20th International Conference on Renewable Energies and Power Quality (ICREPQ'22)
Vigo (Spain), 27th to 29th July 2022
Renewable Energy and Power Quality Journal (RE&PQJ)
ISSN 2172-038 X, Volume No.20, September 2022

Wind Power Forecasting with Machine Learning: Single and combined methods

J. Rosa[1], R. Pestana[1,2,3], C. Leandro[1], C. Geraldes[1,4], J. Esteves[3] and D. Carvalho[1]

[1] Department of Mathematics, Electrical Engineering
Instituto Superior de Engenharia de Lisboa – Lisbon, Portugal
e-mail: A45204@alunos.isel.pt, rui.pestana@isel.pt, carlos.leandro@isel.pt, carlos.geraldes@isel.pt

[2] System Operator Division
REN - Rede Eléctrica Nacional, S.A. – Lisbon, Portugal
e-mail: rui.pestana@ren.pt

[3] R&D NESTER
Centro de Investigação em Energia REN - State Grid, S.A. – Lisbon, Portugal
e-mail: rui.pestana@rdnester.com, joao.esteves@rdnester.com

[4] CEAUL, Centro de Estatística e Aplicações, Universidade de Lisboa – Lisbon, Portugal

Abstract. In Portugal, wind power represents one of the largest renewable sources of energy in the national energy mix. The investment in wind power started several decades ago and is still on the roadmap of political and industrial players. One example is that by 2030 it is estimated that wind power is going to represent up to 35% of renewable energy production in Portugal. With the growth of the installed wind capacity, the development of methods to forecast the amount of energy generated becomes increasingly necessary. Historically, Numerical Weather Prediction (NWP) models were used. However, forecasting accuracy depends on many variables such as on-site conditions, surrounding terrain relief, local meteorology, etc. Thus, it becomes a challenge to obtain improved results using such methods. This article aims to report the development of a machine learning pipeline with the objective of improving the forecasting capability of the NWP's to obtain an error lower than 10%.

Key words. Wind power forecast, feature engineering, machine learning, ensemble models, recurrent neural network.

1. Introduction

The worldwide significant increase of renewable energies is not only due to their important environmental advantages, but also due to their advantages in increasing a country's energetic independence while promoting domestic economic growth. In Portugal alone, by the end of the year 2021, the installed capacity of renewable energies was expected to exceed 14.6 gigawatts (in a total national installed capacity of 19.2 GW). This value represents an increase of 4.5% compared to 2020. In addition, in 2021 65.4% of the electricity generated in mainland Portugal was from renewable energy sources, where more than 26% was generated by wind energy [1]. One of the most essential tasks in power systems operation and control is short and medium-term forecasting. The short and medium-term forecasting of electric power production at wind farms is essential because it allows power production schedules at conventional power plants to be established and also to determine power reserve's needs. Thus, accurate forecasts of electric power production at wind farms play a vital role. However, the random and unstable characteristics of the wind energy source make it difficult to forecast the generated power. Hence, extensive efforts have been devoted to the developments and improvements of wind speed and power forecasting by numerous energy and environment-related research centres and universities [2]. This work tackles the need to develop forecasting models that could provide improved performances and it is divided into two parts: i) Single Machine learning models and, ii) Combination of models. The first part starts with an exploratory data analysis on the NWP data, described in chapter 3. An extensive Exploratory Data Analysis and Feature Engineering process followed (chapter 4). The Feature Engineering is divided into three main sections: i) Feature Selection, ii) Layers' Interactions and iii) Lag Features. After improving the dataset, different machine learning techniques were tested. The developed models are presented in chapter 5 and are designated as base-learns to be used in the second part of this work. The second part, presented in chapter 6, aims to combine the models previously developed to reduce the Normalized Root Mean Square Error (NRMSE) (1), presented in chapter 2. In order to improve the forecasting capacity, two approaches were taken. The first one was Ensemble Modelling, where the base-learns are combined with different algorithms. The second one was Recurrent Neural Network, where we wanted to study the feasibility of combining the base-learns with this strategy.

2. Performance Indicators

The error measure considered is the NRMSE. For each model, the predicted time-series is decomposed in disjoint windows with a horizon of 72h.

