TEMPORAL POST-PROCESSING OF WIND FORECAST USING ENSEMBLE OF MACHINE LEARNING METHODS

Diandra Akemi Alves Kubo

daakubo@inf.ufpr.br

Programa de Pós Graduação em Informática – UFPR

Centro Politécnico, 81.531-980, Paraná, Curitiba, Brasil

Rafael Toshio Inouye

Cesar Beneti

toshio@simepar.br

cesar.beneti@simepar.br

Sistema Meteorológico do Paraná – SIMEPAR

Rua Francisco H. dos Santos, 210, 8.1530-900, Paraná, Curitiba, Brasil

Alana Renata Ribeiro

alanar89@gmail.com

Programa de Pós Graduação em Métodos Numéricos em Engenharia – UFPR

Centro Politécnico, 81.530-900, Paraná, Curitiba, Brasil

Abstract. *Wind forecast accuracy is of extreme importance to several applications, ranging from wind energy generation monitoring to the aviation decision making process. The Weather Research and Forecasting Model (WRF) provides a 72-hour forecast of wind components. Although this model performs fairly well on a mesoscale, it can be biased and too general for specific areas, and therefore the post-processing of this kind of model is needed. A comparison of correction for the U and V components versus wind speed and direction is made in this work. For each variable, several tests were made using five different models plus the average bias correction. All models were trained with WRF data and observed values from five locations in Brazil from 12 September 2016 to 25 July 2017. Our method of forecast correction showed significant improvement over the original forecast in all cases and outperformed the average bias correction.*

Keywords: *Forecast Post-Processing, Machine Learning, Wind Forecast*

1 INTRODUCTION

Accurately knowing in advance wind conditions can be critical to the most diverse applications nowadays. Council et al. (1983) reported how wind shear can be a hazard to aviation, having already caused a plane crashed. The awareness of future wind conditions is imperative for a safe flight.

Knowing the wind forces in advance can also help in preparatory measures for wind erosion that can affect agricultural lands (Saxton, 1995).

However, the most prominent application of wind forecast is for the use of clean and renewable energies in large scale, that has been getting more attention over the past years. This can be seen as a direct result of scientific findings of climate change, mainly affected by greenhouse gas emissions (Carmo et al., 2017).

Besides being a way of reducing CO_2, wind power can also be a way to stimulate economic growth in areas where the wind speed is appropriate. As of 2005, the cost of wind turbines manufactured had decreased about 20% since 1995, and although not emission-free, wind power generation still is one of the best environmentally friendly alternative (Ackermann, 2005).

Notwithstanding all the benefits of wind power, it is not a trivial task to optimally manage it. Zhao & Zeng (2012) assert that "it is risky to introduce a large amount of wind energy into current power systems because the intermittency nature of wind can raise costs for regulation and hamper the reliability of the power system". For this reason, accurate wind forecast is absolutely necessary.

1.1 Justification

Important and advanced as it is, weather forecasts from numerical models are still liable to errors and systematic biases (Madadgar et al., 2014). Most approaches used in the existing literature use the bias correction or are at least similar to it (Moradkhani & Meier, 2010; Candille et al., 2010; Hashino et al., 2006), always trying to identify and quantify errors or to predict uncertainty. It is important to notice that all these works agree that the model post processing is essential.

In face of all that, we chose a machine learning approach for forecast correction, a field that has ben getting more and more attention in recent years. We provide models with the forecast and other variables about the time of the forecast, letting each model decide how to better do the post-processing correction.

2 METEOROLOGICAL CONCEPTS

Numerical weather prediction (NWP) is essetially an initial and boundary condition problem (Daley, 1993), which implies that better initial conditions will produce better prognostics. Most of the time changing initial conditions is tricky and hard to implement. One of the best ways to achieve better results is to correct the model output bias, the goal of this work. One of the most used NWP models is the Weather Research and Forecast model.

2.1 WRF

The Nacional Center for Atmospheric Research (NCAR) and other institutions maintain the processing core of the Weather Research and Forecast (WFR) model. This model is state-of-art, fully compressible, conservative and uses a mass vertical coordinate. The time integration scheme is a third-order Runge-Kutta and the spatial discretization is a second and sixth-order in an Arakawa-C grid (Skamarock et al., 2008).

