installing batteries due to the economic and technical unfeasibility.

The list of attributes can continue with other relevant ones such as the impossibility of this type of energy to be stored, the wind can only be harvested at specific parameters for electricity production, limited proper geographical locations with good wind potential, and in many situations located far away from the consumption centers.

By correlating the wind energy share from the total installed power capacities with the previously mentioned wind characteristics, one can found that power grid can significantly be affected in the case of wind speed changes in terms of network balancing and managing the power flows [3], or in a general speaking term, the stability of the power system can be affected [4]. To overcome this issue, complementary source of energy must be available such as energy storage devices (batteries), pumped-storage hydroelectricity stations, or any traditional electricity generation facilities. In order to operate correctly, with any of the mentioned solutions, due to their technical limitations such as response in time and/or power reserve, forecasting of the wind energy sources availability is a key factor on relying in a sustainable way on this type of energy.

The scope of the next sections is to study different time series modeling methods, recorded in literature, for forecasting production of electrical energy from wind. Two different models are considered: recurrent neural networks (RNNs) and long short-term memory (LSTM) neural networks. The studied time series is the production of electrical energy from wind energy of a national energy system during March 2018.

2. Wind model analysis

In order to understand the behavior of the wind power generation units, a theoretical approach on the mathematical wind model description has to be briefly developed.

One of the most relevant factors in wind modeling description relates to the wind speed.

In order to evaluate the power and electricity generated by a wind power plant, the wind speed analysis is performed by using elements of statistical processing of the measurement data, taking into account its random character. In order to be able to determine the power generated by a wind group, we need an estimation of the wind speed at the height h of the turbine rotor. These data can be obtained by placing an anemometer at some reference height h_{ref} (e.g., 10, 30, or 50 m above the zero wind level). There are two well-known formulas for calculating the wind speed at certain height h [5, 6]:

a. The logarithmic expression:

$$v = v_{ref} \frac{\ln \frac{h}{z_0}}{\ln \frac{h_{ref}}{z_0}} [m/s] \quad (1)$$

b. The power law expression:

$$v = v_{ref} \left(\frac{h}{h_{ref}} \right)^\alpha [m/s] \quad (2)$$

where v is the wind speed at height h above the zero wind level; v_{ref} is the wind speed at h_{ref} (e.g., height of anemometer); and z_0 is the roughness length depending on the surface roughness of a given site (m).

Comparing the above two formulas, the power exponent “ α ” can be calculated from:

$$\alpha = \frac{1}{2.725 - \ln z_0} \quad (3)$$

The average value of v_m speed, relative to a time period T , is calculated with Eq. (4):

$$v_m = \frac{1}{T} \int_0^T v_i(t) dt [m/s] \quad (4)$$

where $v_i(t)$ represents the instantaneous wind speed.

This average wind speed can be hourly, daily, weekly, monthly, quarterly, semi-annual, or annual depending on the reference time interval T , but as a basis of analysis, the hourly average from which the other average values can be determined is used. Despite the observed hourly, daily, and annual average values, the measured wind speeds can vary significantly at different times, at different places, and at different heights relative to the ground.

Therefore, it is very difficult to compare the measured units of time. Thus, an average wind speed is calculated over time intervals Δt whose duration depends on the type of device used and using an N number of average values calculated over that interval. For calculating an hourly wind speed average, for example, if the average speed value is available at 5- or 10-min intervals, 12 values of the first type and 6 of the second will be used.

In practice, however, the average value of wind speed is calculated with Eq. (5), with a better approximation, for shorter the time intervals Δt :

$$v_m \cong \frac{\Delta t}{T} \sum_{j=1}^{\frac{T}{\Delta t}} v_j = \frac{1}{N} \sum_{j=1}^N v_j [m/s] \quad (5)$$

The net electric power at the output of the generator, which takes into account both the efficiency of the electrical part and the efficiency of the mechanical part of a wind group, is given by Eq. (6), according to Ref. [7]:

$$P_{el} = \frac{1}{2} C_e \rho A v_m^3 [W] \quad (6)$$

where A is the swept area by the rotating wind turbine blades; C_e is the total net efficiency factor, which is determined at the terminals of the electric power transformer for the wind power group; ρ is the average air density at hub height; and v_{m3} is the average cube of the instantaneous wind speed.

On the other hand, due to the fact that the power of a wind power group is proportional to the wind speed at the third power, we can consider that the average cubic speed, defined as the root of the third order of the average cube of the instantaneous wind speed, according to Eq. (7), is a measure of the available power and energy for a wind power group:

$$v_{m3} = \left(\frac{1}{T} \int_0^T v_i^3(t) dt \right)^{\frac{1}{3}} \quad [m/s] \quad (7)$$

The irregularity of the wind determines the difference between v_{m3} and v_m , so that the instantaneous wind speeds are all more different from their average value, and the more v_{m3} is greater than v_m . This pattern of the wind is characterized by the irregularity factor that is defined by Eq. (8).

$$k_f = v_m / v_{m3} \quad (8)$$

Often the irregularity factor can be replaced by the mean square deviation denoted as σ and expressed by Eq. (9):

$$\sigma \cong \left(\frac{1}{N} \sum_{j=1}^N \left(\frac{v_j}{v_m} - 1 \right)^2 \right)^{\frac{1}{2}} \quad (9)$$

The measurement units of the wind speed, with modifications of the wind division, are small intervals of 0.5 or 1 m/s; on this basis, they can be easily compared. For this purpose, the measured values of wind speed are classified into different speed classes. For each class, the probability of occurrence of measured values for wind speed allocated to this class is calculated with by taking into account the total number of measured values of wind speed. This frequency distribution always shows a typical course.

Mathematical approximations of such probability distributions can be performed with different functions that can be described by a small number of parameters. For the distribution of wind speeds, for example, either the Weibull distribution or the Rayleigh distribution can be used [7].

3. Wind energy production planning

Considering the assumptions made in the first section of this chapter about wind characteristics, we may note that having in advance information about the wind can be useful in the decision processes related to optimal power system operation. The process of obtaining mentioned information related in our case to the wind energy will be denoted as wind energy production forecasting.

Generally speaking, the forecasting term can be understood as the process of determining a sample P_{k+1} or a set of samples $\{P_{k+1+m} \mid m \in \mathbf{N}\}$ for a specific time t , given the set $\{C_{k-n} \mid n \in \mathbf{N}, n \leq k\}$, where C_{k-n} may consist of P_{k-n} measurements or of a more complex data.

Model class	Type of model	Remarks
Physical	Physical	Models based on meteorological parameters (temperature, atmospheric pressure, geographical and local conditions, environmental conditions, etc.)
Statistical	Auto-regressive (AR)	Approaches that rely on linear statistical models
	Autoregressive moving average (ARMA)	
	Autoregressive integrated moving average (ARIMA)	
Artificial Intelligence	Artificial neural network (ANN)	
	Fuzzy logic (FL)	
	Support vector machines (SVM)	

Table 1.
Classification of forecasting models.

Starting from this assumption, it can be seen that future data can be obtained starting from historical data.

Wind power forecasting respects the above definition, and the literature denotes it as direct forecasting approach. The need of more complex prediction methods to eliminate undesirable uncertainties requires a two-step approach to be adopted resulting in indirect methods for wind power forecast. First step consists in wind speed prediction followed by the usage of the turbine power curve for wind power determination [4]. For the former class of methods, the precision of the forecast is significantly influenced by the process of wind speed prediction [8], or if it is also considered the high degree of variability of wind-to-power curve, both terms play a key role in limited predictability of wind power generation [9].

Generally speaking, the prediction can offer short- or long-term predicted data that correspond to a short window period that contains one or more predicted points or a long-term interval, respectively. In case of wind power forecasting, due to the complexity of the problem, correlated with accuracy of the obtained results, short-term wind power forecast offers an appropriate solution for the task of optimal power system operation that covers but is not limited to power quality, power balance, or economic planning problems. For this purpose, a large number of forecasting models have been developed, which can be classified into three main classes: physical models, statistical models, and artificial intelligence technique-based models [6]. **Table 1** summarizes the classes of previous models.

In the next section, we will focus on exploring forecasting methods from the last class model. The approach was chosen starting from the ability of this type of algorithms that could offer proper results based on learnt patterns that are more appropriate than methods based on linear models.

4. Forecasting based on artificial intelligence

The obtaining of closest possible estimated values from the real ones is the main target. For these purposes, two types of neural networks are investigated, namely, recurrent artificial neural network (RANN) and long short-term memory (LSTM) networks. Their performances are evaluated through the mean absolute error

(MAE), mean absolute percentage error (MAPE), signed mean squared error (SMSE), and normalized mean squared error (NMSE) indexes.

4.1 Recurrent ANN

Feedforward network outputs are calculated based on the network input that is propagated from the input layer to one or more hidden layers and to the output layer through direct connections between the layers. Due to these connections, FFANNs are static networks. A neural network can have inverse connections, from an upper layer to a lower layer (e.g., from the output layer to the input layer), so the output of the network depends on inputs, outputs, and current, previous, and current state of the network, which gives dynamic behavior, and such a network is called a dynamic network [10]. Reverse connections are also called recurrent connections, hence the name of recurrent ANN (RANN).

