Journal of Energy

VOLUME 70 Number 3 | 2021

journal homepage: http://journalofenergy.com/

Josip Đaković [1]
HEP – Distribution System Operator
Slavonski Brod, Croatia
josip.dakovic@hep.hr

Bojan Franc
University of Zagreb
Faculty of electrical engineering and computing
Zagreb, Croatia
bojan.franc@fer.hr

Igor Kuzle
University of Zagreb
Faculty of electrical engineering and computing
Zagreb, Croatia
Igor.kuzle@fer.hr

Yongqian Liu
North China Electric Power University
New Energy School
Beijing, China
yqliu@ncepu.edu.cn

[1] Statements expressed in the paper are author's own opinions, they are not binding for the company/institution in which author is employed nor they necessarily coincide with the official company/institution's positions.

Deep Neural Network Configuration Sensitivity Analysis in Wind Power Forecasting

SUMMARY

The trend toward increasing integration of wind farms into the power system is a challenge for transmission and distribution system operators and electricity market operators. The variability of electricity generation from wind farms increases the requirements for flexibility needed for the reliable and stable operation of the power system. Operating a power system with a high share of renewables requires advanced generation and consumption forecasting methods to ensure the reliable and economical operation of the system. Installed wind power capacities require advanced techniques to monitor and control such data-rich power systems. The rapid development of advanced artificial neural networks and data processing capabilities offers numerous potential applications. The effectiveness of advanced deep recurrent neural networks with long-term memory is constantly being demonstrated for learning complex temporal sequence-to-sequence dependencies. This paper presents the application of deep learning methods to wind power production forecasting. The models are trained using historical wind farm generation measurements and NWP weather forecasts for the areas of Croatian wind farms. Furthermore, a comparison of the accuracy of the proposed models with currently used forecasting tools is presented.

KEY WORDS

Wind power forecasting, deep learning, recurrent neural networks, LSTM, big data analytics, wind farms

INTRODUCTION

Energy generation from renewable energy sources (RES), of which a high proportion is wind farm (WF) generation, will have an increasingly important impact in achieving low-carbon development of the electric power system (EPS). Although the integration of wind farms brings many benefits from an environmental point of view, the unpredictable and variable nature of WF generation poses many challenges for EPS operators (ensuring adequate ancillary services, economic dispatching of power plants, dynamic stability of the system), electricity market operators, and electricity producers and traders. One of the possible solutions to the above challenges is the development of advanced tools and methods for reliable short-term forecasting of wind farm generation [1,2]. Wind power forecasting is becoming increasingly important in grid planning, optimization, and control as more and more energy is generated from inherently intermittent renewable sources. For practical purposes, the forecast time horizon can be divided into short-term (up to 12 hours ahead) and long-term (up to 72 hours ahead) forecasts [3,4]. Short-term forecasts can be used to regulate the system and operate the intraday electricity market, while long-term forecasts are often used to plan power plant dispatch and the day-ahead electricity market [5]. In recent decades, the amount of available information and computer power have grown very rapidly, so forecasting methods have evolved from simple statistical and physical models to much more complex statistical models, including the concepts of machine learning and, more recently, deep learning [6]. The aforementioned methods of Big Data analysis deal with huge amounts of complex data that are not suitable for processing with conventional algorithms. Methods based on a special type of recurrent neural network (RNN) with long short-term memory (LSTM) have proven highly successful in modeling long-term dependencies of meteorological variables and energy generation [7,8,9]. This is because LSTM-based networks are designed to learn dependencies among sequences of data. Weather forecast (Numerical weather prediction-NWP), as the most important input for wind power forecasting, provides time-labeled sequences of forecast data suitable for training recurrent networks. However, the accuracy of the LSTM method depends significantly on the network configuration and pre-training parameterization, which is specific to each type of application.

Fig. 1 shows the classification of commonly used approaches and methods for wind power predictions. Therefore, the methods used in this paper can be classified as data-driven deep-learning methods for short-term point-forecast of a wind power plant production. In addition to the deterministic approach, research is also being conducted on the probabilistic quantification of prognostic results, which aims to reduce the uncertainty of the forecast using confidence intervals [10].

