Very short-term wind energy forecasting based on stacking ensemble

Sinvaldo R. Moreno*, Ramon G. da Silva†, Matheus H. D. M. Ribeiro†‡, Naylene Fraccanabbia§, Viviana C. Mariani§¶ and Leandro dos S. Coelho*†¶
*Electrical Engineering Graduate Program (PPGEE), Federal University of Parana (UFPR), Curitiba, Parana, Brazil. 81530–000
†Industrial & Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR), Curitiba, Parana, Brazil. 80215–901
‡Department of Mathematics, Federal Technological University of Parana (UTFPR), Pato Branco, Parana, Brazil. 85503–390
§Mechanical Engineering Graduate Program (PPGEM), Pontifical Catholic University of Parana (PUCPR), Curitiba, Parana, Brazil. 80215–901
¶Department of Electrical Engineering, Federal University of Parana (UFPR), Curitiba, Parana, Brazil. 81530–000
Emails: sinvaldo.moreno@gmail.com, gomes.ramon@pucpr.edu.br, matheus.dalmolinribeiro@gmail.com, nayfraccanabbia@outlook.com, viviana.mariani@pucpr.br, leandro.coelho@pucpr.br

***Abstract*—Wind power generation is one of the technologies of electric production which still in development in Brazil, however, it already has a great penetration in the national energy matrix, representing 13.98% of the national energy consumption in Brazil. Due to the high level of uncertainty and the chaotic fluctuations in wind speed, predictions of wind energy with high accuracy is a challenge. In this context a stacking ensemble (STACK) model is proposed to forecast the wind power generation of a turbine in a wind farm at Parazinho, RN - Brazil. The proposed model combines four different algorithms as base-learners, such as, eXtreme Gradient Boosting (xgBoost), Support Vector Machine for regression with Linear Kernel (SVR-Linear), Multi-Layer Perceptron with multiple layers (MLP) and K-Nearest Neighbors (K-NN), and one algorithm as meta-learner - Support Vector Machine for regression with Radial Basis Function Kernel (SVR-RBF). To access the performance of adopted methodology, the results of STACK are compared with the results of the base-learners. Four performance measure criteria, as well as statistical tests are adopted. As results, STACK reached better results in all performance measures. Indeed, STACK and SVR-Linear are statistically equals. According to these results, applying the STACK proposed model indeed improved the forecasting when comparing with the other algorithms tested individually.**

***Keywords*—Wind energy, forecasting, time series, machine learning, stacking ensemble.**

I. Introduction

Wind power generation is one of the technologies of electric production which still in development in Brazil, however, it already has a great penetration in the national energy matrix, and it is one of the principal renewable energy sources. In 2018, wind energy represented 13.98% of the national energy consumption and it supplied 38% of the population of Brazil, according to 2018 annual report of the Brazilian Wind Energy Association (ABEEolica) [1].

Due to the high level of uncertainty and the chaotic fluctuations in wind speed, the wind energy is classified as intermittent source, such that wind energy is not able to supply stable demand. Intermittence is mainly driven by continuous and chaotic fluctuations in wind speeds and the lack of tools to provide coherent predictions.

Hence, in literature, wind energy forecast has been the study object in many researches using different techniques [2]–[5]. Because of the chaotic and uncontrollable behavior, predicting wind energy as accurate as possible is a challenge. Wind energy forecasting models can be classified by its prediction time horizon in four categories: very short-term, short-term, medium-term, and long-term as illustrated by Figure 1. In general, shorter forecasting time horizon can provide more detailed and accurate results, but less time left for the deployment of wind power generation [6].

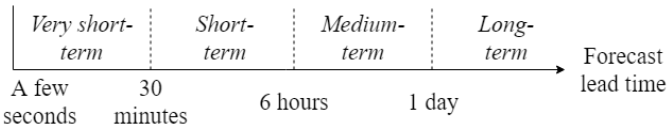


Fig. 1. Time-scale classification of wind energy forecasting, adapted from [6]

To enhance the accuracy of the predictions, stacking ensemble, which comes from ensemble learning, is proposed in this paper. Ensemble learning uses several weak learners (base learners) combined through mean rule (for regression problems) to build a stronger model [7]. Stacking ensemble is one of the many ways to work with ensemble. Stacking ensemble do predictions of several base learners to compose the stacking level 1. These predictions are used as inputs in the next level for a meta-learner on stacking level 2. In summary,

stacking ensemble combines two level models for an accurate prediction: one is the base-learners for preliminarily predicting the posteriori probabilities of samples and the other is a meta-learner for predicting the final value by combining the base-learners [8].

