Deep Learning Applied to Wind Power Forecasting: a Spatio-Temporal Approach

Rubén del Campo[0000−0003−2981−7439], Eloy Anguiano[0000−0001−8516−5629], Álvaro Romero[0000−0002−6087−3917], and José R. Dorronsoro[0000−0002−5271−0616]

Instituto de Ingeniería del Conocimiento,
Francisco Tomás y Valiente, 11 Cantoblanco-UAM. EPS Edificio B pt 5, 28049 Madrid, Spain
{ruben.delcampo,eloy.anguiano,alvaro.romero,jose.dorronsoro}@iic.uam.es
https://www.iic.uam.es/

Abstract. The problem of wind energy production prediction has been one of the most prolific topics of study in the field of machine learning applied to the energy sector. Usually, these models receive data in tabular format. However, in this work we propose to solve the problem of predicting wind power like a spatio-temporal prediction problem as if it were an image or video analysis problem. On the one hand, energy production and the weather variables provided by Numerical Weather Prediction models (NWP) are time series, justifying the temporal treatment. On the other hand, NWP variables are provided in a regular grid format (in terms of latitude and longitude). Thus, the data are arranged as different meteorological variables in the shape of a grid, justifying the spatial treatment as if it were a low-resolution image, where the meteorological points are treated as pixels. For this reason, the goal of this article is to carry out an initial benchmark that compares the performance measured between different types of deep learning architectures that take advantage of these temporal and spatial features. The proposed architectures are: CNN, LSTM, LSTM+CNN (Stacked), LSTM+CNN (Parallel), ConvLSTM and Vision Transformer.

Keywords: Wind, Power, Forecasting, Time Series, NWP, Deep Learning, CNN, RNN, LSTM, ConvLSTM, ViT, Transformer

1 Introduction

In recent years, the development of the electricity system towards a production system based on renewable energies has enabled society to move towards a more efficient and ecologically sustainable way of life. However, the generation of some of these energies cannot be controlled by humans, as their performance is inevitably linked to weather conditions. Both generating agents and the electricity system operators have always shown great interest in obtaining the best possible information on the production capacity that a station will be able to generate in consecutive hours or days. Being able to make this prediction reliably

means great benefits, not only from an economic point of view, but also allows safeguarding the structural integrity of the entire transmission grid.

Given the great importance that this problem has gained in view of the large percentage that renewables already occupy in the energy mix, in recent decades a considerable amount of research has been carried out in the field of machine learning. However, classical machine learning techniques seem to have reached their maximum capacity for improvement with regard to the problem at hand. Thanks to the advances made both in the field of GPU computing, which make it possible to execute complex algorithms in reasonable times, and in the field of artificial intelligence and machine learning with the implementation of new, more efficient optimization algorithms, the state of the art has evolved into what is known as deep learning. However, all these techniques have in common that they use input data in tabular form.

The objective of this work is to review several of the techniques that make up the state of the art of deep learning, applying them to the problem of predicting wind power production from a spatio-temporal point of view. Many of the implementations presented throughout this paper have in common the attempt to take advantage of the geospatial component of weather forecast data and also from the temporal information provided by both the production series and weather forecasts.

2 State of Art

2.1 CNN

Inspired by human visual perception, CNNs [12] apply a series of filters where a kernel is slid through the bidimensional input image, thus extracting the most important features of the image. The fundamental operation here is the convolution which consists in applying to the input image a filter of smaller width and length and equal depth to that image, performing an element by element multiplication of each matrix and adding all the results so that we obtain a single value. In this way, it continues sliding in a sequential way going through all the input pixels several times, resulting in a new coded image with a new dimensionality.

Convolutional layers are often stacked multiple consecutive filters, so the dimensionality of the data increases dramatically. For this reason, it is very common to alternate convolutional layers with pooling layers. These layers are not trainable, but their only function is to compress the information always trying to preserve the most important features found by the filters of the convolutional layers.

2.2 RNN/LSTM

Deep neural networks have demonstrated a great ability to find complex relationships among variables. However, when we confront them with problems that

have temporal information and future instants are related to past instants, we need a type of architecture that takes this feature into account. Recurrent networks (RNN) [13] were proposed with the aim of being able to encode this type of dependencies. In this type of network each past input x_{t-1} is used in the next iteration when x_t arrives. The information propagated from previous states at time moment t is known as *hidden state*.