Josip Đaković, Bojan Franc, Igor Kuzle, Yongqian Liu, Deep Neural Network Configuration Sensitivity Analysis in Wind Power Forecasting, Journal of Energy, vol. 70 Number 3 (2021), p. 19–24
https://doi.org/10.37798/202170389

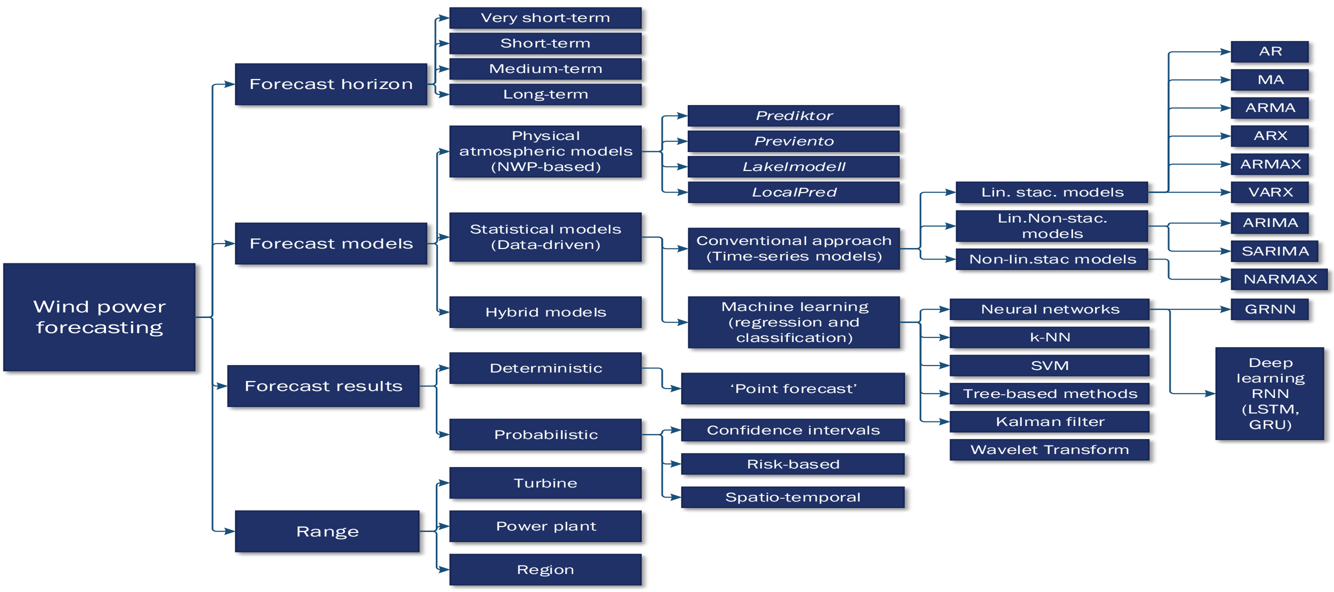


Figure 1: Wind power forecasting approaches [5]

RECURRENT NEURAL NETWORKS

The main feature of conventional neural networks, such as densely interconnected networks and convolutional networks, is that they have no internal memory. Each input is processed independently without determining or comparing conditions between inputs. To process a sequence or time series with these networks, it is necessary to represent the entire sequence at once: turn it into a single data point as the network input [11]. Such a neural network is called a *feedforward neural network* and is often used in load forecasting [12][13]. Unlike traditional neural networks (NN), recurrent neural networks process series (sequences) by iteratively traversing the elements of a sequence and retaining the states that contain information about the previous data. RNN is a type of neural network with an inner loop and memory. The internal state of the RNN is reset between the processing of two different independent sequences, so a sequence is still considered a data point: one input to the network. What changes is that the data point is no longer processed in one step; the iterates internally over the sequence elements. Simple (basic) recurrent networks face the problem of a vanishing gradient when training long sequences using deep networks (networks with multiple layers), which makes them practically useless. The solution to the above problem was proposed in 1997 (Hochreiter and Schmidhuber) in the form of networks with long-lasting short-term memory, but their practical application has been realised only in the last decade. Processing data in the LSTM layer is shown in Fig. 2 LSTM enables the data (hidden state of cell h_t) at any moment t of the input sequence (x_t) to be transferred into long-term memory (C_t) at a later moment in time and deleted from it, if necessary. Stated functions are realized with the help of special gate functions (f_t, i_t, o_t). In short, LSTM-based models learn relevant dependencies across the input sequence, avoiding the vanishing gradient problem during training.

