Reconstitution of Weather Time Series with an Analog Ensemble Model

Maycon Meier dos Santos - a40928

Bragança
2019

Maycon Meier dos Santos

Reconstitution of Time Series with an Analog Ensemble Model

Dissertation presented to the Polytechnic Institute of Bragança as a Requirement for the Master of Science Degree in Industrial Engineering (Mechanical Engineering branch), in the scope of double degree with the Federal Technologic University of Paraná.

Supe rvised by:

Carlos Jorge da Rocha Balsa

Carlos Veiga Rodrigues

Francisco Augusto Aparecido Gomes

Bragança
2019

Acknowledgments

I am very grateful for the Federal Technological University of Paraná (UTFPR), for all the opportunities I have had along my undergraduate and graduate studies. I also thank the Polytechnic Institute of Bragança (IPB) for receiving me in Portugal and for providing me with all the support I needed.

My most honest gratitude for Prof. Carlos Jorge da Rocha Balsa and Prof. Carlos Alberto Veiga Rodrigues for all the advising process. Both professors were of extreme importance on helping me understanding the scientific and programming problems I faced and helped me focus when I lost track of the direction of the work.

I thank Prof. Francisco Augusto Aparecido Gomes, who not only advised me during this dissertation but was also an important advisor along my whole undergraduate studies at UTFPR.

I also thank Prof. José Carlos Rufino Amaro for all the help that you provided to solve the computational difficulties that were found along the way. I'm sure it would have been much harder to finish this dissertation without such assistance.

A special greet to my American friend Jake Betzer for reviewing this dissertation and helping me with my silly English mistakes.

A big shout out to all the friends that Bragança has given me, for all the good experiences we have had as well as the lessons you have made me learn. You helped life here to be smoother.

Finally, I want to thank my family, specially my mom. For supporting me as I put myself in crazy life paths and making sure I know, I have a safe harbor to get back to.

“If you are not willing to be a fool, you cannot become a master. “

Jordan Peterson

ABSTRACT

The numeric weather prediction (NWP) that is currently used is based on global circulation models (GCM), which may be used for weather forecasting within horizons of 15 days, commonly. Yet, GCM lacks the spatial resolution required for engineering applications such as wind energy. Additionally, weather forecasts and hindcasts are often affected by phase errors. This study presents the use of a post-processing technic applied to the forecasting of weather time series. The technic is based on identifying analog ensembles from another time series of observations and using these to refine the forecast. To evaluate the skill of the method it was applied to ten weather stations. The focus of the study is to create data for reanalysis in places that lack weather measurements. To be able to evaluate the skill of the method, data from one station was used to forecast six variables at another station. This study used five years of training data to predict two years of forecast. As the analysis required a significant computational power, the studies were divided into two major approaches. The first approach had only one variable in the training period. The results were good for the variables that are easier to predict but had poor results in predicting variables with high level of abrupt changes. The second approach used multiple variables for the training period. The results were found to be significantly better. Although quantitatively there is error in the forecast characterized by a mean absolute error of 0.49 m/s for the wind speed, qualitatively the forecast was able to follow the behavior of the observed curve. It was found that the method can be very sensitive to the initial calibration, which may hinder the results.

Key words: Analog Ensemble; Weather Forecast; Time Series; Post-processing method.

RESUMO

Os modelos de previsão meteorológica atualmente utilizados são baseados no modelo de circulação atmosférica global. Embora este modelo seja altamente eficiente para previsões de curto prazo é pouco eficiente para previsões de longo prazo, devido ao acumulo sistemático de erros. O presente trabalho utiliza uma técnica de pós processamento aplicada à previsão de séries temporais. A técnica utilizada baseia-se no uso de conjuntos análogos que refinam os resultados. O método foi avaliado através de sua aplicação a estações meteorológicas. O foco do estudo é a geração de *data* para reconstrução de series em locais que não possuam estações meteorológicas. O método foi aplicado de forma que a previsão para a estação A fosse gerada através dos dados da estação B. O estudo utilizou cinco anos de dados para treinamento, e a geração de dois anos de previsões. As análises realizadas demandam de um poder computacional relativamente alto e, portanto, o estudo foi divido em duas partes. Na primeira, o período de treinamento foi gerador por uma única variável. Os resultados foram relativamente bons para as variáveis consideradas de fácil previsão, embora não tenham sido satisfatórios para as variáveis que possuam altos índices de mudanças bruscas. Na segunda análise, múltiplas variáveis compuseram o período de treinamento. Os resultados foram significativamente superiores. Embora as previsões não possuam 100% de precisão, a curva gerada foi capaz de manter o padrão da curva observada em todo o período. Observou-se que o método é eficiente embora bastante sensível à calibração inicial de suas variáveis.

Palavras chave: Conjuntos Análogos; Previsão do tempo; Séries temporais; Método de pós processamento; Previsão Meteorológica numérica.

SUMMARY

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Abbreviations

AnEn	Analog Ensemble
ANN	Artificial Neural Network
ARE	Absolute Relative Error
BIAS	Average of signed errors
ECMWF -EPS	European Center for Medium Range Weather Forecast - Ensemble Prediction System
EMOS	Ensemble Model Output Statistic
FGGE	First GARP Global Experiment
GAM	General Additive Model
GCM	General Circulation Models
HWRF	Hurricane Weather Research and Forecast
LEPS	Limited-area Ensemble Prediction System
LOES	Local Polynomial Regression Filling
MAE	Mean Absolute Error
MAPE	Mean Average Percentage Error
NDBC	National Data Buoys Center
NOAA	National Oceanic and Atmospheric Administration
NWP	Numeric Weather Prediction
PDF	Probabilistic Density Function
QR	Quantile Regression
RMSE	Root-mean Square Error
RPE	Relative Percentage Error

Variables

E_i	Error
A_τ	Analog at a valid past time
F_t	Numerical deterministic forecast at future time
k	Half of the number of additional times computed
n	total number of elements
N_v	Number of Physical Variables
σ	Standard deviation
SK	Skill Score
w	Weight of each variable
x_i	Real/Observed Value
x_{mean}	Mean value of the observed values
y_i	Forecasted value
V_s	Station weight
d	Distance between the sites
Δh	Altitude difference between the sites
μ	Humidity
G_b	Geographic Barriers
t_p	Dew point
I	Inertia

Chapter 1

INTRODUCTION

Time series are an important tool that is heavily used in modern society with a vast range of applicability that includes production management and control, stock´s markets behavior and weather analysis. The main reason for using such a tool is to be able to forecast the behavior of those time series, which is directly related with the capacity of generating profits and solving shortage problems.

The use of forecasting techniques for weather prediction is of great importance due to its direct impact in people lives. That is, by providing information to guarantee safety in situations of drastic weather conditions, allowing safe flights and road drives, and by providing information for decision-making of the implantation of wind and solar power plants (Storm, Dudhia, Basu, Swift, & Giammanco, 2009). The techniques for weather prediction have had great development since the first scientific approaches into the problem, especially after the advancement of technology that finally allowed the processing of more complex and sophisticated attempts into the problem. The biggest issue with the current forecasting systems is related to the uncertainties that are mostly generated by imperfect initial variables inputs that cannot be eliminated (Junk, Monache, Alessandrini, & Cervone, 2015).

Due to the continuous increase of demand for energy in the world and the necessity of reducing the emission of greenhouse gases, renewable energy sources such as wind and solar energy have had a great increase in the number of installed plants. According to the Global Wind Energy Council (GWEC) the total production of wind energy should reach 320 GW by 2020, which will represent a growth of 13% when compared to 2013, with over 25% of that capacity installed in countries of the European Union (GWEC, 2017). The European Wind Energy Association (EWEA) has also stated

that wind power should be responsible for over 12.7% of the total energy consumption in European Union at the same year (EWEA, 2020).

Since it is not possible to control the wind conditions over time, the outputs of wind power plants are not constant. This condition creates another economic challenge for the use of this energy source, since energy companies need to provide the amount of energy predicted to avoid paying fines. Therefore, to be able to have a good energy production plan these companies need to rely on forecasts (Monteiro C et al, 2009). The forecasts in current use are mainly classified in very short-term forecast (up to 9 hours), short-term forecasts (up to 72 hours) and medium-term forecast (up to 7 days). These groups of forecasts are used in the different levels of power production management, as presented in Table 1 (Kim & Hur, 2018). Longer period forecasts could be useful for the management, but the current forecasts do not provide reliable predictions for longer periods.

In order to improve the quality of the forecasts accuracy, post-processing techniques have been developed with a variety of different approaches. The Analog Ensemble technic is a relatively new one and has been presented with a metric calculation tool by Delle Monache et al (2011). This technique has shown great results. It has the ability to use ensembles to make deterministic forecasts and measures of the forecasts uncertainties. Also due to the use of ensembles this technique is not limited by time horizons, and can be applied with good results for long-term forecasts. The limitation of this technique is its inability to predict rare events and abrupt changes but since it uses machine training, it is possible to optimize the results by using proper training data.

1.1 Objectives

The Analog Ensemble post-processing method is still a relatively new method that has only been heavily studied by a specific group of researches. Therefore, this study aims to reanalyze the efficiency of the method and to provide a new application for the method.

Some of the specific objectives of this study are listed below:

1. Analyze literature and research over the Analog Ensemble method;
2. Implement a logic systems for the development of a programming script for the method;

3. Create a script to analyze a single variable with the method;
4. Create a script to analyze multiple variable with the method;
5. Apply the method to Hindcast data for places located between observation stations.

Table 1 – Applications of forecasts by time horizon
Source: Kim and Hur, 2018

Horizon	Time	Applications
Very Short-Term	Up to 9 hours	Intraday/Real-time market/operations
		Ancillary service management
		Transmission congestion management
		Regulation actions
Short-Term	Up to 72 hours	Day-ahead market/operations
		Maintenance planning of power system lines and wind farms
		Transmission congestion management
		Economic Dispatch and Unit Commitment
		Reserve scheduling
Medium-Term	Up to 7 days	Maintenance planning of power system lines and wind farms
		System expanding planning
		Optimal operating cost
		Feasible study for design of VGR

1.2 Outline of the Dissertation

This dissertation is divided into six chapters; the content of each one is described below.

Chapter 1 presents an introduction to this study, along with its objectives and the formulation of the problem. The structure of the dissertation is also presented.

Chapter 2 presents a review of the literature over numeric forecasting. The chapter starts with a brief historic review of the development of this science, and ends by detailing the Analog Ensemble method, which is the focus of this study.

Chapter 3 presents the methodology used for developing this work. The sources for the data presented in the development of this work are described in this section. It also presents the methods used to model the statistical techniques for the purposes of this study.

Chapter 4 presents the first approach over the problem by isolating a single physical variable to simplify the analysis and to evaluate its results. In this section six physical variables are individually analyzed and the results are displayed.

Chapter 5 explores a deeper analysis by combining the variables and using the full method. The chapter shows the results obtained along with a comparison of the results with those of single variables, both in accuracy and the processing power required.

Chapter 6 presents a proposal of future research over the applicability of the analog ensemble technique.

Chapter 7 summarizes the results obtained and lists the conclusions acquired. Suggestions for future works are also presented in this section.

Chapter 2

LITERATURE REVIEW

This chapter presents a review of the current state of art for existing models of weather forecast. The chapter starts with a historic review that illustrates the development of the techniques along the second half of the 20th century. Then, the analog ensemble technique for post-processing predictions is described in detail and the applications for this tool are listed. Final section presents the techniques for measuring error that were later applied for evaluating the results of this study.

2.1 Brief Historic Review

During the World War II meteorology started receiving more attention as it was attributed to providing important information for air attacks planning, but the computer technology and even the mathematics were not able to provide accurate predictions at that time. After the war was over, mathematical models started being developed for weather prediction (Oliveira & Florenzano, 2006). John von Neumann was a mathematician who provided great improvement in the models development and in the programing computer technics that nowadays enables us to have 48 hour accurate weather predictions (Moura, 1996).

When the war was over, in the middle 1940´s, the meteorological system was strongly dependent on the abilities of each operator, and it did not use all the physics correlations through mathematical models to provide the forecast, as it should be. Instead, weather predictors used a two-step analysis proposed by Vilhelm Bjerknes in the beginning of the 20th century. The first step was to determine the initial conditions by

observations of the atmosphere. The second was to use principles of physics to calculate future weather using the equations of mass, momentum and energy conservation (Sampaio & Dias, 2014). Also, the set of tool that were used were very sensitive to the experience of the operator that was collecting the weather observations. That is the reason why weather prediction at that time can be referred as art rather than science.

In 1955, with the support of the NOAA (National Oceanic and Atmospheric Administration), computers started being used to generate forecast maps. This was only possible with the models developed by Von Neumann and the Joint Operational Numerical Weather Prediction Unit, an organization created to help the development of the sciences surrounding the weather prediction (Charney, FjÖrtoft, & Neumann, 1950). The model used in these simulation was the quasi-geostrophic motion system proposed by Jule Charney. This system is based on the atmosphere pressure gradient and the Coriolis force and considering that both forces are almost in balance (Phillips, 1954). Four simulations of 24 hours were ran and the results were, for the first time, satisfying. The biggest problem with these simulations was the poor computational power available. The 24 hours simulation used to take almost 24 hours to be processed. Therefore, they were of no practical use, even though they had a huge theoretical value since they showed that it could be possible to have large scale predictions (Sampaio & Dias, 2014).

As the research continued through the second half of the 20th century, the knowledge over the atmosphere dynamics and climate elements increased. The mathematical models had significant improvements, as well as the computational power available. Table 2 presents some of the significant research developed after the first simulations back in the mid 1950´s.

The Global Circulation model utilizes a tri-dimensional mesh (latitude, longitude, and altitude) and represents the current method used for weather prediction, which considers all events happening around the globe. Figure 1 illustrates the grid used by the Global Circulation method. This model is considerably efficient and can predict, with accurate precision, up to seven days, although it has a limitation of 100 km resolution. To solve that, the downscaling method was created. This method, using regional simulators, depics a very accurate forecast for specific regions.

The current models are well developed but they still have some limitations. From a mathematical perspective, the numerical methods need to use parameterization for some of the variables, such as: the viscosity, the boundary layer, the radiation and the clouds convection. These parameterizations are probably the biggest uncertainty generator for the models. From a practical perspective, the models require very high computational power, which can represent a financial limitation. In some cases, they still require many meteorological stations to provide initial conditions to the models, and that can provide uncertainty in locations far from these stations.

Table 2 - Further contributions in the 20th century
Source – Sampaio and Dias (2014)

Author(s)	Year	Contribution
Manabe e Bryan	1969	Implemented Ocean Multilayer model to the atmospheric GCM´s´
Fels e Schwarzkopf	1975	Radiation of long waves
Manabe e Wetherald	1975	Improved the modeling of the greenhouse gases
Cubasch et al	1994	Integration of the ocean and atmosphere predictions
Miyakoda and Sirutis	1997	Understanding of the boundary layers

2.2 Analog Ensemble

Lorenz (1965) described that, because of the chaotic behavior of the atmosphere, even very sophisticated mathematical models would struggle in long-term weather prediction. According to his calculations, the predictions are limited to a maximum of 16 days with good accuracy. However, as stated before, the current models have problems with imperfect initial conditions and an error growth provided by variables that require parameterization. Altogether, these elements provided a high cost for continuous recalculation of the models, and are not reliable for long terms predictions.

Proper weather prediction can be a key factor in the decision making of investments that rely on climate, such as power generation and agriculture. Vanvyve *et al* (2015) describes the necessity of good long-term prediction for Wind Power industry,

which is in a fast use increase. To provide better efficiency, bigger turbines are being use, which increases the pre-construction costs, and therefore demand a better estimation of payback to justify the investments.

For the purpose of long-term weather predictions that are more reliable for decision making, forecast probability density functions (PDF) have shown better results. In 1969 a stochastic dynamic model of PDF was proposed, but that model requires unviable computational power (Epstein, 1969). The ensemble technique was later proposed by Leith (1974), to solve the stochastic dynamic forecast using the Monte Carlo probability approximation.

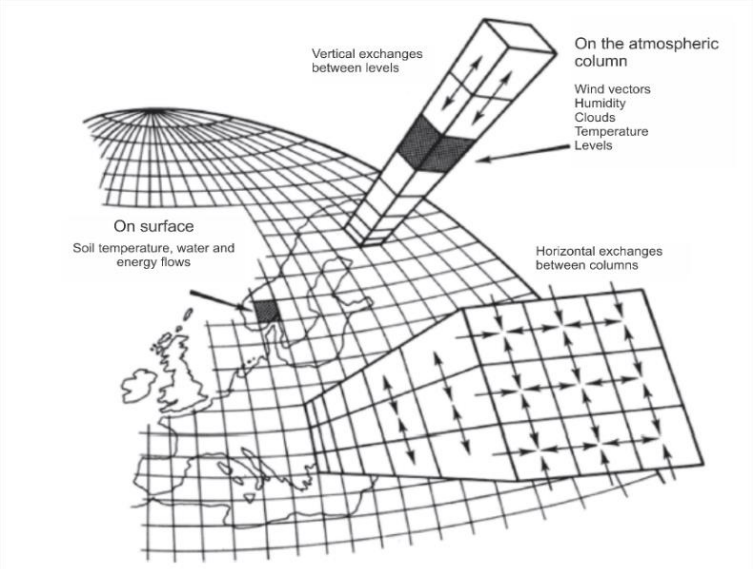


Figure 1 - Horizontal grid of the global numeric model
Adapted from :Sampaio & Dias (2014)

From that, the Analog Ensemble (AnEn) method has shown successful results in its prediction (Delle Monache *et al*, 2013). This method consists in using a set of past observations to identify the best analogs to a pre-established prediction, with the will of reaching a higher accuracy. Monache´s methods seeks to solve the distribution of function $[f(.)]$ that corresponds to a predictive model, showed in Equation 1.

$$f(y|\boldsymbol{x}^f) \tag{1}$$

$$x^f = (x_1^f, x_2^f, \dots, x_n^f) \tag{2}$$

Where:

y = Observed future value of the predictand variable;

x^f = vector of k predictors from the deterministic prediction;

Vanvyve *et al* (2015) describes the use of AnEn as a tree-stage process, where all stages are need to be processed in sequence but can be processed independently for all points that will be forecasted. These stages consists of the analog trend, the analog search window, and the analog ensemble member constitution. The analog trend retrieves the historical value of the variables (such as wind speed) centered on time *t* of the prediction. In this stage, the predictor need to be pre-selected based on anticipated correlations. This stage is composed by a prediction that is intended to be improved. The stage of analog search window, which is also referred as the training period, focus on finding cases with analogous conditions inside the historical data and comparing with those in the target window. The last stage is the selection of the best analogous cases and returns their values. Delle Monache et al. (2013) also mentions that the reliability of the model relies on the fact that the third stage returns not only the variable value but also its estimated error, which can play a major roll on decision-making.

Figure 2 presents an overview of the process, developed by Vanvyve (2015). In his study, Vanvyeve uses the AnEn to reconstruct a set of observed data that had a missing data points, using a set of historical data. The first step for reconstructing the time series is to take a point in the historical data from the period to reconstruct. This point needs to be collected along with the previous and next "*k*" elements of the series, where "*k*" is the size of the analog, as defined in equation 3. Step 2 is to compare the selected pack of elements with all the elements in the training data. From this comparison the best analogs will be taken. Once you know where the best analogs are in the historical series, Step 3 is to select the equivalent elements in the observation series, which are the best analogs for the predicted time to be reconstructed. The result can then be set as the best analog or the mean of the "*Na*" best analogs.

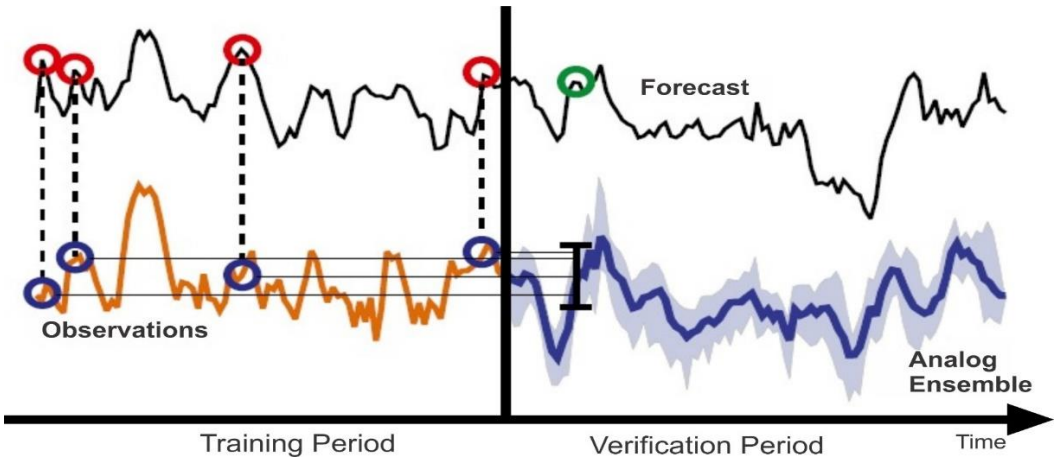


Figure 2 - Analog Ensemble method sketch
Adapted from Keller et al (2017)

In order to create the analogs and compare them with the training period data, Delle Monache et al (2011) developed the following metric:

$$F_t, A_\tau = \sum_{i=1}^{N_v} \frac{w_i}{\sigma_{f_i}} \sqrt{\sum_{j=-k}^{k} (F_{i,t+j} - A_{i,\tau+j})^2} \quad (3)$$

Where:

F_t = Current numerical deterministic forecast at future time t;

A_τ = Analog at same time and place valid at past time t';

N_v = Number of physical variables;

w_i = weight of each physical variable;

σ_{f_i} = Standard deviation of the training time series;

k = half of the number of additional times computed;

The use of the equation 3 proposed by Monache requires two variables whose values need to be defined, the weight of each physical variable (w_i) and the number of additional elements computed (k). In addition, the final result for each forecast can be built as an average of the best results. In that scenario a third variable needs to be optimized. This variable is the number of elements composing the best analogs (Na). According to Monache *et al* (2013), the optimization of these variables require

experimental testing to be acquired. In addition, the number of analogs to be used and the length of the training period can also influence the accuracy of the method. Determining optimal values for these variables also require experimental testing. Optimizing them is important since the increase of the size of the training period would also increase the computational processing required as well as the processing time.

The use of the analog ensemble method has proven itself a very flexible algorithm, as well as being able to research into any specified time window. In addition, the training time for the algorithm does not need to be very large. Vanvyve (2015) has compared multiple-range training time and found that very large periods did not have much better results than the 365 days range. It was notices that a bigger variety of events in the training period was more relevant than a large range. The 365 days range is only inadequate for predicting rare events, such as tornadoes.

There are still some important features to the post-processing analog ensemble method. First, the prediction is calculated based on independent searches for each point that is forecasted. This is an interesting feature since it does not suffer from cumulative error, because the predictions are not affected by previous predictions and it also does not suffer from missing predictions. In addition, the method can have its overall performance improved by using longer datasets to increase the likelihood of emerging similar conditions from predicted events (Delle Monache et al., 2011). Second, the AnEn use can be expanded to any time series with proper dataset. Next section presents the variety of uses that the analogs have been used since its inception in 2011.

2.3 Application of Analog Ensembles

The inception of the Analog Ensemble in 2011 by Delle Monache et al, has paved the way for further research over the applicability of this technique to be developed. This relate to the need of further validation of the accuracy of the method. Alessandrini et al. (2015) performed a study comparing the results of the Analog Ensemble predictions to three other techniques. The European Centre for Medium-Range Weather Forecast Ensemble Predictions System (ECMWF-EPS), the Limited-area Ensemble Prediction

System (LEPS) and a quantile regression (QR). The study was conducted over a 505 day's period in a farm in Italy. After comparing metrics, such as the statistical consistency, the reliability, the sharpness and resolution of the methods (AnEn and the QR techniques) shows better performance in the long-term, where both of them showed similar results. Exceptions in predicting rare weather events, where AnEn outperformed the QR technique.

One of the biggest motivations for further development of statistical methods of weather prediction and post-processing is to give more reliability to the use of renewable energy. Good weather prediction in those terms is important for providing project viability arguments as well as enable negotiation of future energy productions (Vanvyve et al., 2015). Zhang et al. (2018) and Vanvyve et al. (2015) conducted studies over the use of AnEn for Solar Power and Wind Power, respectively. The studies do reinforce the consistency of the method since the results are of high quality and do present the advantage of not having missing points. Although Zhang´s method derives from a Taylor expanded approach over the solar forecasting proposed by Akyurek et al. (2015), its results also help to support the use of methods based on analogs forecasting. That is why further development of method such as AnEn are important, so the wind and photovoltaic energy industries can expect to have more reliable information.