The WRF processing procedure has three main steps. The first is the WRF Preprocessing System (WPS) that arranges the input data for the numerical simulation. After the WPS is done, the vertical interpolation of the meteorological fields to the model levels is calculated, creating the initial and boundary conditions. After this, the model itself is executed.

One main problem in the estimation of surface variables like wind speed is the representation of topography and terrain morphology, which despite the high resolution of only a few kilometers is still problematic (Santos-Alamillos et al., 2013). Probably for that reason, a positive bias over plains and valleys and a negative bias over hills and mountains have been reported (Jiménez & Dudhia, 2012).

Models like the WRF have around variables 150, and a portion of them represent the U and V components of the wind in different situations. And although the model outputs wind in this representation, applications often deal with wind speed and direction.

2.2 U and V components

Meridional (V) and zonal (U) components of wind are the standard output of WRF for wind forecast. The U component is positive for a west to east flow and the V component is positive for a south to north flow. Those components are computed in a staggered grid to avoid numerical instabilities known as odd-even (Neundorf, 2016). Therefore, these values of wind components are actually in the center of the model grid, a result of a mean among the grid edges. These components can also be obtained from a speed SPD and direction DIR point of view, as can be seen on Eq. 1 and Eq. 2. For both these equations, the wind direction must be in radians and the speed in meters per second.

$$U = -SPD * sin(DIR) \tag{1}$$

$$V = -SPD * cos(DIR) \tag{2}$$

2.3 Speed and Direction

Usually, the wind is represented as speed and direction instead of meridional and zonal components. The speed is the norm of U and V (Eq. 3). The direction is defined following the meteorological notation, i.e., the direction from which the wind is blowing, in relation to the true north. It increases clockwise from north when viewed from above (Burt, 2012). Its calculation can be seen on Eq. (4)

$$Speed = \sqrt{U^2 + V^2} \tag{3}$$

$$Direction = \frac{180}{\pi} * arctan(-U, -V) \tag{4}$$

The WRF model forecast data used on this work was processed and provided by SIMEPAR (Paraná Meteorological System).

3 METHODS

In order to better understand a processe being studied, scientific methodology urges us to collect data about said process of interest. Nowadays, this task has become easier, and the computational ability to process it advanced significantly (Alpaydin, 2014). In machine learning, algorithms that have the capacity of learning useful information from a provided database are used.

On a supervised machine learning algorithm, the goal is to obtain a method that can learn from labeled data (Raschka, 2015). This label can be a class that each instance of the data belongs in, or a continuous value representing each instance. When we have classes, we have a problem of classification, and when we have continuous values, regression.

To obtain knowledge about the data and how it behaves, we assume that instances of the same class or approximated value also behave similarly, so when dealing with new data it is safe to predict from what was learned previously (Quinlan, 2014).

In this work, we deal with the problem of regression as the variables we want to predict are continuous. The methods used for this task are explained in the next sections.

3.1 Linear Regression

The most basic approach to the problem of regression is the linear regression. One of the most adopted algorithms for this task even if just for a comparison is the Ordinary Least Squares (Dismuke & Lindrooth, 2006). This method carries this name because its aim is to find regression coefficients ω_1 and ω_0 that minimize the sum of the squared errors calculated on a target variable y_i and the regressed variable $f(x_i)$ when you have an attribute x_i, in every instance i of your data of size N (Tan et al., 2009). This can be seen on Eq. (5).

$$SSE = \sum_{i=1}^{N}[y_i - f(x_i)]^2 = \sum_{i=1}^{N}[y_i - \omega_1 x - \omega_0]^2 \tag{5}$$

To the purpose of finding said regression coefficients, a differentiation is made on Eq. (5), also equaling it to zero. After distributing the summation and solving the remaining equations, the coefficients are obtained (Larose, 2006).