Delays are introduced via reverse connections, so the response of the networks is influenced by the order in which the input vectors are presented. By this delay, information about the input data is stored, and the network may have different answers when, at the input, the same input vector is applied. This behavior makes it possible to approximate dynamic systems and presents an advantage in the field of forecasting [10].

The most used and known recurrent ANN topologies are as follows [11]:

- Jordan ANN (Jordan network or output-feedback recurrent ANN) is a feedforward network with a single-hidden layer and a context neuron for each neuron in the output layer (Figure 1). The purpose of the context neuron is to maintain the activation (output) of a neuron in the output layer at time k until it is used at time $k + 1$. The connections between the output neurons and the context neurons are weighted, as are the direct connections [11]. In Refs. [10, 12], delay blocks are used, with the same purpose as context neurons, when moments $k, k + 1, k + 2, \dots$ are moments of time.
- ANN Elman (Elman network or globally recurrent ANN) uses the same context neurons or delay blocks; the difference with Jordan networks is that each hidden layer will have a layer made up of context neurons that are

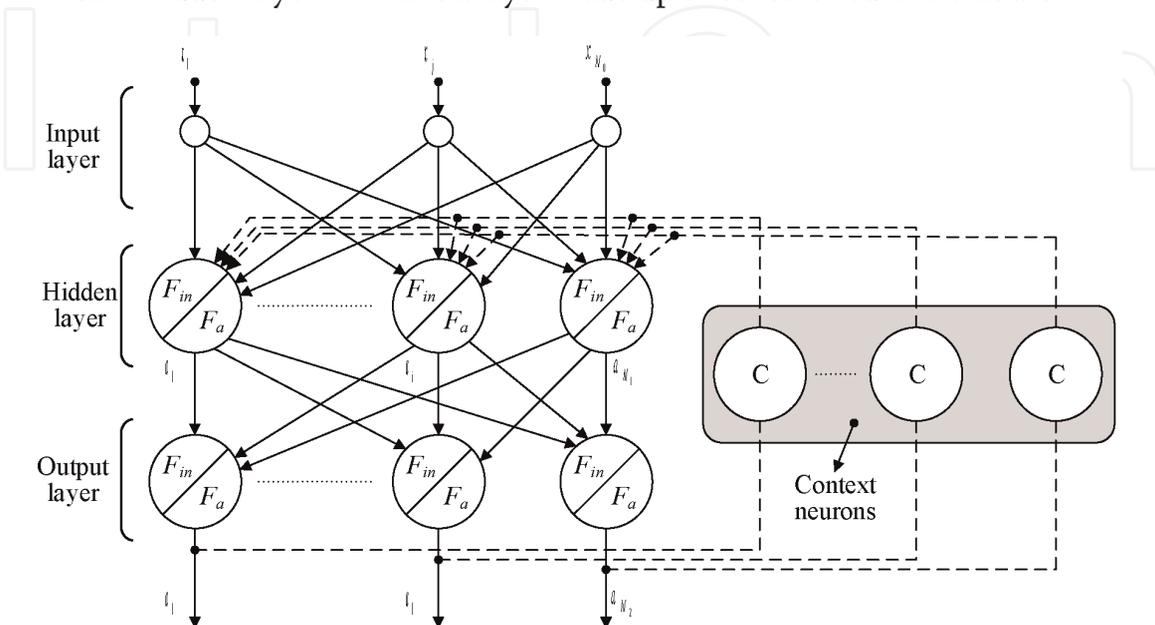


Figure 1. Structure of a Jordan type RANN [11].

connected, further on, to inputs of neurons from the hidden layer (Figure 2) [10, 12].

- Completely recurrent ANN, each neuron in a hidden layer or output layer has one or more context neurons, so more information is retained [12].

The networks shown in the previously mentioned figures are global recurrent networks, where each context neuron connects to the input of each neuron in the hidden layer. If each context neuron links only to the input of the neuron to which it is assigned, the network is locally recurrent [12].

Recurrent ANNs can be trained using the generalized delta learning rule [10]. The weights of the connections between the layers and the displacement weights have a direct effect and an indirect effect on the activation of neurons. The direct effect is created by the weights of the connections between the layers, which can be calculated with the generalized delta rule.

The indirect effect is created by the weights of the connections between the context neurons and the neurons in the hidden layer to which they are connected. The inputs of a layer, which come from the outputs of the context neurons, depend on the same weights on which the outputs of the neurons connected to the context neurons depend. For this reason, the calculation of the gradient

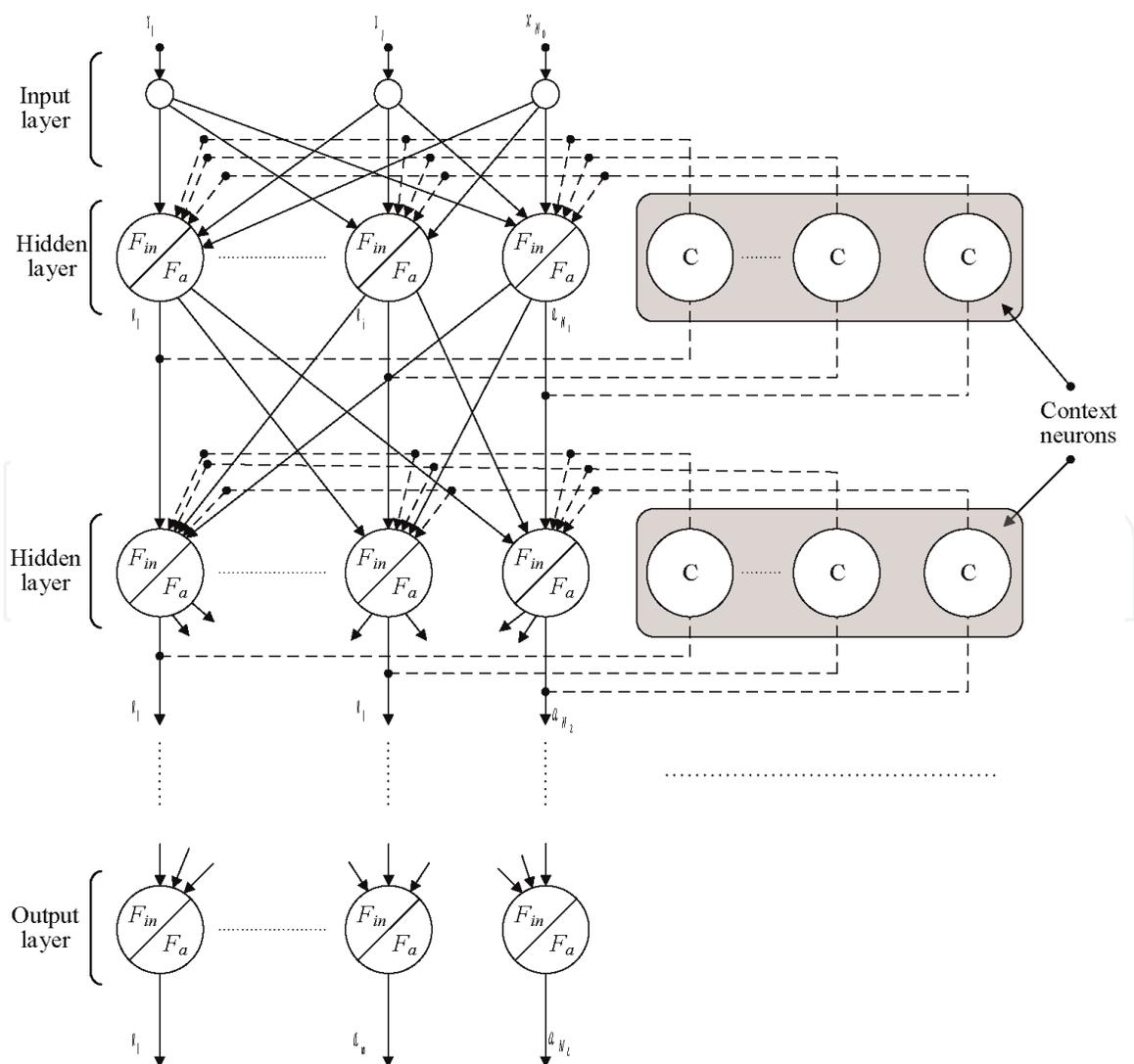


Figure 2.
 Structure of the recurrent Elman type network [11].

depends not only on the weights of the network but also on the previous outputs of the network [12].

There are two different processes for calculating the gradient: the time-propagated generalized delta rule (back propagation through time—BPTT) and real-time recurrent learning (RTRL) [10]. In the first method, the gradient is calculated from the last time moment to the first time moment. For this reason, it is necessary to calculate the network response for each time point before calculating the gradient. In the second method, the gradient is calculated at each time point, together with the network response at that time, continuing with the rest of the time points [10]. The difference between the two methods is that the BPTT algorithm performs offline training and requires less computing power than the RTRL training algorithm that performs online training but requires greater computing power [10, 12].

