

Buturache, Adrian-Nicolae; Stancu, Stelian

Article

Usage of neural-based predictive modeling and IIoT in wind energy applications

Amfiteatru Economic Journal

Provided in Cooperation with:

The Bucharest University of Economic Studies

Suggested Citation: Buturache, Adrian-Nicolae; Stancu, Stelian (2021) : Usage of neural-based predictive modeling and IIoT in wind energy applications, Amfiteatru Economic Journal, ISSN 2247-9104, The Bucharest University of Economic Studies, Bucharest, Vol. 23, Iss. 57, pp. 412-428,
<https://doi.org/10.24818/EA/2021/57/412>

This Version is available at:

<https://hdl.handle.net/10419/281588>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>

**USAGE OF NEURAL-BASED PREDICTIVE MODELING AND IIoT
IN WIND ENERGY APPLICATIONS****Adrian-Nicolae Buturache^{1*} and Stelian Stancu²**¹⁾²⁾ *Bucharest University of Economic Studies, Romania***Please cite this article as:**

Buturache, A.N. and Stancu, S., 2021. Usage of Neural-Based Predictive Modeling and IIoT in Wind Energy Applications. *Amfiteatru Economic*, 23(57), pp.412-428.

DOI: [10.24818/EA/2021/57/412](https://doi.org/10.24818/EA/2021/57/412)

Article History

Received: 29 December 2020

Revised: 14 February 2021

Accepted: 4 March 2021

Abstract

The adoption of wind energy has grown significantly in recent years. New, cost-effective technologies have been developed, led by customer awareness of green technologies and a legal framework proposed at the European Union level. The stochastic nature of wind speed is transferred to wind turbine output, making wind energy difficult to predict. The main scope of predicting wind energy production is to be proactive in balancing and reserving energy to meet demand. When the prediction identifies a potential gap between supply and demand, additional energy from other sources must be generated and supplied. Creating a synergy of physical devices through advanced sensing capabilities, software, storage and analytics capabilities, the Industrial Internet of Things is enabling the effective transition to wind energy through automation by removing many of the disadvantages in a way that has recently become accessible. This research focuses on the data analytics, proposing a fast univariate network-based approach for wind energy prediction, using Feed Forward Neural Networks, Recurrent Neural Networks, Long-Short Term Memory, Gated Recurrent Unit, and Convolutional Neural Networks. Moreover, by introducing the theoretical fundamentals, the implementation method and the hyperparameters of the final models, this article becomes unique in the context of wind energy. At the time of this study, no prior research studies have presented a direct comparison between feedforward, recurrent, and convolutional neural networks – these being the most important in the field of supervised learning.

Keywords: machine learning, artificial neural networks, wind energy, internet of things, industrial internet of things

JEL Classification: O13, O14, O31, Q47, C45

* Corresponding author, **Adrian-Nicolae Buturache** – e-mail: ad.buturache@yahoo.ro

Authors' ORCID:

Adrian-Nicolae Buturache: orcid.org/0000-0002-6146-2992

Stelian Stancu: orcid.org/0000-0002-4727-993X

Introduction

The renewable energy sector has experienced considerable global growth in the last few years. This growth comes with huge challenges, in terms of asset management. Wind turbine site selection can lead to geographically dispersed layouts, which makes the management of these installations a challenging task. By adopting an Internet of Things (IoT) approach, geographical dispersion-related problems can be overcome by enabling remote descriptive, diagnostic, and predictive analytics to minimize operational costs by maximizing production and preventing unplanned, costly downtimes. Predictions of wind energy is one of the core data needed in real-time control of power systems: to be close to the real-time ideal means delivering fast predictions, in sync with entire system dynamics. The higher the prediction accuracy, the higher the efficiency of the system, resulting in savings for all investors. IoT can enable companies across the entire energy supply chain to achieve their targets. The continuously increasing adoption of these renewable technologies has led to a situation that was not previously possible: small installations built by businesses or homeowners. These installations were initially built to cover daily basic energy consumption, but they can also send energy into the power grid. This reinforces the dispersed character of current power systems and creates a new challenge, in terms of optimal grid management. However, data from smart energy meters has the advantage of driving improvements for consumers, for example, by identifying waste, such as power-hungry devices or automated heating, ventilation, and air conditioning systems. The growth of wind power systems is driven by European Commission and Government legislations. For the next 10 years, through the European Green Deal, there are precise targets for the minimum percentage of renewable energy to be used and the reduction of greenhouse gas emissions (European Commission, 2020). Even though most renewable energy resources were identified decades ago, they have not been able to replace fossil fuel-based sources because of their intermittent and variable availability. The solution for this has been to gradually add them to existing power grids, which has been possible due to the development of smart grids that include features, such as power consumption and output power prediction. In this way, a feedback loop between customer and supplier is created. Then, the gap between demand and supply can be covered using fossil fuel-based sources, overcoming the negative impact of availability. The successful integration of wind-based technologies in the existing electricity grid depends on the accuracy of wind prediction. Short-term forecasting plays an important role, for both operational and energy trading activities. The IoT, by definition, covers every piece of technology with the capability to communicate with other devices, systems and networks (Ashton, 2009). Industrial professional and academics split the IoT in two branches: Industrial Internet of Things (IIoT) and Customer Internet of Things (CIoT) (Al-Ali, 2016). IIoT is represented by smart grids, factories, cars, and machines, while CIoT is oriented to the customer and their devices, such as smart home devices, connected cars and wearables. IIoT and CIoT are connected, enabling information transfer between them. This research aims to provide an overview on the use of predictive analytics based on neural networks and IIoT in the wind energy industry. The focus is on analytics, by identifying the theoretical and practical aspects of using network-based algorithms for rapid, short-term univariate predictions of wind energy production, using Romanian-based wind turbines data for the case study. Finally, the results of the best performing models belonging to each of the five selected typologies are compared, both in terms of generalization capacity and training time.