3.2 M5P Tree

Decision and regression trees are hierarchical models of decisions and consequences. At each node of the tree that is not a leaf, a decision is made, as this model is a recursive partition of the instance space. On the leafs of the tree, if the task is to classify, there is a class; if it is regression, a continuous number (Rokach & Maimon, 2014). As Quinlan (1990) stated, tree-based methods are appropriated for data with continuous and discrete attributes.The M5P tree is often used for the task of regression, and Regression trees are built in three main stages: induction, pruning and smoothing.

On the tree induction, the goal is to find the rules that best separate the data according to the goal variable, so that at each tree level, more similar data reaches each node. At each node, a regression model is built. When there is a sufficient group of instances on a node or another metric is achieved, the M5P tree algorithm stops. Different than other regression trees that just

return a value, M5P tree has a regression function on its leafs (Zhan et al., 2011). To control overfitting, a pruning step is added to the tree model. In this step, each node regression model is recalculated with the attributes being dropped one by one to see if the expected error decreases, and when it does, nodes are pruned (Wang & Witten, 1997). Finally, a concern when it comes to regression is to compensate for discontinuities on the data and model, so the M5P tree has a final step of smoothing (Zhan et al., 2011). To smooth the final value reached at a tree leaf, a combination is made with each prediction of the path back to the root from the leaf (Wang & Witten, 1997).

3.3 Random Forest

A Random Forest model consists of a set of decision or regression trees. This model is often used because of how it measures attributes importance and similarities (Svetnik et al., 2003). Although it is a computationally expensive model, it usually achieves a good generalization even for small datasets (Ham et al., 2005).

According to the desired number of trees for the model, each tree is built using a different part of the available data for training. In addition, unlike trees such as the M5P (Section 3.2), the splitting criteria is not the best rule that minimizes the standard deviation tested in all attributes, but only on a set of randomly chosen attributes for that node. It is important to note that the built trees are not pruned. (Breiman, 2001; Liaw et al., 2002).

Liaw et al. (2002) states that after all trees are build, to predict new data for a regression problem, each tree prediction is collected and an average is obtained, which will be the final prediction.

3.4 Multi Layer Perceptron

The Perceptron is the most primitive form of neural networks, representing a single neuron learning from given examples by weighing good and bad predictions with a certain bias. However, a perceptron cannot express decisions that are not linear (Mitchell, 1997). With that in mind, the MultiLayer Perceptron was created, a neural network that is a system of interconnected perceptrons (neurons).

The neurons in this network are arranged in layers, and a neuron from one layer is connected to all other neurons from the next one. Each neuron will pass forward the result of the chosen function calculated on the neuron's input, which can be the environment input or the results from previous layers. The chosen function can vary depending on the problem being studied. This type of processing is called feed forward (Gardner & Dorling, 1998) because of this characteristic of always passing the results to the next layers.

To learn from the given data, the MultiLayer Perceptron uses the backpropagation algorithm (Hecht-Nielsen et al., 1988), which is a way of finding the best weights that connect the neurons on the network.

Gardner & Dorling (1998) states that an advantage to the multilayer perceptron is that it does not make any prior assumptions about the data distribution, different than other statistical approaches.

3.5 Support Vector Regression Machines

Support Vector Machines (SVMs) are hyperplane classification models, that seek to find an optimal hyperplane with maximal distance to the margins of each class. They are built by finding a solution that expands in terms of the data instances that lie on these margins, called support vectors. That is done by solving a constrained quadratic optimization (Hearst et al., 1998).

Using Support Vectors for regression, with the absence of instances for each class used to calculate margins distances, what we analyze is the loss function that will allow a better mapping of the features using the support vectors. The goal is to minimize the error of the predictor function on the mapped values and the target variable (Drucker et al., 1997).

3.6 Gradient Boosting

The combination of predictions from different models usually turns into a better prediction than any of the original ones (Ridgeway, 1999). This is the goal of ensemble models, as they combine different methods in the hope of finding room for improvement. A well-known class of ensembles do boosting, a widely applicable kind of model combination.

Tan et al. (2009) pointed out that boosting techniques consist of an iterative way of sampling on the training data in a manner that makes the classifiers focus on difficult data. This is done by using weak classifiers that are good in specific parts of the data, instead of having one that performs well on all of it.