Cervone et al. (2017) conducted a study combining Artificial Neural Networks (ANN) and the AnEn to generate both probabilistic and deterministic forecasts for photovoltaic energy production. The study focused on short-term prediction, generating 72 hours of prediction using atmospheric NWP data and real observations of photovoltaic power generation from three power plants in Italy. The study showed good reliability for both methods in predicting short-results. The development of Cernove´s study required the use of a supercomputer, due to its approach for defining the best parameters for the analog ensemble. In order to decrease the processing total time, four methods were used: multicore processing, hyper-threading processing, 100% parallelization processing, and 95% parallelization processing. The results are presented in Figure 3. It can be seen that the 100% parallelization method showed the best overall results, and its performance shows continuous increase in equal proportion with the increase of processing cores.

Due to its capacity of better predicting rare events when compared to other methods available in the literature, the Analog Ensemble technique has also been used directed to the prediction of catastrophic events. Alessandrini et al. (2018) applied the AnEn technique to the prediction of the intensity of tropical cyclones. The prediction of this kind of events is of high importance since intense tropical cyclones have a destructive impact over the society and represent a life´s risk to people in the affected regions. Because of the complexity of the problem, the study required the use of an operational configuration from the Hurricane Weather Research and Forecasting (HWRF) model and the AnEn model was constructed over the eastern Pacific and Atlantic Ocean basins. To improve the accuracy prediction of the method, the study used a training dataset of four years. The results showed that the AnEn method performed significantly better than the HWRF for the eastern Pacific and the Atlantic Oceans.

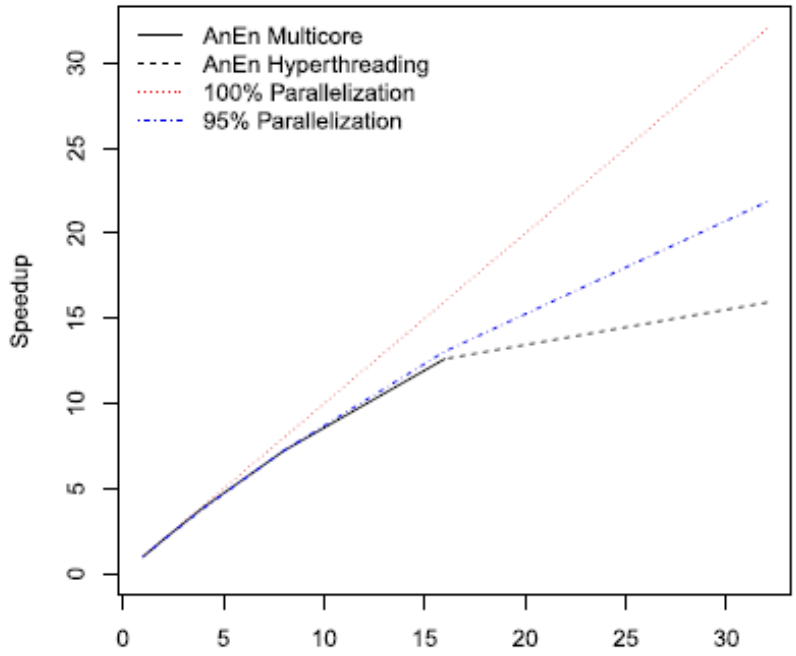


Figure 3 - Processing Speedup per Number of Cores
Source: Cervone et al (2017)

Following the inception of the AnEn for weather prediction, further methodology has been developed by combining the AnEn and other methods to approach specific problems. An example of that is the study performed by Junk et al (2015) where an analog-based method is used to improve ensemble model output statistic (EMOS) for the

calibration of ensemble forecasts. The EMOS is a regression method proposed by Thorarinsdottir and Gneiting (2008). Junk´s approach used Delle Monache (2011) analog ensembles metric to select the training period for the EMOS as well as the variables weights and the set of predictors. The results demonstrated that the analog-based EMOS outperformed the EMOS method. It is worth mentioning that Junk´s results pointed to a high relevance of the predictor weight over the skill of the analog-based EMOS to provide good results. In addition, the amount of variables used and their relevance over the predicted events are of high relevance.

Another use for the AnEn technique is to obtain predictions for past events. Although this may sound redundant, this can be important for fulfilling datasets with missing values. Next section approaches the use of the technique for hindcasting and a discussion of its relevance.

The AnEn model is a flexible tool and it can still be applied to different fields over the forecasting of time series. One of the fronts that has still not been explored is the use of data from one place to compose the training period that will later be used for forecasting results in a different site. That would allow the forecasting and the reconstruction of data for places that lack observation records. Such use of the model can enable the construction of initial variables for the GCM in specific places, improve the planning for both the agriculture and industrial local productions and help on decision making for the implantation of new wind and solar energy generation companies.

2.4 Reanalysis and Hindcasting

Reanalysis was originated in 1979 with the meteorological data exploited for the FGGE. Reanalysis is the prediction of the atmospheric state in the past produced with a single version of data assimilation system. The data was primarily used for learning how make better use of the observations that were being used as initial conditions for numerical weather forecasts models. It was then realized that reanalysis could provide great information for atmospheric research since its data provides coherent, multivariate and spatially complete records of global atmospheric circulation (Dee et al., 2011).

The difference between analysis, reanalysis and hindcasting can be confusing. The analysis approach aims to produce a representation of the atmospheric state over a regular grid. To do so, complex models are used to evaluate the atmospheric behavior, the mathematical physics and variability of the atmospheric and its measurements. The analysis results are snapshots in time, as opposed to forecasts, which shows accumulated parameters, such as the amount of rainfall over a period. Operational systems that run analysis frequently change their software in order to fix bugs. This can cause problems over long analysis that can have data generate by multiple systems (Peng 2014). Reanalysis is a special analysis, where the software system is fixed, and uses only a single version of the data assimilation system. This implies that the results are not affected by method changes (Dee et al., 2011).

Hindcast is an approach to produce numerical meteorological data for locations and periods that do not have past observation data collected. The non-availability of any previous data is what distinguishes hindcast models from reanalysis models (Shi, Schaller, Macleod, Palmer, & Weisheimer, 2015). Although there has been a large research over hindcast models, (see: (Soares, Weisse, Carretero, & Alvarez, 2002); (Cox & Swail, 2001); (Thomas & Dwarakish, 2015) and (Katragkou et al., 2015)), no direct use of analog ensemble models were finding during the development of this literature review.

Meteorological reanalysis is a description of the spatiotemporal distribution of information originated by combining meteorological observation data with numerical weather prediction approaches (Bollmeyer et al., 2015). Due to its model, the reanalysis is the best-estimated four-dimensional atmospheric state for predefining boundaries and it has become a very important tool for monitoring the climate (Trenberth, Koike, & Onogi, 2008). Keller et al(2017) applied the concepts of the analog ensemble to produce reanalysis data for downscaling precipitation. The study's aim was to generate high-quality reconstruction of the retrospective time series and reconstruct synthetic observation datasets for periods with no available observations. Figure 4 shows a schematic illustration of the analog ensemble approach for reanalysis. The approach is similar to that presented in Figure 2, except that, in this case, the training period is a future data to the one that is being post processed by the model. It can be seen that the future

data predicted (green circle) is compared to the best analogs on its on historic series (red circles) data and the equivalent points are selected in the observation data series (blue circles). This results in an average of these results.

The study performed by Keller et al (2017) presents a statistical downscaling for reanalysis precipitation using the AnEn, and concluded that the method is able to outperform the results of the reanalysis and to provide reliable quantification of the underlying uncertainty of reanalysis. It was found that the performance of the AnEn vary for different geographic conditions and on the quality of the prediction that was post-processed by the model.

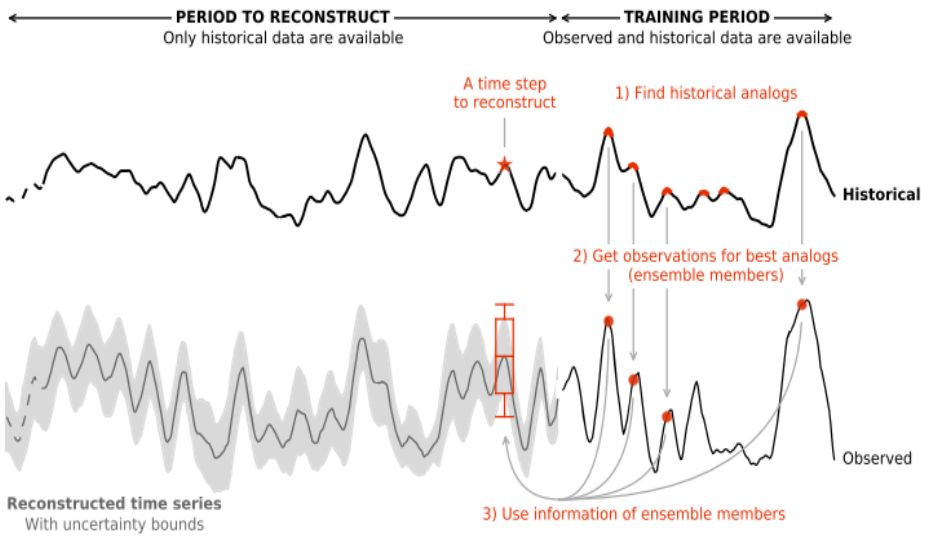


Figure 4 – Analog Ensemble Method for Reanalysis
Source: Vanyeve et al (2015)

2.5 Error Measurement

Due to the necessity of validating the numeric models and of the individual studies, as well as the need of comparing the performance of several distinct approaches over problems of same nature, the estimation of the error produced by the studies is of high importance. Although the error of single predictions can be of simple evaluation, large series of data have been subjected to a variety of error estimation techniques that are explored below. Equation 4 represents the simplest method of approaching the error

of a number, which is simply taking the scalar difference $[E_i]$ between the estimated value $[y_i]$ and the accurate value $[x_i]$.

$$E_i = y_i - x_i \tag{4}$$

Although this technique is of simple evaluation and understanding, its results cannot be directly compared with the results of other studies and approaches that do not use the same inputs or datasets. For that reason, a much more commonly used method is the absolute relative error (ARE) method, described by Equation 5. From this method, we can also use the simple mean (Equation 6) method to have an overall analysis of the results error, which is the BIAS.

$$ARE_i = \left|\frac{y_i - x_i}{x_i}\right| \tag{5}$$

$$BIAS = \frac{1}{n}\sum_{i=1}^{n} E_i \tag{6}$$

Where n is the total number of elements.

In more sophisticated statistical approaches, there are other commonly used error estimation techniques, being the Root-mean-square error (RMSE), the most commonly used, as given by Equation 7.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{7}$$

Although this method is highly used, it can lead to misinterpretation of the results, especially when comparing the error of two different data sets studies, as demonstrated by Willmott & Matsuura (2005). This is due to two elements in the equation, the squared

term and the n^{-1} term. The presence of these two elements in the equation leads to a higher influence of larger elements over the small elements in the final results, which mean that two data error lists with the same mean error will have different error value from the RSME method.

According to Willmott & Matsuura (2005) the Mean Absolute Error (MAE) is not so largely used in the current approaches, even though it is a well-known statistical error measuring tool. They state that this is a more interesting approach since it provides an evaluation of the error that is as good as the RSME. And the MAE provides a more reliable analysis when comparing different studies. Equation 8 presents the MAE method.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i| \tag{8}$$

To be able to apply a good metric for each element of a long vecto, the absolute relative error (ARE) can be applied in the form of Relative Percentage Error (RPE) and the Mean Average Percentage Error (MAPE). Equations 9 and 10 demonstrate these methods, where x_{mean} is the average of all the values from the real values vector.

$$RPE_i = 100\ ARE_i \tag{9}$$

$$MAPE_i = \frac{|y_i - x_i|}{x_{mean}} \tag{10}$$

In addition to the previous measurements, the standard deviation can be used to estimate how the distribution of the error is behaving along the estimated data. The standard deviation (σ) measures the dispersion of the values, and can also be used to evaluate the confidence of the results. Standard deviation is mostly used to set the range, where the results can be acceptable or declined. Equation 11 presents the standard deviation formula.

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} x_i - \bar{x}} \tag{11}$$

The RMSE and the BIAS can be used to evaluate the standard deviation of forecasts. Equation 12 presents the correlation between those variables. The use of those variables together is an important tool in determining the nature of the errors that are generated by forecast models. For that purpose, the RMSE is an indicator the level of randomness in the error, while the BIAS is an indicator of systematic errors (Taylor, 2001).

$$\sigma^2 = RMSE^2 - BIAS^2 \tag{12}$$

The methodology proposed by Taylor to evaluate the quality of the forecasts consists in plotting the standard deviation of an observation in both the x and the y axis. From that, the forecast standard deviation is plotted with a correlation factor. The correlation factor is a measure that varies from 0 to 1. The closer the correlation standard deviation is to 1 the better is the quality of the forecast series. For further deatails over the calculation of the correlation coefficient, see (Taylor, 2011). Also, the distance from the forecast standard deviation to the observation standard deviation indicated the consistence of the values. Therefore, the closer the forecast standard deviation is to the observation standard deviation, the better is the consistence of the calculated series.

When dealing with large amounts of data and with several groups of predictions, some techniques can be used to simplify the comparisons of the results. First, the skill score (SK) of a forecast can be used to compare two results. The skill score can be calculated as shown in equation 13 (Murphy, 1988).

$$SK = 1 - \frac{Result_1}{Result_2} \tag{13}$$

Where the $Result$ variable represents the error values that are aimed to be compared. Negative score values represent scenarios where the $Result_1$ was superior to $Result_2$. Positive score values show that $Result_2$ was higher than $Result_1$.

To provide better visualization of the results behavior for large data sets, moving averages are commonly used. Moving averages are used to smooth the impact of the fluctuation of the results. The basic method for calculating moving averages is taking the average of the n surrounding values of each point. The current calculation power enables us to use more sophisticated methods for estimating moving averages. The R programming language has an integrated smooth-function that uses different techniques according to the size of the vectors. For vectors smaller than 1000 elements, the Local polynomial Regression Filling (LOES) is used. This technique fits polynomial surfaces that are determined by numerical predictors using local fitting. For larger vectors the General Additive Model (GAM) is used, where the smooth-terms are represented using penalized regression splines.

Chapter 3

METHODOLOGY

In this chapter, the data that was used to conduct this study is presented along with a description of the source of the data, .i.e. type and location of the observation stations. The method for preparing and applying the AnEn model with the selected dataset is also presented.

3.1 Dataset

The development of this study requires the use of a large data set of weather information in order to build a training database and to have remaining data to use as a starting prediction for the analog ensemble. For this propose, data from 16 stations were used, where 10 stations are ground stations and six are moored buoys stations (the so-called sentinel of the sea). The United States National Data Buoy Center (NDBC) administrates all 16 stations, though the National Oceanic and Atmospheric Administration (NOAA) runs the grounded stations. Figure 5 and Figure 6 illustrates, respectively, a ground monitoring station and a moored buoy station used by NDBC (National Oceanic and Atmospheric Administration, n.d.).

The selected stations are all located along the in the state of Virginia, in the United States. The weather in this region is classified as humid subtropical according to the Köppen climate classification (Kottek, Grieser, Beck, Rudolf, & Rubel, 2006). The geology of the state includes five regions: Tidewater, Piedmont, Blue Ridge Mountains, Ridge and Valley, and the Cumberland Plateau. The stations used for this study are all located on the Tidewater region. This region mostly embody flat plateaus and rivers that

open into the sea with high tide variants. In addition, the location has a large history of cyclones, tornados and hurricanes (Mitchell et al., 2013). Figure 7 shows a map of the ground station locations. The stations are relatively close in proximity. This proximity was purposely chosen in order to have stations with similar weather conditions along time, and enable interchangeable use of the data for the hindcasting analysis through the ensemble.

Figure 5 – Example of a grounded weather station
Source: NDBC

Figure 6 – Example of a Moored Buoys weather station
Source: NDBC

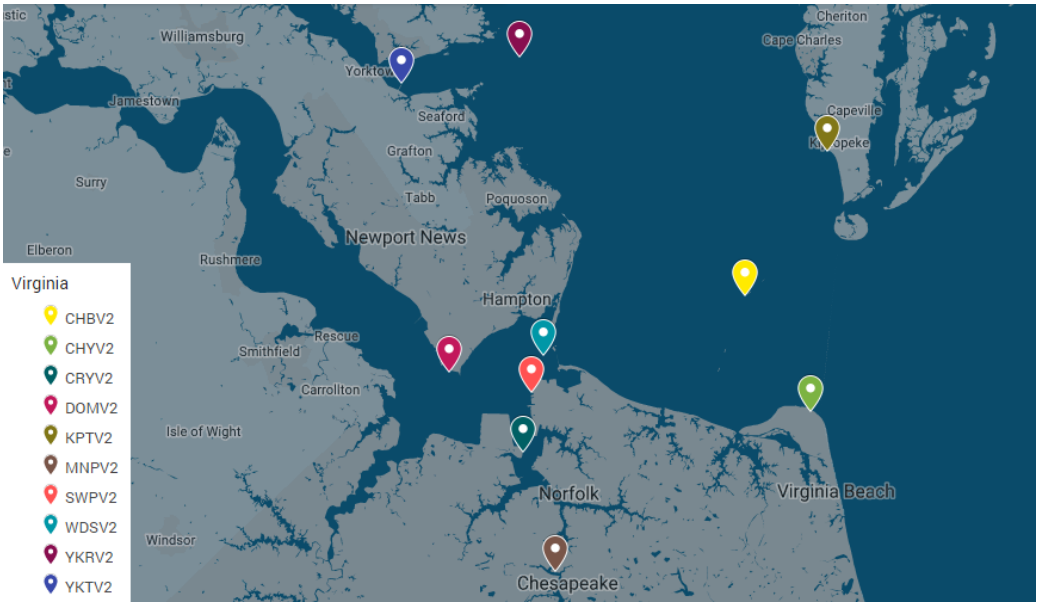


Figure 7 – Geolocation of the NDBC stations in Viginia

Table 3 – Variables Description
Source: NDBC

VARIABLE	DESCRIPTION	UNIT
WDIR	Wind Direction	Degrees (clock wise from true north)
WSPD	Wind Speed	Meters per second
GST	Peak gust speed	Meters per second
WVHT	Significant wave height	meters
DPD	Dominant wave period	Seconds
APD	Average wave period	Seconds
MWD	The DPD wave direction	Degrees (clock wise from true north)
PRES	Sea level pressure	hPa
ATMP	Air temperature	Celsius
WTMP	Wave temperature	Celsius
DEWP	Dewpoint temperature	Celsius
VIS	Station visibility	Nautical Miles
PTDY	Pressure tendency	Plus or minus
TIDE	The water level	Feet

The data available on NDBC hold information from both air and sea stations. There are fourteen weather variables, six from moored buoys stations and eight from ground stations. Table 3 describes all the available variables. The data collected is registered in six minutes intervals, i.e. collecting ten measurements per hour.

The NDBC platform has available data from 2006 to 2017 for the sixteen pre-selected stations, though some of the stations have gaps of data collection. Table 4 shows the availability of data along the years for each station, where *y* stands for available data and *n* for unavailable data. Based on the years with a larger availability of data, this study only used data from 2011 to 2017, considering two factors: there are more data within this time period than before; and the data collected within the duration of seven years is enough for the proposed study (Delle Monache et al, 2013).

During the period studied, the data collected by the stations is not necessarily complete and have punctual gaps and small range of non-collected data along the reference years. These gaps can be due to errors in data collecting as well as non-operating periods of time that are caused by damage, maintenance, and calibration of the stations. These gaps need to be analyzed and considered while processing of the information. Table 5 presents the availability of each of the main six variables for each ground station along the years considered for this study.

Table 4 – Available Data per Station

	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
CHBV2	n	n	n	n	n	n	n	n	n	n	n	y
CHYV2	y	y	y	y	y	y	y	y	y	y	y	y
CRYV2	y	y	y	y	y	y	y	y	y	y	y	y
DOMV2	y	y	y	y	y	y	y	y	y	y	y	y
KPTV2	y	y	y	y	y	y	y	y	y	y	y	y
MNPV2	y	y	y	y	y	y	y	y	y	y	y	y
SWPV2	y	y	y	y	y	y	y	y	y	y	y	y
WDSV2	n	n	y	y	y	y	y	y	y	y	y	y
YKRV2	y	y	y	y	y	y	y	y	y	y	y	y
YKTV2	y	y	y	y	y	y	y	y	y	y	y	y

Table 5 – Variables Availability per Station

	WSPD					PRES				
Station	**Min**	**Mean**	**Max**	**NAN´s**	**Availability**	**Min**	**Mean**	**Max**	**NAN´s**	**Availability**
CHYV2	0	4,6	29,7	37111	**94%**	986,9	1017,1	1042,7	514274	**16%**
YKTV2	0	3,7	23,6	17826	**97%**	974,7	1017,1	1044,3	14872	**98%**
DOMV2	0	4,3	24	20055	**97%**	972,8	1017,4	1044,5	14872	**98%**
KPTV2	0	4,3	22,3	22941	**96%**	0	0	0	613680	**0%**
MNPV2	0	2,4	18,6	25713	**96%**	968,5	1017,2	1044,1	22621	**96%**
WDSV2	0	5,4	25,9	18461	**97%**	970,1	1017,2	1044,9	22621	**96%**
YKRV2	0	5,7	25,9	17241	**97%**	972,6	1016,9	1043,9	14057	**98%**
CRYV2	0	3,8	22,2	136919	**78%**	970,3	1017,4	1044,3	136036	**78%**
SWPV2	0	0	0	613680	**0%**	972	1017	1044	42770	**93%**
CHBV2	0	5	17,6	592427	**3%**	987,7	1019	1036,1	592702	**3%**

	WDIR					WTMP				
Station	**Min**	**Mean**	**Max**	**NAN´s**	**Availability**	**Min**	**Mean**	**Max**	**NAN´s**	**Availability**
CHYV2	0	192	360	37094	**94%**	na	na	na	613680	**0%**
YKTV2	0	210	360	17808	**97%**	-0,3	17,1	32,8	16433	**97%**
DOMV2	0	194	360	20102	**97%**	na	na	na	613680	**0%**
KPTV2	0	176	360	22781	**96%**	0,4	16,8	31,6	22613	**96%**
MNPV2	0	198	360	24979	**96%**	0,9	18,9	34,1	22133	**96%**
WDSV2	0	199	360	18491	**97%**	na	na	na	613680	**0%**
YKRV2	0	194	360	17318	**97%**	na	na	na	613680	**0%**
CRYV2	0	191	360	137087	**78%**	na	na	na	613680	**0%**
SWPV2	na	na	na	613680	**0%**	-0,3	17,1	32,2	11642	**98%**
CHBV2	0	175	360	592427	**3%**	5,8	13,9	23,7	593379	**3%**

	ATMP					GST				
Station	**Min**	**Mean**	**Max**	**NAN´s**	**Availability**	**Min**	**Mean**	**Max**	**NAN´s**	**Availability**
CHYV2	-12,2	17,1	36,5	27060	**96%**	0	6,2	34,9	37709	**94%**
YKTV2	-13,5	17,1	37,8	17751	**97%**	0	5	32,6	17851	**97%**
DOMV2	-12,6	16,8	37,2	17563	**97%**	0	4,9	32,1	20085	**97%**
KPTV2	0	0	0	613680	**0%**	0	5,5	28,9	22752	**96%**
MNPV2	-13,8	18,1	37,3	26206	**96%**	0	3,7	30,7	24834	**96%**
WDSV2	-12,7	17,3	44,4	85380	**86%**	0	6,1	32,1	18552	**97%**
YKRV2	-12,8	16,5	36,3	16232	**97%**	0	6,5	33,5	17356	**97%**
CRYV2	-7,9	16,5	35,2	515310	**16%**	0	5,1	30,5	136741	**78%**
SWPV2	0	0	0	613680	**0%**	0	0	0	613680	**0%**
CHBV2	-4,9	12,6	28,3	593379	**3%**	0	6	20,4	592427	**3%**

3.2 Numeric Approach

The development of this study required a numeric approach for both prepating the data set provided by NDBC and applying Equation 3 into the time series. Preparing the data is a prerequisite in order to organize the available dataset into vectors that could be more easily manipulated later on, to exclude the values of the observations that hadn't been collect and fill them up with *nan´s* values, and to identify the location of big gaps in the time series. This stage of the numerical approach was developed in Python and the code for this can be found in the appendices section. Figure 8 presents a diagram of its main steps. In the final step, it saves its results as netCDF4 files to allow fast read of data and to avoid errors in the read from other computers especially when transitioning from different root languages.