1. Literature review

Methods for wind energy prediction can be grouped depending on the timescale (short-term, medium-term, long-term), model type (physical, statistical, machine learning, hybrid) and the variety of parameters (univariate, multivariate). Physical models are built on the exogenous variables that influence energy production. However, being deterministic, they are dependent on the location and physical properties of the environment in which the wind turbines are located, making them less versatile than other models. In these models, another important aspect is that for all methods of prediction, either the output energy is predicted directly, either firstly wind speed is predicted and then based on power curves, the energy is determined analytically. Most literature uses the indirect method, as described below. An error correction model, based on a bidirectional gated recurrent unit neural network, is proposed to correct the error of the numerical weather prediction of wind speed (Ding, et al., 2019). The results outperformed the selected benchmark models for short-term power prediction and the same approach could also be used for medium and long-term predictions. Using the same univariate machine learning model, Long-Short Term Memory (LSTM) and One-dimensional Convolutional Neural Networks (1D-CNN) can be implemented (Fukuoka, et al., 2018). In this study both LSTM and 1D-CNN provide better performances than the Feed Forward Neural Networks (FFNN). Another way to extract meaningful information from a time series is represented by Empirical Wavelet Transform (Wang and Hu, 2015), then the GPR (Gaussian Process Regression) model combines, in a nonlinear way, the predictions generated by other models like ARIMA (Autoregressive Integrated Moving Average), ELM (Extreme Learning Machine) and SVM (Support Vector Machine). This method is more accurate than the standalone models for predicting short-term wind speed at two sites. An example of ARIMA implementation on real operational data illustrates the improvement in reducing energy buffers, resulting in a cost reduction by accurately predicting wind speed (Eldali, et al., 2016). A similar outcome can be achieved by direct predicting of the wind power (Pant and Garg, 2016). The model performance can also be improved by dividing the year into months and building separate models for each of them (Chen and Lai, 2011). The comparison of ARIMA and FFNN reveals that for each month, and for one hour, two hours, three hours, and four hours ahead, FFNN outperforms ARIMA. Under the direct wind energy prediction, a hybrid approach has been tried that consists of using a non-linear model for the non-linear component of the time series, and a statistical model for the linear factor. An example of this is using ARIMA for the linear and RBFNN (Radial Basis Function Neural Network) for the non-linear component. For large data similarity and a high-density time series, a preprocessing step for extracting the change trend information can be used (Liu, Ding and Jia, 2020). A K-means clustering method is proposed for obtaining a new time series that compresses the data, facilitates storage and utilization, and eliminates noise. The resulting time series is then used as input for ARIMA, SVM, GPR, ESN (Echo State Network), GRU (Gated Recurrent Unit), A-RNN (Attention Recurrent Neural Network), Input-Attn-RNN (Input Attention Recurrent Neural Network) and DA-RNN (Dual-stage Attention based Recurrent Neural Network). Except SVM, the results from all of the selected models were similar, proving the versatility of the network-based models.

Machine learning-based approaches enable researchers to study wind energy production without having much industry experience. These flexible and highly scalable models outperform existing models in fast univariate prediction tasks. Currently, there is a gap in direct univariate wind energy prediction. Furthermore, a comprehensive analysis of the most important network-based models is necessary, specifically to align with the challenges faced

by the wind energy industry. The enabler for implementing these machine learning-based solutions is facilitated by IIoT technologies firstly by the fact that operational data is readily available for consumption, and secondly by the fact that the results of predictive analyzes can be propagated backwards for real-time optimization of energy production.

2. Theoretical fundamentals

2.1. Feed Forward Neural Networks

The artificial neural network (ANN) concept was first introduced by McCulloch and Pitts (1943). Inspired by the human brain, ANN aims to replicate the way information flows between neurons. However, only after backpropagation was introduced (Rumelhart, Hinton and Williams, 1986) did ANN start to demonstrate its capabilities. By implementing backpropagation, the synaptic weights between neurons are updated according to the expected result. The value of the synaptic weight provides information about how important the inputs are to achieve the maximum possible accuracy. FFNN represents an instance of the ANN in which the information is processed while passing forward through the network, traveling from the input layer through hidden layers and, finally, the output layer (figure no. 1). Each layer consists of neurons, representing the computational units of the network.

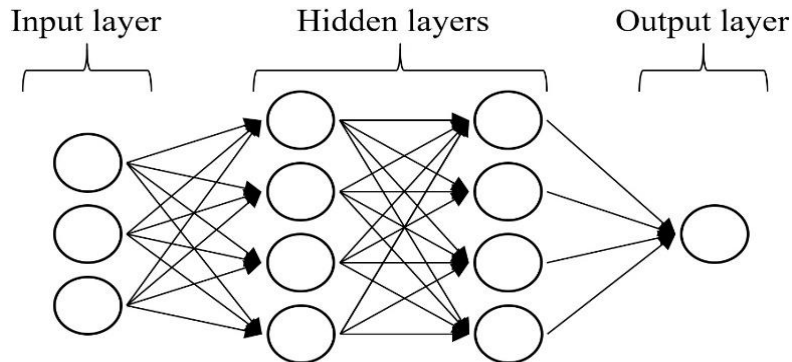


Figure no. 1. Feed Forward Neural Network high level architecture

The neurons only communicate with the outside world if they are in the input or output layer. The neurons of the hidden layers are receiving inputs from the neurons upstream and are sending the information downstream, either to another hidden layer or to the output layer. All neuron inputs have a synaptic weight (w_i), based on the inputs (x_i) and the impact of the output (y_i). The output of a neuron can be calculated as a weighted sum of the inputs (equation 2) on which the activation function f is applied (equation 1).

$$y=f(z) \quad (1)$$

$$z=\sum_{i=1}^n x_i w_i + b \quad (2)$$

Where b represents the bias.

Using the activation function introduces non-linearity into the neuron’s output. The neuron will be fired if the output value is greater than a given threshold, or it can be inhibited if the output is smaller than the threshold (equation 3). For the RELU (rectified linear unit) activation function:

$$y=f(z)=\max(0,z) \tag{3}$$

Backpropagation means the information travels back from the output layer to the input layer. The model’s error is used to update the network parameters, with respect to the objective function (i.e., error minimization). The way network parameters are updated is governed by the optimization algorithm (Kingma and Ba, 2015).