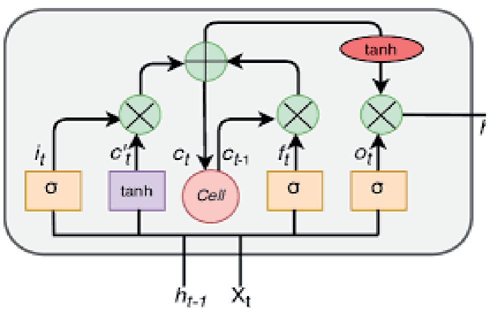


Figure 2: Data flow in LSTM cell in one step

2.1 Forecasting time sequences of wind farm generation

Sequence forecasting is different from other types of supervised machine learning in that it requires to maintain the temporal order of sequence values during model training and testing. Apart from numerical time sequences, sentences in text translation represent another type of commonly used sequential data for supervised machine learning. In the present case of wind power forecasting, time series of Croatian WFs power generation are converted into time sequences by segmenting continuous two-year time series into partially overlapping sequences.[1] For this reason, the paper deals with the time sequences of WF power output.

In general, forecasting problems with sequential data can be divided into four groups:

1. forecasting of the following value of the sequence;
2. sequence classification (forecast of the class according to the input sequence);
3. sequence generation (e.g., by generating text);
4. sequence-to-sequence prediction.

According to the form of available input data that can be used to forecast wind farm generation (sequential forecasts of atmospheric conditions from meteorological models) and obtaining historical power generation data from the SCADA system, a sequence-to-sequence problem can be formed: mapping sequences of (historical) weather forecasts (mostly wind speed and direction) to (historical) wind farm power generation sequences [14]. Fig. 3 shows the process of data preparation for training, validation, and testing of the models used in this work.

Input data, i.e., time series of historical measurements of realized production and historical NWP forecasts are 'cut' into 72-hour long sequences[2], aligned by timestamp, and merged in the form of 3D arrays (tensors) with dimensions: (sequence samples, horizon (72h), predictors). Mathematically speaking, the model of deep learning in this case is a composition of a matrix (tensor functions), which form is defined in advance using the so-called model layers. During the training process, the matrix weights of the predefined structure of the model are adjusted, to achieve an optimal mapping model from input predictors to output WPP production.

During the training and validation process, the available predictors (in this case wind speed and direction) are mapped to the actual power output of

1 The power generation sequences should be aligned according to time steps with overlapping weather forecasts, which in the observed case are generated every 6 hours for the following 72 hours, which means that there are several forecasts for the same moment.
2 In this paper, the sequence is considered as a 72-h long part of the time series of the realized power measurement or NWP forecast.

Josip Đaković, Bojan Franc, Igor Kuzle, Yongqian Liu, Deep Neural Network Configuration Sensitivity Analysis in Wind Power Forecasting, Journal of Energy, vol. 70 Number 3 (2021), p. 19–24
https://doi.org/10.37798/202170389

the WFs under consideration. The difference between training and validation is that validation is used only to monitor the accuracy of the model during the training process. After the model has been trained, new predictors data is inserted into the model for the testing process for which training has not yet been performed, so that the model produces a forecast based on a “learned” relationship between the weather forecasts and the corresponding power output of the wind farm.