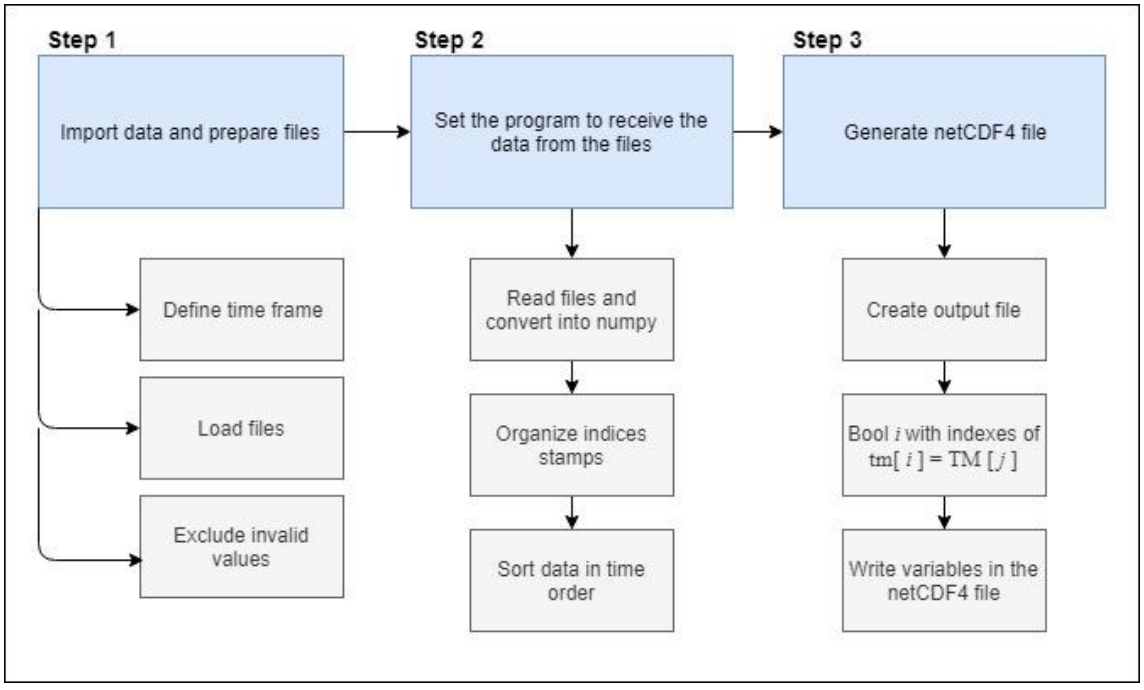


Figure 8 - Logic diagram for setting up the available data

The second stage of the numeric approach is based on applying equation 3 into the prepared data sets. The use of such equation requires two series of data to be functional. One that has a complete historical information and another whose future data will be predicted by the AnEn method. In order to use the best available data and to be

able to estimate the error of the prediction the available data was divided in two periods. The first period matches the training period and is composed with data from January 2011 to December 2015. The second period is the one that will be predicted and is composed by data from January 2016 to December 2017.

Due to the big gaps of data from most of the stations, the first approach of this study analyzed only two station: YKTV2, located in Yorktown USCG Training Center (37°13'36" N 76°28'43" W), and MNPV2, located in Money Point (36°46'41" N 76°18'6" W). These stations were chosen for having the biggest amount of collected data over the studied period, as shown by Table 4. They both have over 96% of collected data for all variables in that period. Figure 9 and Figure 10 present the temperature and wind speed data available after the preparing step for YKTV2 and MNPV2 respectively.

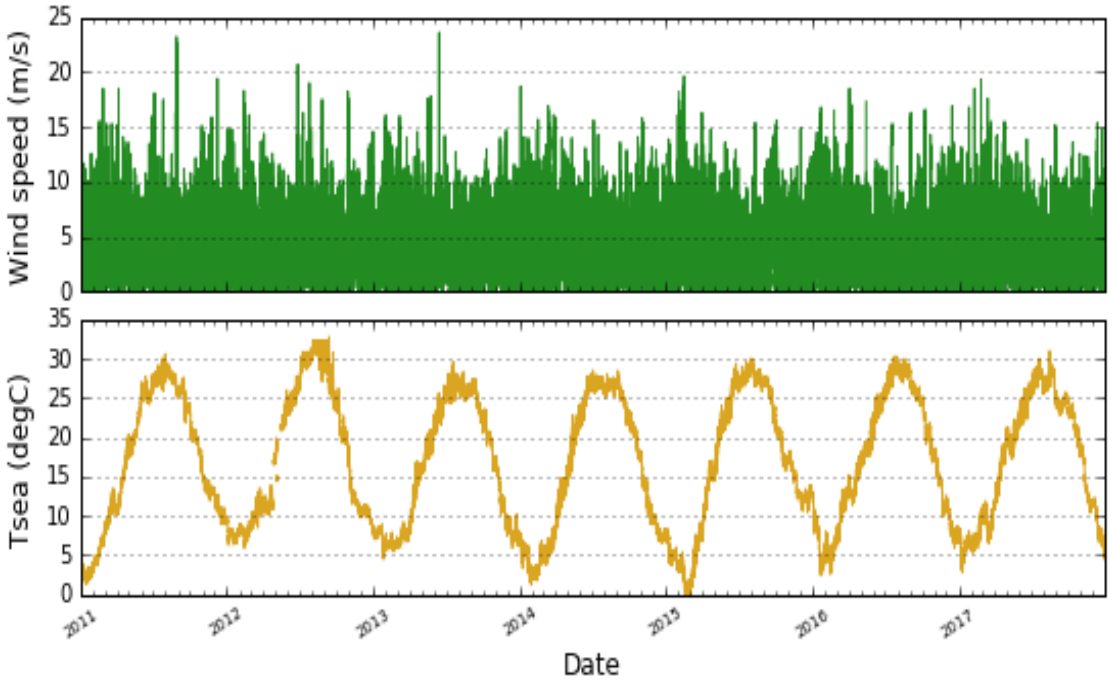


Figure 9 –Station YKTV2 temperature and wind speed dataset

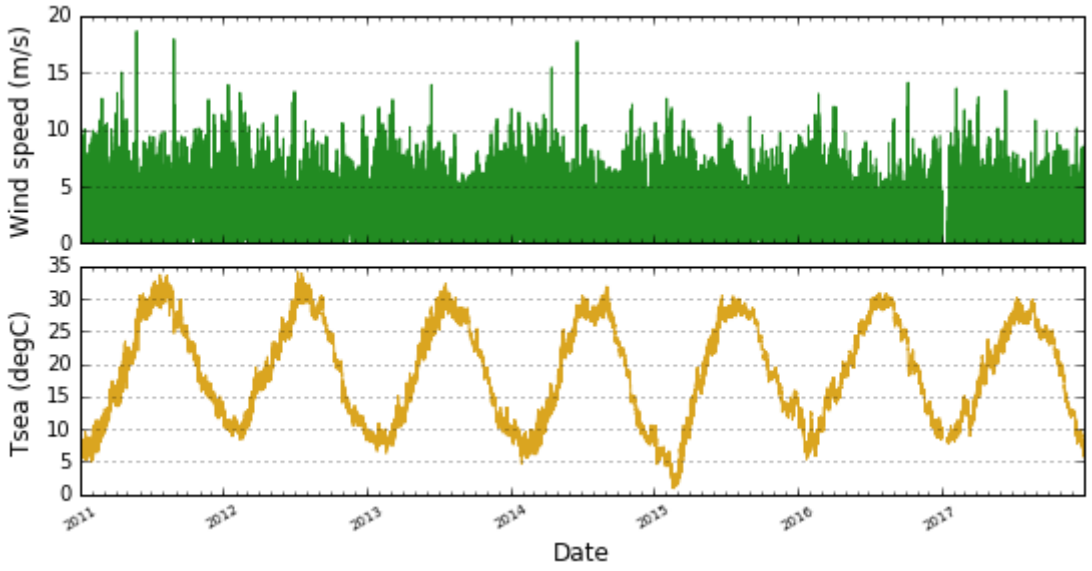


Figure 10 – Station MNPV2 temperature and wind speed dataset

The second stage of the numerical approach was developed in the R language. Figure 11 presents the logic diagram of this stage, and the full code is attached in the appendix section. This stage tested the capacity of each time series to predict each other for three variables: wind speed, wind direction and air temperature. Since two years were predicted in this study (2016 and 2017), there were over 175,440 points to be forecasted, which required a high computation power to process the information in viable time. The computational system initially used for this study was a Ubuntu 16.04 KVM virtual machine with 2.4GHz AMD EPYC 7531 processor cores and 16GB of RAM assigned. This virtual machine is hosted in a cluster at the Polytechnic Institute of Bragança (IPB).

As previously mentioned, this study operated a large amount of data points. Thus, in order to do their processing, the netCDF4 files generated in Python were read into an R program to be processed. Several R scripts were written aiming to reduce the processing time for each prediction. The best script approach was founded to be the one using parallel processing with the *apply* R function. The use of this important function is described below. The full version of the code can be found in appendix 3.

In Figure 11, the diagram demonstrates the steps for applying the metric over the prepared dataset. `A[t[j]]` and `B[t[j]]` represent the input of the historical and observation datasets into the program, respectively. The next step is to create generic

vectors b_j to receive the window frame of length $2*k+1$ to receive the value of each interactions. After that the function is applied into a loop that runs over the training period of the `A[t[j]]` and calculates the metric of each index in the training period. Then, the program selects the best indexes. The final step is to select the equivalent indexes in the `B[t[j]]` vector and take the average of those values to return the forecast of each point.

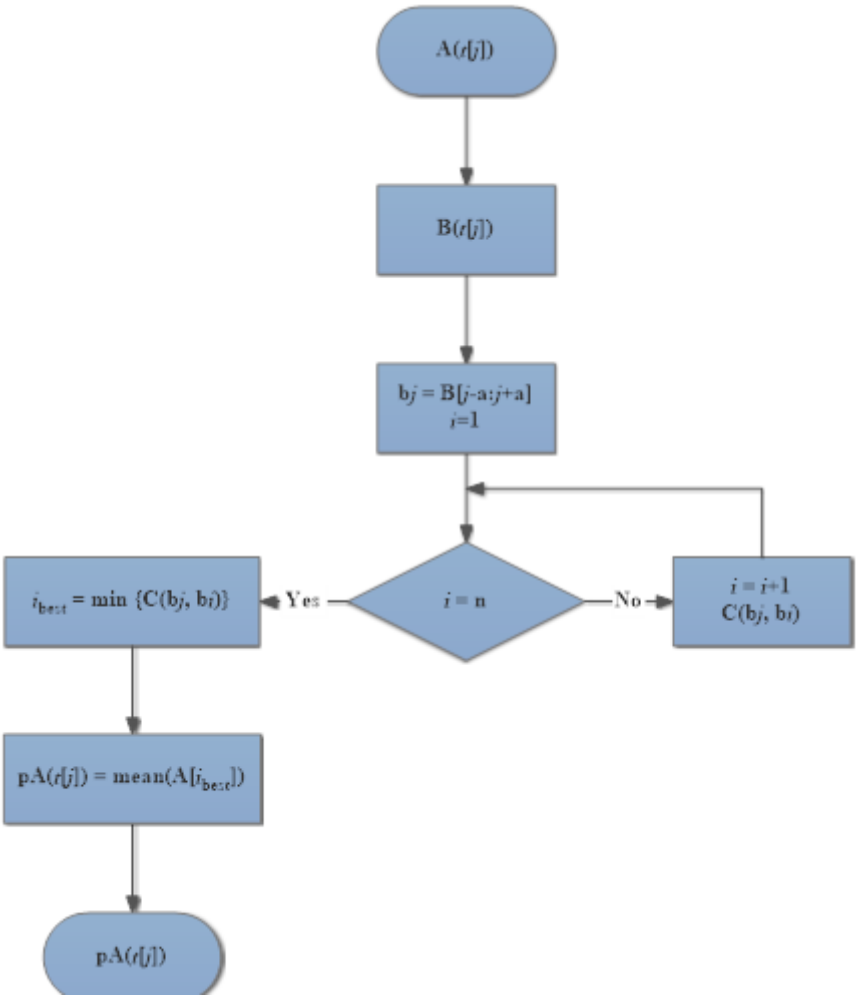


Figure 11 - Forecast script´s diagram

The implementation of a code to apply the diagram that is described in Figure 11 required thet use of some tool that allows the used of *for* loops to be avoided. That is because the developing the code with an excessive use of those loops would drastically increase the computational power required and required a much longer processing time.

The tool that was found to be the most appropriated to use in the R language is the *apply*. This function is adequate since its use allowed the three *for* loops that would be used to be replaced by a single running process. Its use requires the implementation of a function that sweeps the whole data applying the calculation steps for each forecast. It also allows the use of parallel processing that speed up processing time. The parallel cluster and the function that are used in this study is showed below.

```
ncores <- detectCores()
cluster <- makeCluster(ncores-1)  # leave 1 core out for the system
clusterExport(cluster, c("Y", "Ynan", "M", "p", "h", "v", "Na"))  # export sha
red variables
f <- array(data=NA, dim=c(Na, h$N-h$n0+1))  # init forecast array
f[1:Na,1:(p$N-p$n0+1)] <- parSapply(cluster, p$n0:p$N, main)
stopCluster(cluster)
```

With *main* being the following equation:

```
main <- function(n){
  mi <- max(n-M, 1)
  me <- min(n+M, p$N)
  y <- array(data=NA, dim=c(2*M+1))
  y[(M+mi-n+1):(M+me-n+1)] <- p[[v]][mi:me]
  if (sum(is.na(y)) < .5*M) {  # guarantees window has 50% of valid values
    A <- (sweep(Y, 1, y, "-"))**2  # quad error

    ## Get index for analogs
    na <- order(Metric, decreasing=FALSE)[1:Na] + M

    ## Store analogs
    #f[1:Na,n-p$n0+1] <<- h[[v]][na]
    result <- h[[v]][na]
  } else {
    f(best_analogs, time)
    result <- array(data=NA, dim=c(Na))
  }
  return(result)
}
```

Chapter 4

SINGLE PHYSICAL VARIABLE

In this chapter, the first approach to apply the AnEn over different stations is presented. It starts by showing the decision that were made for the independent variables along with the adaptation of the metric that was used. The results that were obtained are presented and discussed in detail, including a deeper analysis of the error measures for all six variables that were studied.

4.1 Time Series

The first approach into this study is to isolate each variable to evaluate its capacity of predict itself. To do so, N_v is set to 1 when applying Equation 3 and therefore w is automatically 1. Equation 3 can be rewritten as shown in Equation 10, the standard deviation values are still included in the equation, but the value of this variable won´t have any effects over the processing values due to the ranking that is used for selecting the best analogs. The training period was the data from 2011 to 2015. Figure 12 shows a diagram of the approach used in this study.

$$F_t, A_\tau = \frac{1}{\sigma}\sqrt{\sum_{j=-k}^{k}(F_{i,t+j} - A_{i,\tau+j})^2} \tag{14}$$

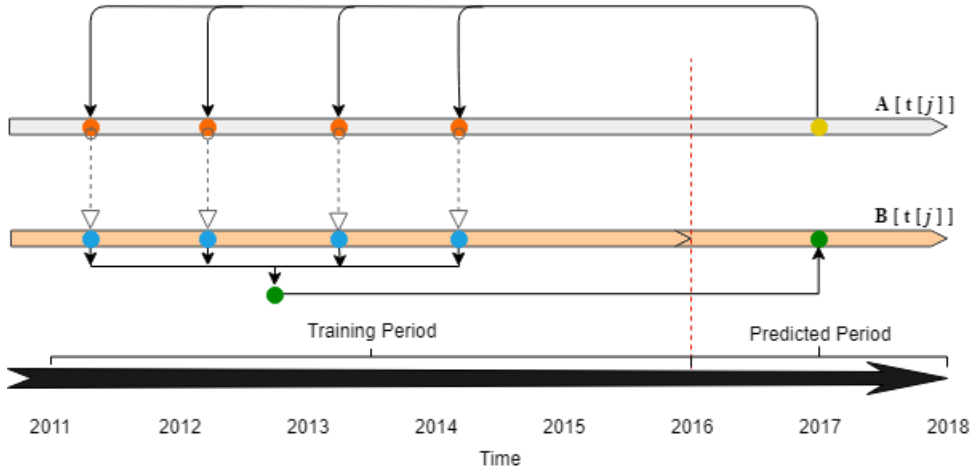


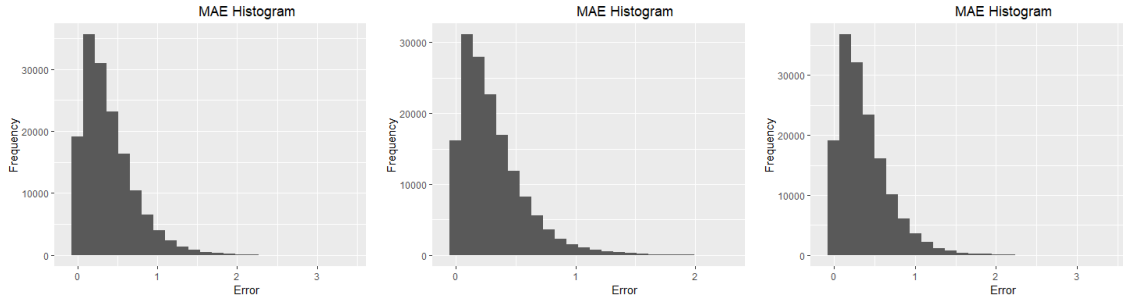
Figure 12 – AnEn Processing Diagram

Due to the conditions of the available data, for the first approach the two stations selected were YKTV2 and MNPV2. The first one provides the predictor time series `A[t[j]]`, where both training and forecast periods are known. The second one provides the historical time series `B[t[j]]` from which the forecasts will be produced through the indexes of the best analogs.

Although evaluating optimal values for the variables k and Na require a computational power that is not available for this study (see: Cervone et al (2017)), the model was applied to a few configurations, as an attempt to observe the effects on both the accuracy of the method and the processing time. To do so, the model was applied to the series with three different sizes for the number of elements composing the average results (Na): 10, 25 and 50 elements. In addition, the analog window frame (k) was tested for 5, 10 and 20 samples. Figure 13 presents the histograms for the MAPE error obtained with each configuration of Na, and Figure 14 presents a density graph of the three sizes. It is shown that the distribution of the error along the time series was not heavily influenced by the analog size, and the density curves have a very similar behavior.

The results for the different sizes of k had similar distributions for the error as shown above. It was noticed that the size of the window affected specific dates for each size but no patterns were identified. In terms of computational power, the increase of the size of the variable k directly increases the size of the vectors that are bean processed which leads to a higher demand of memory and processing power. The memory was the

biggest issue and it limited the tests to windows of 20 elements and that limited this study. Due to these limitations, all the results presented in this chapter were processed for windows of 20 elements. The processing time for all three analog sizes was also very similar, as shown in Table 6. Since the distribution of the error was found to be very similar on the histograms and density maps, an overall error of the time series was calculated to observe its behavior. This error is the BIAS and it was estimated according to Equation 6 and the results are also presented in Table 6. From the table, it can be noticed that the increase in the analog size has a small effect over the processing time but also does not present a too significant decrease in the accumulated error. Due to the small changes in the results, the model was run with analogs of 300 elements. The results showed that using a big analog would not be justifiable and therefore the study was conducted with 25 elements for all the next uses of the model.



A *B* *C*

Figure 13 - Error distribution for different amounts of elements: A) 10 elements; B) 25 elements; C) 50 elements.

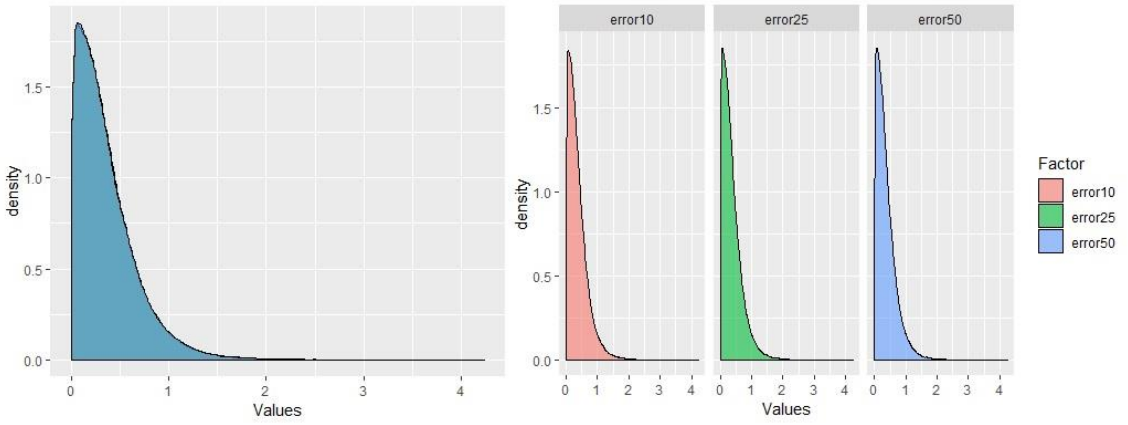


Figure 14 - Density distribution. Grouped error on the left; Separated errors on the right.

Table 6 - Size effect over the processing time and error

Size	Processing time [h]	Error [m/s]
10	1.5542	0.988071
25	1.5538	0.955740
50	1.5553	0.941101
300	1.5602	0.923517

The model was run in the virtual machine allocated in the computer cluster previously described. The processing time was around 1.5 hours for each variable, with each prediction index taking around 50 milliseconds. The results for each variable are presented in the subsections bellow.

4.1.1 Wind Speed

The first physical variable that was analyzed was the wind speed. Although the data was generated for a period of two years, due to the high amount of data the complete time series graph is too dense to provide useful information, as it can be seen in Figure 15. Thus, the results will be shown for smaller periods of time, in order to provide better Visualization of the results. Accordingly, Figure 16 and Figure 17 show the prediction for the wind speed in periods of a week and a month respectively.