The gradient boosting algorithm (Friedman, 2001) has a stagewise strategy, where on each stage the classifiers are trained to fit not only the predictors to the target variable but also the error the classifiers are yielding.

4 METHODOLOGY

The variables which had its correction performed were the wind components described on Section 2: U component, V component, speed and direction of the wind. In order to achieve reliable results, several tests were made with the training data, exploiting the variables as machine learning attributes, disregarding the physics in their relationships.

The ground truth observed data used in this work was available in five points, at two different altitudes, so our results are partitioned in 10 different results for each target variable. The data was collected from 12 September 2016 to 25 July 2017, every 15 minutes, with direction and speed values at both heights in each point. To obtain the U and V observed values, the equations from Section 2 were applied.

The WRF forecast covers a period of 3 days, with predictions for every 30 minutes, totalizing 144 moments of forecast. Due to this interval of 30 minutes, the observed data used were only observations that coincided with the time of forecast provided by the WRF model. One forecast of 144 moments per day was used.

Taking in consideration that we had forecast data for the period we had observed data, we had roughly 316 days of forecast with 144 moments for each point at two heights, which would sum up to approximately 91.000 instances for each point of interest and 455.000 total in our

dataset. This is only an approximation as sometimes data could not be collected in determined points.

Regarding the predictor variables, as it is suggested by Young (1999), the wind can be seasonal, so we added a variable signaling which season of the year is for that instance. We also know that the more away from the initial time the forecast is, the more likely it is to need a stronger correction, therefore we also added a variable of the distance from the initial forecast.

As the forecast distance and season of the year information we added were crucial to our work contribution, we made sure they were balanced on the training and test sets. We also made sure to not split a 3-day forecast between the training and test sets, making sure there was no data with a part in the training and the remaining on the test.

Let us divide these variables into two groups, the first with the U and V components and the other group with the speed and direction variables. For each group, the correction performance was compared when the predictors comprised of: 1) only the target variable forecast in addition to the forecast distance and season; 2) the target variable forecast and also the forecast from the other variable in its group in addition to the forecast distance and season; and finally, 3) the target variable forecast and all other variables forecast in addition to the forecast distance and season. Table 1 shows the predictors for each target variable in each test round.

Table 1: Predictors for each target variable in each test round

Target Variable	Test Round	Predictors
U	1st	U forecast, forecast distance and season
	2nd	U and V forecast, forecast distance and season
	3rd	U, V, Speed and Direction forecast, forecast distance and season
V	1st	V forecast, forecast distance and season
	2nd	U and V forecast, forecast distance and season
	3rd	U, V, Speed and Direction forecast, forecast distance and season
Speed	1st	Speed forecast, forecast distance and season
	2nd	Speed and Direction forecast, forecast distance and season
	3rd	U, V, Speed and Direction forecast, forecast distance and season
Direction	1st	Direction forecast, forecast distance and season
	2nd	Speed and Direction forecast, forecast distance and season
	3rd	U, V, Speed and Direction forecast, forecast distance and season

The metric adopted to compare performance throughout this work was the explained variance score, that can be seen on Eq. (6). In this equation, y represents the ground truth value of the target variable, $\hat{y}$ the predicted value and Var stands for the variance measure. The closer

to 1 this score is, the better is the performance.

$$explained\_variance(y, \hat{y}) = 1 - \frac{Var(y - \hat{y})}{Var(y)} \tag{6}$$

Further on we divided our data into a training and a test set, with 33% of the data for testing and the remaining for training. All methods from Section 3 were used either through the programming language Python, with the Scikit-Learn package (Pedregosa et al., 2011) that provides machine learning tools or by means of the Weka software (Witten et al., 2016).

For each test round, we train our five regressors: Support Vector Regression, Random Forest, Multi Layer Perceptron, Linear Regression and M5P Tree. Finally, after we have 5 different regressors trained, we used their predictions as input for an ensemble method. The chosen ensemble method for this work was the Gradient Boosting.