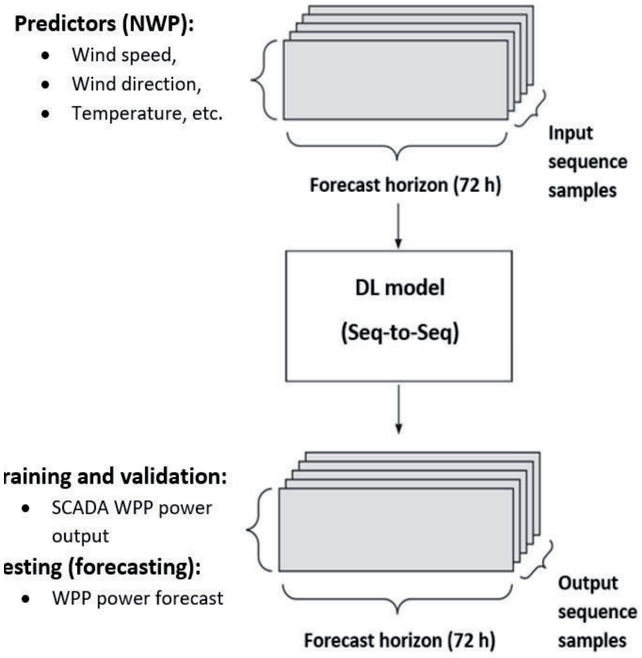


Figure 3: Modeling sequential forecasting model

2.2 Deep learning model for sequential data

The basic structural unit in a deep neural network is a layer. The layer is a module for data processing, which takes one or more tensors (data arrays) as input and returns one or more tensors as output. Some layers have no internal states (dense layers), but RNN-based models have layers with internal states and multiple weight matrices that contain the network’s ‘intelligence’. Sequential data is usually formed as 3D tensors with the following dimensions: (samples, time steps/horizon, predictors/feature) processed by recurrent layers such as LSTM and GRU (Gated Recurrent Unit). A deep learning model is built by “assembling” compatible layers in an appropriate configuration depending on the nature of the problem and the form of the input data. The layers of the model are usually arranged sequentially, meaning that the output of one layer represents the input to the next layer, but other topologies are also possible. In addition to the choice of network architecture, it is necessary to choose a loss function that is minimized during training and represents the accuracy measure between actual values and predictions. The value of a loss function, i.e., the error, is propagated through back-connections in layer weigh matrices (backpropagation through time) by using an optimization algorithm (e.g., Adam) based on stochastic gradient descent. For network training, the Adam optimizer was used because it is considered a very effective and fast training algorithm for deep neural networks. The learning is usually completed when the gradient values of all weight parameters are equal or close to zero. The process of model training is shown in Fig. 4. The choice of the appropriate loss function depends on the nature of the problem (regression, classification), and mean square error is most commonly used for the aforementioned type of sequential numerical data. Training a deep learning model requires considerable computational resources, which determine both the possible ‘depth’ of a model and the speed of training, which is usually performed by advanced graphics processing units [17].

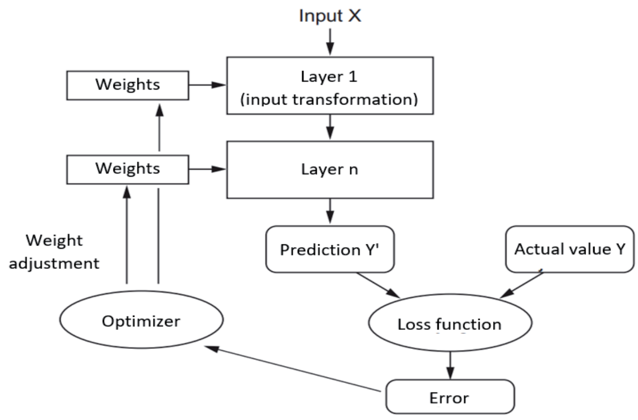


Figure 4: Process of training deep learning model

3. APPLICATION OF DEEP LEARNING TO FORECASTING WF GENERATION

Before the actual Deep Learning model is created, the input data must be prepared, which is usually not in a format suitable for model training. Preparation requires data clearing (e.g., removing unreal values, filling voids, etc.), data timestep alignment (e.g., reducing it to hourly values), and forming appropriate data tensors. In the following section, the process of model training and testing on real two-year data of a wind farm in Croatia is presented.

3.1 Wind farm data

Fig. 5 presents the two-year power generation data (from January 2018 to January 2020) of the considered WF in terms of measured wind speed and direction, i.e., the real wind power dependence curve of the considered WF. The wind rose (wind distribution by directional frequency) is shown in Fig. 6 with average wind speeds on the radial axis (e.g., in the interval from northwest to west wind (135°- 180°) the average wind speed is 7.72 m/s). Moreover, the average wind speed is proportional to the frequency of the wind direction, the predominant winds being bora and mistral. Fig. 7 shows the distribution of wind speed at the site with the marking of the average wind speed of 6 m/s (red line). Finally, Fig. 8 shows the correlation matrix of the measurement parameters in the SCADA system (wind speed and direction, operating power, temperature, and pressure).

It can be observed that wind speed has the highest correlation with power production (more than 0.8), while other meteorological parameters such as air direction, temperature, and pressure have a significantly lower correlation with power. The explanation for the low correlation between wind direction and production can be found in the problem of hourly averaging of wind direction (e.g., a wind whose direction oscillates between 0 and 360 degrees can lead to a mean value that suggests the opposite direction) and the possibility of modern WTG to rotate turbine nacelles in the optimal wind direction.