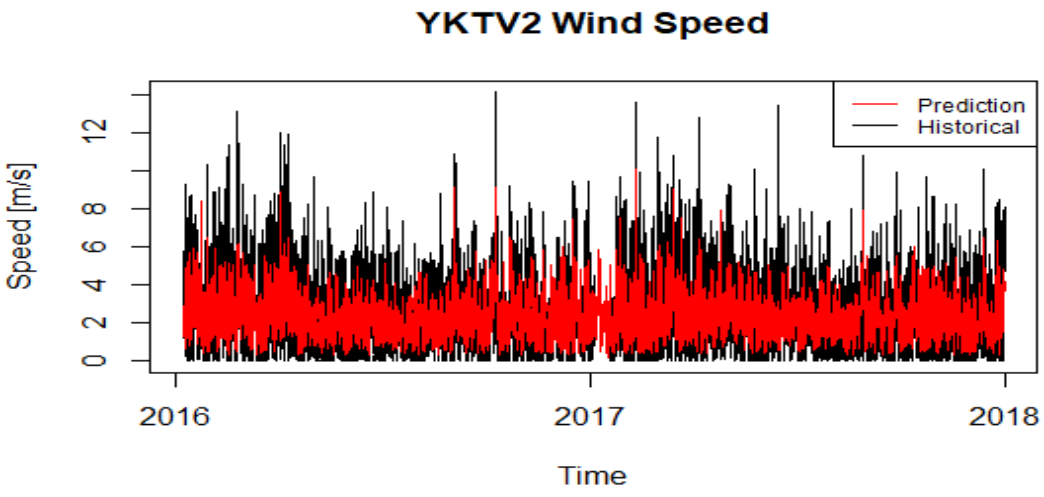


Figure 15 - YKTV2 Wind Speed Prediction

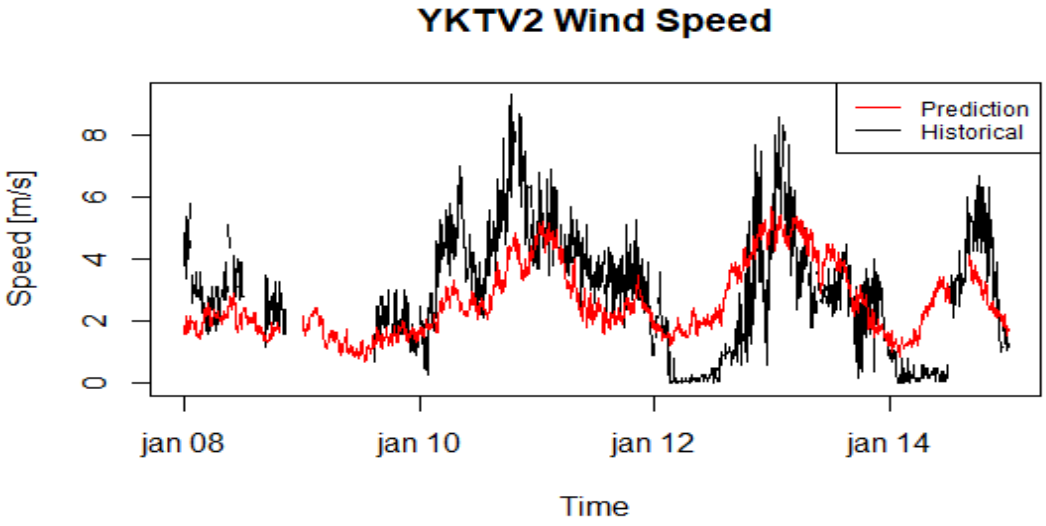


Figure 16 – Weeklong YKTV2 Wind Speed Prediction

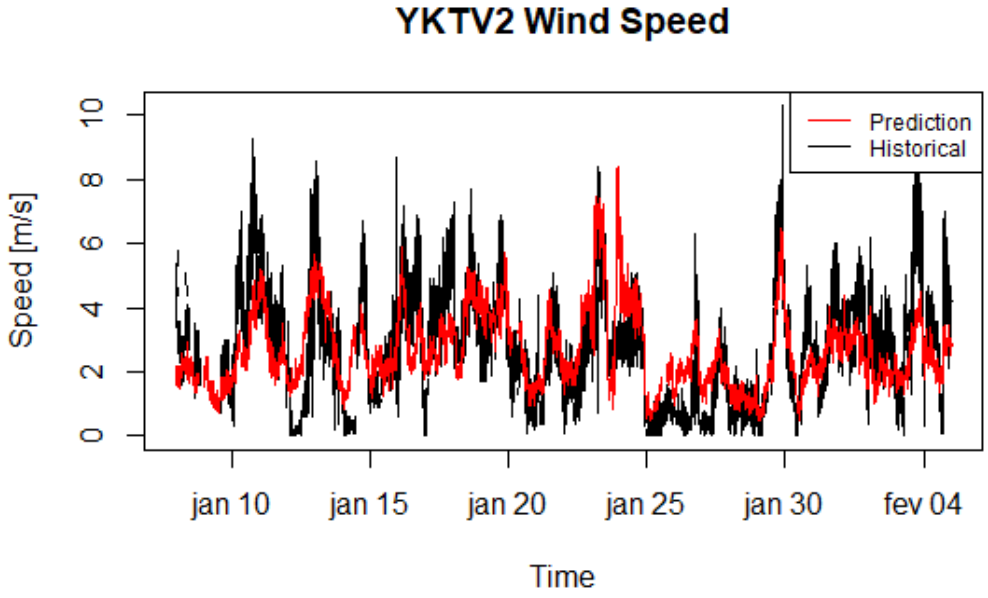


Figure 17 - Monthlong YKTV2 Wind Speed Prediction

The results show good consistence. In Figure 16 it can be clearly observed that, as described in the literature, the method is able to give predictions for periods that have no real observation. It can also be noticed that for other periods were there were no observations the method was not able to return a prediction. That can be explained by the fact that for some points the lack of historical data is too big, which would provide a prediction with no reliability, and therefore those were exclude in the programming model used in this study.

Another interesting fact that can be noticed in both Figure 16 and Figure 17 is that the model was very accurate in keeping the curve pattern of the observation line. On the other hand, it did not have the ability to provide good predictions in period when the speed had big oscillations in small periods. That is even more noticeable in moments where it would decrease too much and then increase back very fast, as it can be seen in Figure 16 in the January 12th.

4.1.2 Pressure

The second variable that was analyzed was the Pressure. Figure 18 presents a weeklong set of the results. Graphs with month and full length for this and the next variables can be found in the attachments section of this work. The variation of the atmospheric pressure in a fixed point is mostly affected by the temperature. Since the air temperature has a thermal inertia it takes longer to change when compared with the wind speed. This leads to a slower change in the atmospheric pressure along the time, meaning that predicting this variable is considerably easier than predicting the wind speed.

The results that were found by applying the method to the pressure, as shown in Figure 18, had a very high accuracy. It can be seen in the picture that the prediction curve is almost overwriting the historical curve for most the predicted period.

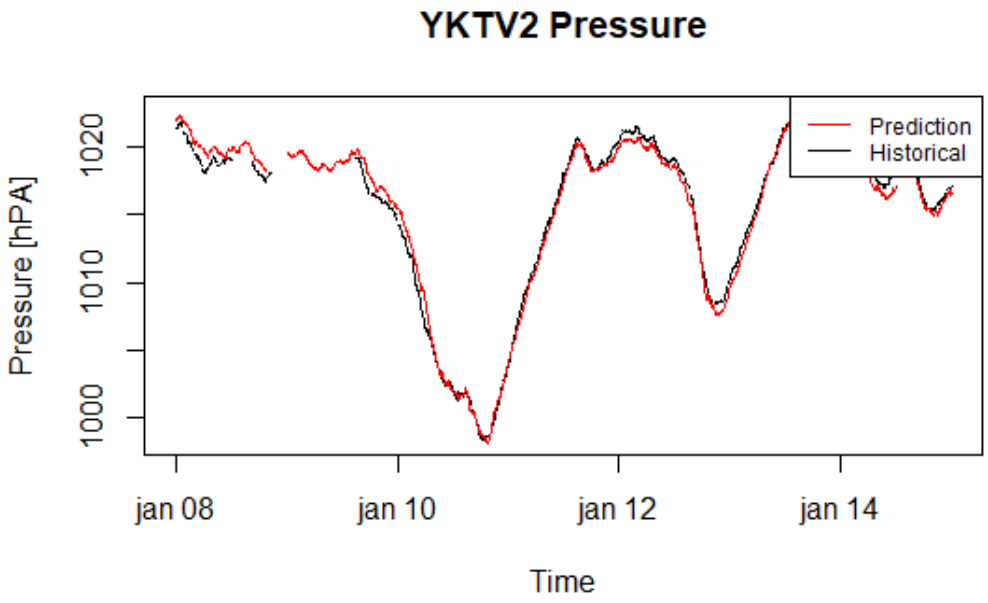


Figure 18 - Weeklong YKTV2 Pressure

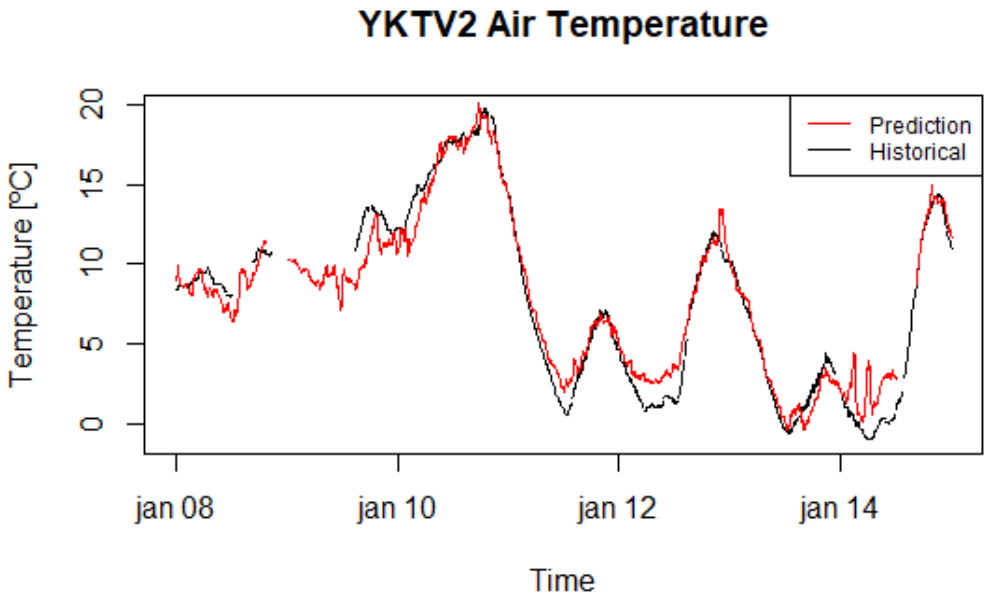


Figure 19 - Weeklong YKTV2 Air Temperature

4.1.3 Air temperature

As previously stated, the air temperature is a variable that varies relatively slower and therefore in of easier predictability by the applied model. Figure 19 presents the weeklong graph of the air temperature prediction. Similarly to the Pressure results, the results for the air temperature had a very good behavior and followed a very similar path to the observation curve. It can be noticed that the accuracy for this variable was a bit slower than those achieved by the pressure. This does make sense since the pressure is a more stable variable even though the air temperature directly influences it.

4.1.4 Wave Temperature

As opposed to the pressure and the air temperature, the wave temperature is a very unstable variable. That is because the wave temperature is influenced by both the air temperature and the wind speed; and is also subjected by to heat transfer effects. It is also worth mentioning that when compared to the air, water is a better thermal conductor and therefore is more easy subjected to faster temperature change. In addition, wave temperature is harder to measure, creating bigger uncertainties in the observation data than other variables. Furthermore, due to the way this study was conducted, the time series that was used to produce the analogs ensemble are not the ideal. Figure 20 presents a weeklong prediction for the wave temperature.

4.1.5 Peak Gust Speed

The peak gust speed is the highest instantaneous value measured in a determined period of time. This is another variable that has a hard predictability. That is because this variable behavior is directly attached to the wind speed, but it presents an even more random behavior. Figure 21 presents the weeklong graph for this variable. It can be noticed that the results are better than those for the wave temperature, and are more similar to those found by for the wind speed.

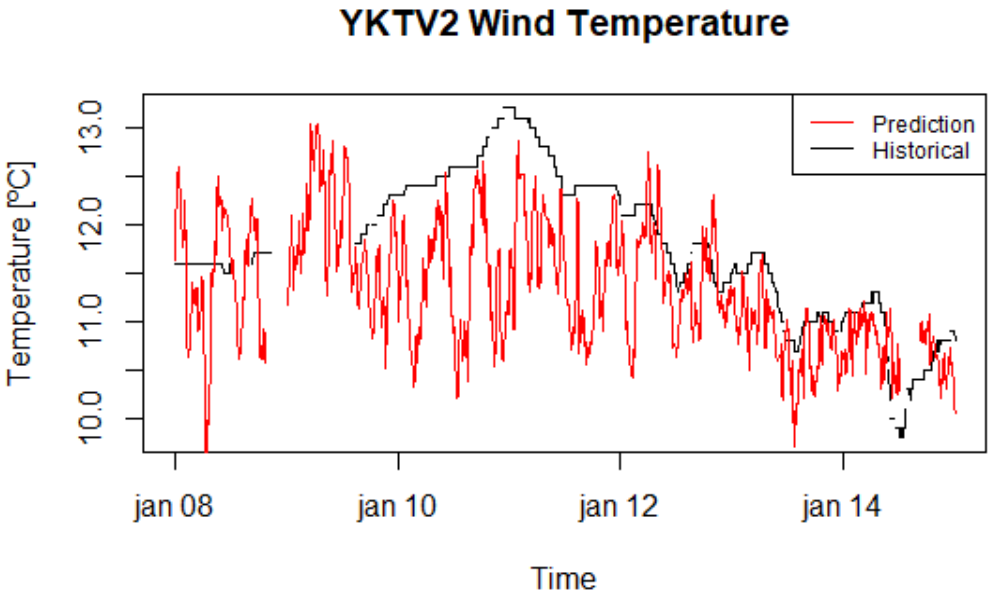


Figure 20 - Weeklong YKTV2 Wave Temperature

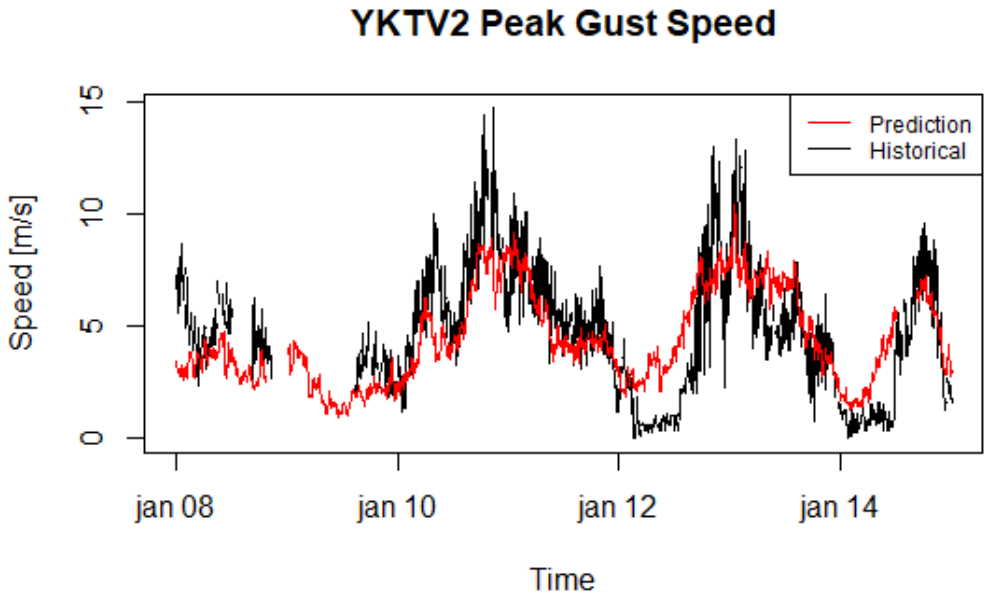
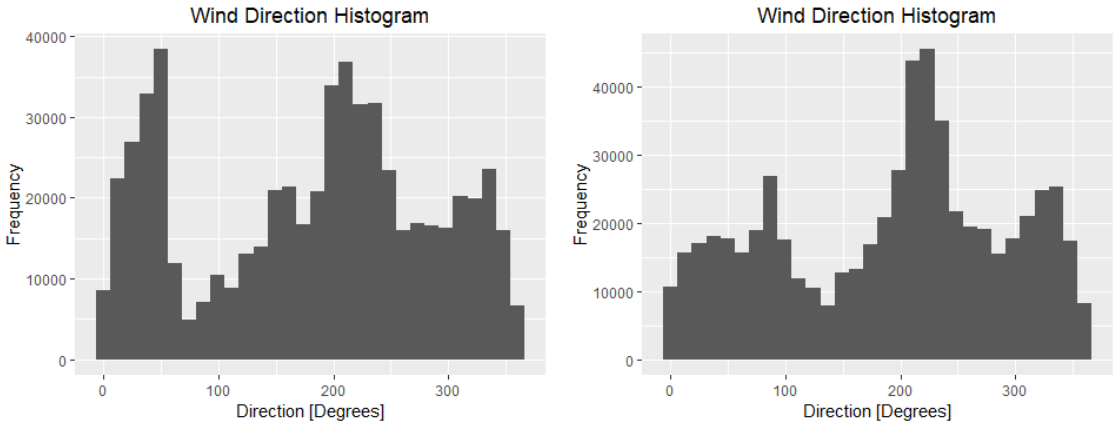


Figure 21 - Weeklong Peak Gust Speed

4.1.6 Wind Direction

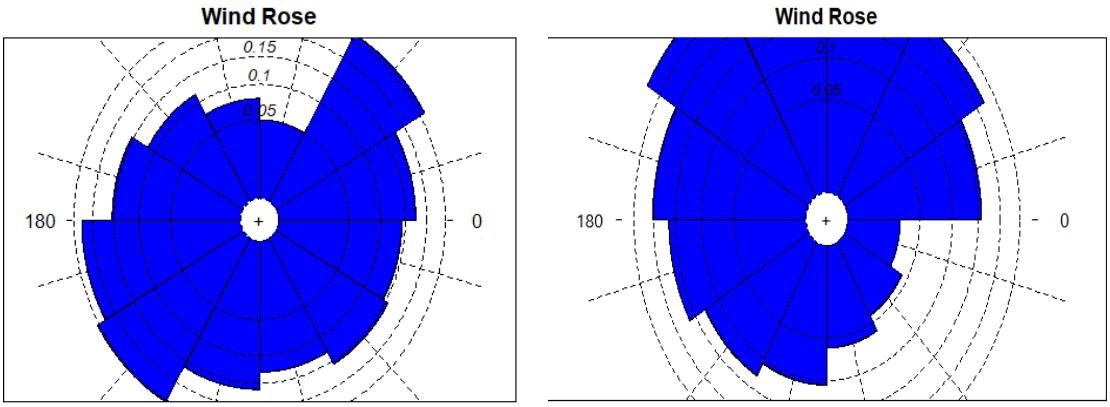
The wind direction is a variable of difficult prediction for being the most unstable of all the variables that were analyzed in this study. That is due to the large amount of change in short periods of time that results in abrupt changes along the time series. Also, this variable has a range of change from 0 to 360 degrees. Because of that it is hard to visualize the behavior of the time series. Thus, the graphs comparing the prediction and the observed values were omitted here for being considered of poor contribution to the analysis. Instead, histograms are shown.

Figure 22A shows a histogram of the predicted values for the wind direction, while Figure 22B shows the equivalent histogram for the observed values. It can be observed that the observed values were more evenly distributed while the prediction values had more peaks for values between 0 and 60 degrees as well as for values between 200 and 250 degrees. To contribute to the visualization of the results, Figure 35 shows a wind rose that related the predicted values of wind direction with the observed values for wind speed.



A *B*

Figure 22 - Histogram of the results for two years period; A) Forecasted values; B) Observed values.



A *B*

Figure 23 - Wind Rose of the wind direction with wind speed magnitude: A) Forecasted; B) Observed

4.2 Error Measures

To have a better insight over the quality of the results presented in section 4.1, this section presents errors from the predictions. To evaluate the error the methods described in this study´s literature review are here applied. Table 7 presents the MAE, RSME and BIAS errors for the six variables analyzed. Except for the wind direction, that had high levels of growth, the other five variables had relatively small error levels. Although the values that were found are similar to each other, they have to be individually analyzed. The results from Table 7 shows that the BIAS error is much smaller than the MAE and RMSE. That indicates that the errors are mostly of random nature and have low levels of systematic error. This agrees to what is found in the literature, and is one of the advantages of the method applied.

The wind speed has a MAE of 0.956 m/s as shown in Table 7, while the values for this variable vary between 0 and 19.3 m/s for the observed series and from 0 to 15.8 m/s for the forecasted series. Therefore, the forecasted time series had a maximum error level of 6.05%. The MAPE for the wind speed is graphically presented in Figure 24**.** The gray line is a plot of the error time series, while the red line is a moving average of its

values. Figure 25 presents the wind speed percentage error (ARE) for the prediction, with the grey line been the percentage error and the red line a moving average of the error.

Table 7 – Accumulated errors for all variables

Variable	MAE	RMSE	BIAS	Unit
Wind Speed	0.956	1.244	-0.384	m/s
Wind Direction	47.581	2438.07	-7.202	Degrees
Wave Temperature	0.963	1.230	0.252	°C
Air Temperature	1.093	1.564	-0.024	°C
Gust Speed	1.319	1.710	-0.671	m/s
Pressure	0.470	0.618	-0.100	hPa

These two graphs show that, although there are some regions with high error and some specific points with extremely high errors, most of the predicted data have a very small error and therefore a good reliability. This is especially indicated by the moving average lines that are both very close to zero. In the other hand, there is a considerable amount of points with high levels of error. The origin of this errors and their effects on the overall skill of the method required further analysis.

It is here worth mentioning that since this prediction is using a single physical variable, the model is not able to take into account the complexity of the atmospheric variation, which reduces the skill of this model. In addition, the wind speed is a variable that has a high variability along short periods of time, making it a harder variable to predict. These are some of the reasons that justify the errors that are found and since the model could still provide solid results, it reinforces the overall reliability of the method.

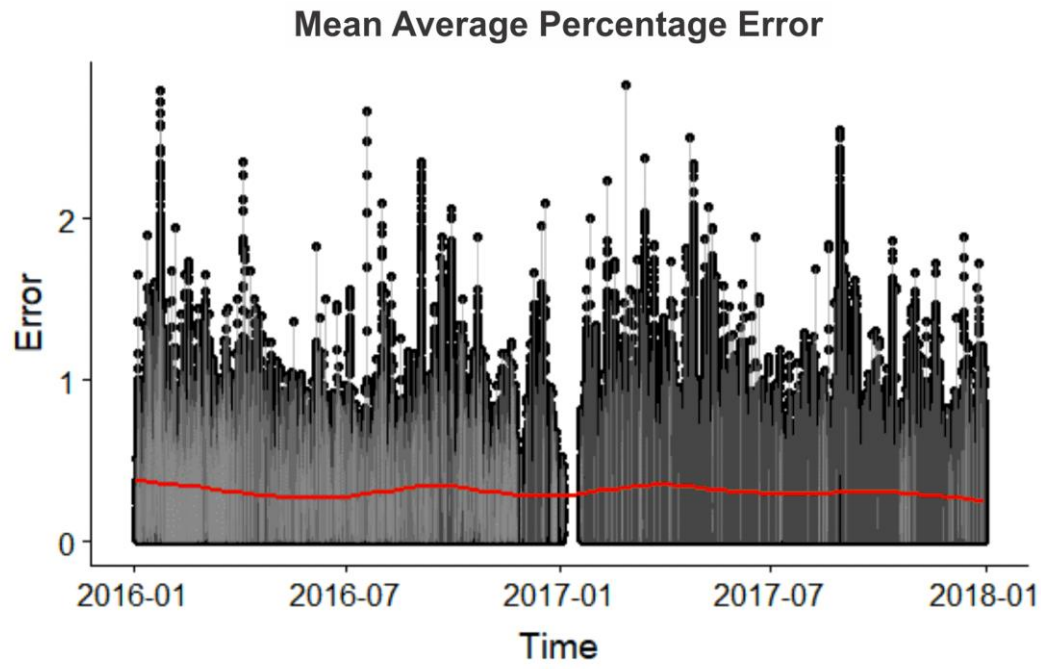


Figure 24 -YKTV2 wind speed forecast mean average percentage error in meters per second