The two methods are detailed in Ref. [10], and briefly, the steps of the two methods are as follows:

- The neural network is initialized, as in the case of the binary preceptor training algorithm and the generalized delta rule. In addition, the RTRL method requires the initialization of the previous values corresponding to the network delays.
- The network response is calculated. For the RTRL method, the response for the first time point is calculated, and for the BPTT method, the network response is calculated for each time point.
- Calculate the total derivatives that take into account the indirect and direct effects and the explicit derivatives that only take into account the direct effects. In the case of the RTRL method, these calculations are repeated for each time point, and in the case of the BPTT method, it is calculated at the last time point, starting from the last time point and continuing until the first time point.
- Calculate the derivatives of the error function. Using the results, the weights are updated, and the algorithm of training the binary preceptor and the generalized delta rule is continued.

Recurrent networks, through inverse connections and dynamic behavior, have a more complex error surface than static feedforward networks. This complexity is due to the nonlinear behavior of the error function, and it has several local minima. Also, a small change in the weights can lead to significant changes in the error in increasing direction [12].

The descending gradient method uses, depending on the network parameters, the partial first-order derivative of the error function, so it is a first-order learning algorithm. When the partial derivative of the second order is used, additional information is obtained on the gradient, and the methods that use this information are called second-order algorithms [12]. Some of these methods are: Newton's method, the conjugated gradient method, and the scaled conjugate gradient method. The last two methods are detailed and described in Ref. [12], and Newton's method is described in Ref. [10].

The main disadvantage of the recurring ANN is given by the inverse connections of the recurring networks. They may have a delay order greater than the first order to store several previous network states. Due to the fact that the value of the gradient may depend on previous values, for a high delay order, the value of the

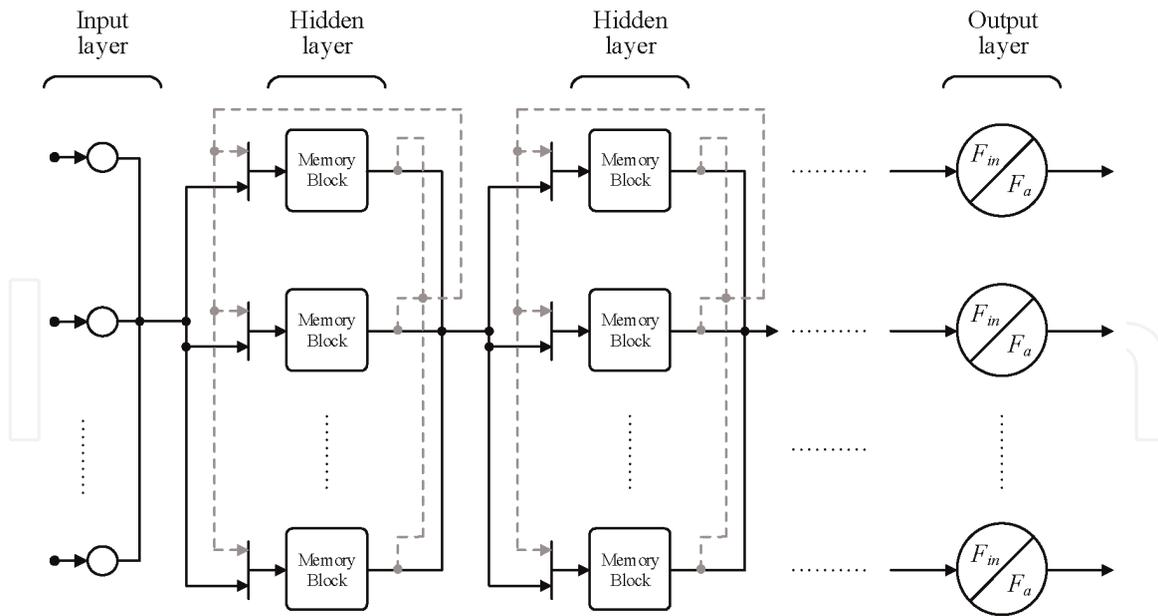


Figure 3.
 The schematic diagram of an LSTM network [13].

gradient may drop very rapidly to an infinitesimal value (vanishing gradient) or increase to a large value (exploding gradient) [13].

4.2 Recurrent ANN of LSTM type

Long short-term memory (LSTM) networks are recurrent networks that have the ability to memorize/learn short-term dependencies but for a long time. It solves the problem of the vanishing gradient by maintaining the local error at a constant value or in a certain domain, so the value of the gradient does not reach infinitesimal values or very large values [13, 14].

Compared to recurrent networks that have neurons in the hidden layer and context neurons or connections with delay blocks, LSTM networks have blocks of memory in the hidden layer. Each memory block contains one or more memory cells, an input gate, an output gate, and, optionally, a forget gate [13]. The schematic diagram of an LSTM network and of a memory block is shown in **Figure 3**.

The role of the cell is to maintain and transmit information from the input of the memory block to the output. The input gate determines the information that enters the cell, and the output gate detects the information coming out of the memory block. The gates control this by calculating the weighted amount of the gate entries and the weight of each entry. This sum is sent to a unipolar sigmoid function, where a value between 0 and 1 is obtained, which controls what information enters the cell and what information exits the memory block [14].

The inputs of the memory block are propagated forward to the input gate, forget gate, and exit. Each gate and circle containing the symbol \sum and a block representation of the unipolar sigmoid function or hyperbolic tangent in **Figure 4** are represented as artificial neurons because the mathematical operations that are applied to the inputs are identical and simplify the graphical representation.

In addition to the entry weights: $\omega_g, \omega_i, \omega_f$, and ω_o , there are also moving weights: b_g, b_i, b_f , and b_o .

To determine the output of the memory block [14], the current state of the cell is calculated before it is affected by the input gate and the forget gate, denoted by g in **Figure 4**:

$$g = F_g(y_g) = F_g\left(\sum_{j=1}^m \omega_{gj} \cdot x_{gj} + b_g\right) \quad (10)$$

where $i = \overline{1, m}$, and m represents the number of inputs. After this, the output of the input port i is calculated:

$$i = F_i(y_i) = F_i\left(\sum_{j=1}^m \omega_{ji} \cdot x_{ji} + b_j\right) \quad (11)$$

The output of the forget gate u can be determined as:

$$u = F_u(y_u) = F_u\left(\sum_{j=1}^m \omega_{ju} \cdot x_{ju} + b_j\right) \quad (12)$$

These three values, g , i , and u , together with the previous state of the cell $c^{<k-1>}$, are used in determining the current state of c cell:

$$c = u \cdot c^{<k-1>} + i \cdot g \quad (13)$$

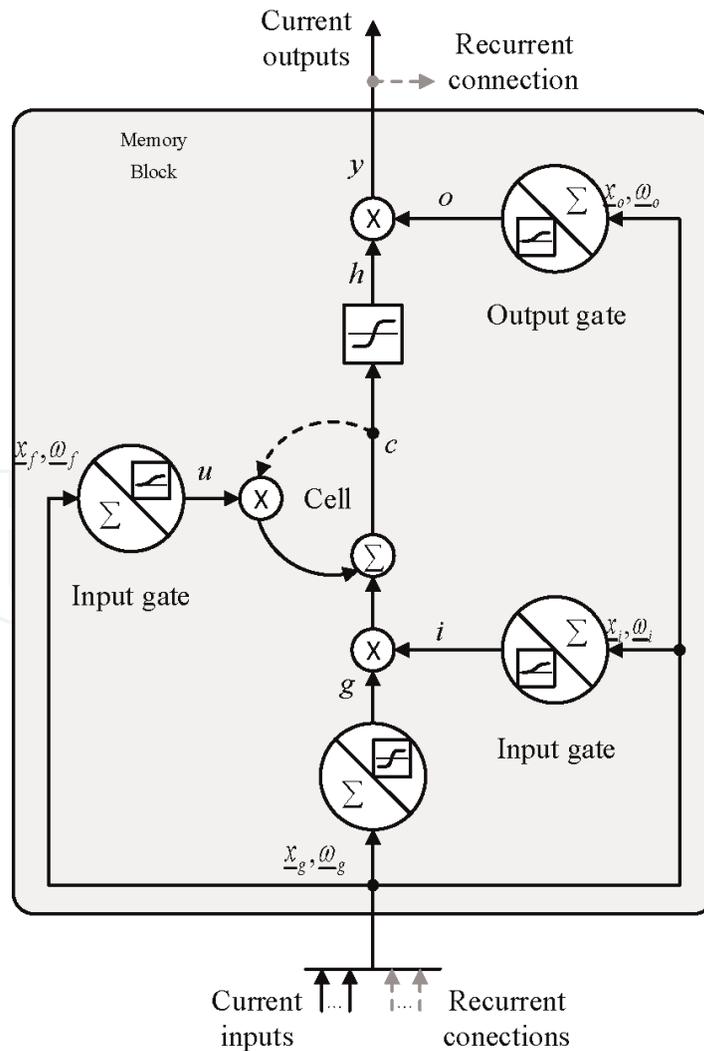


Figure 4. Principle diagram of a memory block (dotted line connections are recursive connections that have first-order delay) [13].