It is expected that wind speed has a positive correlation with the performance of WF power higher than 0.9, while wind direction, pressure, and temperature have a slightly negative correlation with the performance of the wind farm’s power.

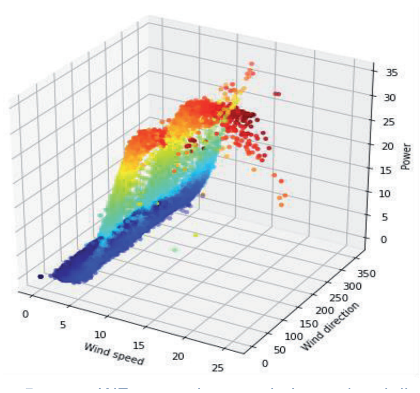


Figure 5: WF generation vs. wind speed and direction

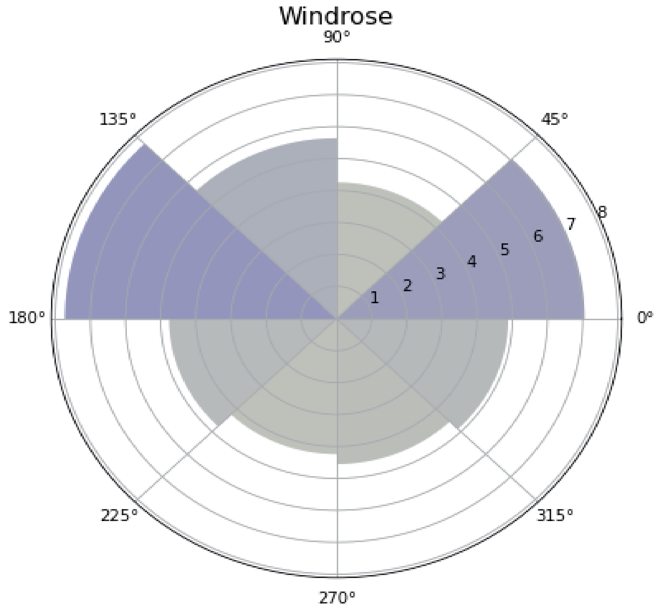


Figure 6: Wind rose with mean wind speed

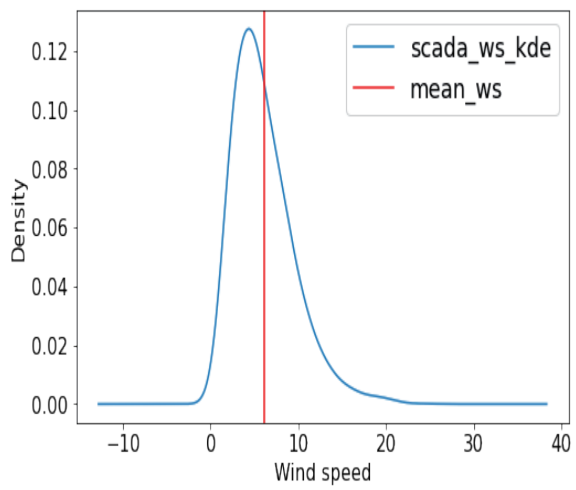


Figure 7: Distribution of WS on WF's location

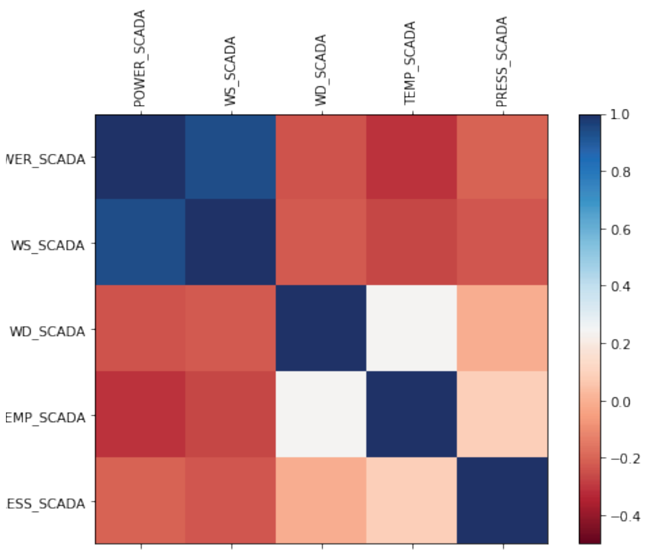


Figure 8: SCADA correlation matrix

3.2 Model

The input predictors of the model were two-year forecasts of wind speed and direction from the Aladin 2 meteorological model (NWP). The wind speed and direction forecasts are available in the form of 72-hour-long hourly sequences computed four times a day, i.e., every six hours. The total number of available sequences in a two-year period is regularly divided into three parts: training, validation, and test in the ratio of 70%, 10%, and 20% (the ratio is randomly selected). The Python 3 programming language was used to build the model, with the specific modules for deep learning, Keras, and TensorFlow for operations with tensors. Fig. 9 shows the structure of the two models used. The first model (Fig. 9a) consists of one LSTM layer and one dense layer. The mentioned model processes sequences only in a chronological way. The other model used is shown in Fig. 9b) (bidirectional LSTM), which processes sequences in a chronological and reverse manner. The internal states of the LSTM cell of forward and reverse sequences are combined with one of the possible functions (summation, multiplication, concatenation, etc.) and forwarded to the next layer (Fig. 9c).