2.2. Recurrent Neural Networks

A Recurrent Neural Network (RNN) can be obtained when feedback connections are added, and are performant while using sequential data. The applications in the time series field are important since the prediction of a time step may depend on multiple steps backwards. The RNN can be defined using the following equations (Pascanu, et al., 2014):

$$h_t=f_h(x_t,h_{t-1})=\phi_h(\mathbf{W}^T h_{t-1}+\mathbf{U}^T x_t) \tag{4}$$

$$y_t=f_o(h_t,x_t)=\phi_o(\mathbf{V}^T h_t) \tag{5}$$

Where x_t = input vector; h_t = hidden state; h_{t-1} = previous step hidden state; y_t = output vector; \mathbf{W} , \mathbf{U} , \mathbf{V} = parameter matrices and f_h , f_o =activation functions. The input x_t and the previous hidden state h_{t-1} are concatenated. The newly created vector contains information of both the current input and the previous state. This vector is passed through a tanh activation function, resulting in output of the current state. The tanh layer regulates the output by fitting the values between -1 and 1 (figure no. 2).

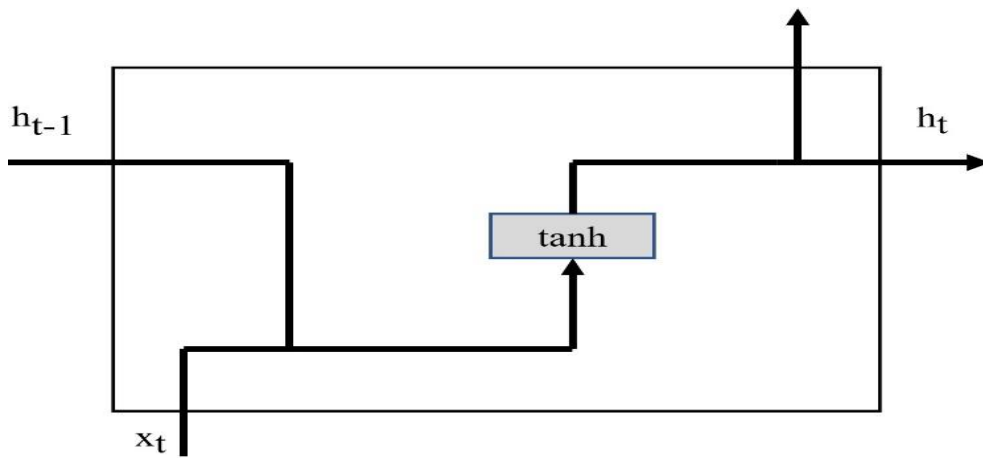


Figure no. 2. Recurrent Neural Network high level architecture

While FFNN is using backpropagation for training, RNN is using backpropagation through time (BPTT) (Rumelhart, Hilton and Williams, 1986). BPTT is suitable for network

applications where model parameters are updated in discrete time steps. The design of the RNN is vulnerable to exploding or vanishing gradient issues (Bengio, Simard and Frasconi, 1994).

2.3. Long-Short Term Memory

Long-Short Term Memory design (LSTM) (Hochreiter and Schmidhuber, 1997) retains the capabilities of the RNN to work with time series data, but it reduces information morphing and the issues related to vanishing and exploding gradients. The novelty of LSTM (figure no. 3) lies in a gating system that can manage the way information flows: the internal gating system ensures that input information can be kept or forgotten, according to its significance to the problem at hand.

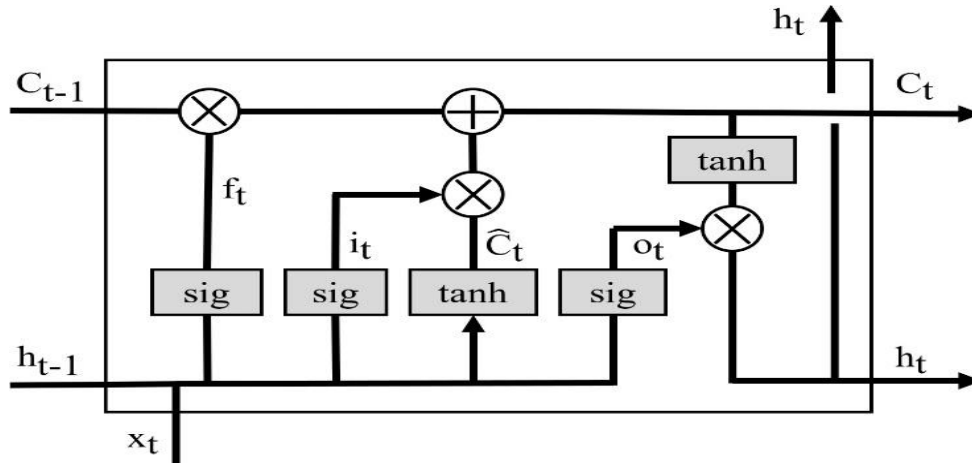


Figure no. 3. Long-short term memory high level architecture

According to the theoretical fundamentals in the cell state relevant information of the earlier steps is stored, reducing the negative impact of information morphing. A network’s gating system consists of three gates: the input gate, the forget gate and the output gate. These three gates are essentially neural networks. As the tanh activation function squashes the results between -1 and 1, the sigmoid activation function performs the same task, but within 0 and 1. The forget gate then decides which information of the previous cell state is to be kept or discarded (equation 6).

$$f_t = \sigma(x_t U^f + h_{t-1} W^f + b^f) \tag{6}$$

Where U^f and W^f are the weights of the current state input and previous cell output, with respect to the forget gate. The input gate consists of two mathematical layers. The first layer decides the new information that will be stored in the cell state (equation 7). This layer acts in a similar way to the forget gate: current state input and previous state cell output are passed through a sigmoid function. The differentiation is made by considering its own bias and weights for the current input and previous state cell output, b^i , U^i and W^i , respectively.

$$i_t = \sigma(x_t U^i + h_{t-1} W^i + b^i) \tag{7}$$

The second layer of the input gate takes the same current input and previous cell state and passes them through a tanh activation function (equation 8). This time, the bias and the weight matrices are specific to this second layer. The new candidate for the cell state is calculated as:

$$\hat{C}_t = \tanh(x_t U^g + h_{t-1} W^g + b^c) \tag{8}$$

The new cell state:

$$C_t = \sigma(f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t) \tag{9}$$

At this point, it decides what information is to be kept from the previous cell state, through the pointwise multiplication of the forget gate vector with the previous state cell vector. The newly computed cell state (equation 9), current input x_t and previous hidden state h_{t-1} are then used to compute a new hidden state. The output is calculated using the current state input and the previous hidden state, both with weights appropriate for this gate:

$$o_t = \sigma(x_t U^o + h_{t-1} W^o + b^o) \tag{10}$$

Through a tanh activation function, the newly calculated cell state C_t is regulated and further multiplied with the output gate result:

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

Now the current cell state C_t , known as long-term memory and the hidden state h_t , known as short-term memory are computed. The logic described above is repeated for all the new time steps considered. The output of each time step is obtained using the short-term memory.