Further, the state of the cell evolves through the hyperbolic tangent function F_h , and h is determined. It is further calculated the activation of the output gate o , after which the two are multiplied to obtain the output of block y :

$$o = F_o(y_o) = F_o\left(\sum_{j=1}^m \omega_{jo} \cdot x_{jo} + b_j\right); y = o \cdot F_h(c) \quad (14)$$

The LSTM networks are trained using a slightly modified version of the BPTT algorithm and of the RTRL algorithm. For the output gate, RTRL is used, and for the rest of the gates and elements of the memory block, RTRL is used. The modification consists in the fact that the errors are considered only for updating the weights of one block, without the error of the other block being modified. The effect created by the gates is that the error can only pass unchanged through the cells. The training algorithm is detailed in Ref. [13].

Following is the implementation of the models discussed for the prediction and the results obtained.

5. Analysis of the neural network-based forecasting models

5.1 RNN-based forecasting model

The chosen RNN network has one input layer, one hidden layer, and one output layer. The hidden layer has recurrent connections from exits to inputs, being a recurrent Elman type network (Figure 2).

The activation function of hidden layer neurons is the unipolar sigmoid function, which is why the time series is normalized in the domain $[0, 1]$, and the activation function of the output neurons is the linear function.

In addition to the weights of feedforward connections, the recurrent network contains weights for the recurrent links. These are initialized with random values in the domain $[-0.1, 0.1]$, and the random values have a uniform distribution (white noise). The weights of feedforward connections are initialized with the value 0, and the displacement weights are initialized with the value 1.

For the network training and the forecasting, the walk-forward method is used, and the parameter assignment is performed by the experimental method. The error statistics for different values of the network parameters are presented below.

By using the graphs shown in Figures 5–8, an RNN model was chosen with the following parameters: a window size of one sample, a single neuron in the

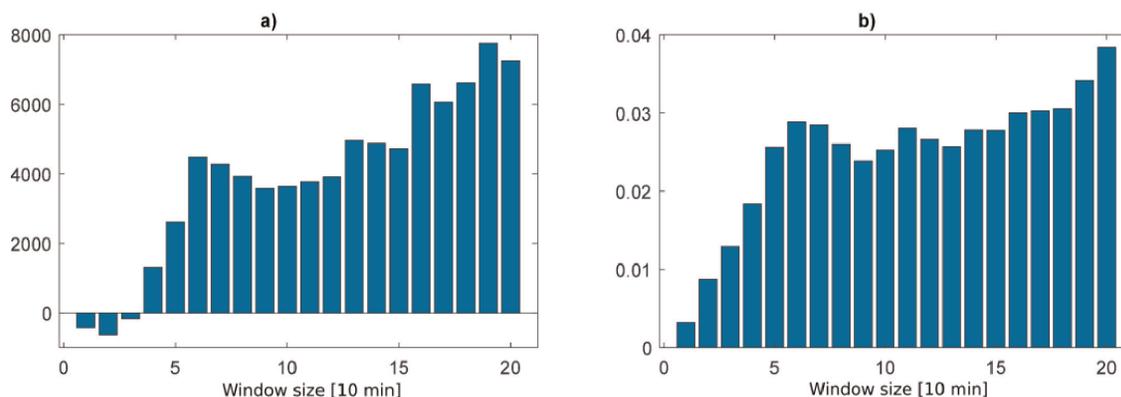


Figure 5. Window size. (a) SMSE, and (b) NMSE.

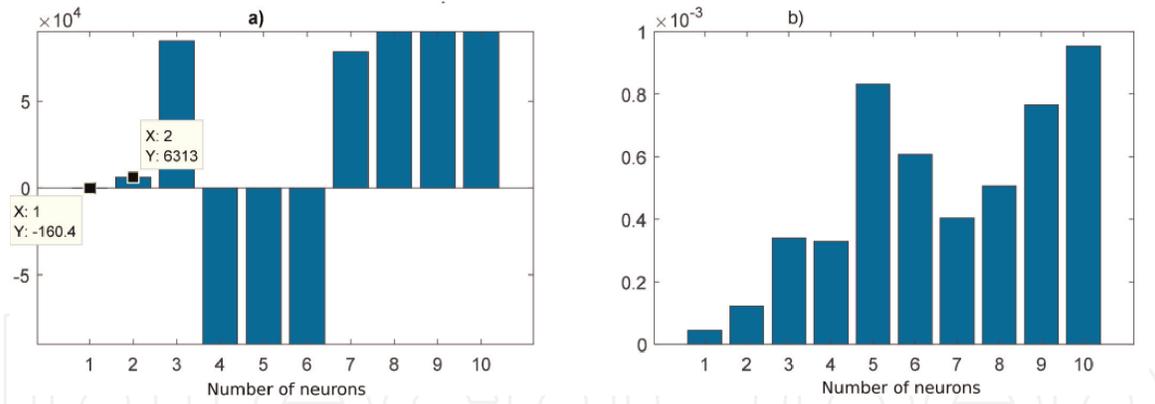


Figure 6. Error statistics for neuron number from the hidden layer. (a) SMSE, and (b) Theil's U statistic.

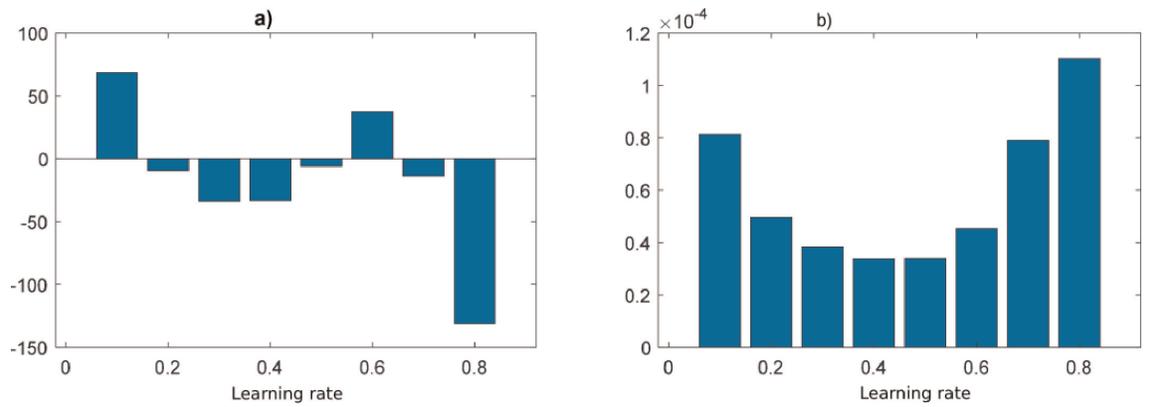


Figure 7. Error statistics for learning rate. (a) SMSE, and (b) Theil's U statistic.

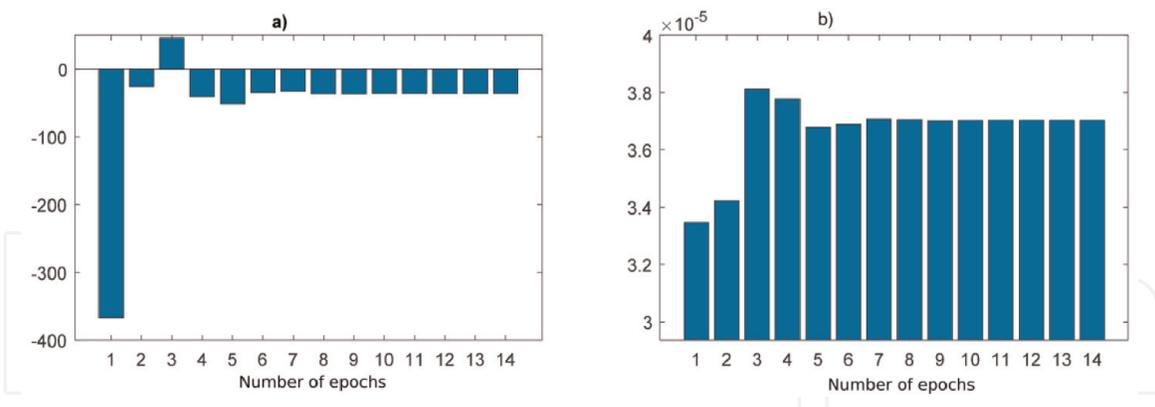


Figure 8. Error statistics for number of training epochs for a window. (a) SMSE, and (b) Theil's U statistic.

hidden layer containing the recurrent links, the learning rate of 0.4, and two-training periods.