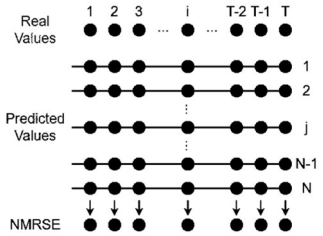

Fig. 1. Error evaluation scheme

In the process of evaluation, NRMSE is calculated for each time-step, as shown in Fig. 1. For the time-step *i*, NRMSE is calculated the following way:

$$\mathbf{NRMSE_i} = \frac{\mathbf{1}}{\boldsymbol{P_{farm}}}\sqrt{\frac{\sum_{j=1}^{N}(y_{i,j}-\hat{y}_{i,j})^2}{N}} \times \mathbf{100\%} \qquad (1)$$

where $y_{i,j}$ is the real value, $\hat{y}_{i,j}$ is the predicted value, P_{farm} is the installed capacity of the wind farm and N is a constant corresponding to the number of 72h windows. The NRMSE is a vector of size T. The model error is evaluated on the average of the NRMSE, one example is shown in Fig. 2.

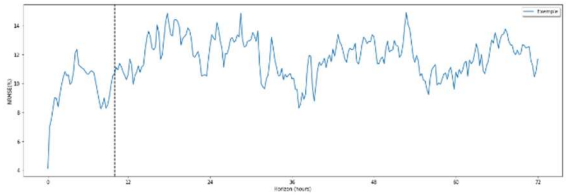

Fig. 2. $NRMSE_i$ representation example. The black line represents a random i, in this case is i = 39

3. NWP data

The original data used in this work are the forecasts of an NWP model (the MM5 model) for a Portuguese wind farm. The MM5 model, short for Fifth-generation Mesoscale Model, is a mesoscale model that can describe the behaviour and evolution of air masses and treat explicitly the inherent phenomena of atmospheric turbulence as well as other types of nonlinear atmospheric phenomena. For these predictions, the model has into consideration the roughness of the terrain as well as previous weather data. The model predictions are made daily starting at 00:00 UTC, with a time horizon of 76 hours, for the years 2018 and 2019. The sampling frequency corresponds to 15 minutes. Lastly, these predictions result in a data set with the 30 atmospheric parameters shown in Table I, which are used to predict the power generated at a given time.

Table I. - Atmospheric features initially considered

Name	Abbreviation	Units
Wind Speed 170m height	ws170m	m/s
Wind Speed 100m height	ws100m	m/s
Wind Speed 30m height	ws30m	m/s
Wind Speed 10m height	ws10m	m/s
Wind Speed Sea Surface height	wssea	m/s
Direction 170m height	d170m	º
Direction 100m height	d100m	º
Direction 30m height	d130m	º
Direction 10m height	d10m	º
Direction Sea Surface height	dsea	º
Temperature 170m height	t170m	^{o}C
Temperature 100m height	t100m	^{o}C
Temperature 30m height	t30m	^{o}C
Temperature 10m height	t10m	^{o}C
Temperature 2m height	t2m	^{o}C
Relative Humidity 2m height	rh2m	%
Momentum Flux	momf	$kg/(ms^2)$
Surface Pressure	psurf	hpa
Mean Sea Level Pressure	mslp	hpa
Cloud Fraction Total	clf	0 to 1 - 0 no clouds; 1 full cloudy
Low Cloud Fraction	clo	0 to 1 - 0 no low clouds; 1 full low cloudy
Middle Cloud Fraction	cmi	0 to 1 - 0 no middle clouds; 1 full middle cloudy
High Cloud Fraction	chi	0 to 1 - 0 no high clouds; 1 full high cloudy
Surface Heat Flux	shf	W/m^2
Latent Heat Flux	lhf	W/m^2
Incoming Short Wave Radiation	swr	W/m^2
Total Precipitation	raint	mm
Convective Precipitation	rainc	mm
Non Convective Precipitation	rainnc	mm

4. Feature Engineering

The single machine learning models are a set of models created to predict the power generated by a wind farm based on data generated by NWP models. The data-modelling pipeline was decomposed into A-Exploratory Data Analysis and B-Feature Engineering (chapter 4), Models (chapter 5) and Ensemble Modelling (chapter 6).

