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 (PPGEPIS), 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].

<table>
<thead>
<tr>
<th>Very short-term</th>
<th>Short-term</th>
<th>Medium-term</th>
<th>Long-term</th>
</tr>
</thead>
<tbody>
<tr>
<td>A few seconds</td>
<td>30 minutes</td>
<td>6 hours</td>
<td>1 day</td>
</tr>
</tbody>
</table>

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](image)

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Unit Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Power</td>
<td>KW</td>
</tr>
<tr>
<td>Input</td>
<td>Generator Bearing Temperature</td>
<td>Celsius</td>
</tr>
<tr>
<td>Input</td>
<td>Generator Bearing 2 Temperature</td>
<td>Celsius</td>
</tr>
<tr>
<td>Input</td>
<td>Generator Speed</td>
<td>RPM</td>
</tr>
<tr>
<td>Input</td>
<td>Wind Speed</td>
<td>m/s</td>
</tr>
<tr>
<td>Input</td>
<td>Wind Direction Absolute</td>
<td>Degrees</td>
</tr>
<tr>
<td>Input</td>
<td>Nacelee Direction</td>
<td>Degrees</td>
</tr>
<tr>
<td>Input</td>
<td>Ambient Temperature</td>
<td>Celsius</td>
</tr>
</tbody>
</table>

B. Methods

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

\[
    f(x) = w^T x + b
\]

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 \( x_i \) and corresponding coefficients \( \alpha_i \in \mathbb{R} \) are required. Hence, for a kernel SVM, the function

\[
    f(x) = \sum_{i=1}^{S} \alpha_i k(x_i, x) + b
\]

has to be evaluated for every new instance \( x \). Furthermore, Radial Basis Function Kernel (RBF kernel) is a default and recommended kernel function for SVM classifier.
The RBF kernel can be defined as

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

(3)

where $\|x - x'\|^2$ may be recognized as the squared Euclidean distance between the two feature vectors, and $\sigma$ 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

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

(4)

where $L(\theta) = l(\hat{y}_i, y_i)$ 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 $\hat{y}_i$ and real value $y_i$, $\Omega(\theta)$ is the regularization term, $\gamma$ is the learning rate, $T$ is the number of leaves in the tree, $\lambda$ 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

$$y = f_2\left( \sum_{j=1}^{N} w_{j} f_1\left( \sum_{i=1}^{n} h_{ij} X_i + b_j \right) + b_o \right)$$

(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 with RBF kernel, by using the linear kernel function [20], which can be define as

$$k(x, y) = x^T y.$$  

(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.

![ACF and PACF plots for lag verification](image)

(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
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].

\[
RRMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2},
\]

\[
MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|,
\]

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

\[
SSE = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2,
\]

where $n$ represents the number of observations of the training and test sets, $y_i$ is the $i$th value observed for series and $\hat{y}_i$ is the $i$th 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

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

in which FD is distributed according to $\chi^2$ with $k - 1$ degrees of freedom, $n$ observations and $k$ groups, and $R_j$ is the sum of the ranks for the $j$th 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)

\[
CD = q_{\infty,k,\alpha} \sqrt{\frac{k(k+1)}{6}},
\]

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 II

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Percentage</th>
<th>Number of observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed</td>
<td>100%</td>
<td>4438</td>
</tr>
<tr>
<td>Training</td>
<td>70%</td>
<td>3106</td>
</tr>
<tr>
<td>Test</td>
<td>30%</td>
<td>1332</td>
</tr>
</tbody>
</table>

**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](image)

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

\[
y(t) = f \{y_{t-1}, x_{it}\}
\]

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 $t$, $x_{it}$ is a matrix of $i$ inputs at time $t$, and $i$ is the number of inputs in the system.

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

\[
y(t) = f \{y_{t-1}, x_{it}\}
\]

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 $t$, $x_{it}$ is a matrix of $i$ inputs at time $t$, and $i$ is the number of inputs in the system.

\[
y(\lambda) = \begin{cases} 
\frac{\lambda^{\lambda-1}}{\log \lambda}, & \text{if } \lambda \neq 0 \\
\log y_i, & \text{if } \lambda = 0
\end{cases}
\]

where $\lambda$ is a parameter, possibly a vector, defining a particular transformation.

...
### TABLE III

**Summary of the Statistical Indicators of the Inputs and Output of the Dataset**

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

### TABLE IV

**RRMSE Criteria**

<table>
<thead>
<tr>
<th>RRMSE (%)</th>
<th>Forecasting power</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 10</td>
<td>Excellent</td>
</tr>
<tr>
<td>10 - 20</td>
<td>Good</td>
</tr>
<tr>
<td>20 - 30</td>
<td>Reasonable</td>
</tr>
<tr>
<td>&gt; 30</td>
<td>Incorrect</td>
</tr>
</tbody>
</table>

### TABLE V

**MAPE Criteria**

<table>
<thead>
<tr>
<th>MAPE (%)</th>
<th>Forecasting power</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 10</td>
<td>Excellent</td>
</tr>
<tr>
<td>10 - 20</td>
<td>Good</td>
</tr>
<tr>
<td>20 - 50</td>
<td>Reasonable</td>
</tr>
<tr>
<td>&gt; 50</td>
<td>Incorrect</td>
</tr>
</tbody>
</table>

### 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, splitting 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

<table>
<thead>
<tr>
<th>Models</th>
<th>Training set</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RRMSE</td>
<td>MAPE</td>
<td>$R^2$</td>
<td></td>
</tr>
<tr>
<td>STACK</td>
<td>0.0980</td>
<td>0.0818</td>
<td>0.8152</td>
<td></td>
</tr>
<tr>
<td>xgBoost</td>
<td>0.0648</td>
<td>0.0527</td>
<td>0.9219</td>
<td></td>
</tr>
<tr>
<td>MLP</td>
<td>0.2389</td>
<td>0.2209</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-NN</td>
<td>0.1283</td>
<td>0.1062</td>
<td>0.6809</td>
<td></td>
</tr>
<tr>
<td>SVR-Linear</td>
<td>0.1310</td>
<td>0.1093</td>
<td>0.6728</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Models</th>
<th>Test set</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RRMSE</td>
<td>MAPE</td>
<td>$R^2$</td>
<td></td>
</tr>
<tr>
<td>STACK</td>
<td>0.1101</td>
<td>0.0934</td>
<td>0.6690</td>
<td></td>
</tr>
<tr>
<td>xgBoost</td>
<td>0.1243</td>
<td>0.1072</td>
<td>0.5938</td>
<td></td>
</tr>
<tr>
<td>MLP</td>
<td>0.2445</td>
<td>2.4529e+35</td>
<td>0.0008</td>
<td></td>
</tr>
<tr>
<td>K-NN</td>
<td>0.1291</td>
<td>0.1160</td>
<td>0.6001</td>
<td></td>
</tr>
<tr>
<td>SVR-Linear</td>
<td>0.1112</td>
<td>0.0971</td>
<td>0.6588</td>
<td></td>
</tr>
</tbody>
</table>

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).

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 ($\chi^2 = 1252.6$, $p$-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.

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 conducting 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 “Cenrais 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” for providing the data for this research. 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