In [15], *bidirectional* recurrent networks were proposed. When they process time series data, the prediction is not made solely on the basis of the past instants following the time order, since in some problems the sequences are not always ordered in time. For example, in our particular case it is quite common for meteorology to have a certain time shift due to the complication of generating such forecasts. The units of this type of network are divided into two parts: one that processes the inputs in the same direction of time t and the other in the reverse direction.

Despite the great popularity of recurrent networks for solving time-dependent problems, they have proven to have major limitations when taking into account dependencies that extend widely in time. *Gradient vanishing* [8] occurs when taking the *backpropagation* so many steps back in time. For this reason, in [9] the LSTM architecture was proposed as a solution to this problem. The main idea behind this solution lies in the use of a memory that is propagated during different time instants [17]. The main parts of the LSTM are:

- *Cell state* is responsible for propagating the information passed along the following time instants. We can understand that it gives the LSTM a kind of memory capacity. The information flow of the cell state is regulated by the different sigmoid gates that limit the information that passes through the different time instants.
- *Forget gate* has the function of establishing how much information from the input and the hidden state reaches the cell state. state reaches the cell state.
- *Input gate* is in charge of deciding how much information from the cell output has to be propagated to the cell state, i.e., how much new information must be taken into account.
- *Output gate* is in charge of establishing the relevance of the new information calculated to be passed as hidden state to the next time instant.

2.3 Transformers

Transformers have become the state of the art of Natural Language Processing. This new architecture is used in this type of problems due to what is called self-attention [19]. This self-attention strategy is able to extract contextual information between the different points of a sequence of information, which fits very directly with text-related problems, since semantic and grammatical relations between words can be captured.

However, there are also adaptations of this architecture that have improved the state of the art in image classification problems, called *Vision Transformers* (ViT) [6]. In this approach, the image is divided into different groups of pixels as

if it were a vector to be introduced into a *Transformer Encoder* with a fictitious parameterized vector added (usually known as CLS Token), so that the output of the encoding of that vector serves as input to the prediction of a series of fully connected layers. This trick of the CLS parameter is commonly used in document classification problems, since it allows us to summarize all the contextual information of the sequence (in this case the image) in a single vector that can be processed later.

Therefore, for our particular use case, we will use this approach as one of the models to consider for that spatial, but not temporal, approach to solving the problem in question.

2.4 Metrics

In this research, a wide variety of neural network architectures have been developed and therefore, each of them has an associated prediction error. This error must be quantifiable and ideally interpretable so that we can compare networks with each other and establish which one is better. Within the field of knowledge of machine learning, this problem falls into the category of *regression*, which is made up of those problems in which the objective is to achieve a single continuous numerical result. Classically, the metrics used in this type of problem are are given in absolute terms. However, in the case of our problem, it is obvious that for a wind farm with a high maximum productive capacity, called *installed power*, these measurements will be much higher than for other farms with a lower installed power. It is for this reason that the previous measurements are always normalized by the installed power of the wind farm. This feature make this type of metrics make the *target* insensitive to changes in installed power over time.

- Normalized Mean Absolute Error:

$$NMAE(Y, \hat{Y}) = \frac{1}{n} \sum \frac{|y - \hat{y}|}{InstalledPower} \tag{1}$$

- Normalized Mean Squared Error:

$$NMSE(Y, \hat{Y}) = \frac{1}{n} \sum (\frac{y - \hat{y}}{InstalledPower})^2 \tag{2}$$

where $y \in Y$ are the observations given and $\hat{y} \in \hat{Y}$ are the predictions made by the model for which its performance is to be measured.

On the other hand, in the search for measures that are independent of the data domain, we will use measures such as:

- Coefficient of determination (R^2 *Score*):

$$R^2(Y, \hat{Y}) = 1 - \frac{\sum (y - \hat{y})^2}{\sum (y - \bar{y})^2} \tag{3}$$

where $\bar{y}$ is the mean of all observations.

- Explained Variance:

$$Exp\_Var(Y, \hat{Y}) = 1 - \frac{Var[y - \hat{y}]}{Var[y]} \tag{4}$$

2.5 Related Work

The problem of renewable production prediction has been widely studied in the field of machine learning. All kinds of papers have been published on the subject using a wide variety of techniques and algorithms.