Although the window is one sample, the number of previous steps is four samples. In addition to these parameters, tests were performed for the moment of rate learning (0.2), the slowing of the learning rate (0), and the number of previous samples (4).

The forecast realized with the RNN (4-1-1) model and the performance of the model is shown in **Figures 9** and **10**.

From the above figures and **Table 2**, it is observed that the RNN model succeeds in a good forecast of the time series when the forecast horizon is a sample ahead, the maximum error being 408.353 MW, which means 15.2% of the maximum electricity.

Next, the aim is to identify the impact of the horizon prediction increasing. To increase the prediction horizon, six samples were considered before corresponding to 1 h. The same iterations of the parameters were repeated, and the chosen model is RNN (6-13-6) with the following parameters: a window size of one sample, 13 neurons in the hidden layer containing the recurrent links, a learning rate of 0.4, the moment of learning rate 0.5, the slowing of the learning rate is omitted (0), and two periods of training.

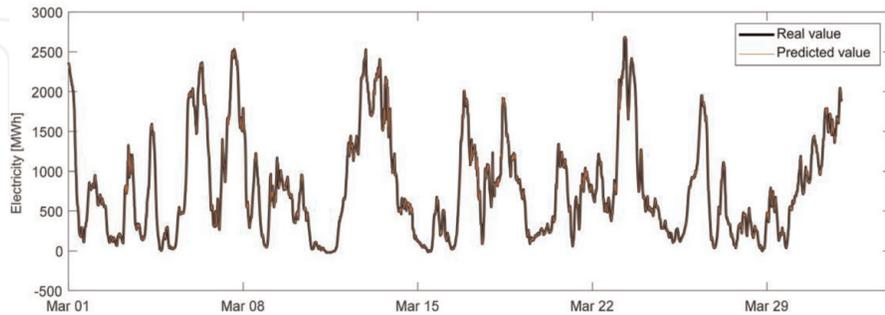


Figure 9.
 Forecast performed with the model RNN (4-1-1).

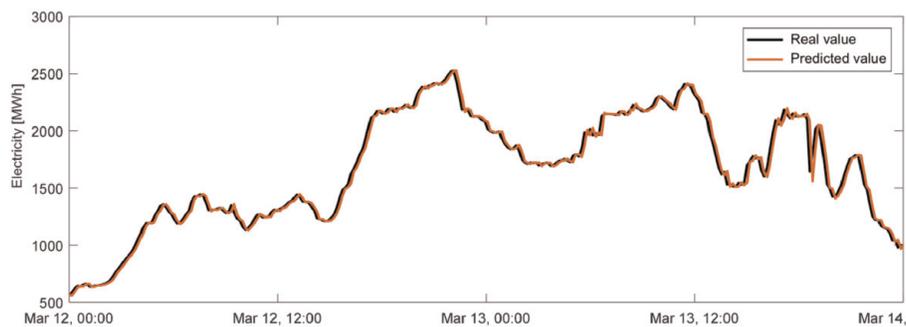


Figure 10.
 Forecast performed with the model RNN (4-1-1), from March 12 to 14.

MAE	MAPE	SMSE	NMSE	Theil's U statistics
23.02	5.66	-25.61	0.0030	3.42×10^{-5}

Table 2.
 RNN (4-1-1) model performances.

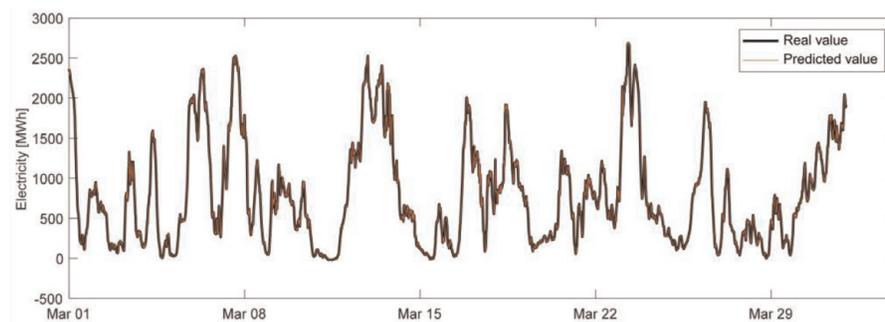


Figure 11.
 Forecast performed with the model RNN (6-13-6).

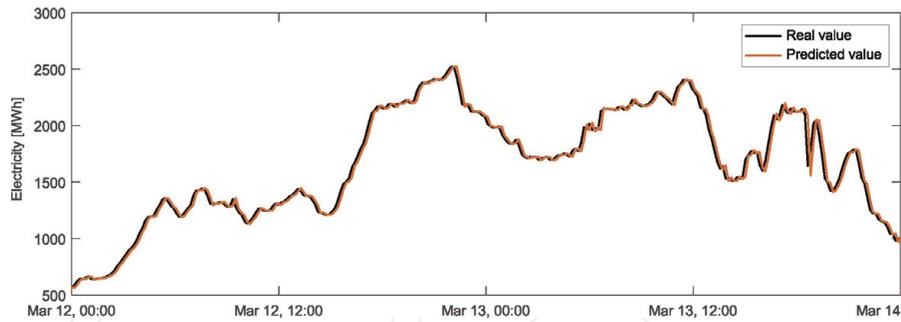


Figure 12. Forecast performed with the model RNN (6-13-6), from March 12 to 14.

MAE	MAPE	SMSE	NMSE	Theil's U statistics
21.35	5.27	-9.28	0.0028	3.27×10^{-5}

Table 3. RNN (6-13-6) model performances.

Although the window is one sample, the number of previous steps is six samples. It can be observed that at every six samples or window moving, a forecast of the next six samples is made.

The performance and forecast achieved with the RNN model (6-13-6) for a forecast horizon of six samples are presented in **Figure 11**.

From **Figure 12**, it can be seen that the RNN model manages to approximate the time series, the performance being synthesized in **Table 3**.

Although the forecast horizon was widened by five samples, performances similar to those obtained for the single-sample prediction were obtained, with the difference that the model requires more neurons in the hidden layer.

5.2 LSTM implementation and analysis of the model based on LSTM networks

The LSTM model is similar to the RNN model; it contains an input layer, a hidden layer, and an output layer. **Figure 3** represents the schematic of the model, with the difference that there is only one hidden layer.

Instead of neurons, the LSTM network contains memory blocks that have inputs, outputs, and different gates (input, output, and forget). For the inputs and outputs, the hyperbolic tangent transfer function is used, and for the gates, an estimate of the unipolar sigmoid activation function is used. Since the inputs and outputs of a memory block use the hyperbolic tangent function, the time series is normalized in the interval $[-1, 1]$.

For the network training and the forecasting, the walk-forward method is used, and the parameter assignment is performed by the experimental method. The error statistics for different values of the network parameters are determined further.

By using the graphs shown in **Figures 13–16**, a LSTM model was chosen with the following parameters: a window size of one sample, five neurons in the hidden layer containing the recurrent links, a learning rate of 0.4, and two-training periods. Although the window is one sample, the number of previous steps is three samples. In addition to these parameters, tests were performed for the moment of learning rate (0.3) and the slowing of the learning rate (0).

The forecast realized with the LSTM model (3-5-1) and the performance of the model is described below (**Figures 17 and 18 and Table 4**).

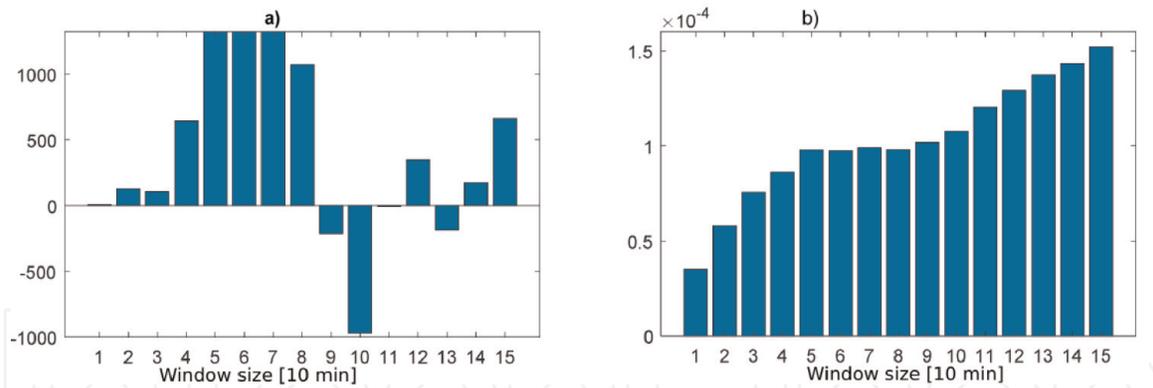


Figure 13. Statistics for window size. (a) SMSE, and (b) Theil's U statistic.