Hence, this paper proposes an application of an efficient stacking ensemble forecasting model that combines different and heterogeneous algorithms as learners. The stacking ensemble model combines four algorithms - xgBoost, MLP, K-NN and SVR-Linear. As meta-learner heading the stacking ensemble, SVR-RBF were used. The proposed STACK model has been trained to forecasting the wind power generation one-step ahead (10 minutes ahead).

The remainder of this paper is structured as follows: Section II-A presents the dataset adopted in this paper. Section II-B presents a brief description of the models used on this paper. Section III details the procedures of the research methodology applied. Section IV presents the results obtained and discussions. Finally, Section V presents the final considerations and some proposals of future works.

II. Material & Methods

A. Material

The collected dataset refers to the observation of the wind turbine power generation every 10 minutes. This wind power comes from a turbine in a wind farm located at Parazinho, RN - Brazil, Figure 2. The dataset period starts in August 01 2017 00:00h, and ends in August 31 2017 23:50h. Therefore, the number of observations in this case is 4439, as presented on Figure 3.

Fig. 2. Wind farm location map

The dataset is composed by eigth variables, as follows on Table I, where Power is the system output, and the others are the system inputs.

B. Methods

1) Stacking Ensemble: Stacking is originally presented by [9] as stacked generalization, which has been proposed

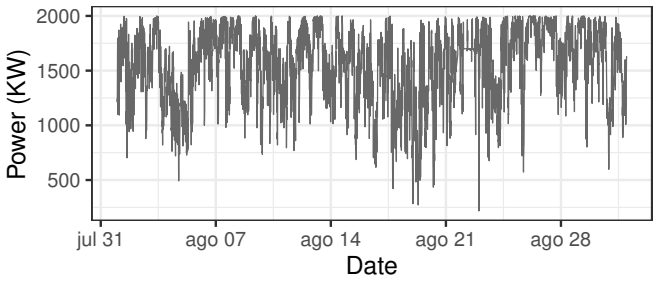


Fig. 3. Observed time series

TABLE I
INPUTS AND OUTPUT OF THE SYSTEM

Type	Description	Unit Measure
Output	Power	KW
Input	Generator Bearing Temperature	Celsius
Input	Generator Bearing 2 Temperature	Celsius
Input	Generator Speed	RPM
Input	Wind Speed	m/s
Input	Wind Direction Absolute	Degrees
Input	Nacelee Direction	Degrees
Input	Ambient Temperature	Celsius

to improve prediction accuracy by integrating a number of diverse sub-models. Hence, Stacking algorithm is an ensemble learning technique, in which the predictions of a group of individual learners (base learners) are given as inputs to a second-level learning algorithm (meta-learner) which combines the model predictions optimally to form a final set of predictions [2]. In this paper the base-learners are combined by stacking, in which a meta-learner is served to combine the predictions of base learners which are called meta-data [8].

2) Support Vector Machines with Radial Basis Function Kernel: Support Vector Machine (SVM) is a classification algorithm based on statistical learning theory [10]. The linear SVM is an efficient algorithm for classification and regression in linearly structured data [11]. Let x_i, $i = \{1, 2, \ldots, n\}$ be n training samples.

$$f(x) = w^T x + b \tag{1}$$

where x_i is an n-dimensional input, w is the weight vector and b the bias

In nonlinearly structured data, the SVM can also be applied if a kernel k is employed. In this case, a certain number S of support vectors $\mathbf{x}_i$ and corresponding coefficients $\alpha_i \in \mathbb{R}$ are required. Hence, for a kernel SVM, the function

$$f(\mathbf{x}) = \sum_{i=1}^{S} \alpha_i k\left(\mathbf{x}_i, \mathbf{x}\right) + b \tag{2}$$

has to be evaluated for every new instance $\mathbf{x}$.