A. Exploratory Data Analysis

Exploratory Data Analysis is used to evaluate and investigate datasets and summarize their main characteristics, often employing data visualization methods.

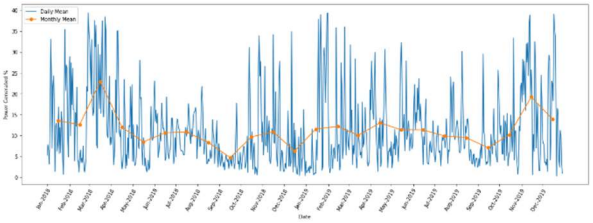

Fig. 3. Daily/monthly average of power generated

By analysing the data, shown in Fig. 3, it was concluded that the 2018 NWP data appears to be representative of the 2019 NWP data, apart from March of 2018.

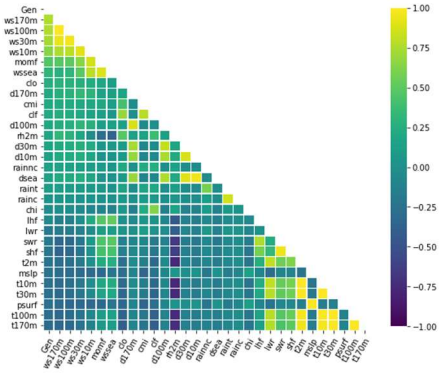

Fig. 4. Pearson correlation matrix

Through an observation of the Pearson correlation matrix, depicted in Fig. 4, for all pairs of variables, it was possible to conclude that the power generated is strongly correlated only with the wind speed at 170, 100, and 30 meters. It is also possible to detect a potential problem, in which some of the explanatory attributes have high correlation. In addition, it was also created a wind rose graph between the direction of the wind and the energy produced and was concluded that if the wind comes from the northwest, there is greater intensity of wind speed, producing a larger amount of energy.

B. Feature Engineering

Feature engineering is a process of transforming already existing features into new ones to have a better problem description, decreasing the model error. From removing unnecessary features to adding new ones that represent the interaction between altitudes and, lastly, adding features to deal with the autocorrelation problem. Feature Selection is the first step of Feature engineering. Feature selection refers to the techniques that support the selection of features to use, removing features that do not contribute with any valuable information to our model or even harm the performance, which reduces the number of input variables in a dataset. Two different strategies are used to detect the variable contribution: i) Extreme Gradient Boosting (XGBoost) and ii) GAM with Auto Regressive (GAMAR) models. Even if both models agree on removing a certain feature, it is still important to analyse the effect of removing said feature on the error. If the error increases with the removal of the attribute, then it is not worth removing the variable and it should be maintained in the dataset, despite the results of the model. From repeating this process multiple times, it was possible to conclude that several features presented in our dataset were not contributing to the prediction, as initially speculated in the exploratory data analysis. In total, just by removing these features, our models improved, on average, 0.28% on the error. The removed features are shown in Table II.

Table II. - Features removed and their impact on the error.

Feature removed	Mean error reduction
Features that represent precipitation	0.057
Features that represent cloud coverage	0.036
Wave radiation	0.020
Heat flux	0.022
Surface pressure	0.042
Wind speed, direction and temperature at 100 meters	0.077
Wind speed, direction and temperature at 10 meters	0.025

Layers' Interactions, the second step of Feature engineering, has the goal of correcting the existence of any possible outliers and modelling the turbulence. The layer's interaction were applied for the wind speed, wind direction, and temperature at every consecutive pair of altitudes, with the following rationales:

1) Wind speed: To calculate the wind speed interaction between consecutive layers $(\psi_{i,j})$, an approximation of the Shear velocity was used as shown in (2).

$$\boldsymbol{\psi_{i,j}} = \sqrt{\frac{|\boldsymbol{WS_i} - \boldsymbol{WS_j}|}{\boldsymbol{\rho}} \times \boldsymbol{dist_{i,j}}} \quad (2)$$

In (2), WS_i is the wind speed at the altitude i, ρ is the air density, and $dist_{i,j}$ is the distance between altitudes.