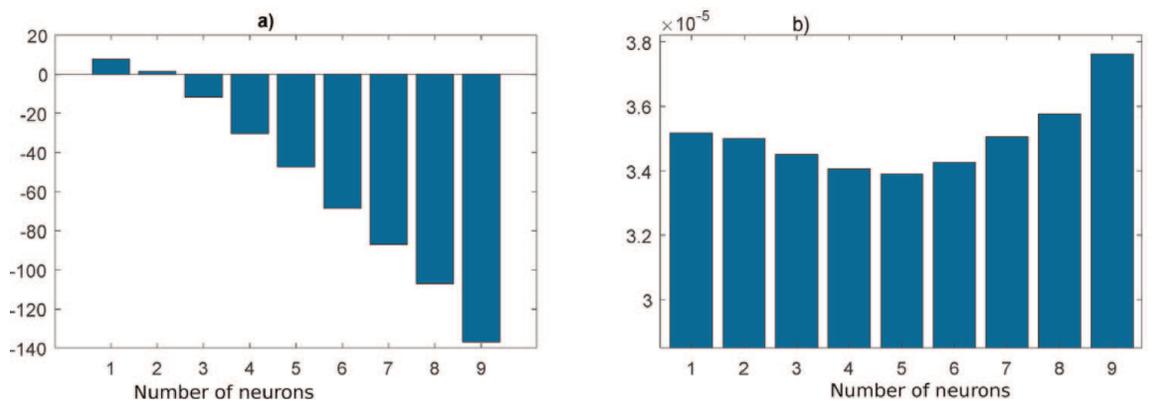


Figure 14. Statistics for neuron number from the hidden layer. (a) SMSE, and (b) Theil's U statistic.

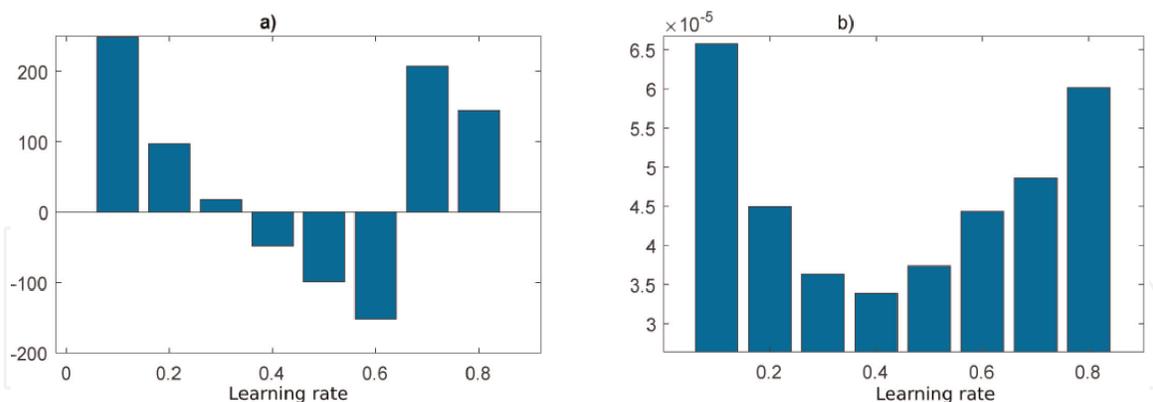


Figure 15. Statistics for learning rate. (a) SMSE, and (b) Theil's U statistic.

The LSTM model has performances almost identical to the RNN model, the maximum error being 408,441 MW meaning 15.21% of the maximum electricity.

Similar to the previous models, to increase the prediction horizon, six samples were considered before corresponding to 1 h. The same iterations of the parameters were repeated, and the chosen model is LSTM (1-29-6) with the following parameters: a window size of one sample, 29 neurons in the hidden layer containing the recurrent links, the learning rate of 0.6, the moment of learning rate 0.5, the slowing of the learning rate is omitted (0), and two training times, and only one previous sample is used. Similar to the RNN model, at every six samples or window movement, a forecast of the next six samples is made.

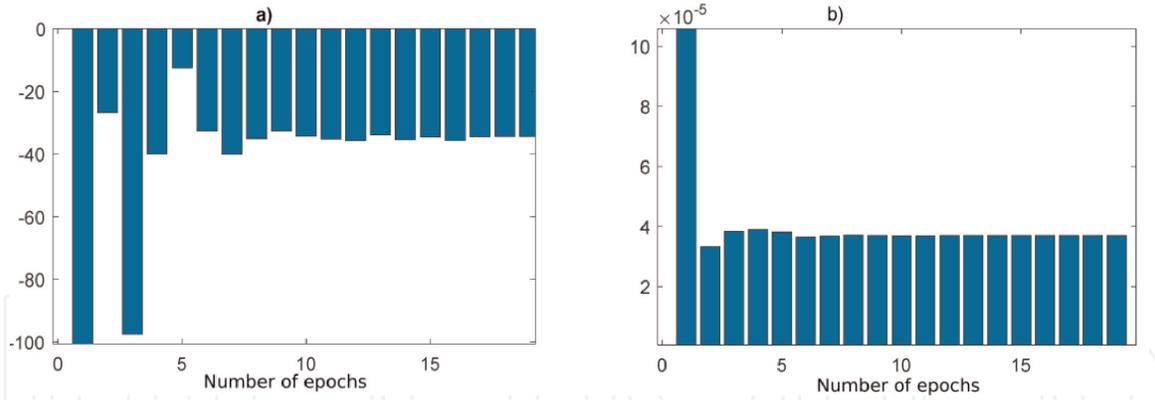


Figure 16. Statistics for number of training epochs for a window. (a) SMSE, and (b) Theil's U statistic.

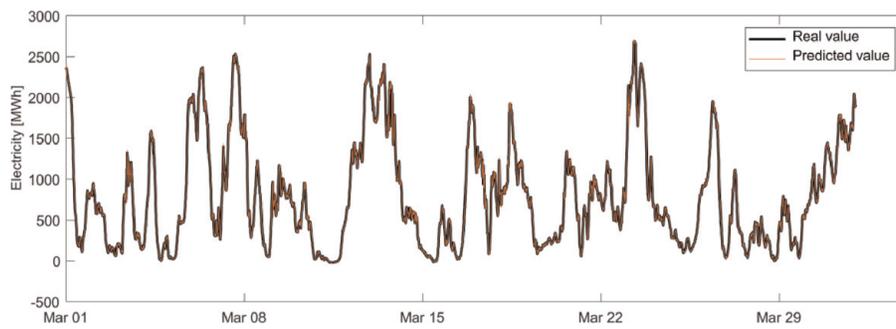


Figure 17. Forecast performed with the model LSTM (3-5-1).

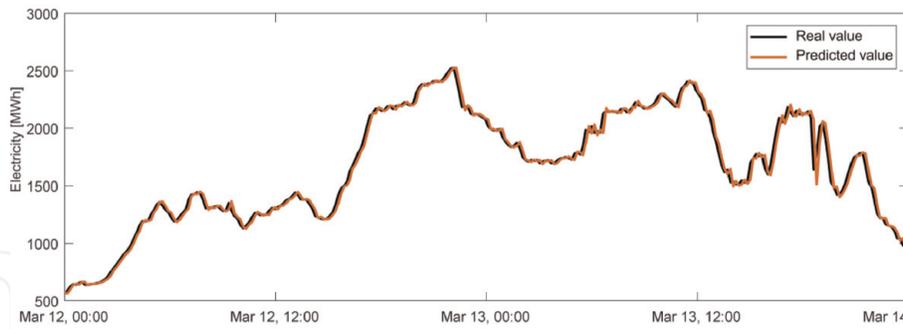


Figure 18. Forecast performed with the model LSTM (3-5-1), from March 12 to 14.

MAE	MAPE	SMSE	NMSE	Theil's U statistics
22.17	5.42	-26.82	0.0029	3.33×10^{-5}

Table 4. LSTM (3-5-1) model performances.

The performance and forecast achieved with the LSTM model (1-29-6) for a six-sample forecast horizon are presented below (Figure 19 and Table 5).

The performances are similar to the RNN and LSTM models with a sample prediction horizon and the RNN model with six forward samples. The maximum error is 391.435 MW, which corresponds to 14.57% of the maximum electricity.

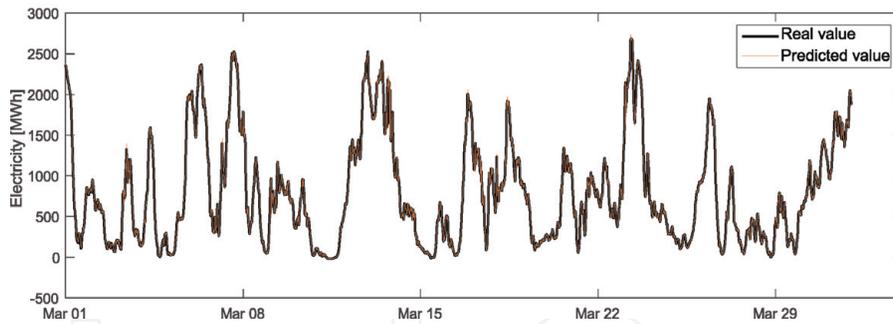


Figure 19.
 Forecast performed with the model LSTM (1-29-6).