2.4. Gated Recurrent Unit

GRU it is a relatively new design (Cho, et al., 2014), which has been gaining popularity since its inception. As with the LSTM, this design aims to reduce the issues related to long-term dependencies and vanishing or exploding gradients. GRU logic is similar to that implemented in the LSTM, in that the information flow is regulated by a gating system (figure no. 4).

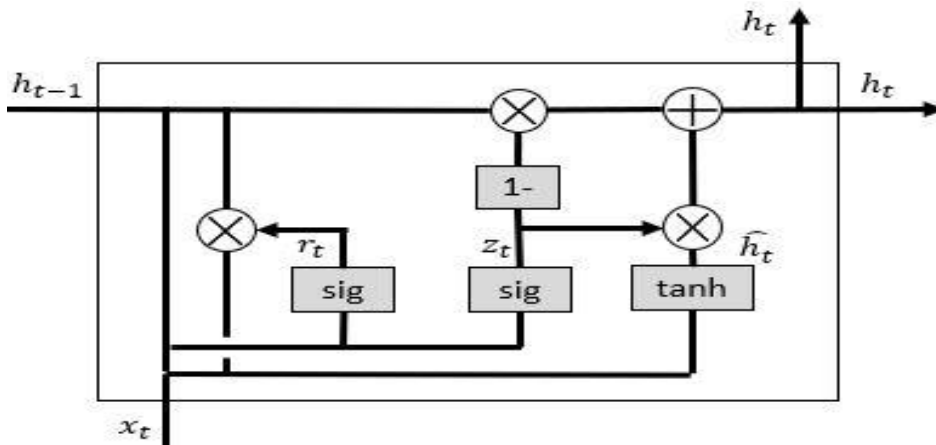


Figure no. 4. Gated recurrent unit high level architecture

However, the gating system of GRU consists of only two gates: an update gate and a reset gate. Another difference is represented by the cell state, which is not part of GRU. Both the reset and update gate functionality are governed by the sigmoid activation functions. In this way, only relevant data is kept. The reset gate has a similar functionality as LSTM's forget gate, deciding what information from the previous hidden state is to be discarded.

$$r_t = \sigma(x_t U^r + h_{t-1} W^r + b^r) \quad (12)$$

The update gate works in the same way as LSTM's input gate, filtering the information coming from the previous state and the information of the current input, selecting the new information to be added.

$$z_t = \sigma(x_t U^z + h_{t-1} W^z + b^z) \quad (13)$$

Where r (equation 12) and z (equation 13) indicate that the weights are proper for the reset gate and the update gate. If the values of the update gate are close to one, then the information of the old state is kept, while the current state input is ignored. The reset gate ensures the short-term dependencies are captured, while the update gate does the same but for long-term dependencies. The new hidden state candidate is governed by the following equation:

$$\hat{h}_t = \tanh(x_t U^{\hat{h}} + (h_{t-1} \cdot r_t) W^{\hat{h}} + b^{\hat{h}}) \quad (14)$$

After assimilating the effect of the reset gate into the new hidden state candidate, then the impact of the update gate output is incorporated into the current hidden state:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h}_t \quad (15)$$

In this way, GRU manages to deal with short and long-term dependencies and gradient-related issues with less calculations, being less computationally expensive than the LSTM.

2.5. Convolutional Neural Networks

Convolutional Neural Networks (CNN) represent a branch of neural-based models invented for computer vision tasks, initially for handwritten digit recognition (Le Cun, et al., 1990). Modern CNNs started to be used on one-dimensional (1D CNN) sequential data of applications, like time series, text, or audio analysis (Zhang, et al., 2020). CNNs might result in more computationally efficient architectures due to the ease of computation parallelization across graphical power unit cores and fewer parameters, compared to fully-connected architectures.

A typical 1D CNN configuration for a time series prediction problem consists of: input data, a convolutional layer, a pooling layer, a concatenation layer, a dense layer and an output layer (figure no. 5). Convolutional layer processes the input data and learn to extract the features appropriate for the regression made by the dense layer (Abdeljaber, et al., 2017).

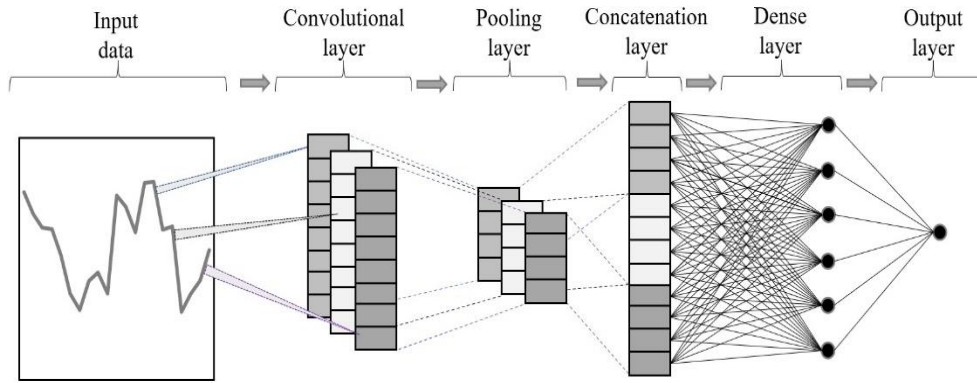


Figure no. 5. One-dimensional convolutional neural network (1D CNN) high level architecture

The forward propagation in a 1D CNN layer uses the following equation (Kiranyaz, et al., 2020):

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} \text{conv1D}(w_{ik}^{l-1}, s_i^{l-1}) \tag{16}$$

Where b_k^l = bias of the k^{th} neuron at layer l ; w_{ik}^{l-1} = kernel from the i^{th} neuron at layer $l-1$ to the k^{th} neuron at layer l ; s_i^{l-1} = output of the i^{th} neuron at layer $l-1$; x_k^l = input. By passing through the activation function, the input x_k^l , the intermediate output y_k^l can be obtained.

$$y_k^l = f(x_k^l) \tag{17}$$

As a step-by-step approach, first a kernel must be selected, which is then translated along the time series, one step at a time. For each step, the dot product of the kernel is calculated and fitted to the time series. The convolution is represented by the resulting sequence of dot products between the kernel and time series (End to End Machine Learning School, 2020). The pooling layer functions to reduce the number of trainable parameters and to retain the information. Flattening the pooling layer output results in a one-dimensional array, which has the right shape for using in the fully connected layer, which is essentially a feed forward neural network.