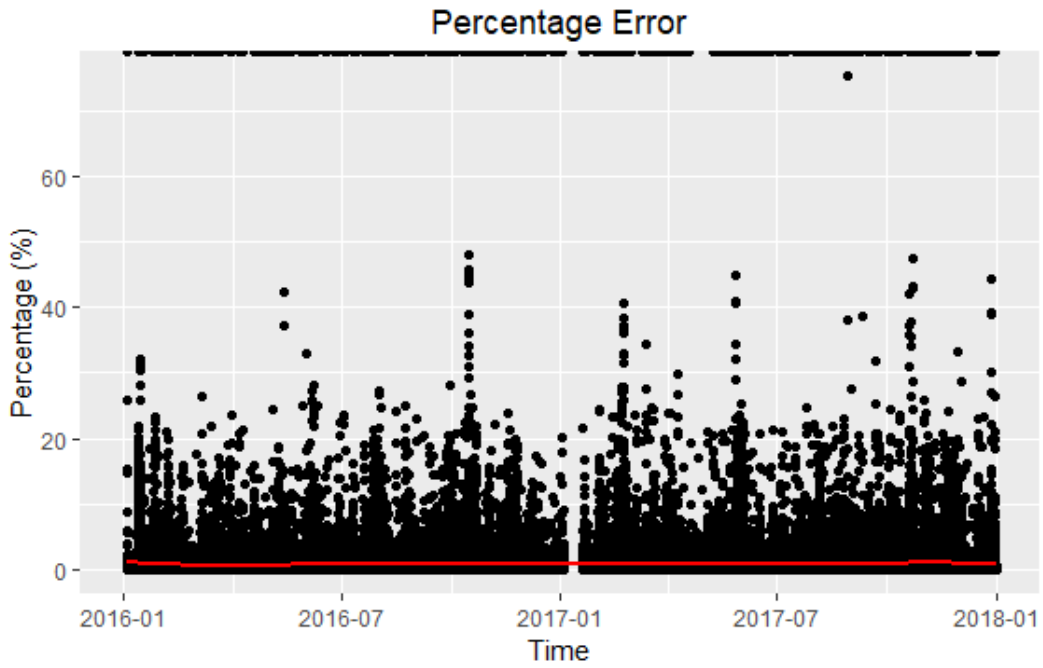


Figure 25 - YKTV2 wind speed forecast percentage error

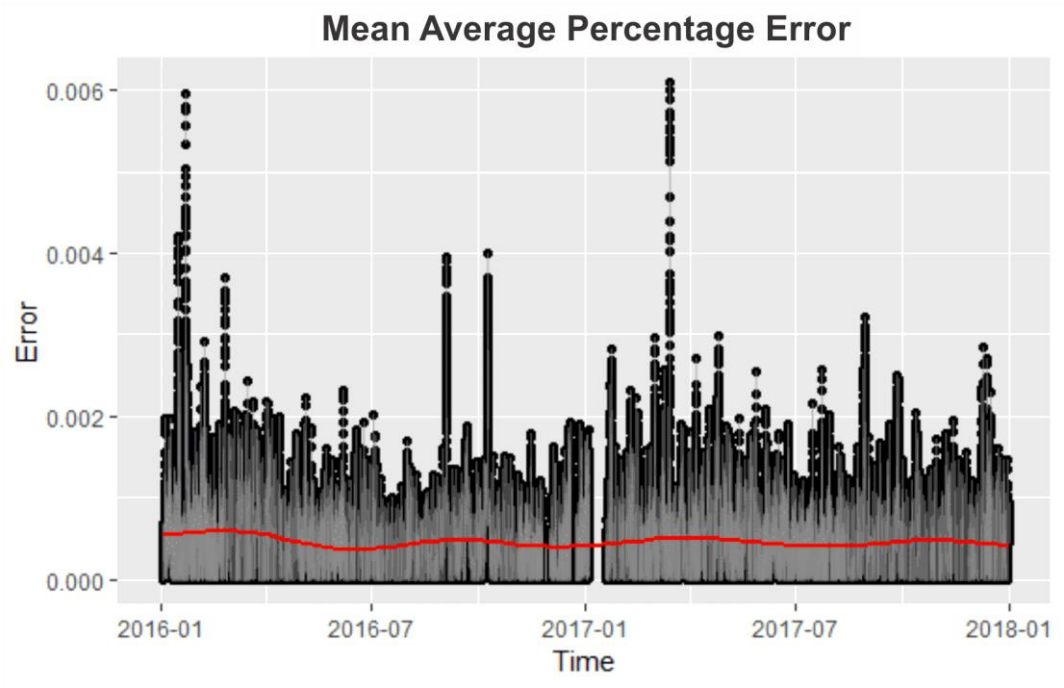


Figure 26 - Pressure Mean Average Percentage Error

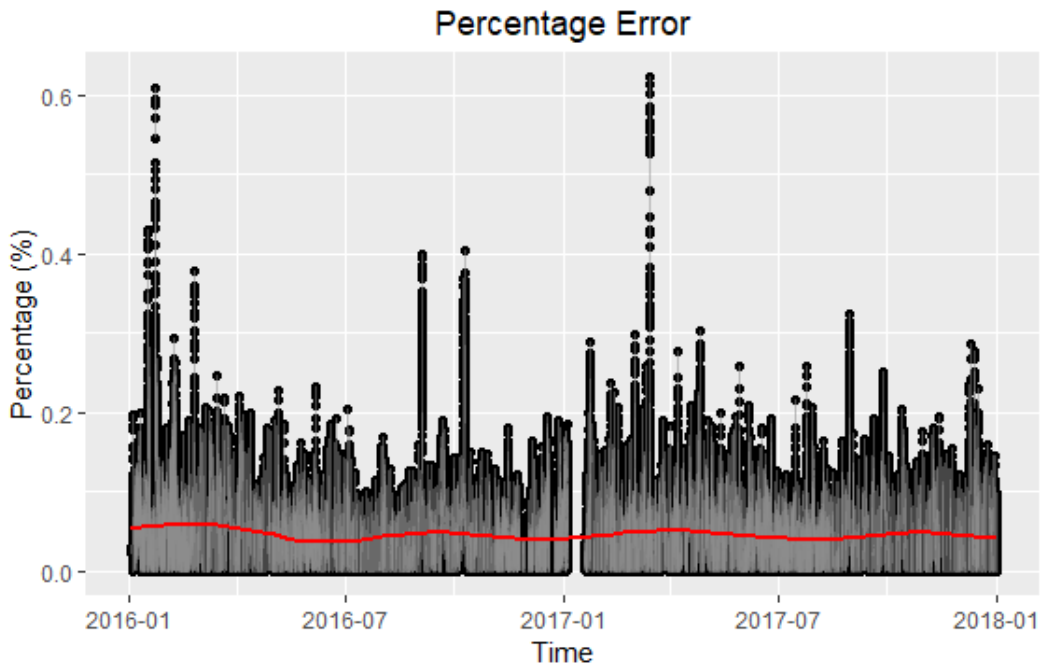


Figure 27 - Pressure Percentage Error

The forecasted values for the pressure range from 987 to 1040 hPa. With this range and the MAE error from Table 7, the pressure had 0.88 % of error. Such value could be expected since the pressure was the variable with the best result, as shown in section 4.1. Figure 26 shows the mean absolute error for this variable and Figure 27 shows a percentage error, both with their respective moving averages shown in a red line. As expected the graphs show small levels of error for the pressure. This variable is easier to predict for having smoother changes and for having smaller variance from one station to another due to the similar geolocations.

The air temperature is another variable with good predictability for having smaller abrupt changes over short periods. The error results demonstrate that. The values variation ranges from -8.808 to 34.72 °C, and with the MAE of 1.093 °C, it returns a 2.51% of maximum error. Like the pressure, the geolocation of both stations lead to conditions that are similar enough to facilitate the predictability from one station to another. As shown in section 4.1.3 the method was also able to reproduce this curve with great accuracy.

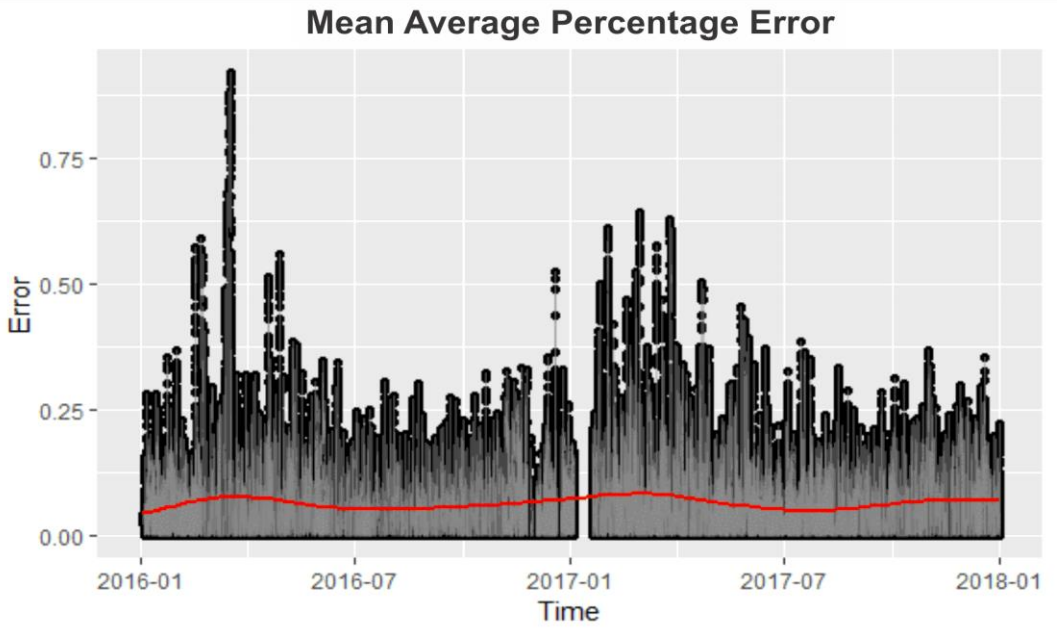


Figure 28 - Air temperature Mean Average Percentage Error

Figure 28 and Figure 29 present the MAPE and the percentage error for this variable. The error levels were small; although the percentage error shows a much larger amount of big errors then it was found in the pressure graphs. To put it into a better visual

perspective, Figure 30 is a histogram of the MAPE values distribution. The histogram shows that although there are more points with elevated error, the majority of the MAPE values are under 0.25%.

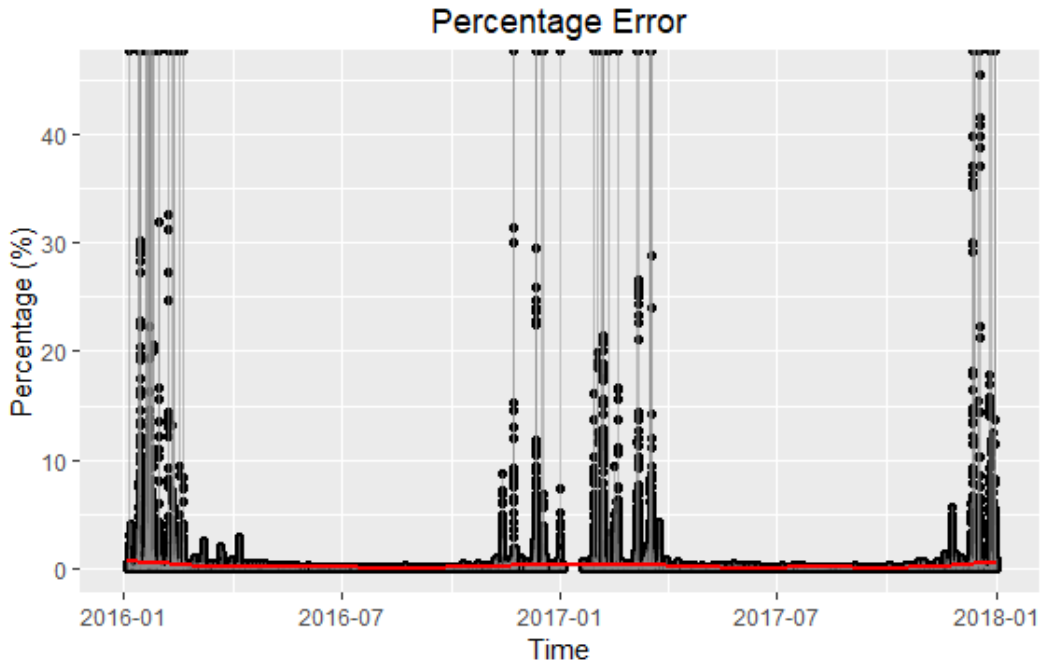


Figure 29 - Air temperature percentage error

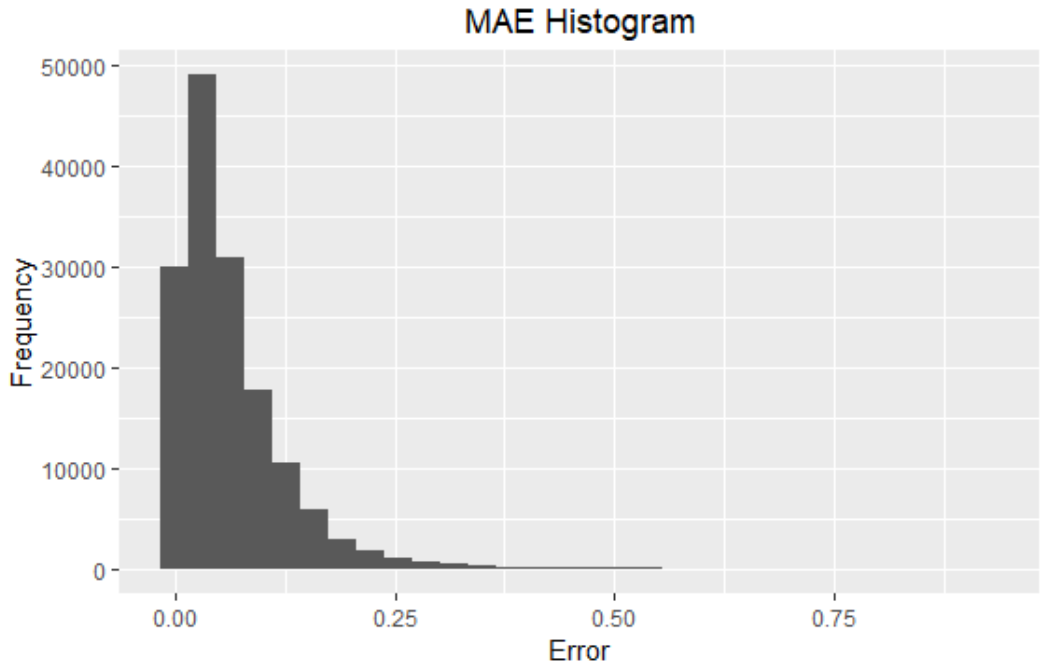


Figure 30 – Air temperature histogram of the MAPE (2016-2017)

The wave temperature, as shown, is much more subjected to abrupt changes, which lead to higher forecast instability. The forecast value ranges from 2.38 to 31.3°C returning a 3.3% error. Although this value is not much larger than the air temperature results, Figure 31 and Figure 32 shown a distribution of the error that is less concentrated than those found in air temperature graphs. That indicated the presence of abrupt changes that consequently lead to the dispersion of the forecasted values compared to the observed values.

As it can be noticed from the figures, the mean absolute error that was produced by the model was not very high, but that contrasts with the results that can be observed in Figure 20. That is due the fact that the wave temperature had only varied in a short range of temperatures but with a higher frequency than the previous variables. Therefore, the error was able to remain a short number, but the reality that is shown by Figure 20 indicates that the values are not very reliable.

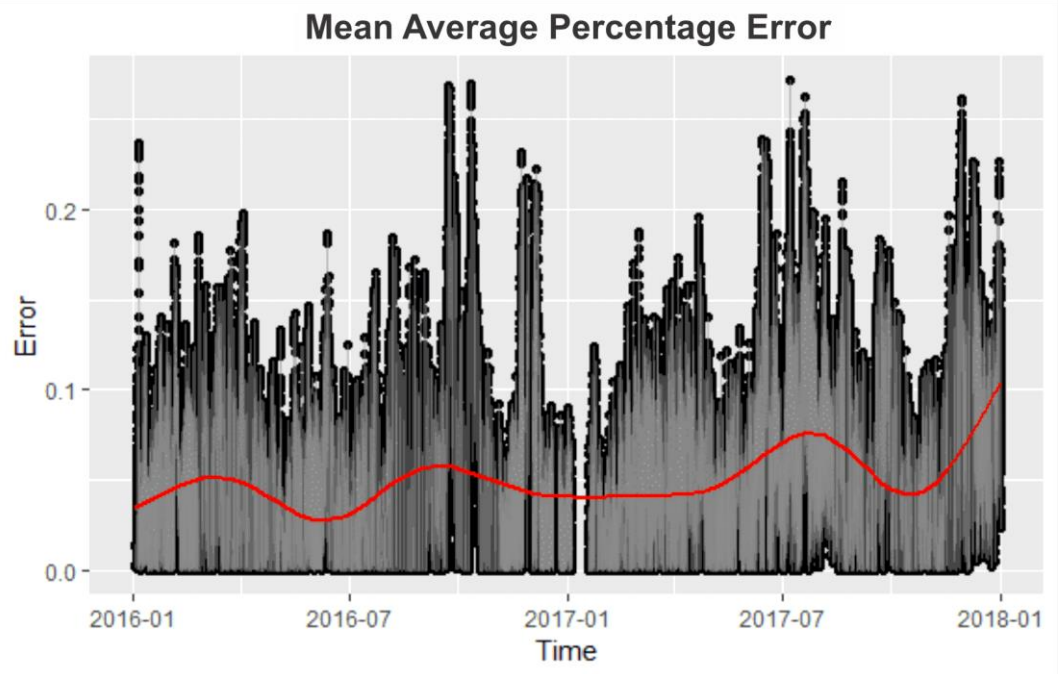


Figure 31 - YKTV2 Wave Temperature MAPE

In contrast to the results of the mean absolute error, which were relatively small for this study, the percentage error was able to provide a more accurate state of the

prediction by showing extremely high error levels and sustaining the fact that the results for this variable are not reliable.

It is certainly important to reinforce that since the YKTV2 and MNPV2 stations are located about 50 kilometers from each other and one of them is located by a sea shore while the other on is located by a river shore. This are elements that can provide a significant difference on the wave temperatures patterns, and are likely to be responsible for the differences that are found in the prediction of this variable.

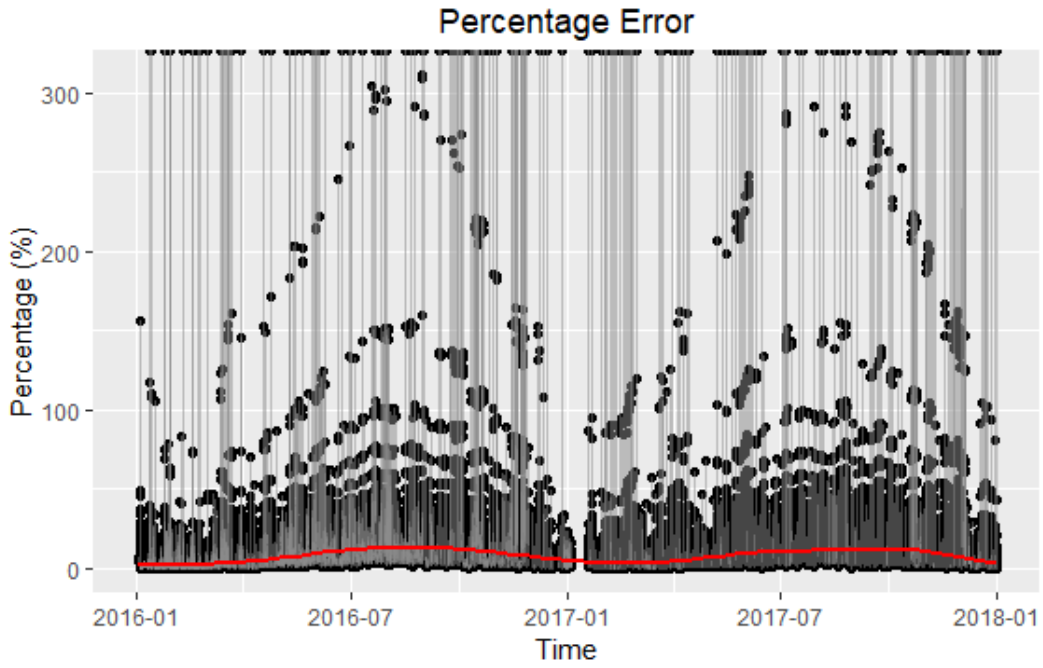


Figure 32 - Wave Temperature Percentage Error

The error results for the gust speed have shown a good consistency along the time series, although there are a considerable amount of data with very high percentage error. This can be explained by the fact that the peak values can often be resulted by sudden high speed gust that can be generated by a series of factors, such as the wind direction change. These sudden events are harder to predict, as it was previously mentioned. In spite of that, there are a good amount of data with good prediction values. The forecasted values vary from 0.8 to 22.6 m/s producing 6.06% error for the corresponding. Figure 33 shows that the error is considerably compacted with few values of super errors. Those errors can be better visualized in the percentage error histogram in Figure 35. The

histogram gives a clear indication of some reliability in the prediction of this variable, since the majority of the results percentage error are concentrated below 10%.

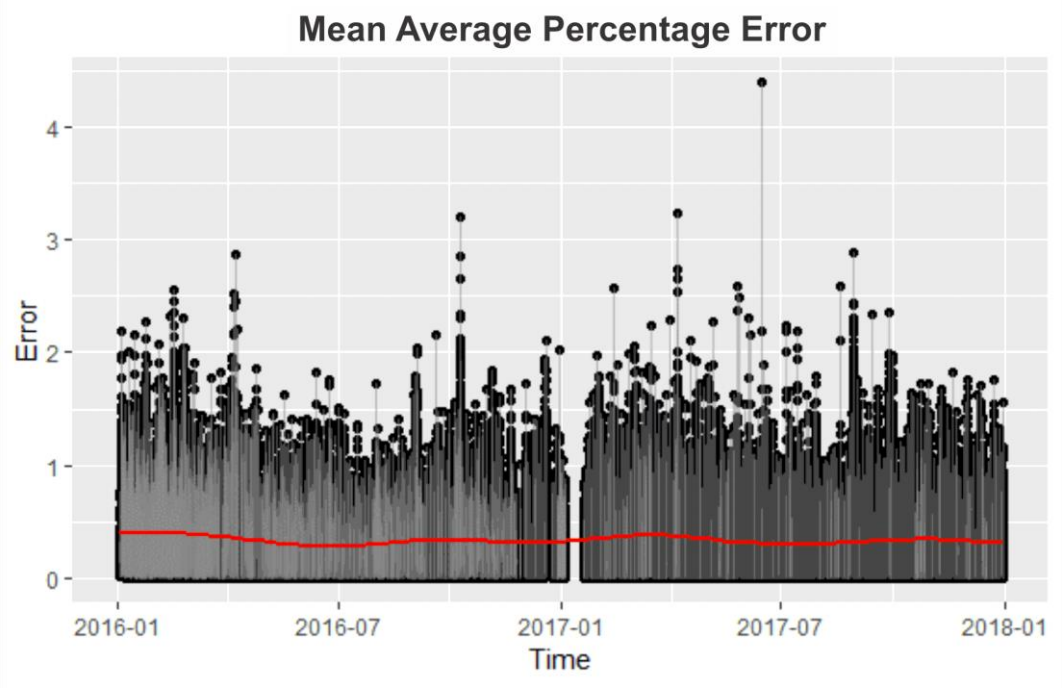


Figure 33 - Peak Gust Speed MAPE

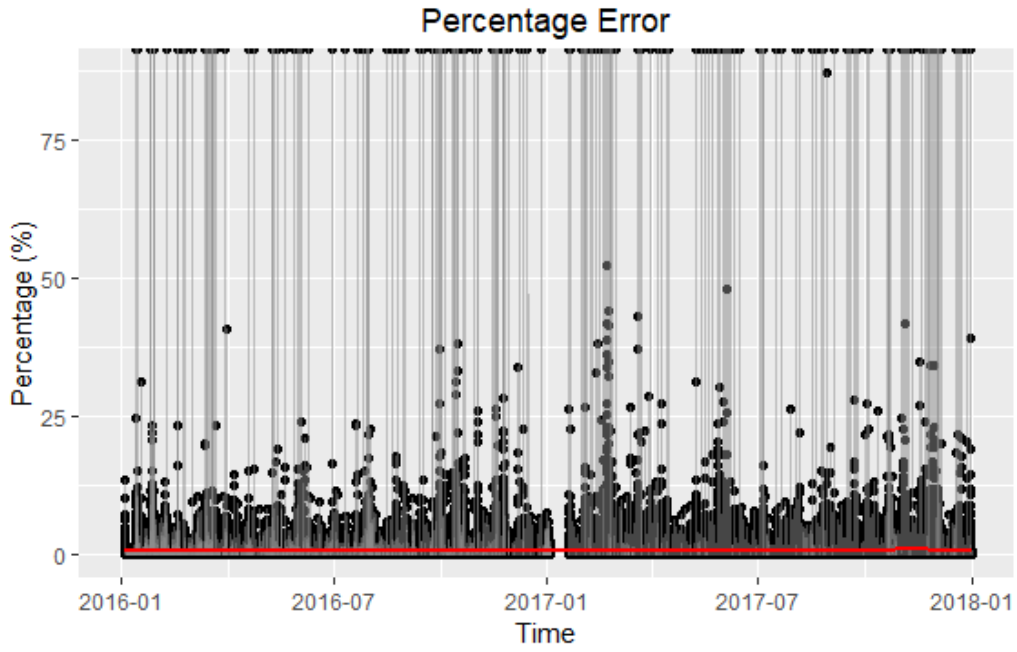


Figure 34 - Peak Gust Speed Percentage Error

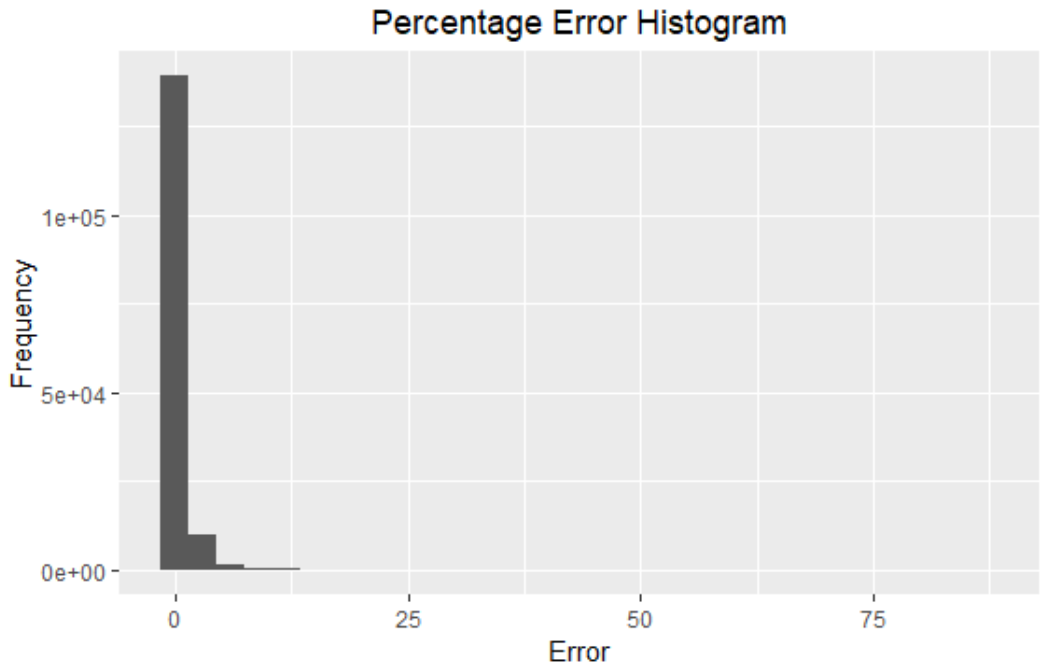


Figure 35 - Percentage Error Histogram (2016-2017)