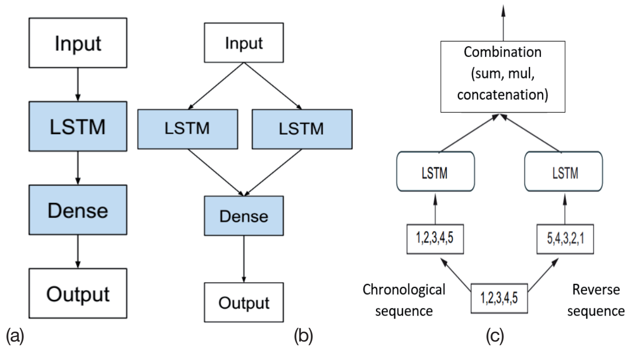


Figure 9: a) one-directional LSTM model (model 1); b) bidirectional LSTM model (Bi-LSTM) (model 2); c) Working principle of BI-LSTM

Fig. 10 presents layers and belonging parameters (report from Tensorflow environment) that are adjusted during training. The bidirectional LSTM model has almost twice as big internal memory. The size of the internal memory is a random (hyper) parameter of the model, like many other parameters that should be set before training.

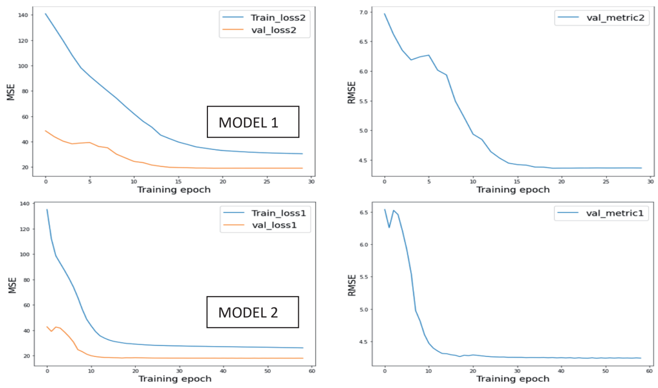
```
Model: "sequential_15"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_15 (LSTM)               (None, 72, 50)            10600
_________________________________________________________________
dense_26 (Dense)             (None, 72, 50)            2550
_________________________________________________________________
dense_27 (Dense)             (None, 72, 1)             51
=================================================================
Total params: 13,201
Trainable params: 13,201
Non-trainable params: 0
```

```
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional_9 (Bidirection multiple                  21200
_________________________________________________________________
dense_16 (Dense)             multiple                  2550
_________________________________________________________________
dense_17 (Dense)             multiple                  51
=================================================================
Total params: 23,801
Trainable params: 23,801
Non-trainable params: 0
```