Furthermore, Radial Basis Function Kernel (RBF kernel) is a default and recommended kernel function for SVM classifier

[12]. The RBF kernel can be defined as

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \quad (3)$$

where $\|\mathbf{x} - \mathbf{x}'\|^2$ may be recognized as the squared Euclidean distance between the two feature vectors, and σ is a free parameter.

3) eXtreme Gradient Boosting: The eXtreme Gradient Boosting (xgBoost) is an improved version of the gradient boosting decision tree algorithm that constructs boosted trees in an efficient and parallel manner [13]. xgBoost is based on gradient boosting, which generates a strong classifier by iteratively updating parameters of the former classifier to decrease the gradient of loss function [14].

The formulation of xgBoost algorithm is defined by [4] as

$$F_{obj}(\theta) = L(\theta) + \Omega(\theta) \quad (4)$$

where $L(\theta) = l\left(\widehat{y_i}, y_i\right)$ and $\Omega(\theta) = \gamma T + \frac{1}{2}\lambda\|w\|^2$.

$F_{obj}(\theta)$ is the objective function, $L(\theta)$ is the loss function between prediction $\widehat{y_i}$ and real value y_i, $\Omega(\theta)$ is the regularization term, γ is the learning rate, T is the number of leaves in the tree, λ is the regularization parameter, and w is the weights of the leaves.

4) Multi-Layer Perceptron with multiple layers: Multi-Layer Perceptron (MLP) is the most common and applicable type of feedforward neural networks [15]. MLP networks consist of input layer, one or more hidden layers and output layer. Each layer has a number of neurons (processing units) and each neuron is fully interconnected with weighted connections to the neuron in the subsequent layer [16]. The general expression can be defined as follows

$$y = f_2\left(\sum_{j=1}^{N} w_j f_l\left(\sum_{i=1}^{n} h_{ij} X_i + b_j\right) + b_o\right) \quad (5)$$

where h_{ij}, b_j and f_1 are the weight matrix, the bias vector and the activation function of the hidden layer, respectively, and w_j, b_o and f_2 are the weight vector, the bias scalar and the activation function of the output layer.

5) K-Nearest Neighbors: K-Nearest Neighbors (K-NN) is algorithm to solve regression and classification problems which was proposed by [17].

The K-NN algorithm works by categorizing data via correlating inputs to similar outputs. The number of nearest neighbors (k) and the distance between data points were adjusted while developing the K-NN model. [18].

Basically, K-NN works by finding the distance between a query and all the examples in the dataset, selecting the number of examples (k) closest to the query, then votes for the most frequent label (as classification) or calculates the averages of the labels (regression).

In regression, the K-NN algorithm follows: (i) Compute the Euclidean or Mahalanobis distance from the query example to the labeled examples; (ii) Order the labeled examples by increasing distance; (iii) Find a heuristically optimal number k of nearest neighbors; and (iv) Calculate an inverse distance weighted average with the k-nearest multivariate neighbors [19].

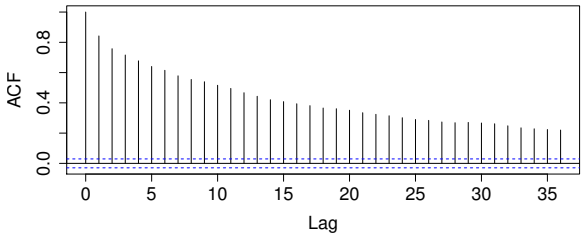
6) Support Vector Machines with Linear Kernel: SVM with Linear kernel differs from SVM wit RBF kernel, by using the linear kernel function [20], which can be define as

$$k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}. \quad (6)$$

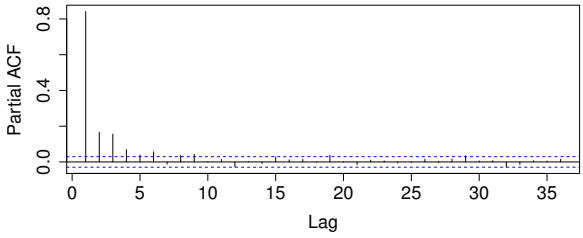
III. Methodology

A. Data processing

The dataset has 4439 observations collected since August 01 2017 until August 31 2017, every 10 minutes. By dividing the dataset into input and outputs of the system, as explained in Section II, an Auto-Correlation Function (ACF) and Partial Auto-Correlation Function (PACF) plots were conducted in the output (Power) to determine how many lags would be chosen for predictions calculation as seen on Figures 4a and 4b, respectively. The lag equal 1 was chosen then.