2) Wind direction: The wind direction interaction between consecutive layers $(\phi_{i,j})$ is calculated as in (3) and (4).

$$\boldsymbol{\phi_{i,j}} = \left|\boldsymbol{Ang_i} - \boldsymbol{Ang_j}\right| \quad (3)$$

$$\boldsymbol{\Phi_{i,j}} = \begin{cases} \boldsymbol{360 - \phi_{i,j}}, & \textbf{If} \quad \boldsymbol{\phi_{i,j} \geq 180} \\ \boldsymbol{\phi_{i,j}}, & \textbf{If} \quad \boldsymbol{\phi_{i,j} < 180} \end{cases} \quad (4)$$

In (3), Ang_i represents the wind direction at altitude i, in degrees, Ang_j represents the wind direction at altitude j, in degrees.

3) Temperature: Lastly, the interaction between the temperature of consecutive layers $(\tau_{i,j})$ is given by the difference between both, as in (5).

$$\boldsymbol{\tau_{i,j}} = \boldsymbol{T_i} - \boldsymbol{T_j} \quad (5)$$

In (5), T_i represents the temperature at the altitude i, and T_j represents the temperature at the altitude j, both in Celsius degrees.

Lag Features is the last step of Feature engineering. Bearing in mind that, past observations have a great influence on the future, they should be considered in any form in the prediction. A common strategy to solve this problem is to try to replicate an autoregressive component. However, instead of considering all past values, as is usually done by an autoregressive component, we will consider a window with a statistical summary of the interval. This type of feature is usually called lag feature.

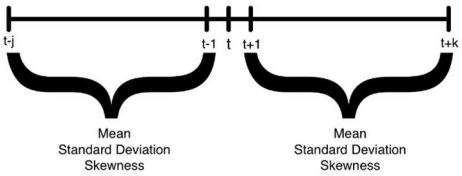

Fig. 5. Backwards and forwards lag features

It was concluded that wind speed at 170 meters, 30 meters and sea level should have lag features, as well as the temperature and wind direction at 170 and 30 meters. The statistical measures used to describe these features were the mean; the standard deviation; the maximum and minimum value; the skewness, and the robust coefficient of variation with exception of the coefficient of variation being replaced by the phase of the angular momentum [3] for the direction features. The chosen window' horizon was 6 hours for the backward and forward lag features, exemplified in Fig. 5.

Lastly, lag features for the wind power generated were also created, however it's used two different window' horizons: half an hour and an hour. The only statistical measure used was the mean. It is important to reinforce that for the wind power generated, creating lag features are only based on previous observations. The models showed better results when the lag features for the target variable, the wind power generated, had a prediction horizon

smaller than 6 hours. Thus, every algorithm tested has two versions: i) one for short term predictions, the prediction horizon smaller than 6 hours, where the model extrapolates the results and, ii) a second version for medium-term predictions, prediction horizon after 6 hours, in which case the model does not use these lag features.

5. Single machine learning models

After improving the dataset, various algorithms were evaluated. In total, six algorithms were used in this stage: 1) Persistence, 2) XGBoost, 3) Light Gradient Boosting Machine (LightGBM), 4) Support Vector Machine (SVM), 5) Autoregressive Integrated Moving Average with Exogenous Variable (ARIMAX) and, 6) GAMAR.

1) Persistence uses the last known value as the forecast for every future point, as shown in (6).

$$\hat{y}_{t+k|t} = y_t, \qquad k \geq 1 \tag{6}$$

In (5), y_t represents the real value of the power generated at t and $\hat{y}_{t+k}$ represents the predicted value of the power generated at the k-step ahead prediction.

2) XGBoost implements machine learning algorithms under the Gradient Boosting framework. XGBoost tends to over-fit the data. Regularization, amongst other techniques, is a technique used to avoid over-fitting through both LASSO (L1) and Ridge (L2) regularization.
3) LightGBM is an improved version of gradient learning framework based on decision trees and the idea of "weak" learners [4].
4) SVM aims to fit as many instances as possible on the decision boundary while limiting margin violations. The kernel function models the interaction between the features and the target variable. It was chosen a linear Kernel for this work. The feature vectors were normalized prior to feeding them to the SVM.
5) ARIMAX is a model suitable to deal with the high autocorrelation of the variables due to its autoregressive component. Before the data is feed to the model the data is first standardized and after that a Principal Components Analysis (PCA) is applied to the data where we conclude that 30 out of 103 components explain more than 80% of the data variance, which was the minimum required for the PCA to be considered well applied.
6) GAMAR fits a general additive model (GAM) [5] to the data while keeping an autoregressive part. Parameters in GAMAR are estimated by maximum partial likelihood using modified Newton's method [6].