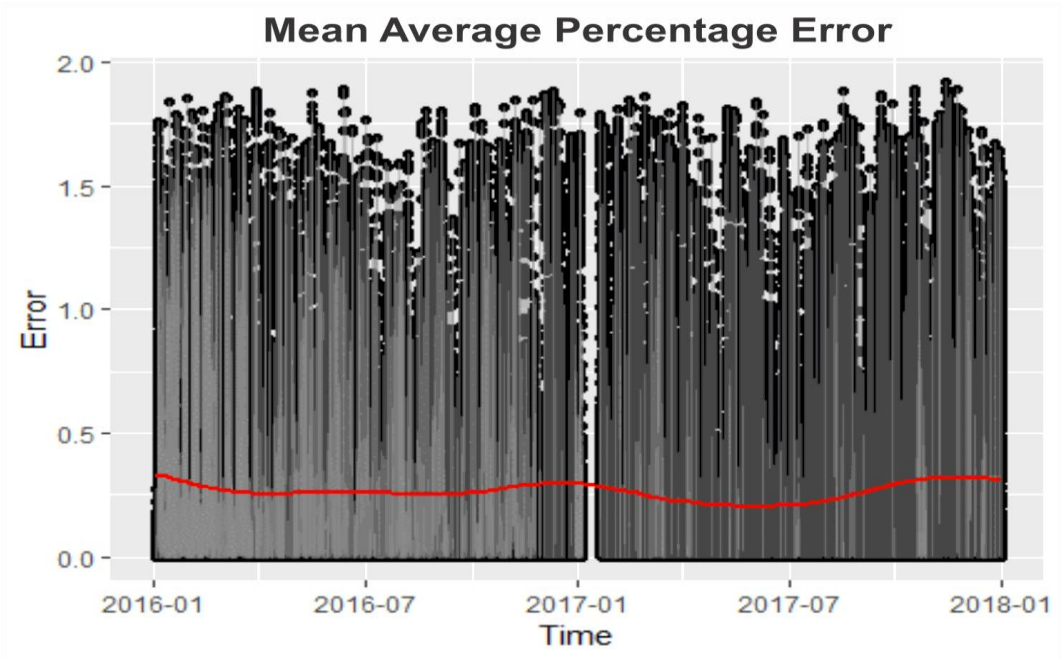


Figure 36 - Wind Direction MAPE

Figure 36 presents a graph for the wind direction error. Although the moving average line is located at a considerably low level of error, there is a large amount of data with very high levels of error. Also, since the range of values of these variable is very large, values that were far from the observed value could still create a small level of error. Therefore, the results for this variable are considered to be the worst from the variables analyzed in this study, with prediction results of no reliability. Further studies should be conducted to attempt an optimization of the results for this variable.

Finally, to have a graph view of the result's variance, Figure 37 presents a Taylor Diagram for all six variables. The plots were generated according to the method proposed by Karl Taylor and implemented by CRAN plotrix in R (Taylor, 2001). These graphs further reinforce the results that were previously described. For the pressure and the air temperature, the correlation was near to 1, indicating the precision of the results. In addition, for these variables, it can be noticed that the red dot is very near the observation standard deviation curve, indicating great reliability for those forecasts. The diagram also shows great results for the wave temperature, indicating that although the error for this variable was considerably high, the method was efficient in its predicting. This contradiction can be explained by the high variability of the wave temperature that has been previously discussed.

The wind speed and the gust speed had intermediary results. Both variable presented a standard deviation that is considerably low. However, the correlation for these variables are situated around 0.7 and 0.8, which is significantly smaller than the pressure and air temperature results, but it is still a reliably range. Moreover, the red dot is located far from the observation curve, indicating that there is a significant amount of that with results that are far from the observed and that do not fit the standard deviation range.

For the wind direction, the diagram indicates a very high standard deviation of the observation values. That reinforces that predicting this variable is hard, due to its instability. In addition, the correlation factor for this variable was the smallest. In the other hand, the red dot is positioned closer to the observation curve than in the wind speed diagram, indicating similar variability in both the forecasted and the observed cases.

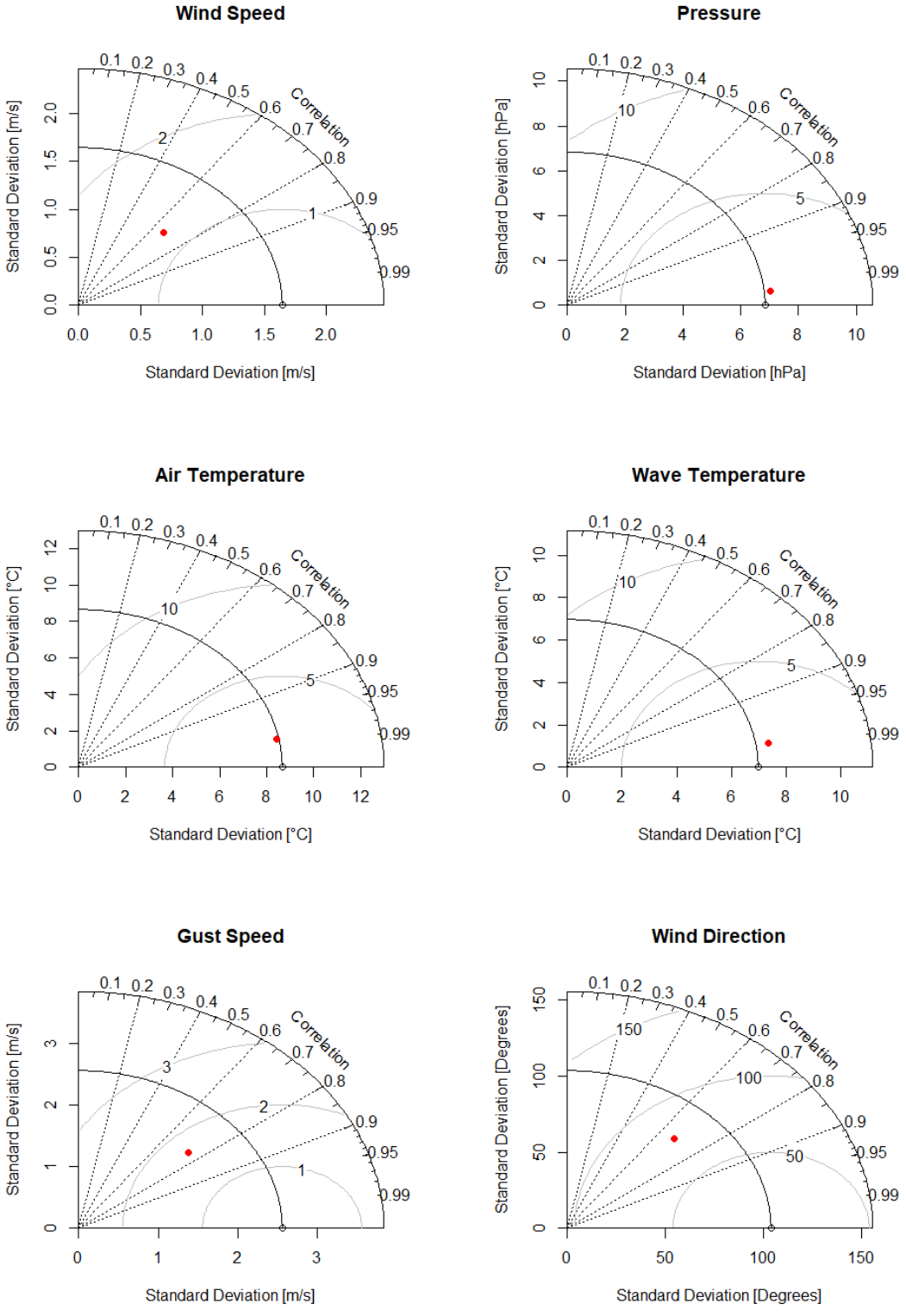


Figure 37 - Taylor Diagram of the standard deviation for forecasted series

Chapter 5

MULTI PHYSICAL VARIABLES

The second approach of this study was to apply the method using multiple variables. In this case, Equation 3 will be fully applied. Considering the individual results of the six variables as described in Chapter 4 and the computational limitations for the study, the method was applied using three prediction variables. The evaluation of ideal values for the variable *w* is beyond the scope of this study, and therefore its values were evenly distributed. Although this decision affects the quality of the results, the standard deviation of each prediction variable will assign different influence to them in the results and therefore the results can still be considered reliable.

Table 8 – Analyzed periods description

	Season	**Start Date**	**Start Index**
Week 1	Winter	February 4th	446400
Week 2	Spring	May 5th	468240
Week 3	Summer	August 6th	490560
Week 4	Fall	November 6th	512680

As previously mentioned in Chapter 4, the method showed a high demand of memory allocation. To be able to process the method with multiple variables, the memory of the virtual machine in use was upgraded to 64GB. Also, the analog windows used was reduced to 10 and the prediction time was of 4 months instead of 2 years. To have a larger range of results, the predictions were set to the middle week of each season of 2016. Table

8 presents the starting dates of each week period and their respectively vector indexes into the program. Every week period was composed of 1680 points of 6 minutes intervals.

5.1 Different Stations

The multivariable version of the code was applied for two cases. The first used prediction variables from a different station and the second one used prediction variables from the same station. In the case with different stations, the stations used were the same as in Chapter 4, where MNPV2 provides the historical data, and YKTV2 is the station to be predicted. The three predicting variables chosen were the Wind Speed (WSPD), the Pressure (PRES), and the Air Temperature (ATMP), and the variable to be predicted was the Wind Speed. It was decided to use the ATMP and the PRES based on their performance on the single variable approach. The WSPD was included in order to have the same variable in both the historical and in the prediction.

As previously mentioned, the Wind Speed is a variable that has a medium level of predictability when compared to the other variables used in this study. The results that were obtained for the WSPD in the single variable approach presented significant error and the prediction curve was only able to maintain the observed curve for part of the graph. Figures Figure 38, Figure 39, Figure 40 and Figure 41 present the results for the WSPD that were obtained using the multivariable approach. The results were found to be significantly better than those from the single variables approach, and that can be clearly observed when looking at the same week period on the single variable results.

Although the results were significantly better, it can still be observed that the prediction curve does not follow the observation behavior for all the periods and some high levels of error are still observed. There are three major factors that are still leading to this error. The first one is the calibration of the weights of each variable since the results here presented were generated by setting the same weight for all variables. The second is due to the use of historical data from a different station. The third is the nature of the variable, that as in the single variable approach, is of hard prediction due to high changes in small periods of time.

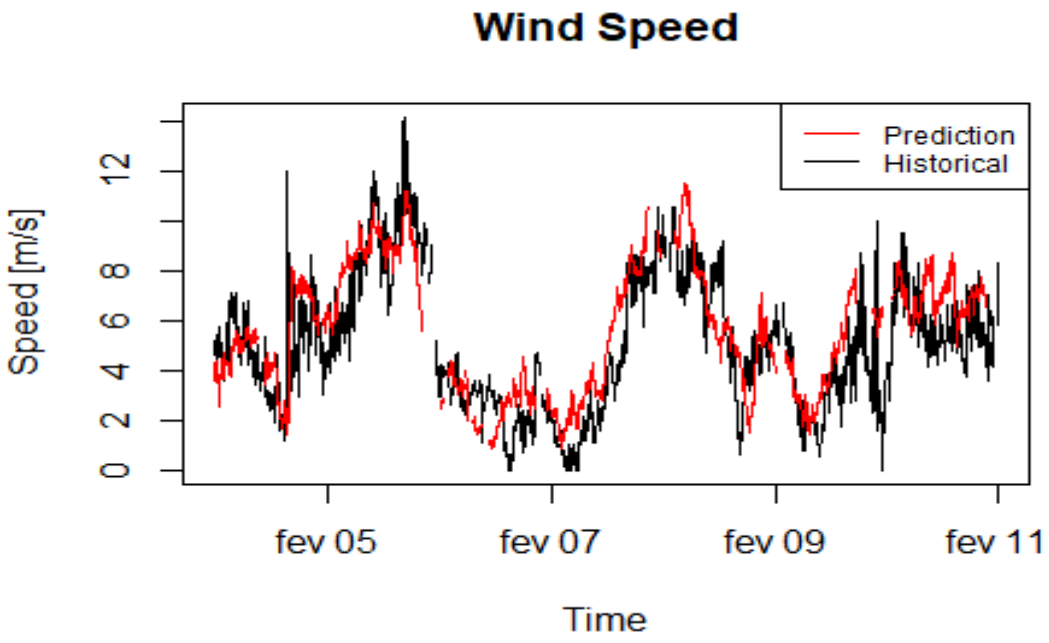


Figure 38 - Wind speed week 1 forecast

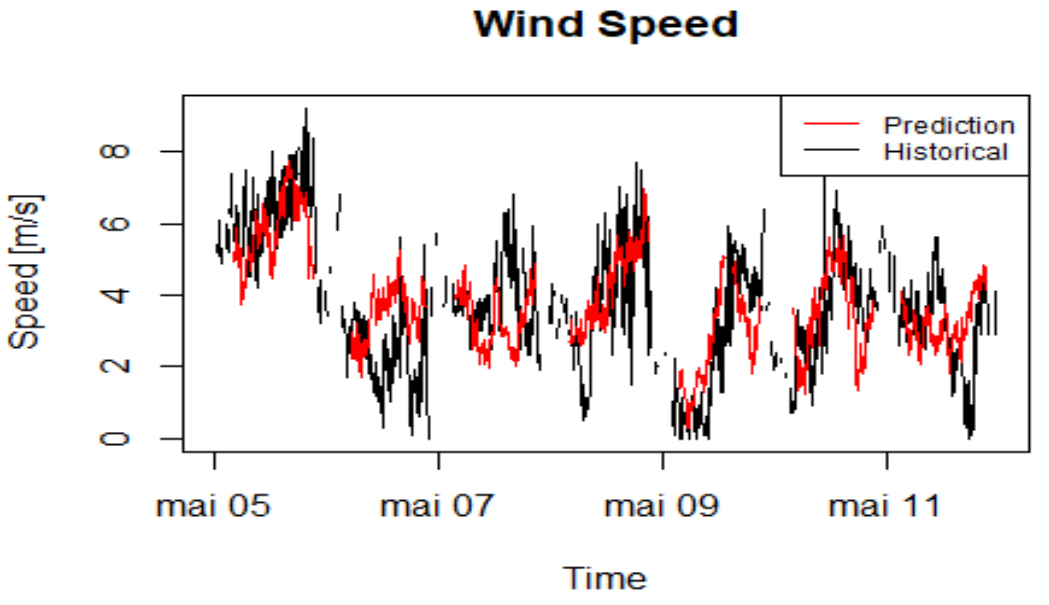


Figure 39 - Wind speed week 2 forecast

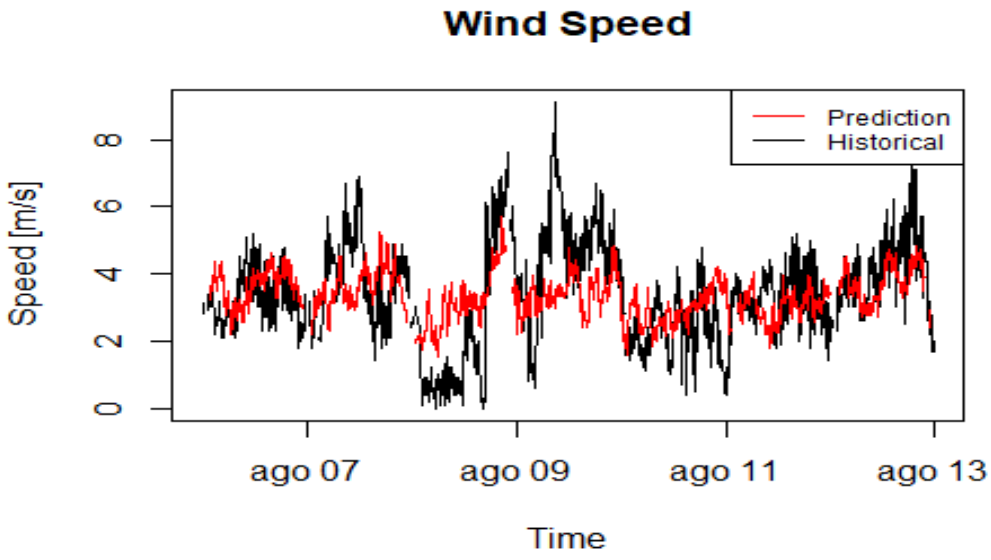


Figure 40 - Wind speed week 3 forecast

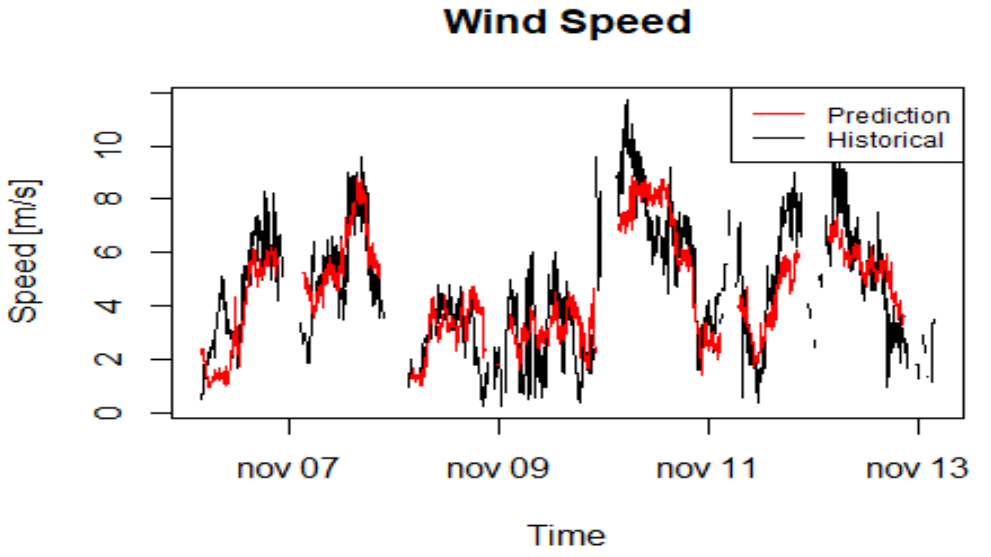
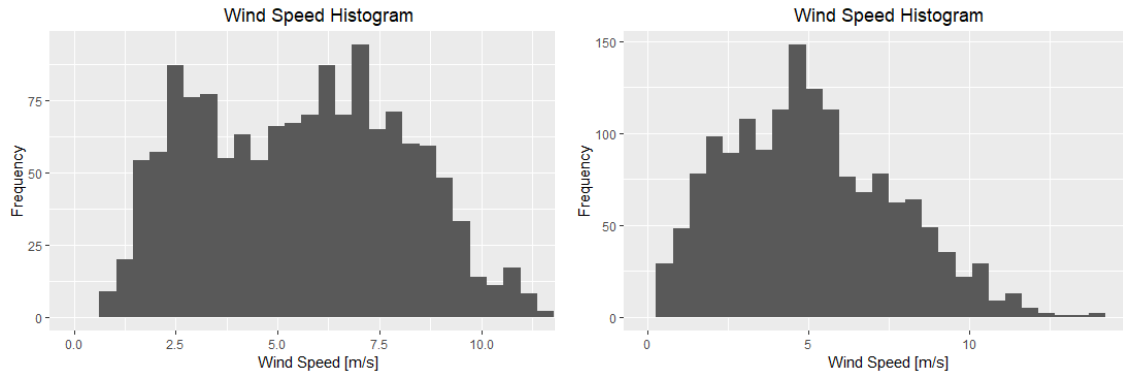


Figure 41 - Wind speed week 4 forecast

Figure 42 presents the histograms for the forecasted and the observed values of wind speed for week 1. Although the results with the multivariable approach may not seem much more accurate at first, comparing the values distribution gives a different

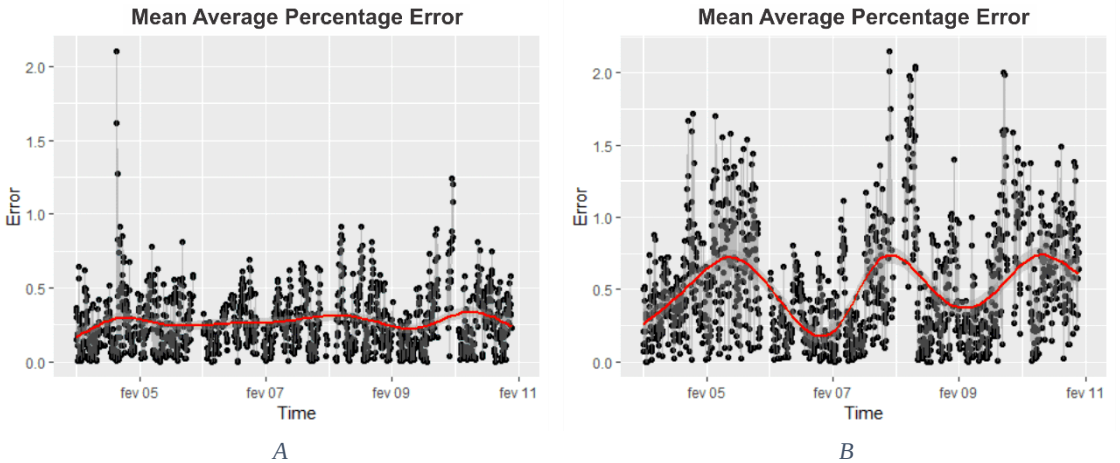
perspective. When compared to the results from Figure 22 it can be observed that this approach provide a better distribution of the values. Therefore, this approach is more efficient to predict events that are less frequent than the other one.



A *B*

Figure 42 – Week 1 wind speed histogram: A) Forecasted values; B) Observed values

Figure 43A shows the error results for the multivariable approach for Week 1 and Figure 43B shows the results for the single variable approach in the same period. There is a big reduction in the error obtained in the multivariable results and the majority of the values are concentrated around 0.25 where in the single variable the results not only are overall higher but are also much more spread.



A *B*

Figure 43 - Week 1 MAPE error: A) Multi variables results; B) Single variable results

To provide an overall view of the error, Figure 44 presents histograms for the MAPE for the four weeks that were analyzed, where Figure 44 A, B, C and D represent

Week 1, 2, 3 and 4, respectively. It can be observed that Week 3 was the one that presented the higher amount of value that are higher than 0.5, but still those values represent only 18% of the predictions.