5 RESULTS AND DISCUSSION

Our initial results can be seen in Table 2. This table shows the explained variance values for each of the target variables at each test round. The methods of Linear Regression (LR - Section 3.1), M5P tree (Section 3.2), Random Forest (RF - Section 3.3), MultiLayer Perceptron (MLP - Section 3.4) and Support Vector Regression (SVR - Section 3.5) were trained and tested.

Table 2: Initial Results, explained variance of best method.

	Test Round		
Target Variable	**1st**	**2nd**	**3rd**
U	0.62318595 (SVR)	0.6368428 (M5P)	0.64017916 (RF)
V	0.58117256 (M5P)	0.59938795 (RF)	0.60253958 (RF)
Speed	0.5199313 (M5P)	0.54444089 (M5P)	0.54997971 (RF)
Direction	0.61638534 (M5P)	0.62768586 (M5P)	0.63298906 (M5P)

What we can infer from Table 2 is that initially, the M5P tree handled very well the correction, losing only in the U component correction to the SVR. But as the tests progressed and more variables were added as predictors, we can see the Random Forest algorithm is able to better predict the corrected forecast. We also note that the third round of test, the one with more predictors, is where the performance was higher for all target variables.

For that reason, in Fig. 1 we can see the explained variance score distribution for each method. As was explained in Section 3, the closer to 1 the results are, the better the method performed.

We can see a clear difference between any of the methods and the bias correction. And in order to compare the best method, the bias correction, and the original forecast, we also have the plots of the average score of these 3 approaches for each forecast time. The plots for each target variable can be seen in Fig. 2.

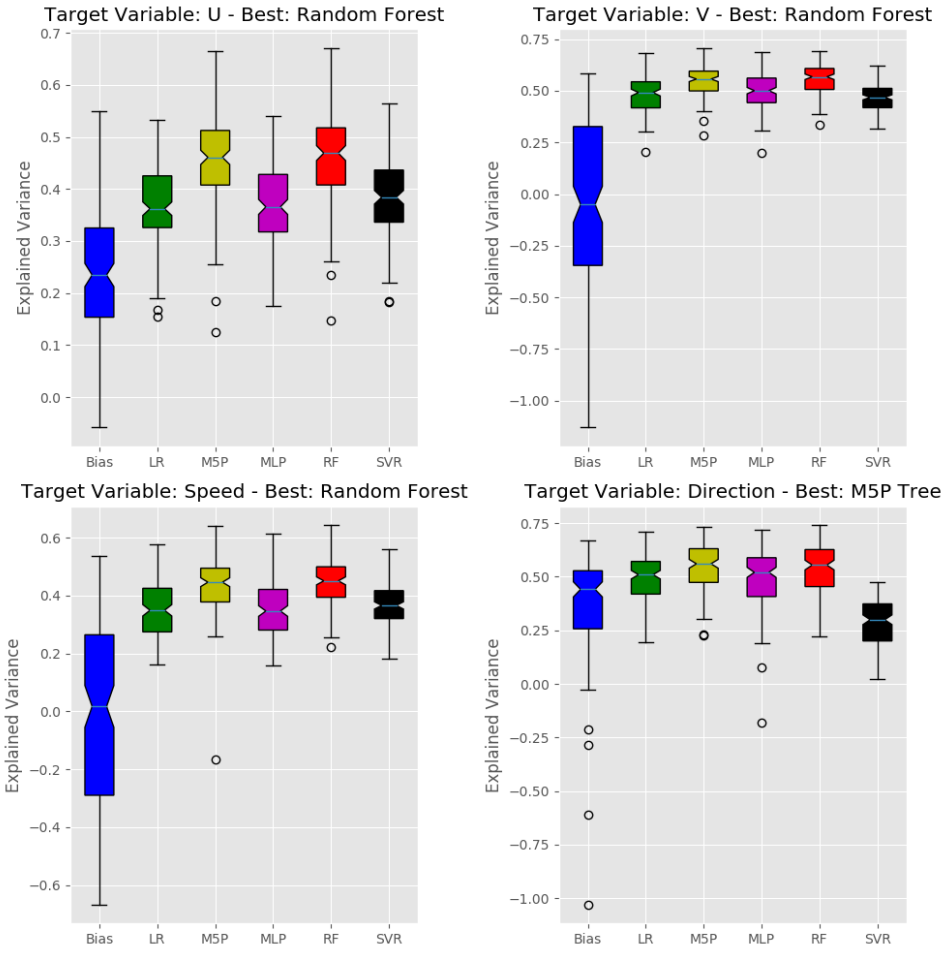

Figure 1: Third round of tests where the best results were achieved by a method alone. Each subplot shows the score distribution for each method trained and the bias correction.