Fiure 10: a) Model 1 parameters; b) Model 2 parameters

4. RESULTS

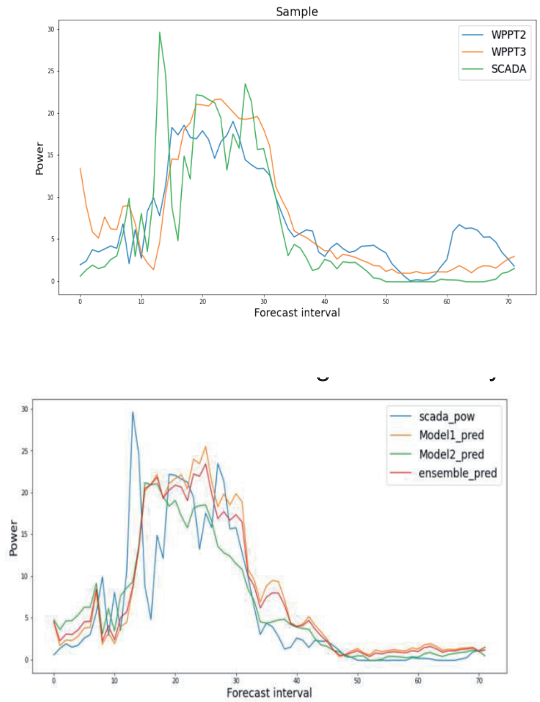
Fig. 11 shows the process of training and validating models 1 and 2. The training of the model is terminated when the loss function has not changed over a certain number of epochs (e.g., 10 epochs) in the validation data. Fig. 11b) shows the root mean square error (RMSE) between the forecasted sequences and the real values on the validation data (the root of the loss function on the validation data), i.e., the validation metrics of the model. It is obvious that model 2 reaches the minimum of the loss function faster, so the overall RMSE is more favorable in the case of bidirectional LSTM, which is due to the larger internal memory of model 2.



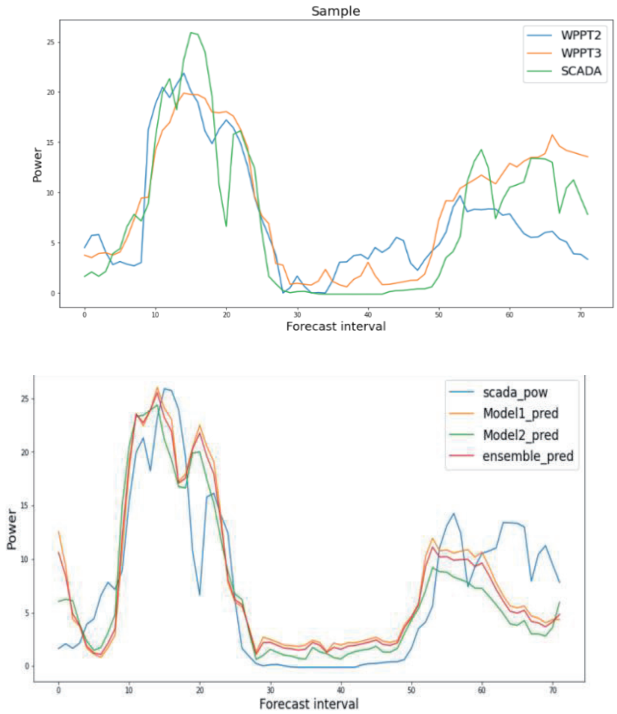
a) b)

Figure 11: a) Loss function of models 1 and 2 (training and validation) b) metrics of model's accuracy on validation data (RMSE)

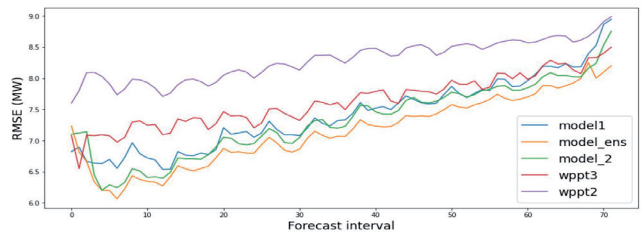
Fig. 12 presents the comparison of commercial tools (WPPT2 and WPPT3) with two test data samples, in comparison with real measurements of the SCADA system and comparison with forecasts of the presented deep learning models. In addition, it is possible to combine the forecasts of both models to obtain an average prediction that can provide better results (red curve - ensemble_pred). It can be seen that the proposed models provide forecasts of commercial accuracy with a relatively shallow model structure. Of course, more complex, and 'deeper' models could provide better results. It is interesting to note that the WPPT2 tool uses the same NWP forecasts (Aladin 2) as the input data used in this work, while WPPT3 uses multiple NWP forecast sources (Enfor). Fig 12 c) presents a horizon RMSE performance comparison on the test dataset (500 sequences) where it can be seen that the WPPT2 model had the worst while the ensembled model had the highest accuracy.



a)



b)



c)

Figure 12: Model comparison with commercial tools and SCADA measurements; a) sample 1; b) sample 2; c) Horizon RMSE comparison (test dataset)

CONCLUSION

This paper shows one of the approaches to applying deep learning to wind power forecasting using recurrent networks for sequential data. The process of data preparation and the data structure, as well as the model structure, are explained. Finally, a comparison of the forecasts obtained by the proposed methodology and commercial tools is presented using two samples, which provides insight into the accuracy of the forecasts obtained with methods of deep learning. The results showed that deep recurrent LSTM-based networks can outperform commercially available forecasting tools when trained using only the wind speed forecast as an input feature. Future research will focus on developing more complex models of deep learning to increase overall accuracy.