MAE	MAPE	SMSE	NMSE	Theil's U statistics
21.36	5.22	-41.33	0.0027	3.25×10^{-5}

Table 5.
 LSTM (1-29-6) model performances.

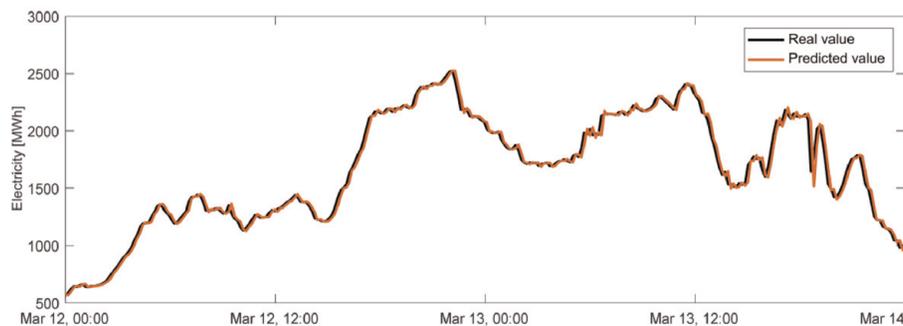


Figure 20.
 Forecast performed with the model LSTM (1-29-6), from March 12 to 14.

Model	MAE	MAPE	SMSE	NMSE	Theil's U statistics
RNN (4-1-1)	23.02	5.66	-25.61	0.0030	3.42×10^{-5}
LSTM (3-5-1)	22.17	5.42	-26.82	0.0029	3.33×10^{-5}

Table 6.
 Models performances for a horizon of one sample.

5.3 Models comparison

5.3.1 Models comparison for a prediction horizon of one sample (10 min)

The compared models are RNN (4-1-1) and LSTM (3-5-1). The used comparison period is 1 day (March 13), and the performances are determined on the entire time series.

Between the RNN and LSTM models, the differences are very small as can be seen in **Figure 20** and **Table 6**, the LSTM model being with MAE—3.7%, MAPE—4.24%, NMSE—3.33%, and Theil's *U*—2.63% more efficient than the RNN model, but the SMSE statistics is 4.72% higher.

For the forecast horizon of a sample and the time series used, the RNN and LSTM models present the best performances (**Figure 21**).

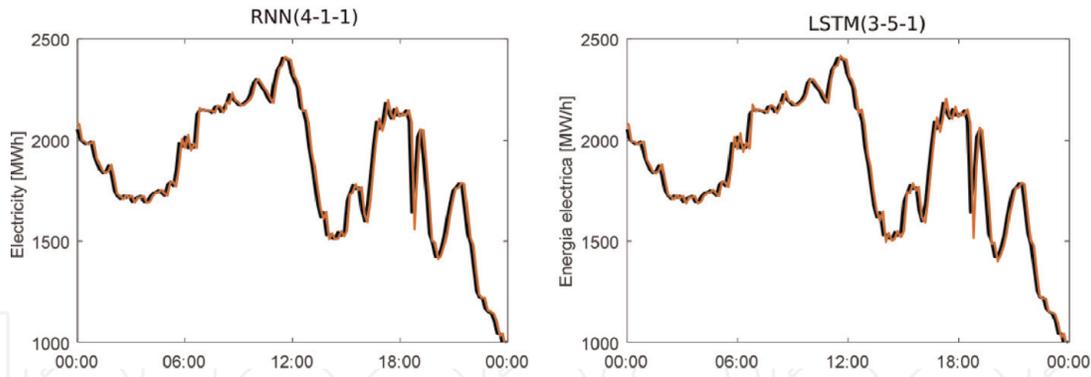


Figure 21. Comparison of the two models in achieving the forecast with a horizon of one sample.

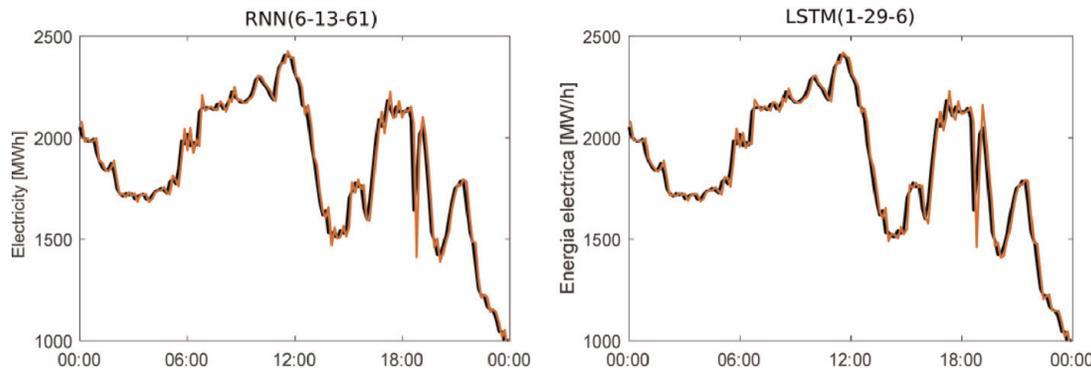


Figure 22. Comparison of the two models in achieving the forecast with a horizon of six sample.

Model	MAE	MAPE	SMSE	NMSE	Theil's U statistics
RNN (6-13-6)	21.35	5.27	-9.28	0.0028	3.27×10^{-5}
LSTM (1-29-6)	21.36	5.22	-41.33	0.0027	3.25×10^{-5}

Table 7. Models performances for a horizon of six samples.

5.3.2 Performance comparison for a prediction horizon of six samples (1 h)

The compared models are RNN (6-13-6) and LSTM (1-29-6). The period used for comparison is the same, and 1 day (March 13) and the performances are determined for the entire time series (Figure 22 and Table 7).

Models based on recurrent networks manage a better approximation of the time series and achieve similar or even better performance than the one-sample prediction before. Between RNN and LSTM, the differences are small between MAE, MAPE, NMSE, and Theil's U statistics, and the largest difference appears in SMSE statistics. It is worth mentioning that the LSTM model has performances similar to the RNN model and has a single sample before and the same size as the window of a sample.

5.3.3 Performances comparison of the RNN and LSTM models for different values of the forecast horizon

The increasing forecast horizon leads inevitably to higher errors and lower performance. For the forecast, the same models were used: RNN (6-13-6) and

LSTM (1-29-6), and the forecast horizon ranges from 6 to 144 samples with a step of six samples between tests.

From the figure above, it can be observed that the two models have similar performances up to a prediction horizon of 50 samples, after which the difference between the models increases, the performances being better for the RNN model.

The forecasts made with the two models for a prediction horizon of 144 samples (1 day) are presented below.

In **Figure 23**, a significant prediction error is observed for the first forecast made, after which the next forecast has a much lower prediction error. Although the models are trained on each window, the forecast is only performed on multiples of 144 samples. This simplifies the arrangement of the resulting values, each of the 144 future samples being saved in a vector that is added to the previous prediction vector. To eliminate this initial error, more frequent predictions can be made, even at each window (10 min in this time series), but manipulating the results is more difficult.

From **Figures 23** and **24**, it can be observed that the two models have similar values for the small variation zones of the time series and for the larger variation

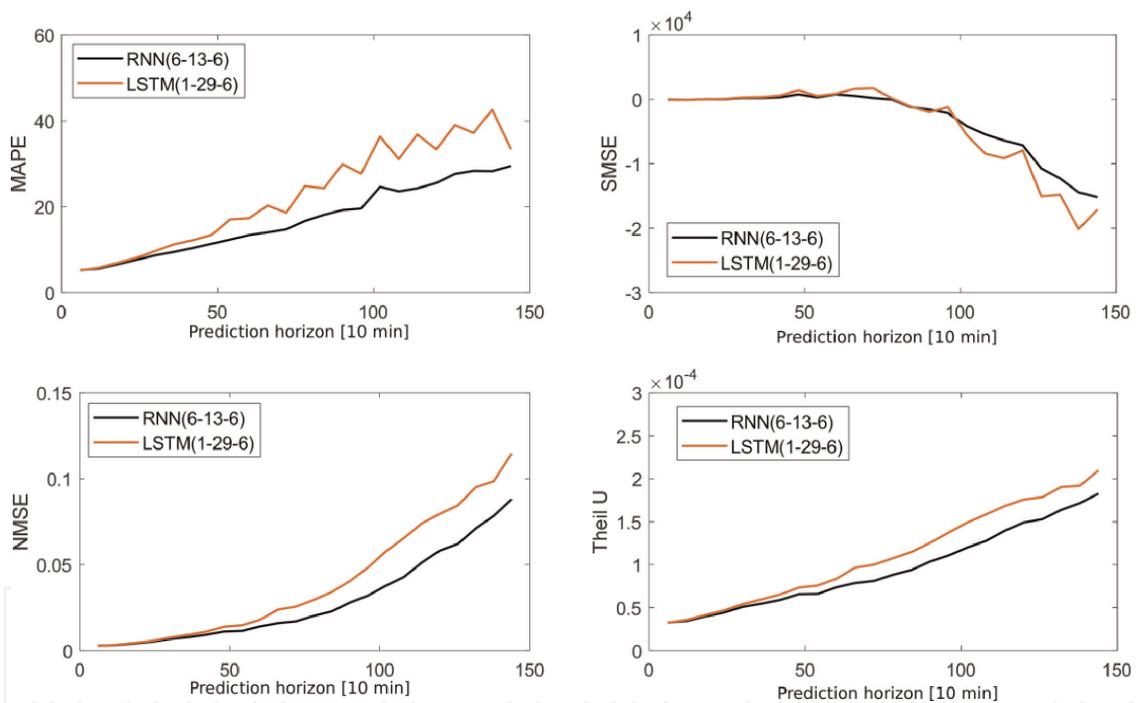


Figure 23. Models RNN (6-13-6) and LSTM (1-29-6) comparison for different values of the prediction horizon.