(a) Auto-correlation function for wind power



(b) Partial auto-correlation function for wind power

Fig. 4. ACF and PACF plots for lag verification

Furthermore, the new dataset was splitted into two sets (training and test) to perform the machine learning algorithms, predictions and performance measures analysis. The training set corresponds of 70% of the whole dataset (3106 observations), and test set is the remainder (1332 observations), according to Table II.

Moreover, Table III presents statistical indicators of the inputs and the output, such as maximum (max), minimum (min), mean and standard deviation (std).

In addition, a BoxCox transformation [21] was applied on the training set to preprocessing the set for the algorithm training. The BoxCox transformation is defined as

TABLE II
NUMBER OF OBSERVATIONS USED FOR WIND POWER DATA ANALYSIS

Dataset	Percentage	Number of observations
Observed	100%	4438
Training	70%	3106
Test	30%	1332

$$y^{(\lambda)} = \begin{cases} \frac{y^{\lambda}-1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log y, & \text{if } \lambda = 0 \end{cases} \tag{7}$$

where λ is a parameter, possibly a vector, defining a particular transformation.

B. Training, forecasting and evaluation

The training set was performed by each base-learner model – xgBoost, MLP, K-NN and SVR-Linear – as well the SVM with RBF kernel (SVR-RBF), which is the meta-learner, using a 5-fold cross-validation.

The base-learners are used in the first layer of STACK. The outputs of those base-learners are then used to train the second layer, the meta-learner. The second layer model (SVR-BRF) is used to combine the outputs from the first layer, giving the final predict values of the training. The framework of the stacking ensemble (STACK) is presented in Figure 5.

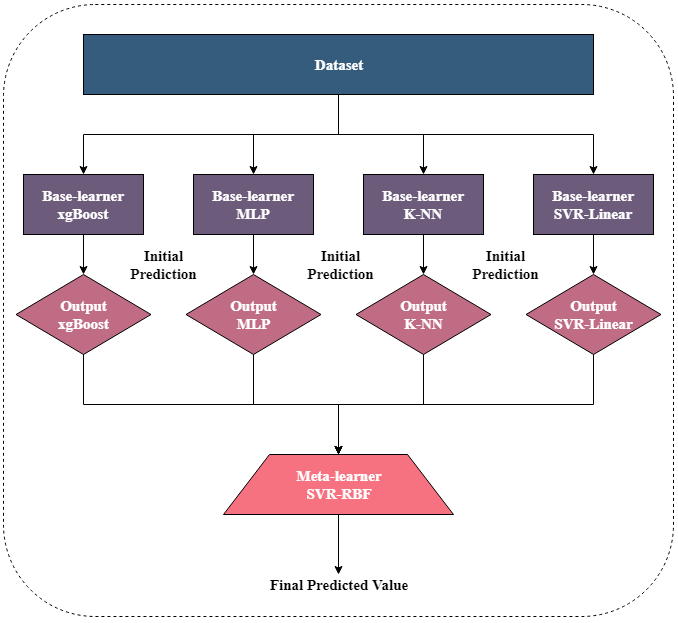


Fig. 5. Framework of the proposed forecasting model

The forecasting of the wind power generation is given by (8), as follows

$$\hat{y} = f\{y_t, \mathbf{x}_{it}\} \tag{8}$$

where f is a function related to model adopted in training process, $\hat{y}$ is the forecast value for one-step ahead, y_t is the observed output value at time t, $\mathbf{x}_{it}$ is a matrix of i inputs at time t, and i is the number of inputs in the system.