In recent years, following the popularization of deep learning, we have started to see some research using this type of novel techniques. Focusing on publications that use data similar to those available for this work, we find some research such as [1], where they simply extend the idea of the multilayer perceptron by increasing the number of layers and neurons per layer. On the other hand, in [18] they continued with the idea by increasing the number of layers even more.

Architectures based on LSTM have also begun to be used in the renewable prediction problem, as they are particularly well suited to deal with time series. For example, in [7] they use a LSTM to predict the production of a photovoltaic farm using meteorological data.

One of the first publications that began to try to exploit the geospatial component of meteorological data was [5]. In it, a three-dimensional dataset is constructed where the first dimension is the meteorological variables, the second dimension is the latitude and the third is the longitude. It showed that this type of approach taking into account the location of each weather point brought new information to the model that the other traditional methods were unable to appreciate.

Extending this idea of encoding the input data by means of CNNs, some researches proposed that the output of the last convolutional, instead of attaching it to a fully-connected layer, should be attached to an LSTM layer such as [3] or [14]. We will call this type of architecture *CNN-LSTM Stacked*. Another approach of combining CNN with LSTM is found in [11], where instead of connecting the output of one network to the input of another, what they propose is that both process the same input in parallel, concatenating their output to pass it to a *fully-connected* layer. We will call this type of architecture *CNN-LSTM Parallel*.

Following the approach of trying to encode both temporal and geospatial information of meteorological data, the idea of ConvLSTM [16] arises. In essence, what is proposed is an architecture similar to LSTMs except that the input data, instead of being one-dimensional data on which tensor multiplication operations are applied, are two-dimensional data on which the convolution operation is applied using convolutional layers. That is the reason why they have been used in video analysis problems.

3 Methodology

3.1 Data sources

To solve the problem of renewable production forecasting we need two sources of data: productions and weather forecasts.

- Production data: We have used the series of actual productions measured in a wind farm located in Spain for every hour of the years 2019, 2020 and 2021.
- Meteorological data: Meteorological forecasts made with Numerical Weather Prediction models (NWP) have been obtained for 24 horizons corresponding to the years 2019, 2020 and 2021. For all these time instants, meteorological variables have been obtained forming a 9×9 regular grid around the location of the park. We have used the following variables: u and v wind components at 10m and 100m altitude with its module ($V10$ and $V100$), the temperature at 10m altitude (T) and the surface pressure (p). So we can understand this grid of points as if it were an "image" where each meteorological point would be a pixel and each variable would be a channel of it.

3.2 Exploratory Analysis

It has been previously emphasized throughout this work that one of the purposes of this work is to try to take advantage of the spatial characteristics of the weather forecast data. In order to justify this decision, an analysis has been carried out, which is shown in figure 1, where the Pearson correlation of each of the variables in the grid is measured with the wind production target. In this way, we can clearly see how, depending on the variable, there will be certain geographical areas that are much more relevant than others. The winds that have more impact on the target are those measured in points of the northwest-southeast diagonal. For this reason, neural network architectures that take into consideration this type of geospatial characteristics such as CNN or ViT seem more than adequate to solve this problem. In other variables such as pressure or temperature, although their correlation also occurs in specific areas, this value is very small. Nevertheless, it is convenient to keep in mind that the Pearson correlation only finds linear relationships, so we will not exclude these variables from the study.

3.3 Experimentation

In order to arrive at the neural network architectures that will be described later, we have carried out an exhaustive experimentation process. During this process, multiple variations on the networks have been tested, varying their structure (number of units, shape of the layers, lags passed, etc.), their hyperparameters (dropout, bidirectionality, hidden size, etc.) or even parameters affecting the training (learning rate, batch size, etc.).

4 Implemented Neural Architectures

4.1 LSTM

The first of the implemented architectures is the one based on LSTM networks, which, as previously explained, are especially appropriate for time series related

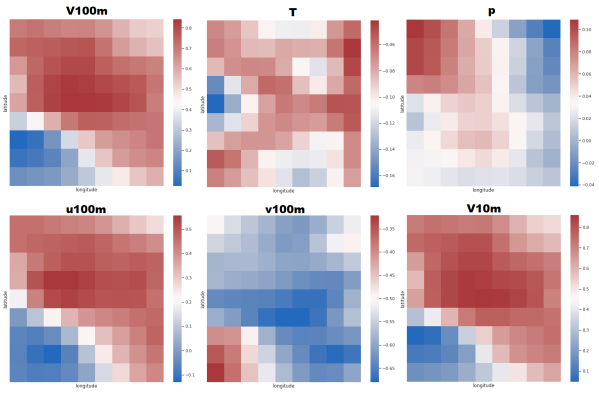

Fig. 1. Correlation of the meteorological variables with the wind farm production in each one of the coordinates.

problems. We note that this network simply receives the data records in one-dimensional form by applying time delays of the received data. Furthermore, they have been used in their bidirectional mode which is capable of analyzing the data sequence in both temporal directions.