The results achieved at this point were good compared to other approaches, but we wanted to see if there was room for improvement. Therefore, the next step we took was splitting our data once more, this time the test dataset, in order to get the predictions from each regressor and feed it to a boosting regression method. The results of this step can be seen on Table 3.

Table 3: Explained variance of corrected forecast after ensemble.

	Test Round		
Target Variable	**1st**	**2nd**	**3rd**
U	0.66382903	0.68493536	0.68143604
V	0.62158359	0.64559839	0.64235024
Speed	0.55938822	0.58295944	0.58760735
Direction	0.67597393	0.68150216	0.69018498

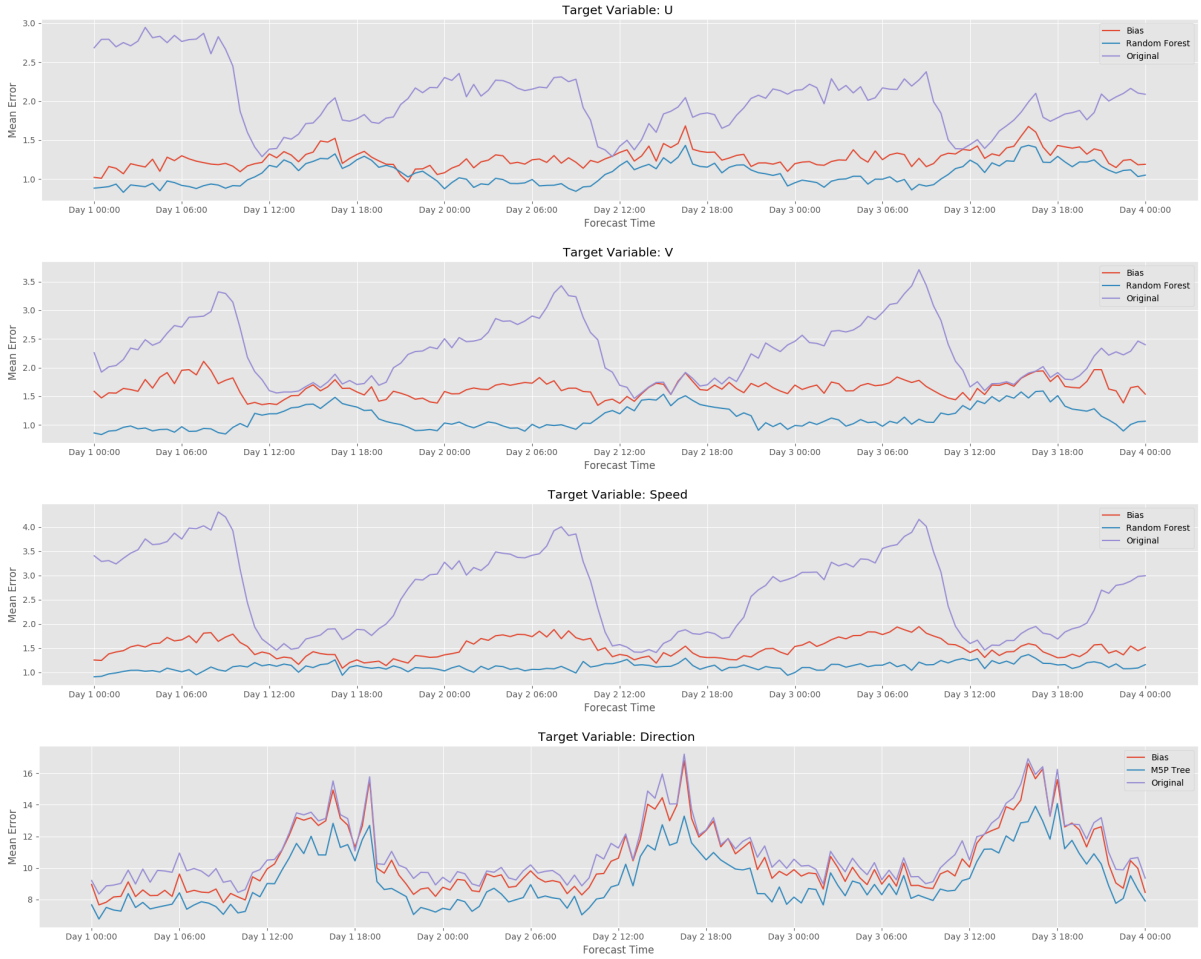

Figure 2: Mean error comparison by forecast time between the best method for the round, the bias correction and the original forecast.

The final comparison by target variable of the best method alone and the ensemble of all methods can be seen on Fig. 3.

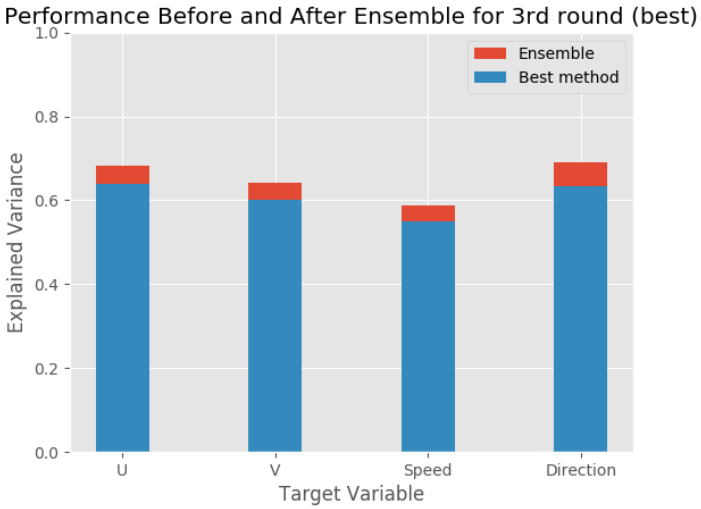

Figure 3: Performance comparison

6 CONCLUSION

Wind forecast correction is a well-established stage for processes that depend on wind forecast. With that in mind, we undertook the task of doing so with a machine learning approach.

We did several tests in order to get a full picture of the possibilities available when it came to predictors and approaches. The Support Vector Regression and the M5P tree did better when there were only a few predictors. In the other hand, Random Forests did well when there were more.