Moreover, the forecasting is performed using the training results with the test set and the results were evaluated using relative root mean square error (RRMSE) (9), mean absolute percentage error (MAPE) (10), coefficient of determination (R^2) (11), and sum of squared errors (SSE) (12) to determine which is the better model for this problem, according to criteria on Tables IV [22] and V [3].

$$\text{RRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{\frac{1}{n}\sum_{i=1}^{n} y_i}, \tag{9}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{10}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}[y_i(t) - \hat{y}_i(t)]^2}{\sum_{i=1}^{n}[y_i(t) - \overline{y}_i(t)]^2}, \tag{11}$$

$$\text{SSE} = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2, \tag{12}$$

where n represents the number of observations of the training and test sets, y_i is the ith value observed for series and $\hat{y}_i$ is the ith value predicted by the adopted model.

Friedman test can be applied to evaluate the squared errors (SE). This approach is able to show if there is difference between the average ranks for more than two samples [23]. It can be applied to answer if on a set of k algorithms (greater than two) at least two of the methods present different results. The statistic of test is stated as follows

$$\text{FD} = \frac{12n}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{13}$$

in which FD is distributed according to χ^2 with $k-1$ degrees of freedom, n observations and k groups, and R_j is the sum of the ranks for the jth group. Under null hypothesis, there is no difference between groups SE.

In addition, the Nemenyi test can be applied. In this approach a threshold is obtained according to (14)

$$\text{CD} = \frac{q_{\infty,k,\alpha}}{\sqrt{2}}\sqrt{\frac{k(k+1)}{6}}, \tag{14}$$

where CD is the critical difference for which it is possible to infer that the results are statistically equals or not for the compared groups, $q_{\infty,k,\alpha}$ are the quantiles based on the Studentized range statistic, k is the groups number and n is the observations number. If the critical differences rank sums $|R_i - R_j|$ are greater than CD, there is difference between results from groups i and j [24]. Therefore, these two approaches are applied to compare the SE of applied algorithms [25], [26].

The results presented in Section IV are generated using the processor Intel(R) Core(TM) i5-7200U Central Processing Unit @ 2.50GHz 2.70GHz in Windows 10 64 bits operating system. The R software [27] is adopted to performing the modeling.

TABLE III
SUMMARY OF THE STATISTICAL INDICATORS OF THE INPUTS AND OUTPUT OF THE DATASET

Variable	Samples	Number	Statistical indicator			
			Max	Min	Mean	std
Power	All samples	4438	2000.30	217.50	1553.00	338.8628
	Training samples	3106	2000.30	272.60	1506.40	342.2418
	Test samples	1332	2000.10	217.50	1662.00	304.2551
Generator Bearing Temperature	All samples	4438	76.00	46.00	63.46	6.5940
	Training samples	3106	76.00	46.00	62.31	6.4972
	Test samples	1332	75.00	50.00	66.17	6.0016
Generator Bearing 2 Temperature	All samples	4438	59.00	39.00	49.98	4.5902
	Training samples	3106	59.00	39.00	49.28	4.5592
	Test samples	1332	58.00	41.00	51.63	4.2253
Generator Speed	All samples	4438	1345.00	335.00	1282.00	75.0849
	Training samples	3106	1345.00	793.00	1277.00	77.2921
	Test samples	1332	1345.00	335.00	1294.00	68.2772
Wind Speed	All samples	4438	15.40	5.10	9236.00	1.1825
	Training samples	3106	15.40	5.10	9169.00	1.2235
	Test samples	1332	13.00	6.30	9393.00	1.0649
Wind Direction Absolute	All samples	4438	167.20	97.60	134.20	13.1943
	Training samples	3106	167.20	97.60	134.00	13.2954
	Test samples	1332	166.00	103.80	134.80	12.9402
Nacelle Direction	All samples	4438	165.60	98.70	134.10	13.2749
	Training samples	3106	164.60	98.70	133.90	13.4646
	Test samples	1332	165.60	104.20	134.60	12.8116
Ambient Temperature	All samples	4438	32.00	22.00	26.05	2.4878
	Training samples	3106	31.00	22.00	25.97	2.4816
	Test samples	1332	32.00	23.00	26.22	2.4947

TABLE IV
RRMSE CRITERIA

RRMSE (%)	Forecasting power
< 10	Excellent
10 - 20	Good
20 - 30	Reasonable
> 30	Incorrect