In this problem where the input data are meteorological predictions with their associated error, bidirectionality can be an improvement since it can happen that the predictions received are correct in the magnitude of the meteorological variables, but these are wrongly ordered in time. The bidirectional network, by evaluating the sequence in both directions, can mitigate this effect that could worsen the training capacity of our network.

4.2 CNN

It has been justified during the previous sections that considering the spatial component in meteorological data by treating our data in a two-dimensional form can help our models to find relationships that would be difficult to appreciate using a tabular format. For this reason, it is proposed to use a CNN as shown in the CNN parts of the figure 2.

For this architecture we start changing input data shape to a three-dimensional representation (D variables $\times L$ latitudes $\times M$ longitudes). In the first block a convolutional layer is used applying *zero-padding* so that our network can take into consideration the information presented by the variables corresponding to the points located at the ends of the grid of points. Subsequently, an *average-pooling* layer is used to reduce the dimensionality of the input data. This is followed by a series of identical blocks consisting of two convolutional layers followed by batch normalization layers in charge of reducing the *internal covariate shift* and favoring a fast convergence during training [10]. Although the *batch normalization* layer may sometimes make it unnecessary to apply *dropout* layers [2], it has been seen that in this case they are convenient, since they help

to reduce the *overfitting* that can be caused in a complex network such as the one proposed. Finally, before passing the data through *fully-connected* layers, an *average-pooling* layer is applied, also with the aim of reducing the data dimension, which at that moment is very high after having applied a large number of convolutional filters. The activation function used is ELU [4], that has proved to be one of the most reliable activation functions in the state of the art.

4.3 Combinations

As discussed above, CNNs are particularly suitable for capturing the spatial information provided by the input data, while LSTMs are particularly suitable for capturing the temporal information of the input data. For this reason, it has been decided to apply combinations between the CNN and LSTM networks detailed in the previous subsections:

- CNN+LSTM Stacked: in this solution, the networks are connected in such a way that the output of the CNN is immediately directed to the input of the LSTM network, as can be seen in the first figure 2. It should be noted that the CNN output has to be flattened so that it can be treated by the LSTM input.
- CNN+LSTM Parallel: this solution is similar to the one previously explained except that, as shown in the second figure 2, in this case the input of the neural network is passed equally to the CNN and the LSTM, except that the former will receive the data in three-dimensional form and the latter will not. The output of both networks will be flattened and concatenated into a single one-dimensional vector.

4.4 ConvLSTM

The ConvLSTM presents an architecture that instead of treating the CNN and the LSTM as independent networks, merges the underlying idea of both of them trying to achieve a conceptually more complete architecture that by itself is able to process both spatial and temporal information of meteorological data.

We have used two ConvLSTM units separated by dropout and *batch normalization* layers that help to maintain a stable training avoiding *overfitting*. In the first unit the convolutional layers use a 5×5 filter, while in the second unit they use a 3×3. In this way, the former can focus on more general features of the input data while the latter can focus on more specific ones.

4.5 ViT

As mentioned above, the ViT model architecture groups chunks of pixels disjointly. For the current implementation we have grouped the pixels into three vectors to which a last parameterizable vector (called CLS) is added. To these four vectors we add a vector representing their positional encoding to move the