As a last resort, we used a boosting ensemble technique to evaluate if performance would increase. The results we obtained showed that the boosting technique was better in all case scenarios, but not by far.

We also noted that the WRF model consistently has the worst predictions for the 9th hour of each day of forecast, while the corrections methods used do generally well in all forecast times, learning how to deal with peaks of wrongful predictions like these.

REFERENCES

Ackermann, T. (2005). *Wind power in power systems*. John Wiley & Sons.

Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.

Breiman, L. (2001). Random forests. *Machine learning*, vol. 45(n. 1), pp. 5–32.

Burt, S. (2012). *The weather observer's handbook*. Cambridge University Press.

Candille, G., Beauregard, S., & Gagnon, N. (2010). Bias correction and multiensemble in the naefs context or how to get a "free calibration" through a multiensemble approach. *Monthly Weather Review*, vol. 138(n. 11), pp. 4268–4281.

Carmo, F., Martins, J. F., & Sănduleac, M. (2017). A methodology to assess home pv capacity to mitigate wind power forecasting errors. In *Young Engineers Forum (YEF-ECE), 2017 International* (pp. 47–52).: IEEE.

Council, N. R. et al. (1983). *Low-altitude wind shear and its hazard to aviation*. National Academies Press.

Daley, R. (1993). *Atmospheric Data Analysis*. Cambridge Atmospheric and Space Science Series. Cambridge University Press.