The metaparameters of these models were tuned using Optuna [7].

C. Models results

After improving the NWP data, these models are trained with the 2018 data and then tested in the 2018 and 2019 data. The models' predictions in 2018 are used as the Ensembles' base-learns and the models' predictions in 2019 is used to evaluate the models' performance. The training aimed to minimize the NRMSE of production for the next 15 minutes.

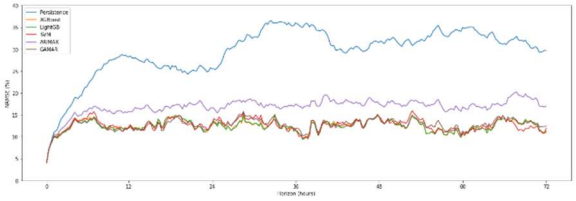

Fig. 6. All base-models results

As seen in Fig. 6 models perform better in short-term predictions, however, the results are showing more discrepancy in the medium-term predictions.

Table III. - Single ML results.

	Average NRMSE
Persistence	29.46 %
XGBoost	12.51 %
LightGBM	12.45 %
SVM	12.62 %
ARIMAX	16.86 %
GAMAR	12.76 %

Based on the results obtained, Table III, it is possible to conclude that for a model to obtain a good performance, it needs to be able to capture non-linear relationships between the various attributes and the wind power generated. For this reason, ARIMAX does not manage to obtain as good results as other models tested: XGBoost, LightGBM, SVM, and GAMAR. Thus, only these four models are considered for the Chapter 6.

6. Combination of models

The objective of combining the models developed in Chapter 5 is to continue improving the forecast capability in order to obtain an even lower error.

A. Ensemble Modelling

Ensemble models is a machine learning approach that combines multiple learners and synthesizes the results into a single prediction by using many different modelling algorithms or using different training data sets [8].
The next three models are a form of the ensemble strategy called Stacking. Stacking is an ensemble learning technique that combines multiple models via a meta-model. The base-learners are learned in parallel and taken as new features to re-train a new learner, the meta-model. The meta-model inputs the predictions as the features and the target being the ground truth values in data and attempts to learn how to best combine the input predictions to make a better output prediction.

1) In Weighted Ensemble, a weight is associated with each model while ensuring the sum of all weights is equal to one. The forecast is the sum of the product of all weights with the respective model.
2) The second model is designated Stacking or STCK. The meta-model is a Support Vector Machine for Regression Problems (SVR) with an radial basis function (RBF) kernel. Before the data is feed to the model the data is first standardized and after that PCA is applied to the data where we conclude that 1 out of 4 components were chosen [9].
3) Linear SVR is a Stacking of models. The motivation for using Linear SVR as a meta model instead of linear kernel SVR is because it has more flexibility in the choice of penalties and loss functions and works better with a large number of samples [10].

The next two models are a form of the ensemble strategy called Boosting. Boosting considers homogeneous weak learners but learns them sequentially in a very adaptive way and combines them following a deterministic strategy.
Boosting is described as a stage-wise additive model. This is because one new weak learner is added at a time and existing weak learners in the model are frozen and left unchanged.

4) Boosted Regression Tree (BTR) combines the strengths of two algorithms: Ada Boost and a Regression Tree. Ada Boosting is a meta-estimator used for performance improvement [11]. Regression Tree uses the tree representation to solve the problem in which each leaf node corresponds to a numeric value and attributes are represented on the internal node of the tree.
5) XGBoost was explained in chapter 4.

B. *Ensemble models results*

The model was trained and tested in a dataset predicted in Chapter 5 with four attributes with the predictions of the models XGBoost, LightGBM, SVM, and GAMAR. Training was done with the data from 2018 and then was tested with the data from 2019. The training aimed to minimize the NRMSE of production for the next 15 minutes.