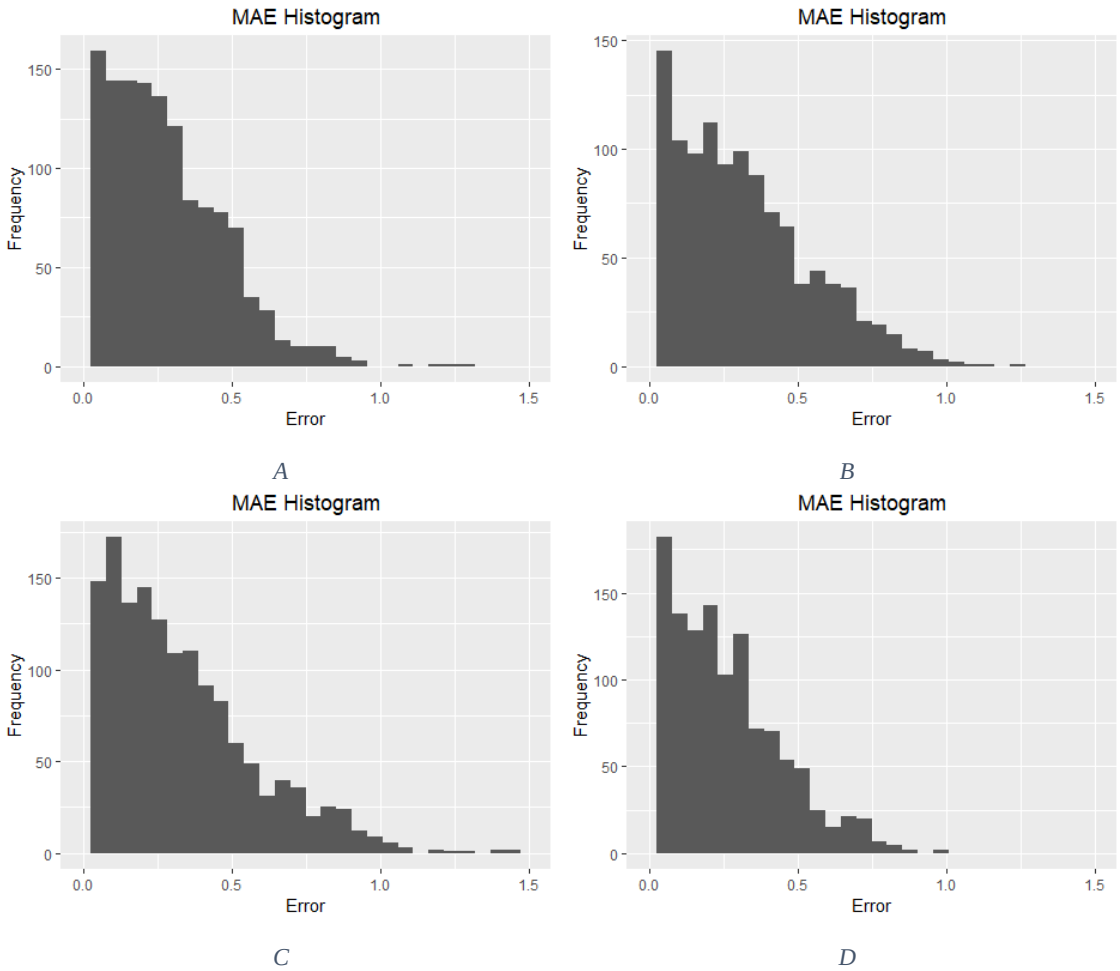


Figure 44 – Multivariable approach MAPE histograms; A) Week 1; B) Week 2; C) Week 3; D) Week 4.

To be able to better compare the results from chapter 4 with the multivariable approach, Figure 45 presents a Taylor diagram for week 1. Compared to the wind speed diagram from section 4.2, the correlation was found to be a bit superior. In addition, the proximity of the standard deviation was significant better. This reinforces that although the results were not visually better, the quality of the results was superior.

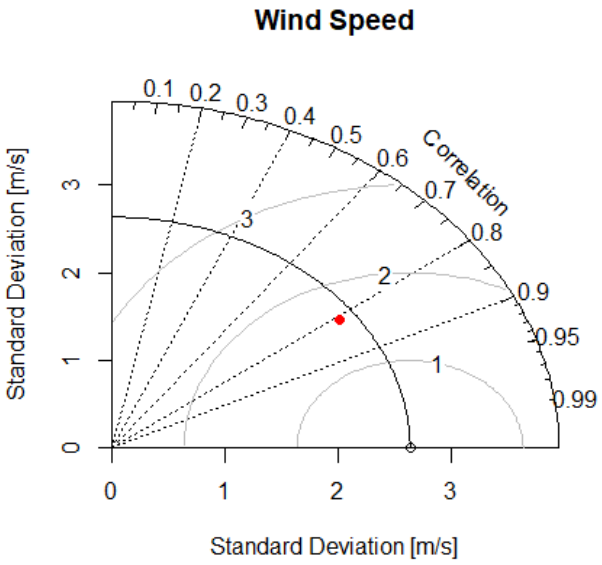


Figure 45 - Week 1 wind speed Taylor diagram

5.2 Single Station

For the second part of the multivariable analysis, both historical and prediction data are from the same station: the YKTV2 station was used for this section, to enable an easier comparison to the Section 5.1 results. The predicting variables chosen were the Gust Speed (GST), the Pressure (PRES) and the Air Temperature (ATMP), and the variable to be predicted was kept as the Wind Speed. The GST was also chosen based on its performance in the previous analysis and the other variables were kept to provided easy comparison of the results.

The results for the Wind Speed are shown in Figure 46 for Week 1, Figure 47 for Week 2, Figure 48 for Week 3 and Figure 49 for Week 4. The results that were obtained were of great quality: the prediction curves followed the behavior of the historical curve for nearly all the period analyzed.

Figure 50 presents the histograms for the forecasted data and the observed data. Like in section 5.1 it can be observed that the results are more evenly distributed than the single variable approach. Although the results in this section are better than those from

section 5.1, the distribution of the values of the data is similar. That only emphasizes that the use of different stations can have great value with proper calibration of the variables.

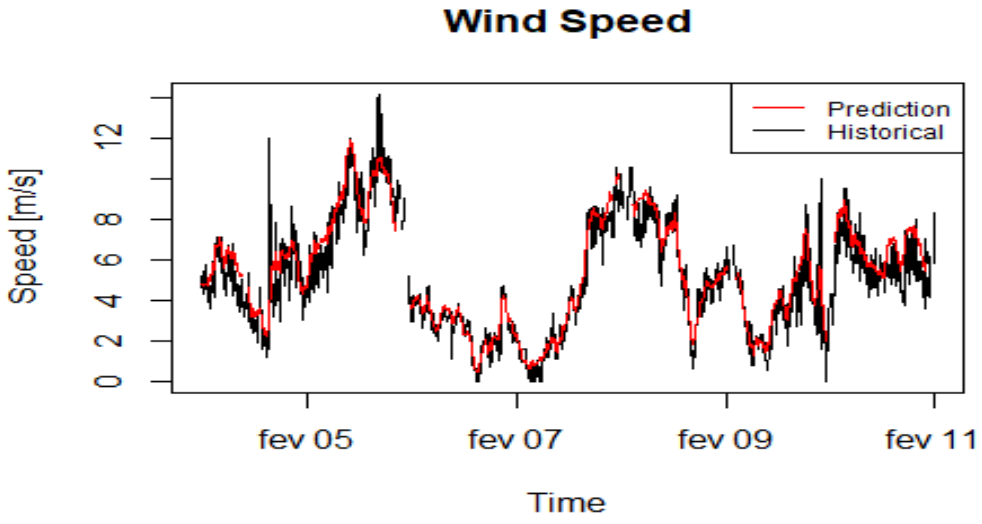


Figure 46 - Wind speed week 1 forecast

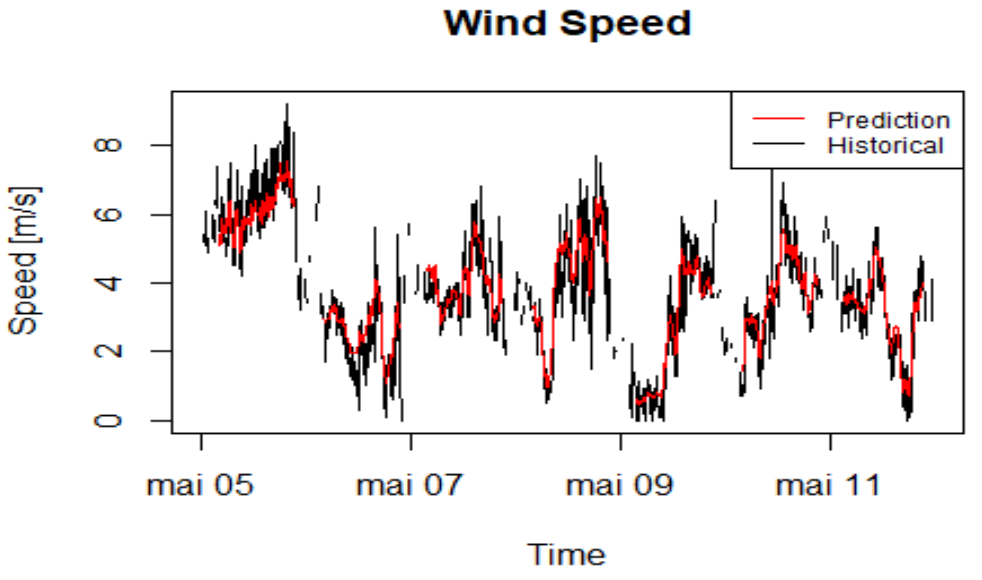


Figure 47 - Wind speed week 2 forecast

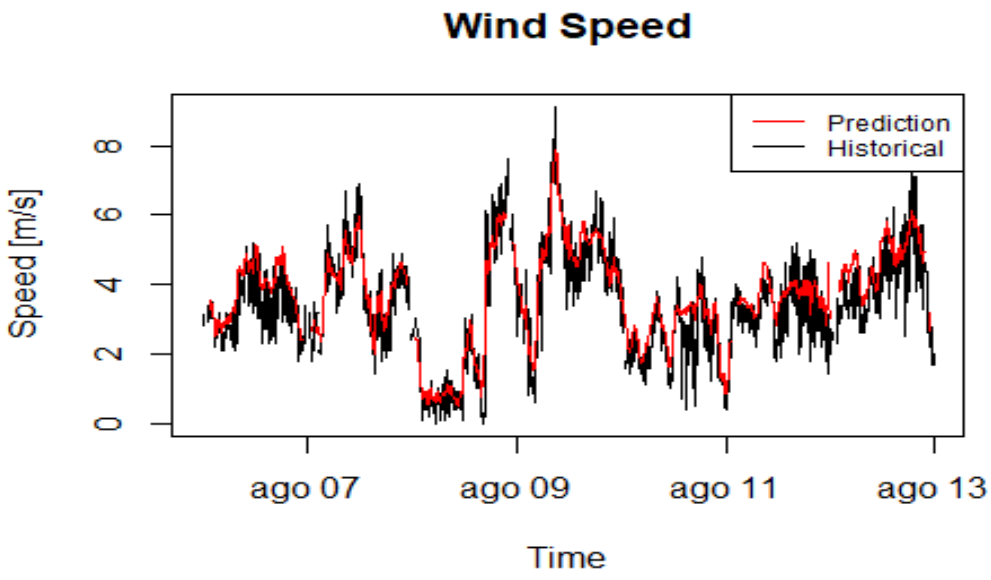


Figure 48 - Wind speed week 3 forecast

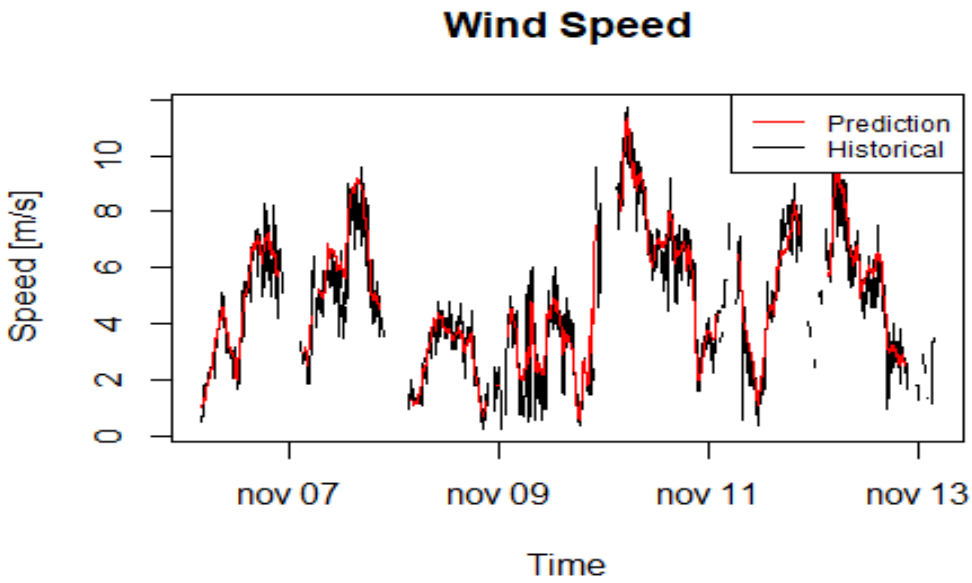
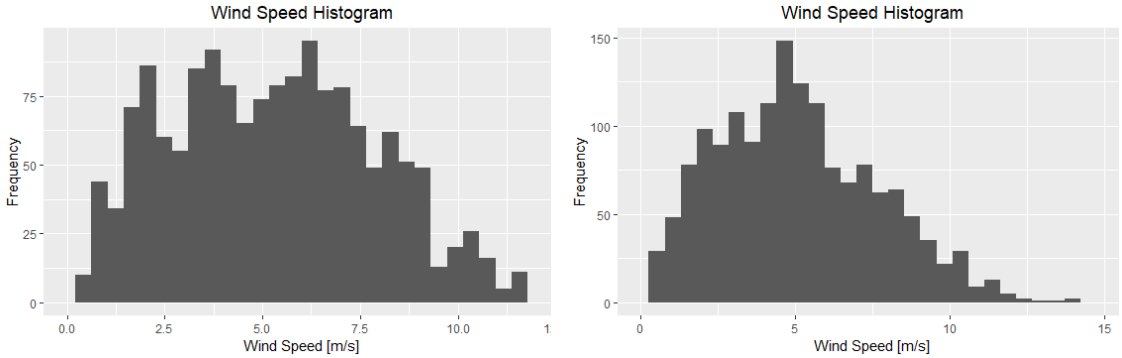


Figure 49 - Wind speed week 4 forecast



A *B*

Figure 50 – Week 1 wind speed values histogram: A) Forecasted values; B) Observed values

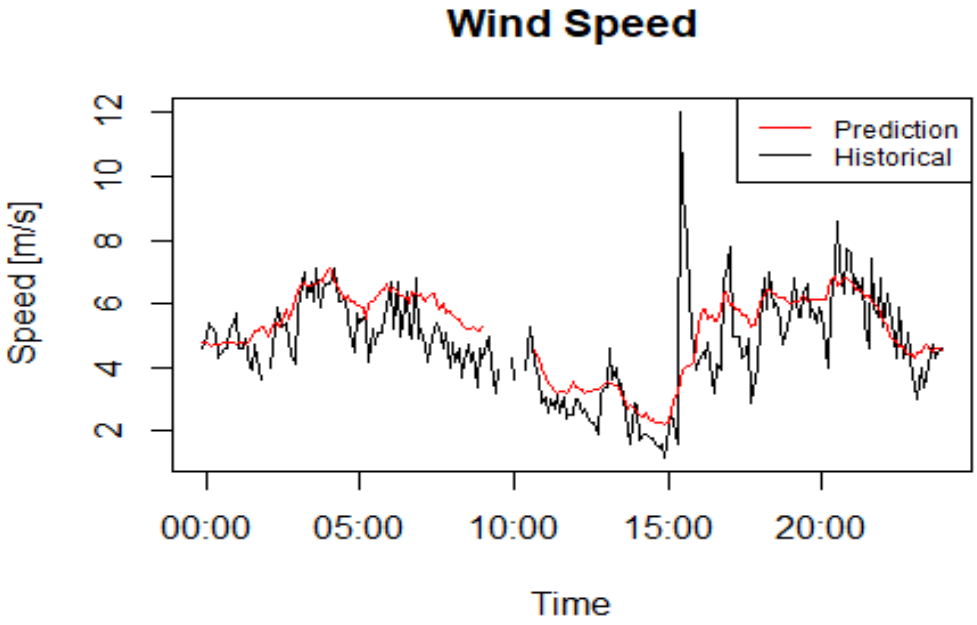


Figure 51 - 24 hours long wind speed forecast

Although the results showed themselves to be of great precision when compared to those previously acquired, they still carry some divergences with the historical values. Figure 51 shows a zoom into the first 24 hours of Figure 46. In this zoomed in version of the graphic it is possible to see that although the results were very efficient in following the behavior of the historical data and had a small level of error, the results were still not

completely precise, specially in points of greater changes over small periods of time. Implementing more accurate values for the weight of each variable could still be an alternative to have even more accurate results but it is not expected to have results of much greater values. Based on the literature, the best way to achieve results of higher accuracy is by having longer training data, which could provide better analogs for specific events and for the abrupt changes.

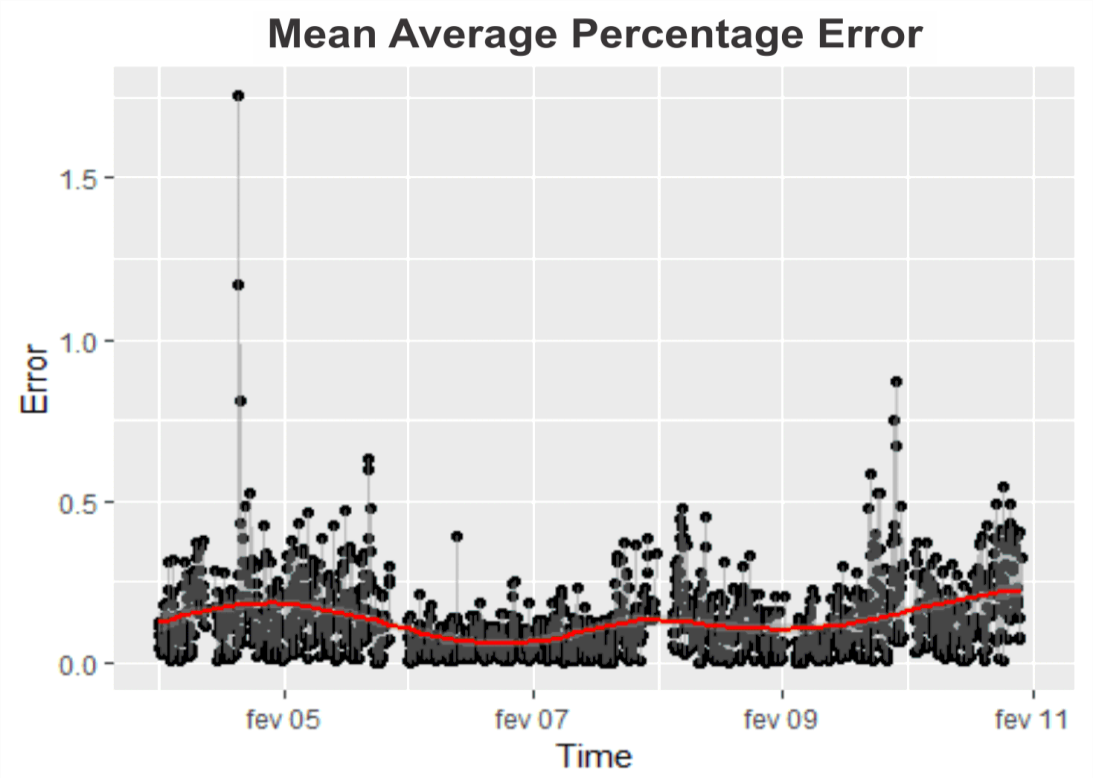


Figure 52 – Week 1 wind speed MAPE

Figure 52 presents the MAPE error for the first week of this study. It can be observed that the majority of the results are bellow 0.25. These results are considerably better than those from section 5.1 where they were located around 0.25 but with a great part of the values reaching up to 0.5. Figure 53 shows the error distributions for the four weeks that were predicted. Similarly to the results in section 5.1; it can be observed that Week 3 had the higher concentration of values over 0.25, but those only represent 16% of the observations. Only 18 points had an error higher than 0.5.

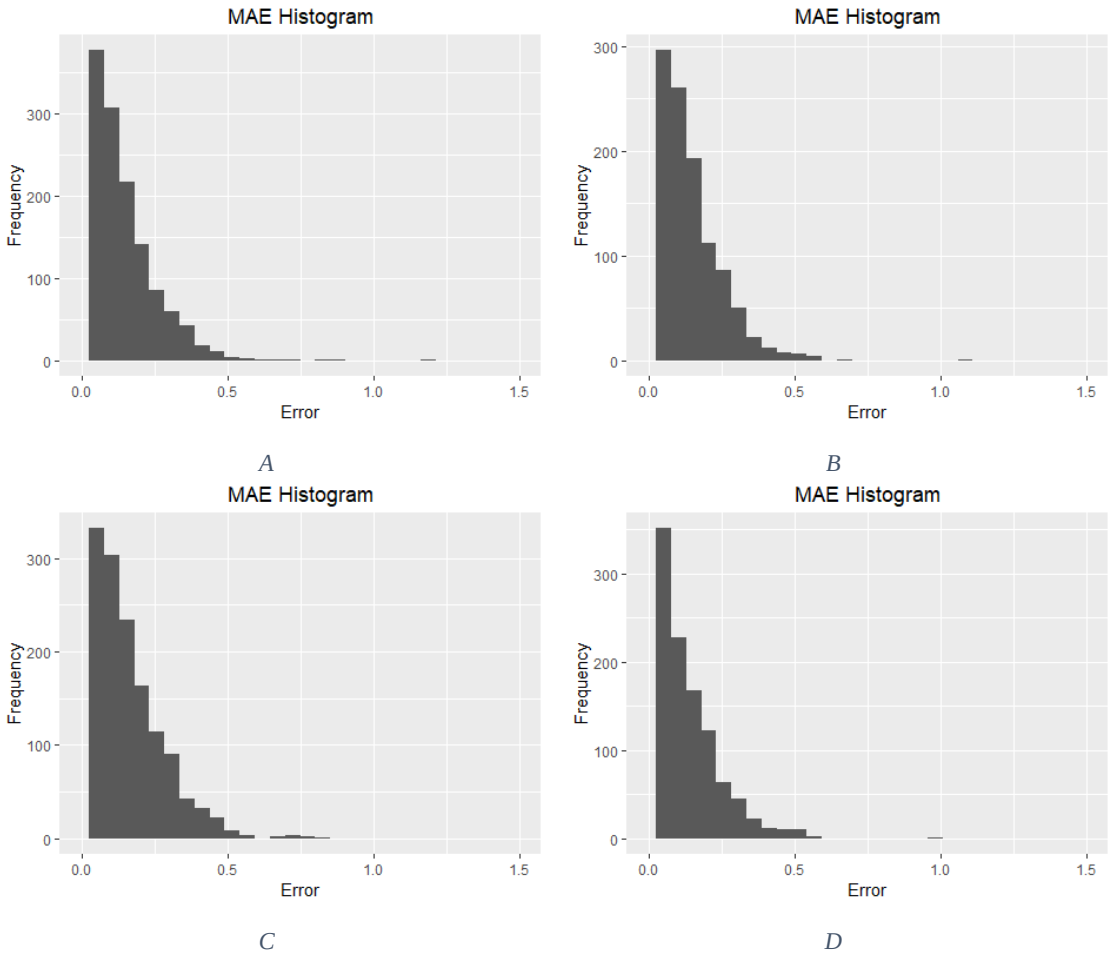


Figure 53 – Wind speed MAPE Histograms: A) Week 1; B) Week 2; C) Week 3; D) Week 4

To be able to have a more clear comparison between the results, Table 9 presents the accumulated error values for both the sections 5.1 and 5.2 represented by Week B and Week A respectively. All variables in Table 9 were calculated with the equations listed in section 2.5. Since section 5.2 had better results the Skill Score was calculated with those values in the denominator, which generated negative values as a consequence. The table shows that the accumulated errors are considerably high, specially the percentage error. That is justified since the predictions from this section were able to maintain the behavior of the observed curves but did not have a high precision for the exact observed values.

Figure 54 presents the Taylor diagram for week 1. As it would be expected, the results here were considerably superior to those previously presented. The correlation was

significantly better and the standard deviation proximity was also improved when compared to the results from section 5.1.