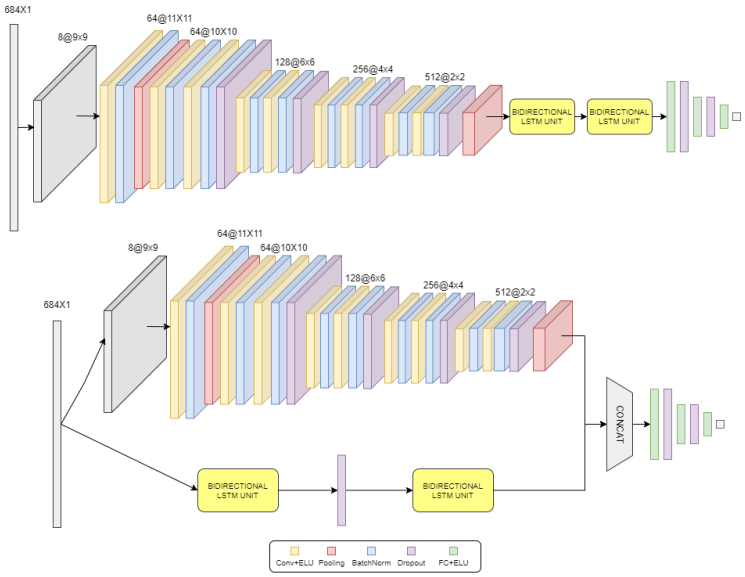

Fig. 2. Implemented CNN-LSTM Parallel and Stacked neural network architectures

projection to the latent space of the next dense layer of the network. After encoding this information through a Transformer Encoder, the output corresponding to the position of the CLS vector is used for a fully-connected network to give us the prediction value we are looking for.

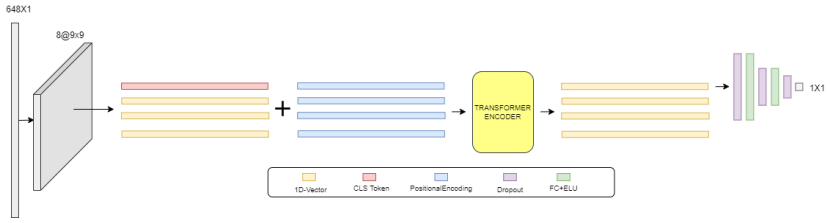

Fig. 3. Implemented ViT neural network architecture

5 Results

5.1 Benchmark Results

During this research we have carried out a multitude of executions with the aim of arriving at neural network architectures that offer the best performance

in accordance with the metrics we have previously explained. All of them have been trained and tested under the same conditions: we have used the 2019 and 2020 data for training and validation and the 2021 data for testing.

Table 1. Test metrics of every model averaged over the first 24 prediction horizons

	NMAE	**NMSE**	**R2**	**VarScore**
CNN	0.084	0.019	0.778	0.779
LSTM	**0.080**	**0.017**	**0.802**	**0.801**
CNN+LSTM Parallel	0.082	0.018	0.788	0.791
CNN+LSTM Stacked	0.082	0.018	0.795	0.796
ConvLSTM	0.081	0.018	0.788	0.789
ViT	**0.080**	**0.017**	**0.799**	**0.802**
Ridge	0.104	0.020	0.762	0.765
MLP	0.103	0.020	0.765	0.800
SVM	0.092	0.018	0.791	0.797

In the table 1 we can see the comparison among all the implemented architectures. We can see that the results obtained in the comparison do not differ much between the different proposed solutions. In spite of this the neural network architectures that have finally shown the best performance in all metrics are the LSTM and ViT. The LSTM, thanks to its bidirectional characteristic, is able to minimize the intrinsic temporal error in weather predictions. On the other hand, ViT also presents a high performance, so the hypothesis of applying in this problem networks that take into account spatial information is justified. However, although this problem has both a spatial and a temporal component, the combinations between CNNs (spatial) and LSTMs (temporal) do not end up being the best performers. This leaves the door open for further investigation of new combination techniques.

Nevertheless, it has been proven that the deep learning methods proposed during this research are a considerable improvement over other classical machine learning methods applied to regression problems, such as Ridge, MLP or SVM.

5.2 Degradation Study

Since the weather-based forecasting problem is based on source data that are also forecasts, models may have different performances at different forecast horizons. Therefore, comparing models solely by a metric averaged over such prediction horizons seems to us to lack all the analysis that a model's performance might receive, since one model might perform better for near prediction horizons and another might perform better at far horizons.

This is why this part of the study focuses on comparing the performance of the different models over the prediction horizons and the error distributions of the models.