3. Methodology

Between 2005 and 2017 the amount of wind energy produced by the European Union increased by 414%. Romania is one country which has reached the European Commission target for renewable energy for 2020 (European Court of Auditors, 2019), having a total wind power capacity of 3040 MW (Sava, 2020). In 2018, 42% of Romania’s energy consumption was represented by renewable energy, while the average in the European Union is 32%. Of this, 15% of Romania’s renewable energy is created by wind energy (Botea, 2020). This

impressive increase has resulted in growing complexity of the tasks needed to ensure optimal and safe operation of wind energy, adding uncertainty to the power systems. The uncertainty and volatility have a significant impact, since the contribution of wind power to the total power is increasing. Wind turbine operators need to deal with these previously mentioned challenges, without impacting the grid or creating disruptions to the customers. IoT technologies can create mechanisms for automatic monitoring and controlling systems, enabling the operator to implement strategies with respect to these challenges. Ultimately, the wind energy prediction results are used to support production, in order to deliver energy without suffering gaps in supply.

A high-level view of a typical IIoT architecture consists of three major layers: the connect layer, the acquisition layer and the analytics layer (figure no. 6).

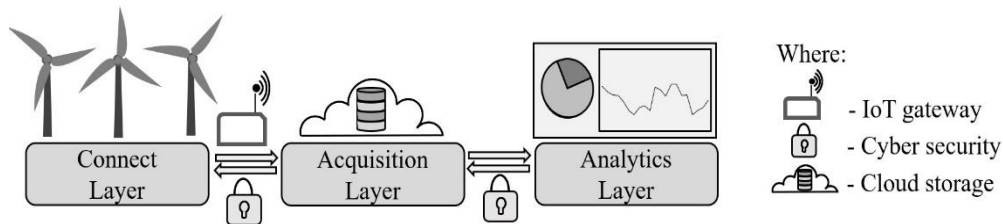


Figure no. 6. High-level overview of the proposed IIoT architecture

The connections of these layers are bidirectional (Krishna, et al., 2018) and compliant with cyber security protocols. The connect layer consists of physical devices, such as sensors for monitoring and actuators for control. Besides sensors to record data, these devices must be able to connect to the IT infrastructure for sending and receiving data. Additionally, data standardization and transformation can be made. The connection to the internet is done using an IoT gateway. This device has bigger computational capabilities compared to the existing devices on the connect layer and is capable of aggregating data from multiple sensors and sending it in the cloud. As part of the acquisition layer, cloud hosted servers host the data received from IoT gateways. Cloud storage has advantages, such as scalability, usability, accessibility, security, cost-efficiency, and automation (Singh, 2020). In the analytics layer, the real-world operational data is transformed into actionable insights for managing and improving business operations. The analytics layer can contain various types of data analytics (Hanski, et al., 2018): this study focuses on the descriptive and predictive analytics. On top of the analytics layer, the Cross-Industry Standard for Data Mining (CRISP-DM) is implemented (IBM, 2019). This robust methodology gives a structured approach for data mining projects. The step-by-step guidance describes the main tasks to be completed during each phase and the interactions between these phases. CRISP-DM is essential in real-world business cases and consists of the following six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. The first phase, business understanding, has been covered extensively in the previous sections. The second phase of CRISP-DM is represented by data understanding. The data was made available by Open Power System Data (Neon Neue Energieökonomik, 2020). The wind energy production time series is aggregated by hour and consists of 15336 data records between 01/01/2019 and 30/09/2020 (table no. 1). Open Power System Data is a platform for data gathered from the European power system, with freely available data for researchers.

Table no. 1. Wind energy production time series description

	count	mean	std	min	25%	50%	75%	max
Wind energy production	15336	774.03	665.99	0	237	573	1165.5	2809

Note: Count = total number of data points; std = standard deviation; min = minimum value; 25% = first quartile; 50% = median value; 75% = third quartile; max = maximum value.

Figure no. 7 shows the average wind energy production, plotted by day. This data confirms the intermittent nature of wind energy.

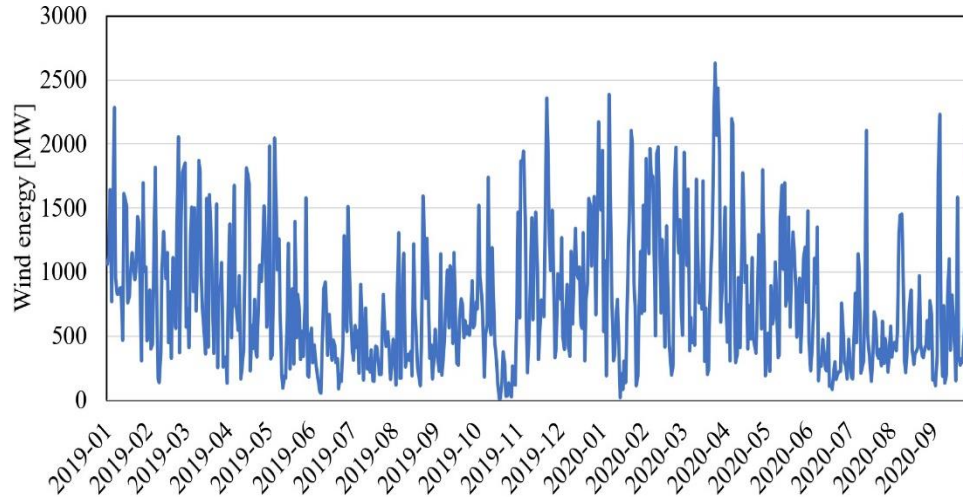


Figure no. 7. Wind energy production daily average

For this time series there are no quality issues, such as missing data, outliers or differences in data format. In the modeling phase, five techniques were considered: FFNN, RNN, LSTM, GRU and CNN, described in detail in Section 2. Selected metrics for the evaluation phase enable an objective comparison. The three metrics selected are: coefficient of determination, mean absolute error (MAE) and training time. The coefficient of determination (R^2) measures how much of the variability in the outcome can be explained by the model (equation 18). This coefficient is bound between 0 and 1. The closer to 1, the better the model is in explaining the variability in the outcome. The MAE averages the absolute values of the prediction errors (equation 19). This mean does not take into account if the error value is negative or positive.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)}{\sum_{i=1}^n (y_i - \hat{y}_i)^2} \tag{18}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \tag{19}$$

Where y_i is the actual value and \hat{y}_i is the predicted value. Training time assesses the resources spent on training and if the model training can happen within consecutive timestamps. The sixth phase of CRISP-DM, deployment, has not been covered in this research. The entire experiment happened offline.