REFERENCES

[1] A. M. Foley, P. G. Leahy, A. Marvuglia and E. J. McKeogh, *Current methods and advances in forecasting of wind power generation*, Renewable Energy, vol. 37, pp. 1-8, January 2012 https://www.sciencedirect.com/science/article/pii/S0960148111002850

[2] J. Jung and R. Broadwater, *Current status and future advances for wind speed and power forecasting*, Renewable and Sustainable Energy Reviews, vol. 31, pp. 762-777, March 2014 https://www.sciencedirect.com/science/article/pii/S1364032114000094

[3] G. Giebel, *The state-of-the-art in short-term prediction of wind power - A literature overview*, Project ANEMOS, August 2003 https://www.osti.gov/etdeweb/servlets/purl/20675341

[4 Y. Ding, *Data science for wind energy*, Chapman and Hall/CRC, p. 424, December 2020

[5] J. Đaković and I. Kuzle, *Status and classification of methods for forecasting electricity generation from wind farms*, 13th Symposium on power system management HRO Cigre, Rovinj, Croatia, November 2018 https://www.bib.irb.hr/971989

[6] I. Kuzle, M. Klarić and H. Pandžić, *Feasibility assessment of a wind power plant with insufficient local wind data using cascade-correlating neural network*, Strojarstvo, vol. 53, no. 6, pp. 455-462, December 2011 https://hrcak.srce.hr/file/126033

[7] V. Bali and A. Kumar, *Deep learning based wind speed forecasting-A review*, 9th International Conference on Cloud Computing, Data Science & Engineering, Noida, India, January 2019, https://ieeexplore.ieee.org/document/8776923

[8] L. Han, R. Zhang, X. Wang, A. Bao and H. Jing, *Multi-step wind power forecast based on VMD-LSTM*, IET Renewable Power Generation, vol. 13, no. 10, pp. 1690-1700, July 2019, https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-rpg.2018.5781

[9] J.-F. Toubeau, J. Bottieau, F. Vallée and Z. D. Grève, *Improved day-ahead predictions of load and renewable generation by optimally exploiting multi-scale dependencies*, IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia), December 2017 https://ieeexplore.ieee.org/document/8378396

[10] H. Zhang, Y. Liu, J. Yan, S. Han, L. Li and Q. Long, *Improved deep mixture density network for regional wind power probabilistic forecasting*, IEEE Transactions on Power Systems, vol. 35, no. 4, pp. 2549-2560, July 2020, https://ieeexplore.ieee.org/document/8982039

[11] F. Chollet, *Deep learning with Python*, Manning, p. 384, November 2017 https://www.manning.com/books/deep-learning-with-python

[12] I. Sičaja, A. Previšić, M. Zečević and D. Budiša, *Evaluation of load forecast model performance in Croatian DSO*, Journal of Energy, vol. 67, no. 2, pp. 54-62, June 2018 http://www.journalofenergy.com/index.php/joe/article/view/80

[13] N. Holjevac, C. Soares, I. Kuzle, *Short-term power system hourly load forecasting using artificial neural networks*, Journal of Energy, vol. 66, no. 1-4, pp. 241-254, December 2017 http://www.journalofenergy.com/index.php/joe/article/view/107

[14] K. Jurković, H. Pandžić and I. Kuzle, *Review on unit commitment under uncertainty approaches*, 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2015), Opatija, Croatia, May 2015 https://ieeexplore.ieee.org/abstract/document/7160438

[15] D.P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, ICLR, p. 1-15, January 2017 https://arxiv.org/abs/1412.6980

[16] V. Bushaev, *Adam — latest trends in deep learning optimization*, October 2018, https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

[17] J. Brownlee, *Long short-term memory networks with Python*, ebook, p. 229, July 2018 https://machinelearningmastery.com/lstms-with-python/