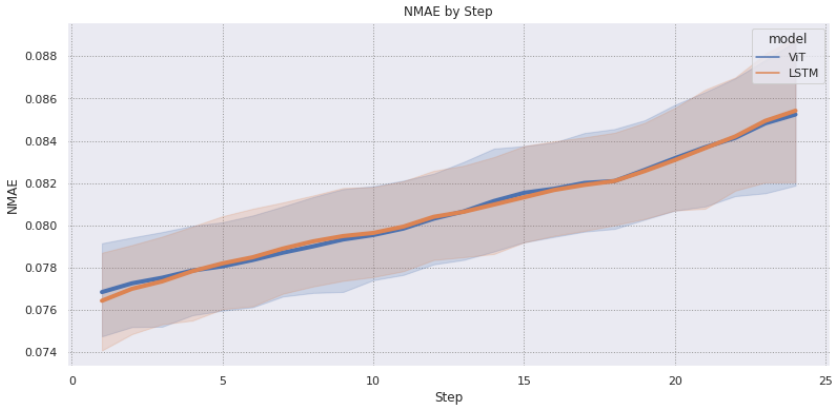

Fig. 4. Comparison of NMAE by prediction step between LSTM and ViT with its confidence interval

6 Conclusion

The prediction of wind energy production is a complex task, as it is influenced by meteorological and geographical factors. For this reason, the aim of this paper is to provide a new perspective on the problem by applying the latest Deep Learning techniques that have been classically used in other disciplines such as NLP or Computer Vision.

After an extensive research on the state-of-the-art techniques in neural network architectures applied to time series and spatial data together with their combinations and variants, we have proposed many architectures to solve the problem. These approaches have been: CNN, LSTM, LSTM+CNN (Stacked), LSTM+CNN (Parallel), ConvLSTM and Vision Transformer.

Despite the high computational cost of training this type of architectures based on deep neural networks, an exhaustive benchmark has been elaborated in order to compare, applying different metrics, the performance of the proposed architectures. All the implementations have obtained results of similar quality, although the LSTM and ViT have stood out, obtaining an NMAE of 0.080. Thus, with the former, the temporal approach to the problem has been justified, while with the latter, the spatial approach has been justified. Moreover, all the proposed solutions have been shown to be a clear improvement over classical machine learning models using tabular data.

References

1. Badrinath Krishna, V., Wadman, W., Kim, Y.: Nowcasting: Accurate and precise short-term wind power prediction using hyperlocal wind forecasts (2018)
2. Baldi, P., Sadowski, P.J.: Understanding dropout. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) Advances in Neural Information Processing Systems. vol. 26. Curran Associates, Inc. (2013)

3. Chen, Y., Wang, Y., Dong, Z., Su, J., Han, Z., Zhou, D., Zhao, Y., Bao, Y.: 2-d regional short-term wind speed forecast based on cnn-lstm deep learning model. Energy Conversion and Management **244**, 114451 (2021)
4. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus) (2015)
5. Díaz-Vico, D., Torres-Barrán, A., Omari, A., Dorronsoro, J.R.: Deep neural networks for wind and solar energy prediction. Neural Processing Letters **46**(3), 829–844 (2017)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. CoRR (2020)
7. Gao, M., Li, J., Hong, F., Long, D.: Short-term forecasting of power production in a large-scale photovoltaic plant based on lstm. Applied Sciences **9**, 3192 (2019)
8. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **6**, 107–116 (1998)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**, 1735–80 (1997)
10. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 448–456. PMLR, Lille, France (2015)
11. Karim, F., Majumdar, S., Darabi, H., Chen, S.: Lstm fully convolutional networks for time series classification. IEEE Access **6**, 1662–1669 (2018)
12. Lecun, Y., Haffner, P., Bengio, Y.: Object recognition with gradient-based learning (08 2000)
13. Lipton, Z.: A critical review of recurrent neural networks for sequence learning (2015)
14. Ren, J., Yu, Z., Gao, G., Yu, G., Yu, J.: A cnn-lstm-lightgbm based short-term wind power prediction method based on attention mechanism. Energy Reports **8**, 437–443 (2022), iCPE 2021-The 2nd International Conference on Power Engineering
15. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on **45**, 2673 – 2681 (12 1997)
16. SHI, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.k., WOO, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015)
17. Staudemeyer, R., Morris, E.: Understanding lstm – a tutorial into long short-term memory recurrent neural networks (2019)
18. Torres, J., Aguilar, R., Zúñiga, K.: Deep learning to predict the generation of a wind farm. Journal of Renewable and Sustainable Energy **10** (2018)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017)