All the proposed algorithms were implemented in Python, using Keras (Chollet, 2017) with TensorFlow on the backend. The hardware setup consisted of a Dell Precision 7350 equipped with a Nvidia Quadro P2000 GPU, 32 GB RAM, Intel Core i5-8400H @ 2.5 GHz CPU, Windows 10, and Python 3.6.10. The learning process used a supervised paradigm. Pairs of input-output training data subset records were fed into the models during the training process. The models then learned to generalize based on this training data. Testing happened using unseen data – i.e., a test data subset that was not part of the training. During the testing phase, the performance of the models was assessed (Fawcett and Provost, 2013).

4. Results and discussion

For the selected models, the final performance and training times represented the average of three trials, with different ratios between training and testing data subsets (table no. 2 and figures no. 8-10). This ensured meaningful results, unrelated to the data split.

Table no. 2. Parameter selection

Model	Parameters	Results		
		R^2	MAE	Training Time[s]
FFNN	batch=8, e=197, nn=47, hl=2, opt=Adam, lr=0.001, wi=Xavier Uniform, activation=RELU	0.963	82.3	350.7
RNN	batch=128, e=20, nn=128, hl=2, opt=Adam, lr=0.001, wi=Xavier uniform, activation=RELU	0.971	70	93.3
LSTM	batch=128, e=29, nn=128, hl=2, opt=Adam, lr=0.001, wi=Xavier uniform, activation=RELU	0.972	69.9	701.8
GRU	batch=128, e=25, nn=128, hl=2, opt= Adam, lr=0.001, wi=Xavier uniform, activation=RELU	0.971	70	427.5
1D CNN	batch=32, e=12, nndl=64, hndl=1, filters=64, kernel=3, pool=2, opt=Adam, lr=0.001, wi=Xavier uniform, activation=RELU	0.961	82.5	17.6

Note: FFNN = feedforward neural network, RNN = recurrent neural network, LSTM = long short-term memory, GRU = gated recurrent unit, 1D CNN = one dimensional convolutional neural network, batch = batch size, e = number of epochs, nn = number of neurons on hidden layer, hl = number of hidden layers, opt = optimization algorithm, lr = learning rate, wi = weights initialization, activation = activation function, nndl = number of neurons dense layer, hndl = number of hidden dense layers, filters = number of filters, kernel = kernel size, pool = max pooling size.

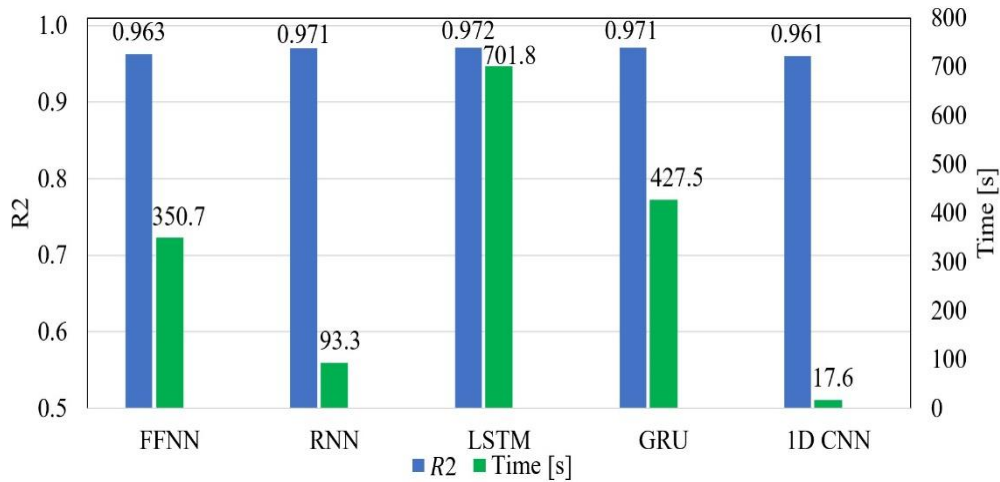


Figure no. 8. Comparison of coefficient of determination and training time

Note: FFNN = feedforward neural network, RNN = recurrent neural network, LSTM = long short-term memory, GRU = gated recurrent unit, 1D CNN = one dimensional convolutional neural network, R2 = coefficient of determination.

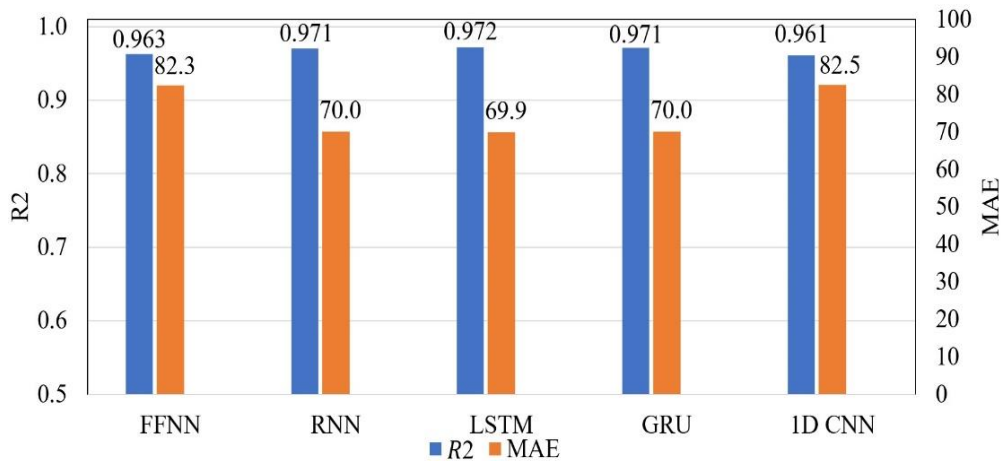


Figure no. 9. Comparison of coefficient of determination and mean absolute error

Note: FFNN = feedforward neural network, RNN = recurrent neural network, LSTM = long short-term memory, GRU = gated recurrent unit, 1D CNN = one dimensional convolutional neural network, R2 = coefficient of determination, MAE = mean absolute error.