TABLE V
MAPE CRITERIA

MAPE (%)	Forecasting power
< 10	Excellent
10 - 20	Good
20 - 50	Reasonable
> 50	Incorrect

TABLE VI
CONTROL HYPER-PARAMETERS FOR META AND BASE-MODELS

Model	Control Hyperparameters	
SVR-RBF (STACK)	Kernel	Radial
	Sigma	150
	Cost	0.1
xgBoost	Boosting Iterations	50
	L2 Regularization (λ)	0.1
	L1 Regularization (α)	0.001
	Learning Rate	0.3
MLP	Hidden Units layer1	3
K-NN	Neighbors	13
SVR-Linear	Kernel	Linear
	Cost	4

IV. RESULTS

In training phase, the hyper-parameters presented on Table VI was the best tunes for all base-learners and the meta-learner to obtain better results. The hyper-parameters were chosen by a Grid-Search.

The models performance measures to forecast one step-ahead the wind power generation are shown on Table VII. The best results for training and test set are stated in bold. In training phase, xgBoost presented a significant difference on the results compared to the others, it performed excellent in all performance measures. However, on test phase, STACK presented better results, which according to Tables IV and V, it had a good performance on RRMSE and an excellent performance on MAPE.

Furthermore, Figure 6 illustrates the SSE calculated for each approach. STACK showed to have the lower SSE between the models.

Figure 7a shows the complete dataset with the STACK prediction, spliting the dataset into training and test by the vertical blue line. Once dataset has many observations, it is quite difficult to analyze as a whole, so for a better visualization of the STACK forecasting performance, the Figure 7b shows the samples from 4200 to 4300.

Furthermore, Figure 8 illustrates the observed values of

TABLE VII
PERFORMANCE MEASURES OF THE BASE-MODELS AND STACKING ON TRAINING AND TEST SETS

Training set			
Models	RRMSE	MAPE	R^2
STACK	0.0980	0.0818	0.8152
xgBoost	**0.0648**	**0.0527**	**0.9219**
MLP	0.2389	0.2209	-
K-NN	0.1283	0.1062	0.6809
SVR-Linear	0.1310	0.1093	0.6728
Test set			
Models	RRMSE	MAPE	R^2
STACK	**0.1101**	**0.0934**	**0.6690**
xgBoost	0.1243	0.1072	0.5938
MLP	0.2445	2.4529e+35	0.0008
K-NN	0.1291	0.1160	0.6001
SVR-Linear	0.1112	0.0971	0.6588

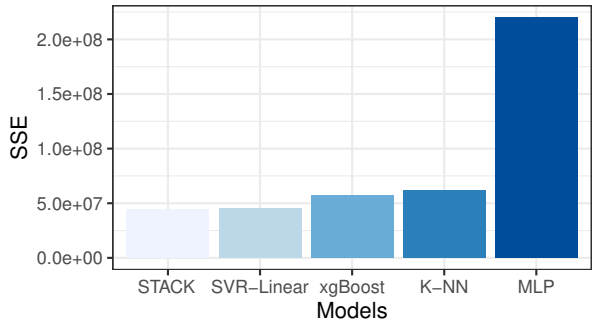
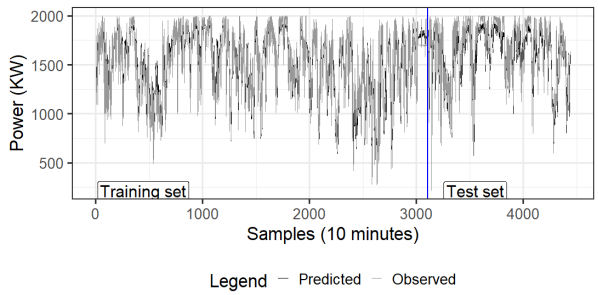
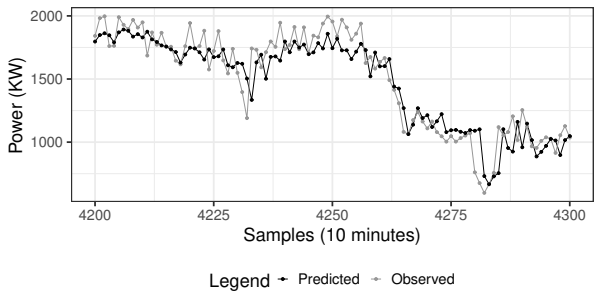


Fig. 6. Sum of squared errors of the models



(a) Prediction for whole dataset



(b) Samples from 4200 to 4300

Fig. 7. STACK predictions

the time series versus the predicted values for wind power generation obtained by STACK results both for training (red triangles) and test sets (black dots).