Dismuke, C. & Lindrooth, R. (2006). Ordinary least squares. *Methods and Designs for Outcomes Research*, vol. 93, pp. 93–104.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., & Vapnik, V. (1997). Support vector regression machines. In *Advances in neural information processing systems* (pp. 155–161).

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, (pp. 1189–1232).

Gardner, M. W. & Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, vol. 32(n. 14), pp. 2627–2636.

Ham, J., Chen, Y., Crawford, M. M., & Ghosh, J. (2005). Investigation of the random forest framework for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43(n. 3), pp. 492–501.

Hashino, T., Bradley, A., & Schwartz, S. (2006). Evaluation of bias-correction methods for ensemble streamflow volume forecasts. *Hydrology and Earth System Sciences Discussions*, vol. 3(n. 2), pp. 561–594.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, vol. 13(n. 4), pp. 18–28.

Hecht-Nielsen, R. et al. (1988). Theory of the backpropagation neural network. *Neural Networks*, vol. 1(n. 1), pp. 445–448.

Jiménez, P. A. & Dudhia, J. (2012). Improving the representation of resolved and unresolved topographic effects on surface wind in the wrf model. *Journal of Applied Meteorology and Climatology*, vol. 51(n. 2), pp. 300–316.

Larose, D. T. (2006). *Data mining methods & models*. John Wiley & Sons.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, vol. 2(n. 3), pp. 18–22.

Madadgar, S., Moradkhani, H., & Garen, D. (2014). Towards improved post-processing of hydrologic forecast ensembles. *Hydrological Processes*, vol. 28(n. 1), pp. 104–122.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill Education.

Moradkhani, H. & Meier, M. (2010). Long-lead water supply forecast using large-scale climate predictors and independent component analysis. *Journal of Hydrologic Engineering*, vol. 15(n. 10), pp. 744–762.

Neundorf, R. L. (2016). Aplicação do método multigrid paralelizado em gpus no modelo wrf. Master's thesis, UFPR, Curitiba.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830.

Quinlan, J. R. (1990). Decision trees and decision-making. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2), 339–346.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Raschka, S. (2015). *Python machine learning*. Packt Publishing Ltd.

Ridgeway, G. (1999). The state of boosting. *Computing Science and Statistics*, (pp. 172–181).

Rokach, L. & Maimon, O. (2014). *Data mining with decision trees: theory and applications*. World scientific.

Santos-Alamillos, F. J., Pozo-Vázquez, D., Ruiz-Arias, J. A., Lara-Fanego, V., & Tovar-Pescador, J. (2013). Analysis of wrf model wind estimate sensitivity to physics parameterization choice and terrain representation in andalusia (southern spain). *Journal of Applied Meteorology and Climatology*, vol. 52(n. 7), pp. 1592–1609.

Saxton, K. (1995). Wind erosion and its impact on off-site air quality in the columbia plateau—an integrated research plan. *Transactions of the ASAE*, vol. 38(n. 4), pp. 1031–1038.

Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D. M., Duda, M. G., Huang, X.-Y., Wang, W., & Powers, J. G. (2008). *A Description of the Advanced Research WRF Version 3*. NCAR, Boulder, Colorado, USA.

Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., & Feuston, B. P. (2003). Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, vol. 43(n. 6), pp. 1947–1958.

Tan, P.-N., Steinbach, M., & Kumar, V. (2009). *Introdução ao datamining: mineração de dados*. Ciência Moderna.

Wang, Y. & Witten, I. H. (1997). Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*: Springer.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Young, I. (1999). Seasonal variability of the global ocean wind and wave climate. *International Journal of Climatology*, vol. 19(n. 9), pp. 931–950.

Zhan, C., Gan, A., & Hadi, M. (2011). Prediction of lane clearance time of freeway incidents using the m5p tree algorithm. *IEEE Transactions on Intelligent Transportation Systems*, vol. 12(n. 4), pp. 1549–1557.

Zhao, L. & Zeng, B. (2012). Robust unit commitment problem with demand response and wind energy. In *Power and Energy Society General Meeting, 2012 IEEE* (pp. 1–8).: IEEE.