Due to the highly configurable character of network-based models and their outstanding capability to learn and generalize, the results can be considered meaningful and reliable. There are no significant differences between RNN, LSTM and GRU, in terms of R^2 and MAE. However, the LSTM training time is expected to be the longest, due to its complex

mechanisms for capturing short – and long-term dependencies. Among the selected models, LSTM is the most computationally expensive. FFNN can be considered the least sophisticated model, but it provides a reasonable R^2 in half the training time of the LSTM. According to the theoretical considerations, 1D CNN was the least expensive, in terms of computational power. Even if R^2 results of the considered models appear similar, MAE provides another perspective (figure no. 9): for a real-world business case the increase of MAE, even the training time takes just few seconds, might not be feasible.

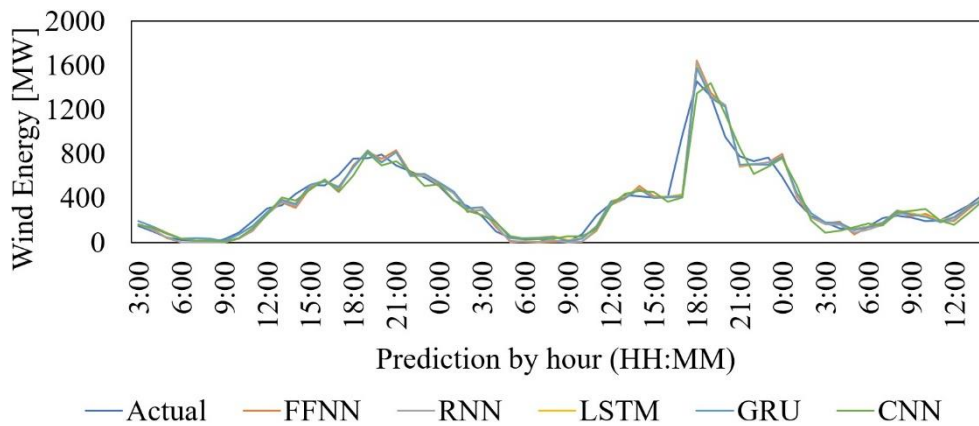


Figure no. 10. Wind energy prediction by hour

Note: FFNN = feedforward neural network, RNN = recurrent neural network, LSTM = long short-term memory, GRU = gated recurrent unit, CNN = convolutional neural network.

Conclusions

The energy industry must keep pace with the other industries that it is setting in motion. Energy demand, and the way customers use it, is becoming more complex. The pattern of consumption is highly influenced by the devices used, while the distributed character of the power systems is influenced by household or small business renewable energy systems. The key to safe and optimal operation of power grids relies on predictability across the entire energy supply chain. IoT technologies can act as enablers, gathering together all the “things” to create synergy. Geographically dispersed assets can, with the help of IIoT technologies, be managed more efficiently, preventing and minimizing the costly downtimes, while maximizing the output and reducing the negative impact of production volatility. Through IIoT, multiple energy resources, such as solar, geothermal, hydro and biomass, can be integrated and managed. As well as the IIoT, the CIoT is crucial for providing an end-to-end solution. The predictive analysis results from this case study show that LSTM produced the best prediction. Even with the long training time, the impact of such a prediction in a business will create the premises for a proper hardware upgrade to reduce the training time. This will allow the prediction to be made within consecutive data points for a more discrete time scale. MAE gives a measure of the performance in relation to an absolute value, namely the energy produced. In the case of RNN, LSTM, and GRU, the MAE values are differentiated at the decimal level and even if the unit of measurement is MW, at the time of deployment in production other elements must be considered. For a fully informed deployment, it is

important that the three models are used in parallel to determine the best solution. The complexity of the tasks for identifying the parameters and the stability of the models or the frequency with which the training must be repeated are decisive in selecting the best solution. The values of R^2 are close between the selected models and are also close to the value 1, meaning that the models can, to a large extent, explain the variability in the results. Comparing the results with those obtained by other researchers was not possible due to the way the metrics are typically selected; specifically, the metrics allow a comparison of models that are trained using the same data set, but do not allow a comparison of models trained on different data sets. In this regard, the use of the coefficient of determination, the complete description of the models' parameters, and the software and hardware configuration will allow other researchers to use this article for comparative studies. The performance of the models might be increased by adding exogenous variables, such as wind speed, wind shear, ambient temperature and pressure, dew point temperature and humidity.

References

- Abdeljaber, O., Avci, O., Kiranyaz, S., Boashash, B., Sodano, H. and Inman, D., 2017. 1-D CNNs for Structural Damage Detection: Verification on a Structural Health Monitoring Benchmark Data. *Neurocomputing*, [e-journal] 275, pp.1308-1317. 10.1016/j.neucom.2017.09.069.
- Al-Ali, A.R., 2016. Internet of Things Role in the Renewable Energy Resources. *Energy Procedia*, [e-journal] 100, pp.34-38. 10.1016/j.egypro.2016.10.144.
- Ashton, K., 2009. That "Internet of Things" Thing: In the Real World Things Matter More than Ideas. *RFID Journal*, 22(7), pp.97-114.
- Bengio, Y., Simard, P. and Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, [e-journal] 5(2), pp.157-166. 10.1109/72.279181.
- Botea, R., 2020. *Energiile regenerabile au acoperit 42% din consumul de energie al României, cu 10 puncte procentuale peste media europeană*. [online] Available at: <<https://www.zf.ro/eveniment/energiile-regenerabile-au-acoperit-42-din-consumul-de-energie-al-romaniei-cu-10-puncte-procentuale-peste-media-europeana-18764797>> [Accessed 19 October 2020].
- Chen, L. and Lai, X., 2011. Comparison between ARIMA and ANN models used in short-term wind speed forecasting. In: *IEEE, 2011 Asia-Pacific Power and Energy Engineering Conference*. Wuhan, China, 25-28 March 2011. S.l.: IEEE. 10.1109/APPEEC.2011.5748446.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In: s.n., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, 25-29 October 2014. S.l.: s.n. 10.3115/v1/D14-1179.
- Chollet, F., 2017. *Deep Learning with Python*. Greenwich: Manning Publications.
- Ding, M., Zhou, H., Xie, H., Wu, M., Nakanishi, Y. and Yokoyama, R., 2019. A gated recurrent unit neural networks based wind speed error correction model for short-term wind power forecasting. *Neurocomputing*, [e-journal] 365, pp.54-61. 10.1016/j.neucom.2019.07.058.