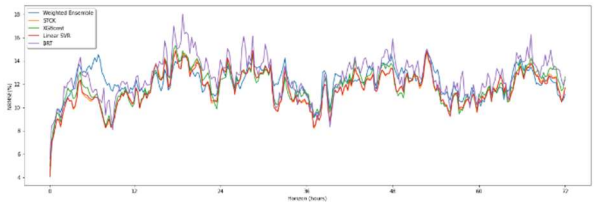

Fig. 7. All Ensemble results

As seen in Fig. 7, models also perform better in short-term predictions. The results for medium-term predictions have less of a discrepancy.

Table IV. - Ensemble models results

	Average NRMSE
Weighted Ensemble	12.18 %
STCK	11.60 %
Linear SVR	11.59 %
BTR	12.60 %
XGBoost	11.83 %

Ensembling the weak learners made an improvement on predicting the power generated, Table IV. Linear SVR presents the best performance with an average NRMSE of 11.59 %.

C. *Recurrent Neural Networks*

Recurrent neural networks (RNN) are a class of neural networks that are helpful in modelling sequence data, like time series, derived from feedforward networks.
Because of its' internal memory, they memorize essential aspects regarding the input, which allows them to be very precise at predicting the future. RNN can form a much deeper understanding of a sequence and its context compared to other algorithms.

1) Data Preprocessing: The dataset is first normalized to prevent the network from ineffectively learning the problem. After that a PCA is applied to the data where we conclude that 1 out of 4 components.
2) Layers: Keras is an open-source software library that provides a Python interface for the TensorFlow library [12]. In keras, the basic building blocks of neural networks are called layers. The same layers were used on each RNN and these are: i) Bidirectional layer; ii) TimeDistributed layer with a Dense layer of single output value to design a many-to-many RNN and lastly, iii) weight layer regularizer.

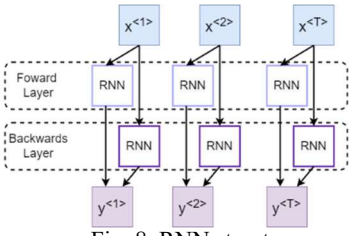

Fig. 8. RNN structure

In the RNN structure, as shown in Fig. 8, x_i and y_i are the values of each time-step, where $\vec{x}$ is the input, the j^{th} window, and $\vec{y}$ is the output of the RNN of the same window.

Three different types of RNN were used: i) Simple RNN, ii) Long Short-Term Memory (LSTM) and, iii) Gated Recurrent Unit (GRU).

1) Simple RNN uses the previous information in the sequence to produce the current output. The training data was divided into 48h horizon windows to be feed to the RNN [13] [14].

2) LSTM is a sequential network that allows information to persist. It can handle the vanishing gradient problem faced by RNN. LSTM has three gates: i) input gate, ii) forget gate and iii) output gate. The training data was divided into 48h horizon windows to be feed to the RNN [15].
3) GRU, similar to the LSTM, has two units gates that modulate the flow of information inside the unit having a separate memory cell: i) update gate and, ii) reset gate. The training data was divided into 24h horizon windows to be feed to the RNN [16].

D. RNN Results

The model was trained and tested in a dataset predicted in chapter 5 with four attributes with the predictions of the models XGBoost, LightGBM, SVM, and GAMAR. Training was done with the data from 2018 and then was tested with the data from 2019. The model was trained to minimize the NRMSE of production for the next 72 hours.

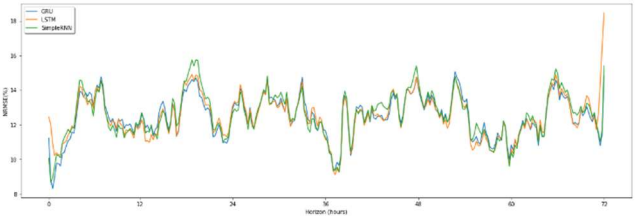

Fig. 9. RNN results

As seen in Fig. 9, the results of the RNNs are very similar to each other. While testing for different horizons and with different layers, the NRMSE results have shown a minor variation. These methods also show difficulty in predicting the first few time-steps and the last few time-steps.

Table V. - RNN results

	Average NRMSE
Simple RNN	12.55 %
LSTM	12.56 %
GRU	12.45 %

GRU presents the best performance with an average NRMSE of 12.45 %, Table V. In the end, the RNN performance does not outperform the ensemble models nor its' base learners.