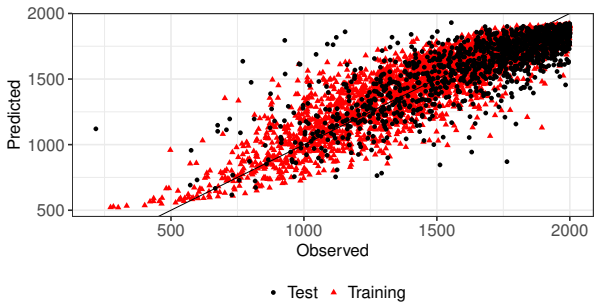


Fig. 8. Observed versus predicted values for wind power generation

In addition, comparing the results of the performance on Table VII, it can be seen that xgBoost had a better performance on training phase. However, in test phase STACk presented the black dots more centralized with fewer outliers. Both results, diagram and performance measures, shows that STACK prediction performed better than other models.

According to Friedman test, considering observations that predictions are different, there is statistically difference between the squared error (SE) for five adopted approaches (χ^2_4 = 1252.6, p-value < 0.05).

Moreover, the Figure 9 illustrates the comparisons between five approaches. In this representation, those algorithms that are not joined by a line can be regarded as different.

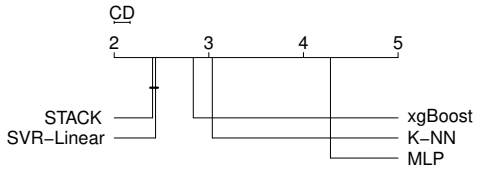


Fig. 9. Comparison of the CD of the predicted models

According to Figure 9, there is no statistical difference between STACK and SVR-Linear (CD = 0.16717, degrees of freedom = 6655, p-value < 0.05). However, STACK presents statistically lower error than others models.

Therefore, after conducing the experiments, STACK presented better results when compared with the base-models analyzed individually. Furthermore, the stacking ensemble presented better results on all performance measures tested, showing good and excellent performance on RRMSE and MAPE, respectively. Also, STACK presented the lower error in the statistical analysis compared to the others.

V. Conclusion

The paper proposed a stacking ensemble of 4 heterogeneous base-learner models and 1 meta-learner. The proposed framework was developed with the objective to forecast the wind power generation one observation ahead. The stacking ensemble was composed by xgBoost, MLP, SVR-Linear and K-NN, as base-learners in the first layer, and using SVR-RBF as meta-learner in the second layer. The models were compared using RRMSE, MAPE and R^2 criteria, as well as statistical tests. The results showed that stacking ensemble had a better performance than the other approaches with individually analysis in all performance measures criteria.

For future works is intend to adopt different combinations of models in both layers of the stacking ensemble. Increasing the number of base-learners and compare them would be a value contribution. Also, increasing the number of steps ahead to forecasting.

Acknowledgments

The authors would like to thank the company "Centrais Eólicas Renascença" for providing the data for this research. Moreover, this work was financed by National Council of Scientific and Technologic Development of Brazil – CNPq (Grants number: 303908/2015-7-PQ, 303906/2015-4-PQ, 404659/2016-0-Univ, 405101/2016-3-Univ), PRONEX '*Fundação Araucária*' 042/2018, Graduate Support Program of the Federal University of Paraná (PROAP-UFPR), and in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* - Brasil (CAPES) - Finance Code 001.

Furthermore, the authors would like to thank the Program Chairs and Reviewers for their constructive comments and recommendations, which have significantly improved the presentation of this paper.