- Eldali, F., Hansen, T., Suryanarayanan, S. and Chong, E., 2016. Employing ARIMA models to improve wind power forecasts: A case study in ERCOT. In: *IEEE, 2016 North American Power Symposium (NAPS)*. Denver, CO, 18-20 September 2016. S.l: IEEE. 10.1109/NAPS.2016.7747861.
- End to End Machine Learning School, 2020. *Convolution in one dimension for neural networks*. [online] Available at: <https://e2eml.school/convolution_one_d.html> [Accessed 20 December 2020].
- European Commission, 2020. *EU climate policies and the European Green Pact*. [online] Available at: <https://ec.europa.eu/clima/policies/eu-climate-action_ro> [Accessed 20 October 2020].
- European Court of Auditors, 2019. *Wind and solar energy for electricity generation: significant action is needed to achieve EU targets*. [online] Available at: <<https://op.europa.eu/webpub/eca/special-reports/wind-solar-power-generation-8-2019/ro/index.html#h2table5>> [Accessed 19 December 2020].
- Fawcett, T. and Provost, F., 2013. *Data Science for Business*. Newton: O'Reilly.
- Fukuoka, R., Suzuki, H., Kitajima, T., Kuwahara, A. and Yasuno, T., 2018. Wind Speed Prediction Model Using LSTM and 1D-CNN. *Journal of Signal Processing*, [e-journal] 22(4), pp.207-210. 10.2299/jsp.22.207.
- Hanski, J., Uusitalo, T., Vainio, H., Kunttu, S., Valkokari, P., Kortelainen, H. and Koskinen, K., 2018. *Smart asset management as a service Deliverable 2.0*. [pdf] Available at: <<http://doi.org/10.13140/RG.2.2.31027.94244>> [Accessed 19 October 2020].
- Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, [e-journal] 9(8), pp.1735-1780. 10.1162/neco.1997.9.8.1735.
- IBM, 2019. *IBM SPSS Modeler CRISP-DM Guide: CRISP-DM Help Overview*. [online] Available at: <https://www.ibm.com/support/knowledgecenter/SS3RA7_sub/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html> [Accessed 19 October 2020].
- Kingma, D. and Ba, J., 2015. Adam: A Method for Stochastic Optimization, In: S.n., *3rd International Conference for Learning Representations*. San Diego, CA, 7-9 May 2015. S.l.: s.n.
- Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. and Inman, D.J., 2020. 1D Convolutional Neural Networks and Applications: A Survey. *Mechanical Systems and Signal Processing*, [e-journal] 151. 10.1016/j.ymssp.2020.107398.
- Krishna, P.G., Ravi, K.S., Kishore, K.H., Veni, K.K., Rao, K.N.S. and Prasad, R.D., 2018. Design and development of bi-directional IoT gateway using ZigBee and Wi-Fi technologies with MQTT protocol. *International Journal of Engineering & Technology*, [e-journal] 7(2.8), pp.125-129. 10.14419/ijet.v7i2.8.10344.
- Le Cun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1990. Handwritten Digit Recognition with a Back-Propagation Network. In: D. Touretzky, *Advances in Neural Information Processing Systems (NIPS 1989)*. Denver, CO, 27-30 November 1989. Burlington: Morgan Kaufmann.
- Liu, Y., Ding, S. and Jia, W., 2020. A novel prediction method of complex univariate time series based on k-means clustering. *Soft Computing*, [e-journal] 24. 10.1007/s00500-020-04952-2.

- McCulloch, W.S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, pp.115-133.
- Neon Neue Energieökonomik, 2020. *Open Power Systems Data: Load, wind and solar, prices in hourly resolution*. [online] Available at: <https://data.open-power-system-data.org/time_series/2020-10-06> [Accessed 19 October 2020].
- Pant, P. and Garg, A., 2016. Forecasting of Short Term Wind Power Using ARIMA Method. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 4(3), pp.356-361.
- Pascanu, R., Gulcehre, C., Cho, K. and Bengio, Y., 2014. How to Construct Deep Recurrent Neural Networks. In: *ICLR, 2nd International Conference on Learning Representations*. Banff, Canada, 14-16 April 2014. S.l.: s.n.
- Rumelhart, D., Hinton, G. E. and Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature*, [e-journal] 323, pp.533-536. 10.1038/323533a0.
- Sava, J.A., 2020. *Onshore wind energy capacity in Romania 2008-2019*. [online] Available at: <<https://www.statista.com/statistics/870766/onshore-wind-energy-capacity-in-romania/>> [Accessed 21 December 2020].
- Singh, V., 2020. *10 Benefits of Using Cloud Storage*. [online] Available at: <<https://cloudacademy.com/blog/10-benefits-of-using-cloud-storage/>> [Accessed 19 October 2020].
- Yun, M. and Yuxin B., 2010. Research on the architecture and key technology of Internet of Things (IoT) applied on the smart grid. In: *IEEE. 2010 International Conference on Advances in Energy Engineering*. Beijing, China, 19-20 June 2010. S.l.: s.n. 10.1109/ICAEE.2010.5557611.
- Wang, J. and Hu, J., 2015. A robust combination approach for short-term wind speed forecasting and analysis – Combination of the ARIMA (Autoregressive Integrated Moving Average), ELM (Extreme Learning Machine), SVM (Support Vector Machine) and LSSVM (Least Square SVM) forecasts using a GPR (Gaussian Process Regression) model. *Energy*, [e-journal] 93(P1), pp.41-56. 10.1016/j.energy.2015.08.045.
- Zhang, A., Lipton Z.C., Li, M. and Smola, A.J., 2020. *Dive into Deep Learning*. [online] Available at: <https://d2l.ai/chapter_convolutional-neural-networks/index.html> [Accessed 20 December 2020].