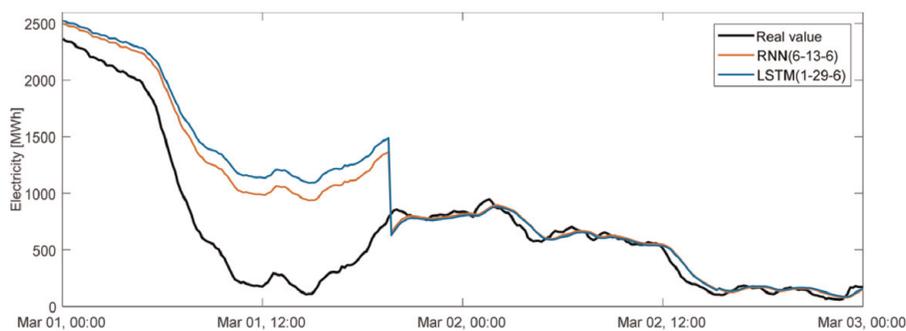


Figure 24. Models RNN (6-13-6) and LSTM (1-29-6) comparison and a 144 samples horizon, March 1 to 3.

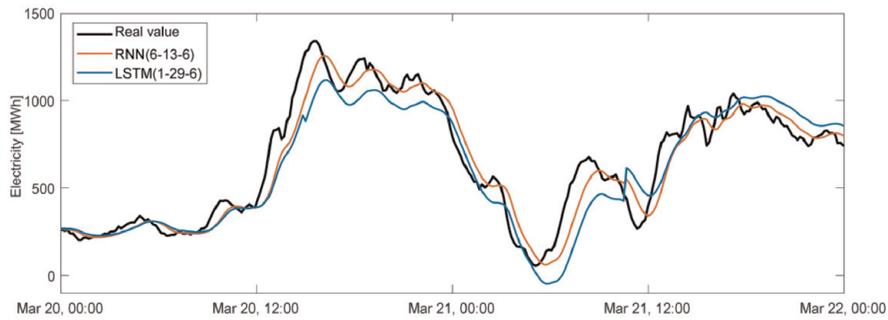


Figure 25. Models RNN (6-13-6) and LSTM (1-29-6) comparison and a 144 samples horizon, March 20 to 22.

Model	MAE	MAPE	SMSE	NMSE	Theil's U statistics
RNN (6-13-6)	111.86	29.47	-1514.3	0.0881	1.83×10^{-4}
LSTM (1-29-6)	127.98	33.40	-1708.3	0.1147	2.10×10^{-4}

Table 8. RNN (6-13-6) and LSTM (1-29-6) models performances comparisons for a 144 samples prediction horizon.

zones, the values differ, the more efficient model being the RNN model (**Figure 25** and **Table 8**).

The parameters of the models have not been reconfigured, most likely, if a longer training (more epochs), a larger window, and a number of previous samples are allowed, the model's performances would increase.

For the chosen parameters of the models and the time series taken into account, the RNN model presents the best performances for a prediction horizon of 144 samples (1 day). For a better performance, it may be considered the reconfiguration of parameters, the networks structure changing such as adding more hidden layers, the activation function changing, or the usage of another type of neural network.

6. Conclusions

In this chapter, two prediction models based on artificial neural network (recurrent type neural networks: RNN and LSTM) for wind power forecasting were studied. Experimental results exhibited were obtained through the data processing performed with the aim of Python platform with Keras library and Tensorflow.

It can be noted that there are a multitude of models recorded in literature, and some are generalized and can be applied to multiple domains, and others are more specific to one domain or application. Furthermore, no other variables, except the production of electrical energy, have been considered. For example, the wind speed and weather forecasts can be used to forecast, and results with and without these added variables can be compared.

To achieve the proposed objective of forecasting, only one variable was used, namely, historical records of electricity production from wind energy, but several variables that influence this size can be considered.

In addition to highlighted problems, other directions of research consist in identifying stochastic models and models based on neural networks, capable to approximate the seasonal and trend components of time series. There is possible to adopt models that reduce the number of parameters to be assigned, such as the model presented in Ref. [15].

The RNN and LSTM models were compared for different time horizons. For the chosen parameters, the RNN model presents the best performances in the case of the short-term horizon, the higher the forecast horizon, the greater the difference between the RNN and LSTM model results. It is worth mentioning that, the performances of the models depend on the chosen parameters, for this reason, different methods of parameter optimization can be engaged, resulting new methods that can further be investigated.

IntechOpen

IntechOpen

Author details

Cristian-Dragos Dumitru* and Adrian Gligor
George Emil Palade University of Medicine, Pharmacy, Science and Technology of
Targu Mures, Targu Mures, Romania

*Address all correspondence to: cristian.dumitru@umfst.ro

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. Distributed under the terms of the Creative Commons Attribution - NonCommercial 4.0 License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited. 

References

- [1] Eurostat. Renewable energy statistics. Data extracted in January 2019 [Internet]. 2019. Available from: https://ec.europa.eu/eurostat/statistics-explained/index.php/Renewable_energy_statistics#Wind_power_becomes_the_most_important_renewable_source_of_electricity [Accessed: 01 September 2019]
- [2] Wagner H-J, Mathur J. Introduction to Wind Energy Systems. Green Energy and Technology: Springer International Publishing AG; 2018. 105 p. DOI: 10.1007/978-3-319-68804-6_1
- [3] Gilbert C, Browell J, McMillan D. Leveraging turbine-level data for improved probabilistic wind power forecasting. *IEEE Transactions on Sustainable Energy*. 2019;1-9, forthcoming. DOI: 10.1109/TSTE.2019.2920085
- [4] Wang Y, Hu Q, Srinivasan D, Wang Z. Wind power curve modeling and wind power forecasting with inconsistent data. *IEEE Transactions on Sustainable Energy*. 2018;**10**(1):16-25
- [5] Vatră CA. Renewable sources productivity—Particular case wind turbine. In: *Lucrările Conferinței Naționale și Expoziția de Energetică CNEE 2007*; 7-9 November 2007; Sinaia: Romania; 2007. pp. 917-922
- [6] Gasch R, Twele J, editors. *Wind Power Plants: Fundamentals, Design, Construction and Operation*. Heidelberg, Dordrecht, London, New York: Springer; 2012. 548 p. DOI: 10.1007/978-3-642-22938-1
- [7] Golovanov N, Postolache P, Toader C. *Eficiența și calitatea energiei electrice*. București, Romania: AGIR; 2007
- [8] Zhang Y, Wang J, Wang X. Review on probabilistic forecasting of wind power generation. *Renewable and Sustainable Energy Reviews*. 2014;**32**: 255-270. DOI: 10.1016/j.rser.2014.01.033
- [9] Naik J, Dash S, Dash PK, Bisoi R. Short term wind power forecasting using hybrid variational mode decomposition and multi-kernel regularized pseudo inverse neural network. *Renewable Energy*. 2018;**118**: 180-212. DOI: 10.1016/j.renene.2017.10.111
- [10] Hagan MT, Demuth HB, Beale MH, De Jesús O. *Neural Network Design*. Stillwater, Oklahoma: Martin Hagan; 2014
- [11] Kriesel D. A Brief Introduction to Neural Networks, 27 May 2005. [Internet]. 2019. Available from: http://www.dkriesel.com/en/science/neural_networks [Accessed: 29 April 2018]
- [12] Medsker LR, Jain LC. *Recurrent Neural Networks: Design and Applications*. Boca Raton, FL: CRC Press; 2001
- [13] Gers F. Long Short-Term Memory in Recurrent Neural Networks
- [14] Olah C. Understanding LSTM Networks, 27 August 2015. [Internet]. Available from: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> [Accessed: 23 May 2019]
- [15] Ghiassi M, Saidane H, Zimbra DK. A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting*. 2005;**21**:341-362