References

[1] Brazilian Wind Energy Association, "2018 wind power generation annual report," May 2019. [Online]. Available: http://abeeolica.org.br/wp-content/uploads/2019/05/Boletim-Anual_2018.pdf

[2] M. Tahir, R. El-Shatshat, and M. Salama, "Improved stacked ensemble based model for very short-term wind power forecasting," in *53rd International Universities Power Engineering Conference (UPEC)*. Glasgow, UK: IEEE, 2018, pp. 1–6.

[3] Q. He, J. Wang, and H. Lu, "A hybrid system for short-term wind speed forecasting," *Applied Energy*, vol. 226, pp. 756–771, 2018.

[4] H. Demolli, A. S. Dokuz, A. Ecemis, and M. Gokcek, "Wind power forecasting based on daily wind speed data using machine learning algorithms," *Energy Conversion and Management*, vol. 198, pp. 1–12, 2019.

[5] H. S. Dhiman, D. Deb, and J. M. Guerrero, "Hybrid machine intelligent svr variants for wind forecasting and ramp events," *Renewable and Sustainable Energy Reviews*, vol. 108, pp. 369–379, 2019.

[6] H. Liu and C. Chen, "Data processing strategies in wind energy forecasting models and applications: A comprehensive review," *Applied Energy*, vol. 249, pp. 392–408, 2019.

[7] M. H. D. M. Ribeiro, V. H. A. Ribeiro, G. Reynoso-Meza, and L. S. Coelho, "Multi-objective ensemble model for short-term price forecasting in corn price time series," in *International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, July 2019, pp. 1–8.

[8] Y. Wang, D. Wang, N. Geng, Y. Wang, Y. Yin, and Y. Jin, "Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection," *Applied Soft Computing*, vol. 77, pp. 188–204, 2019.

[9] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[10] V. Vapnik, *The nature of statistical learning theory*. New York, USA: Springer Science & Business Media, 2013.

[11] M. Ring and B. M. Eskofier, "An approximation of the gaussian rbf kernel for efficient classification with svms," *Pattern Recognition Letters*, vol. 84, pp. 107–113, 2016.

[12] Q. Liu, C. Chen, Y. Zhang, and Z. Hu, "Feature selection for support vector machines with rbf kernel," *Artificial Intelligence Review*, vol. 36, no. 2, pp. 99–115, 2011.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, USA: ACM, 2016, pp. 785–794.

[14] S. Lim and S. Chi, "Xgboost application on bridge management systems for proactive damage estimation," *Advanced Engineering Informatics*, vol. 41, no. 100922, 2019.

[15] M. Khishe and H. Mohammadi, "Passive sonar target classification using multi-layer perceptron trained by salp swarm algorithm," *Ocean Engineering*, vol. 181, pp. 98–108, 2019.

[16] N. Messikh, S. Bousba, and N. Bougdah, "The use of a multilayer perceptron (mlp) for modelling the phenol removal by emulsion liquid membrane," *Journal of Environmental Chemical Engineering*, vol. 5, no. 4, pp. 3483–3489, 2017.

[17] T. M. Cover, P. Hart *et al.*, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[18] B. Saçlı, C. Aydınalp, G. Cansız, S. Joof, T. Yilmaz, M. Çayören, B. Önal, and I. Akduman, "Microwave dielectric property based classification of renal calculi: Application of a knn algorithm," *Computers in Biology and Medicine*, pp. 1–8, 2019.

[19] J. Strickland, *Predictive analytics using R*. Morrisville, USA: Lulu.com, January 2015.

[20] L. Hu and J. Cui, "Digital image recognition based on fractional-order-pca-svm coupling algorithm," *Measurement*, vol. 145, pp. 150–159, 2019.

[21] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 26, no. 2, pp. 211–243, 1964.

[22] M. Despotovic, V. Nedic, D. Despotovic, and S. Cvetanovic, "Evaluation of empirical models for predicting monthly mean horizontal diffuse solar radiation," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 246–260, 2016.

[23] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[24] P. Nemenyi, "Distribution-free multiple comparisons," in *Biometrics*, vol. 18, no. 2. Washington, DC, USA: International Biometric Society, 1962, p. 263.

[25] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. Jan, pp. 1–30, 2006.

[26] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[27] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2018. [Online]. Available: http://www.R-project.org/