7. Conclusion

The objective of this project was to decrease the average NRMSE error by single ML methods (12.45%) and combining them to reduce even more the error (11.59%). By providing, a better wind power forecast the amount of operational reserves is reduced, decreasing the overall cost of system operation. The results obtained throughout the work demonstrate that it is possible to obtain an average error between 11% and 13% for medium-term forecasts.

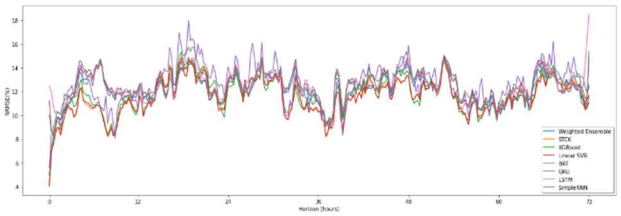

Fig. 10. All results

As seen in Fig. 10, the error has almost always very similar behavior. In conclusion, the ensemble model Linear SVR showed the best results with an average NRMSE of 11.59 %. The improvement of these models is still a work in progress, efforts are being put in action in order to further increase their performances.

Acknowledgement

The work was supported by Science and Technology Project of SGCC (Research on the Short-term Wind Power Ensemble Learning Forecasts Based on Multi-source Heterogeneous Data Fusion) (4200-201955510A-0-0-00).

References

[1] APREN. "Balanço da produção de eletricidade de Portugal continental". https://www.apren.pt/pt/energias-renovaveis/producao. Visited on February 2022.
[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Y Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". NIPS 2014 Workshop on Deep Learning, December 2014.
[3] Jeff Sanny William Moebs, Samuel J. Ling. University Physics Volume 1, chapter 11 - Angular Momentum, pages 539–565. OpenStax, 2019.
[4] Fan, Junliang & Ma, Xin & Wu, Lifeng & Zhang, Fucang & Yu, Xiang & Zeng, Wenzhi, 2019. "Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data," Agricultural Water Management, Elsevier, vol. 225(C).
[5] Trevor Hastie and Robert Tibshirani. Generalized additive models. Statistical Science, 1(3):297 – 310, 1986. URL: https://doi.org/10.1214/ss/1177013604
[6] Yang, L., Qin, G., Zhao, N. et al. Using a generalized additive model with autoregressive terms to study the effects of daily temperature on mortality. BMC Med Res Methodol 12, 165 (2012). https://doi.org/10.1186/1471-2288-12-165
[7] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. https://doi.org/10.48550/arXiv.1907.10902
[8] Vijay Kotu, Bala Deshpande. "Chapter 2 - data science process". In Vijay Kotu, Bala Deshpande, editor, *Data Science (Second Edition)*, pages 19–37. Morgan Kaufmann, second edition edition, 2019.
[9] Diogo Camilo de Carvalho. "Machine learning in wind power forecast". ISEL, July 2021.
[10] Joana Lopes Rosa. "Wind Power Forecast with Machine Learning". ISEL, March 2022.
[11] Harris Drucker. "Improving regressors using boosting techniques" . Proceedings of the 14th International Conference on Machine Learning, 08 1997. August 1997.
[12] Wikipedia contributors. "Keras - Wikipedia". https://en.wikipedia.org/w/index.php?title=Keras&oldid=%201068761032, 2022. Visited on February 2022.
[13] Apeksha Shewalkar, Deepika Nyavanandi, Simone A. Ludwig. "Performanceevaluation of deep neuralnetworks applied to speech recognition: rnn,lstm and gru" . Department of Computer Science, North Dakota State University,Fargo, ND, USA, 9(4):235–245. October 2019.
[14] Victor Zhou. " An Introduction to Recurrent Neural Networks for Beginners". https://victorzhou.com/blog/intro-to-rnns/. Visited on February 2022.
[15] Kaisheng Yao, Trevor Cohn, Katerina Vylomova, Kevin Duh, Chris Dyer. "Depth-Gated Recurrent Neural Networks" . August 2015.
[16] Rahul Dey, Fathi Salem. "Gate-variants of Gated Recurrent Unit (GRU) neural networks" . pages 1597–1600. August 2017.