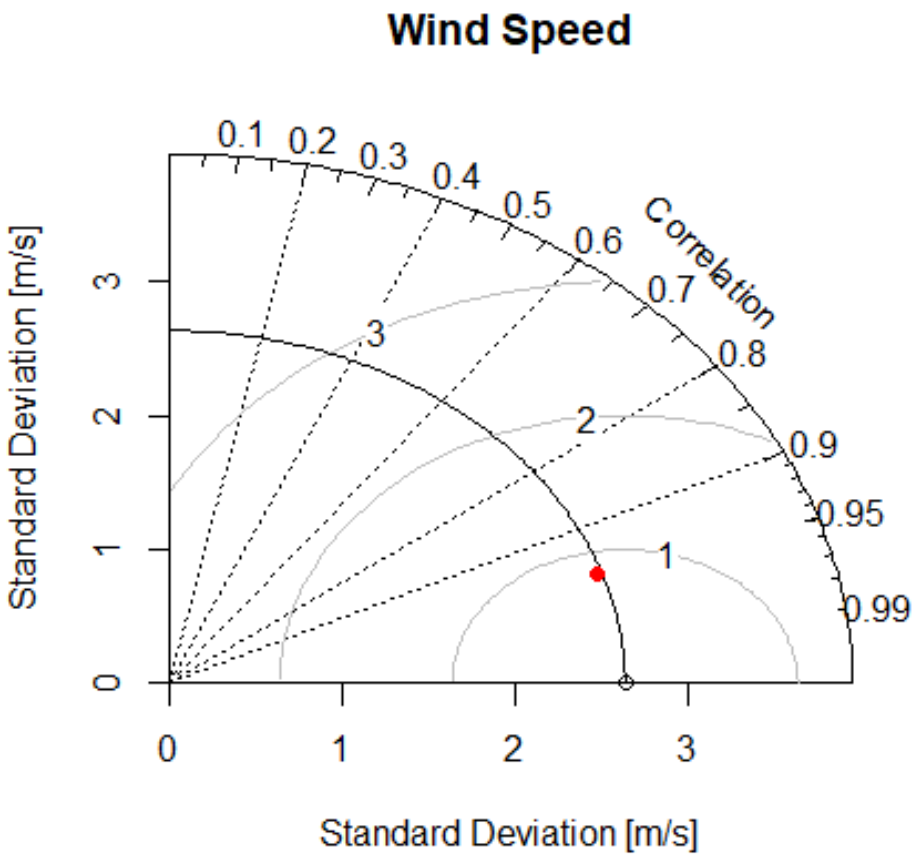


Figure 54 - Week 1 Taylor diagram

To have an overall view of the results for both the single variable and the multivariable approaches Table 10 presents the skill score of each analysis compared to the results for the wind speed and for the pressure from the single variable approach. The skill score was calculated using the MAE results. This variable was used over the other for being better for the description of large datasets, according to the literature previously described. The Wind speed was chosen for being the most frequent variable presented in this study and the pressure for being the overall best result found.

Table 9 - Accumulated error for the multivariable approach

Variable	MAE [m/s]	RMSE [m/s]	ARE [%]	Skill Score
Week 1 A	0.655	0.892	15.494	-1.08
Week 1 B	1.362	1.706	27.962	
Week 2 A	0.490	0.629	15.475	-1.33
Week 2 B	1.139	1.412	25.357	
Week 3 A	0.504	0.650	17.546	-1.17
Week 3 B	1.094	1.397	26.192	
Week 4 A	0.562	0.757	14.081	-1.05
Week 4 B	1.154	1.458	23.820	

The skill score results for both Tables 9 and 10 give a more clear perspective of the results. It can be observed that the forecast of the pressure was highly consistent even with the single variable approach. The results also indicate that without refining the variables *w* and *k* values, the use of multivariable might not provide much better results than those for single variable use. That indicated that the increase in the computational power required might not always be justifiable.

Table 10 - Skill Score of the results

Variable	Skill Score	
	Wind Speed	**Pressure**
Wind Speed	0	-1.032
Wind Direction	-48.785	-100.142
Wave Temperature	-0.007	-1.047
Air Temperature	-0.143	-1.323
Gust Speed	-0.380	-1.804
Pressure	0.508	0

Continuation Table 10

Variable	Skill Score	
	Wind Speed	**Pressure**
Week 1 A	0.315	-0.392
Week 1 B	-0.425	-1.895
Week 2 A	0.488	-0.041
Week 2 B	-0.191	-1.420
Week 3 A	0.472	-0.072
Week 3 B	-0.145	-1.326
Week 4 A	0.412	-0.195
Week 4 B	-0.207	-1.453

Chapter 6

CONCLUSIONS

6.1 Conclusions

This study presented the use of an Analog Ensemble methodology applied to the forecasting of time series. The Analog Ensembles is a post processing technique that allows long-term forecasts to acquire higher precision. The methodology was applied to a group of weather stations in the state of Virgina in the United Stated of America. The main goal was to produce good forecasts for each station using the data provided by other stations.

Focused on finding the reliability of the use of the Analog Ensemble technique for multiple scenarios and paying attention to the computational power required, this study was divided in three major approaches. In the first one, six weather variables were individually forecasted for one station using data from a different station. In the second approach the wind speed for one station was forecasted by a set of variables from another station. Lastly, the wind speed was forecasted for one station by a set of different variables from the same station.

The results for the single variable approach varied for each variable. That was expected since the Analog Ensemble technique is sensible to abrupt changes. That being, the pressure, the air temperature, the wind speed and the gust speed presented good results. These variables can be considered well behaved along time, with smaller amounts of abrupt changes and therefore easier to predict. On the contrary, the wind direction and the wave temperature have a much higher amount of abrupt changes and are harder to predict and therefore had worse results in this section.

For the multivariable approach, it was found that using a set of variables from a different station to predict the wind speed can lead to a good prediction. However, the results were not much better than those acquired by the single variable approach for the

same variable. On the other hand, the set of variables from the same station were able to generate great results. The time series predicted by this approach was of great quality and highly similar to the observed series.

In summary, the use of the Analog Ensemble technique showed itself to be a powerful technique for post processing of forecasts. Also, the use of this technique to generate data for regions that lack observation stations have a great potential, but it is of great importance to have a well calibrated algorithm with a refined set of variables.

6.2 Method Proposal and Future Work

The Analog Ensemble method still is a considerably new technique and its use can still be expanded. The main purpose of the current work was to evaluate the efficiency of the method in forecasting data for one place based on observations from another place. The continuation of the study requires the use of data from multiple stations to predict a central station. There are two motivations to do it. The first one is to evaluate the increase in the accuracy of the method and compare it to the computational power required. The second is to be able to both reconstruct and forecast data for places with no weather measurements records.

The results that were presented in Chapters 4 and 5 showed that there is a great potential for forecasting in a site with data from a different sites. To improve the capability of the model multiple stations can be use. In order to do so, Equation 3 needs to be adapted to support the input of multiple stations forecasts and combine them in a single result. Equation 15 is a proposal for how the problem can be approached.

$$F_t, A_{t'} = \sum_{s=1}^{N_s} V_s \sum_{i=1}^{N_v} \frac{w_i}{\sigma_{f_i}} \sqrt{\sum_{j=-k}^{k} (F_{i,t+j} - A_{i,t'+j})^2} \qquad (15)$$

Where V_s is the weight of each stations over the final result and N_s is the number of stations in use. Determining the value for V_s would be the biggest challenge for this

approach. Like the variable w_i, the values of V_s can be found by a loop model to optimize the results. Although this method demands great processing power it is relatively simple and provides the best set of results. In the other hand, using loop interactions would limit the range of application for the model.

The advantage of using equation 15 to extract results from multiple stations is the possibility of generating data for reanalysis and hindcasting for places that don´t have observation stations. In that condition it would not be possible to evaluate optimal values for V_s with loop interaction. For that end, it would be necessary to evaluate what are the physical variable that would affect the weather differences between two sites. Equation 16 shows the variables that would have physical effects over the value of V_s.

$$V_s = V_s\left(d, \Delta h, \mu, G_b, t_p, I\right) \tag{16}$$

Where:

$d =$ Distance between the sites;

$\Delta h =$ Altitude difference between the sites;

$\mu =$ Humidity;

$G_b =$ Geographic Barriers;

$t_p =$ Dewpoint;

$I =$ Inertia.

It is important to mention that the effects of Latitude variation were neglected because the distance between the stations should not be large. The variance of the weather conditions along a radius is of difficult prediction and therefore the higher the distance between the stations to be compared, the harder it is to maintain a correlation between the values. The same applies for the variance of altitude, since it leads to pressure and temperature variance and those variables would cause great changes in weather conditions. The humidity and the dewpoint are believed to be variables of smaller impact, since the stations would be somewhat near each other, the values for this variables should be considerably similar.

The geographic barriers refers to ground changes between the stations. The presence of mountains between the two sites could completely change the weather behavior from one place to another. Also, it should consider the existence of different air masses and wind currents. Other elements that could impact this variable are the presence of cities, lakes and rivers between the stations.

The inertia conditions refers to the location of each variable. Different local elements could lead to slower or faster change in weather conditions. If one of the stations is located close to big masses of water (i.e. big lakes or the sea) or close to big cities, those elements would drastically change the weather behavior when compared to a site that do not have them. To evaluate this variable other variables can be considered, such as the soil heat, the superficial flow rate (run off), sub superficial flow rates and the available potential energy.

6.2.1 Topics for further research

From the development of this dissertation, there were relevant topics that could not be more deeply studied due to limitations that were found. The following topics are proposal for future studies that could complement and improved the results found in this dissertation.

1) The multivariable approach requires better definition of the weights for each variable in study. The variables weight can be obtained by interaction process. The literature reviewed suggests that good improvements in the precision of the method can be obtained by the proper definition of the optimal weights.
2) Further investigation on the forecasting of the variables for a station using a set of different stations with the definition of a parameter to adjust the weights of each station in the forecast. This parameter should consider physic characteristics of the stations, such as the distance between the stations as the altitude difference. With well defined parameters, it is believed that the forecast skill of the model to produce data for reanalysis could be very high.

REFERENCES

Akyurek, B. O., Akyurek, A. S., Kleissl, J., & Rosing, T. S. (2015). TESLA: Taylor expanded solar analog forecasting. *2014 IEEE International Conference on Smart Grid Communications, SmartGridComm 2014*, (November 2014), 127–132. https://doi.org/10.1109/SmartGridComm.2014.7007634

Alessandrini, S., Delle Monache, L., Rozoff, C. M., & Lewis, W. E. (2018). Probabilistic Prediction of Tropical Cyclone Intensity with an Analog Ensemble. *Monthly Weather Review*, *146*(6), 1723–1744. https://doi.org/10.1175/MWR-D-17-0314.1

Alessandrini, S., Delle Monache, L., Sperati, S., & Nissen, J. N. (2015). A novel application of an analog ensemble for short-term wind power forecasting. *Renewable Energy*, *76*, 768–781. https://doi.org/10.1016/j.renene.2014.11.061

Bollmeyer, C., Keller, J. D., Ohlwein, C., Wahl, S., Crewell, S., Friederichs, P., … Steinke, S. (2015). Towards a high-resolution regional reanalysis for the european CORDEX domain. *Quarterly Journal of the Royal Meteorological Society*, *141*(686), 1–15. https://doi.org/10.1002/qj.2486

Cervone, G., Clemente-Harding, L., Alessandrini, S., & Delle Monache, L. (2017). Short-term photovoltaic power forecasting using Artificial Neural Networks and an Analog Ensemble. *Renewable Energy*, *108*, 274–286. https://doi.org/10.1016/j.renene.2017.02.052

Charney, J. G., FjÖrtoft, R., & Neumann, J. Von. (1950). Numerical Integration of the Barotropic Vorticity Equation. *Tellus*, *2*(4), 237–254. https://doi.org/10.3402/tellusa.v2i4.8607

Cox, A. T., & Swail, V. R. (2001). A global wave hindcast over the period 1958-1997: Validation and climate assessment. *Journal of Geophysical Research: Oceans*, *106*(C2), 2313–2329. https://doi.org/10.1029/2001JC000301

Dee, D. P., Uppala, S. M., Simmons, A. J., Berrisford, P., Poli, P., Kobayashi, S., … Vitart, F. (2011). The ERA-Interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, *137*(656), 553–597. https://doi.org/10.1002/qj.828

Delle Monache, L., Eckel, F. A., Rife, D. L., Nagarajan, B., & Searight, K. (2013). Probabilistic Weather Prediction with an Analog Ensemble. *Monthly Weather Review*, *141*(10), 3498–3516. https://doi.org/10.1175/MWR-D-12-00281.1

Delle Monache, L., Nipen, T., Liu, Y., Roux, G., & Stull, R. (2011). Kalman Filter and Analog Schemes to Postprocess Numerical Weather Predictions. *Monthly Weather Review*, *139*(11), 3554–3570. https://doi.org/10.1175/2011MWR3653.1

Epstein, E. S. (1969). Stochastic dynamic prediction. *Tellus*, *21*(6), 739–759. https://doi.org/10.3402/tellusa.v21i6.10143

EWEA. (2020). *Wind Energy Scenarios for 2020*. Institutional report, EWEA (European Wind Energy Association), Rue d'Arlon 80, 1040 Brussels, Belgium.

GWEC. (2017). *Global Wind Report 2017*. Institutional report, GWEC (Global Wind Energy Council), Rue d'Arlon 80, 1040Brussels, Belguim.

Junk, C., Delle Monache, L., & Alessandrini, S. (2015). Analog-Based Ensemble Model Output Statistics. *Monthly Weather Review*, *143*(7), 2909–2917. https://doi.org/10.1175/MWR-D-15-0095.1

Junk, C., Monache, L. D., Alessandrini, S., & Cervone, G. (2015). Predictor-weighting strategies for probabilistic wind power forecasting with an analog ensemble, *24*(4), 361–379. https://doi.org/10.1127/metz/2015/0659

Katragkou, E., Garciá-Diéz, M., Vautard, R., Sobolowski, S., Zanis, P., Alexandri, G., … Jacob, D. (2015). Regional climate hindcast simulations within EURO-CORDEX: Evaluation of a WRF multi-physics ensemble. *Geoscientific Model Development*, *8*(3), 603–618. https://doi.org/10.5194/gmd-8-603-2015

Keller, J. D., Monache, L. D., & Alessandrini, S. (2017). Statistical downscaling of a high-resolution precipitation reanalysis using the analog ensemble method. *Journal of Applied Meteorology and Climatology*, *56*(7), 2081–2095. https://doi.org/10.1175/JAMC-D-16-0380.1

Kim, D., & Hur, J. (2018). Short-term probabilistic forecasting of wind energy resources using the enhanced ensemble method. *Energy*, *157*, 211–226. https://doi.org/10.1016/j.energy.2018.05.157

Kottek, M., Grieser, J., Beck, C., Rudolf, B., & Rubel, F. (2006). World Map of the Köppen-Geiger climate classification updated. *Meteorologische Zeitschrift*, *15*(3), 259–263. https://doi.org/10.1127/0941-2948/2006/0130

Leith, C. E. (1974). Theoretical Sskill of Monte Carlo forecasts. *Monthly Weather Review*. https://doi.org/10.1175/1520-0493(1974)102<0409:TSOMCF>2.0.CO;2

Lorenz, E. N. (1965). A study of the predictability of a 28-variable atmospheric model. *Det Norske Meteorologiske Institutt*.

Mitchell, M., Hershner, C., Julie, H., Schatt, D., Mason, P., & Eggington, E. (2013). Recurrent Flooding Study for Tidewater Virginia. *Virginia Institute of Marine Science, Center for Coastal Resources Management, William and Mary*, 1–141. https://doi.org/10.21220/V5TG79

Monteiro C, Bessa R, Miranda V, Botterud A, Wang J, C. G. (2009). *Wind Power Forecasting : State-of-the-Art 2009*. Report ANL/DIS-10e11. Argonne National Laboratory; November 2009.

Moura, A. D. (1996). Von Neumann e a previsão numérica de tempo e clima. *Estudos Avançados*, *10*(26), 227–236. https://doi.org/10.1590/S0103-40141996000100021

Murphy, A. H. (1988). Skill Scores Based on the Mean Square Error and Their Relationships to the Correlation Coefficient. *Monthly Weather Review*. https://doi.org/10.1175/1520-0493(1988)116<2417:SSBOTM>2.0.CO;2

National Oceanic and Atmospheric Administration. (n.d.). National Data Buoys Center. Retrieved November 22, 2018, from https://www.ndbc.noaa.gov/

Oliveira, G. S. de, & Florenzano, T. G. (2006). Satélites e o Meio Ambiente. In *Salto Para o Futuro / TV Escola* (pp. 79–96). Rio de Janeiro.

Phillips, N. A. (1954). Energy Transformations and Meridional Circulations associated with simple Baroclinic Waves in a two-level, Quasi-geostrophic Model. *Tellus*, *6*(3), 274–286. https://doi.org/10.3402/tellusa.v6i3.8734

Sampaio, G., & Dias, P. L. da S. (2014). Evolução dos Modelos Climáticos e de Previsão de Tempo e Clima. *Revista USP*, (103), 41. https://doi.org/10.11606/issn.2316-9036.v0i103p41-54

Shi, W., Schaller, N., Macleod, D., Palmer, T. N., & Weisheimer, A. (2015). Impact of hindcast length on estimates of seasonal climate predictability. *Geophysical Research Letters*, (42), 1554–1559.

https://doi.org/10.1002/2014GL062829.Abstract

Soares, C. G., Weisse, R., Carretero, J. C., & Alvarez, E. (2002). A 40 Year Hindcast of Wind, Sea Level and Waves in European Waters. *21st International Conference on Offshore Mechanics and Arctic Engineering, Volume 2*, 669–675. https://doi.org/10.1115/OMAE2002-28604

Storm, B., Dudhia, J., Basu, S., Swift, A., & Giammanco, I. (2009). Evaluation of the weather research and forecasting model on forecasting low-level jets: Implications for wind energy. *Wind Energy*, *12*(1), 81–90. https://doi.org/10.1002/we.288

Taylor, K. E. (2001). Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research*, *106*(1), 7183–7192. https://doi.org/10.1007/BF00139495

Thomas, T. J., & Dwarakish, G. S. (2015). Numerical Wave Modelling – A Review. *Aquatic Procedia*, *4*(Icwrcoe), 443–448. https://doi.org/10.1016/j.aqpro.2015.02.059

Thorarinsdottir, T. L., & Gneiting, T. (2010). Probabilistic Forecasts of Wind Speed : Ensemble Model Output Statistics using Heteroskedastic Censored Regression Technical Report no . 546, 1–22.

Trenberth, K. E., Koike, T., & Onogi, K. (2008). Progress and prospects for reanalysis for weather and climate. *Eos*, *89*(26), 234–235. https://doi.org/10.1029/2008EO260002

Vanvyve, E., Monache, L. D., Monaghan, A. J., & Pinto, J. O. (2015). Wind resource estimates with an analog ensemble approach. *Renewable Energy*, *74*, 761–773. https://doi.org/10.1016/j.renene.2014.08.060

Willmott, C., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model perfomance. *Climate Research*, *30*, 79–82. https://doi.org/10.3354/cr00799

Zhang, X., Li, Y., Lu, S., Hamann, H., Hodge, B. M. S., & Lehman, B. (2018). A Solar Time-based Analog Ensemble Method for Regional Solar Power Forecasting. *IEEE Transactions on Sustainable Energy*, *3029*(c). https://doi.org/10.1109/TSTE.2018.2832634

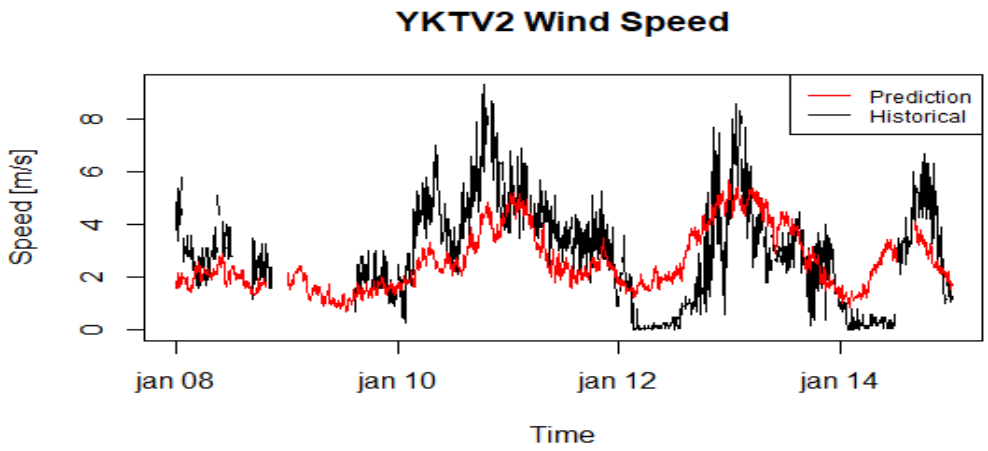
Appendices

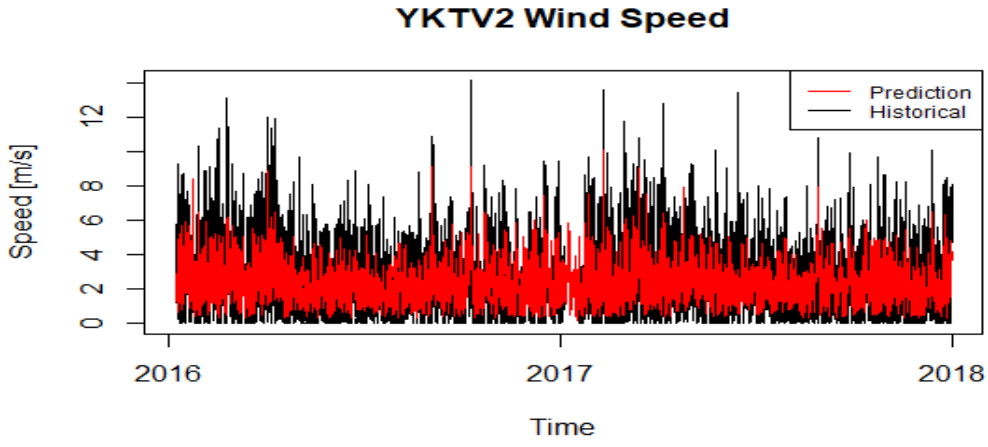
Appendix 1

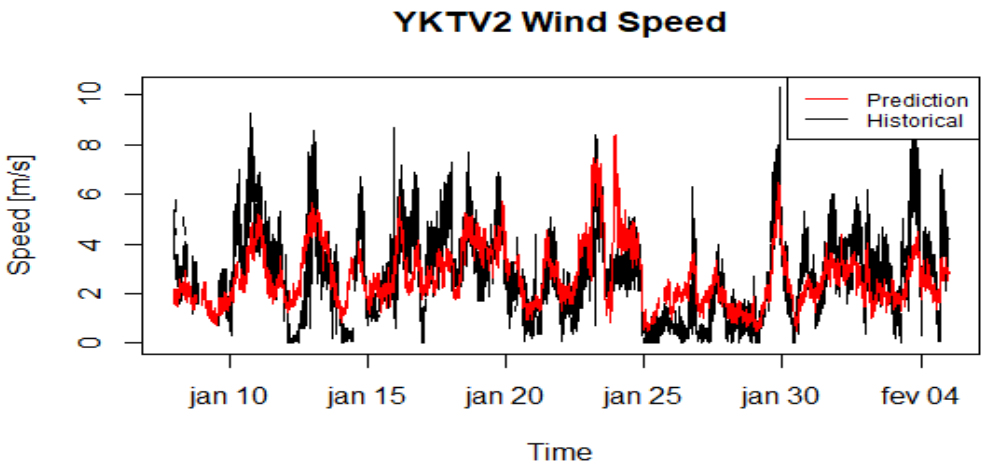
List of Graphics

This section presets all the graphics for each variable.

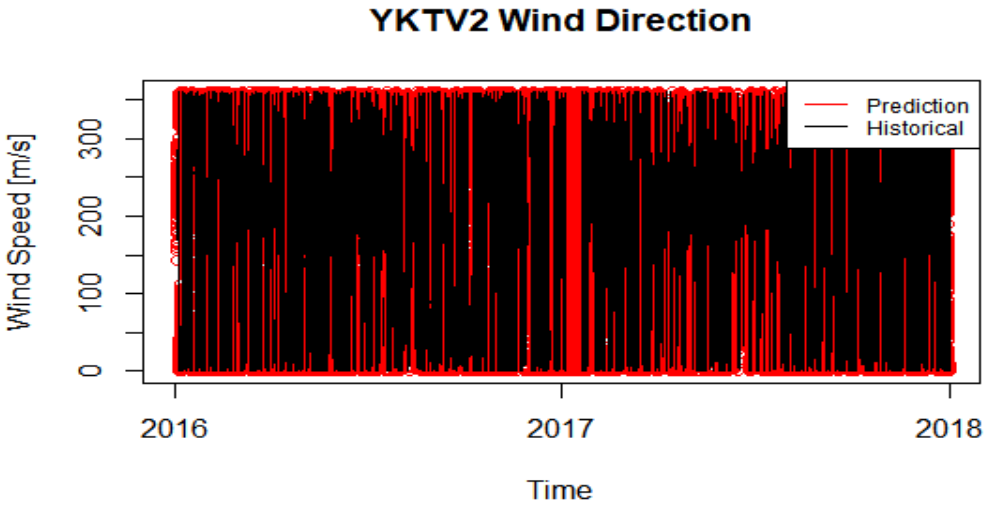
1 Wind Speed (WSPD)

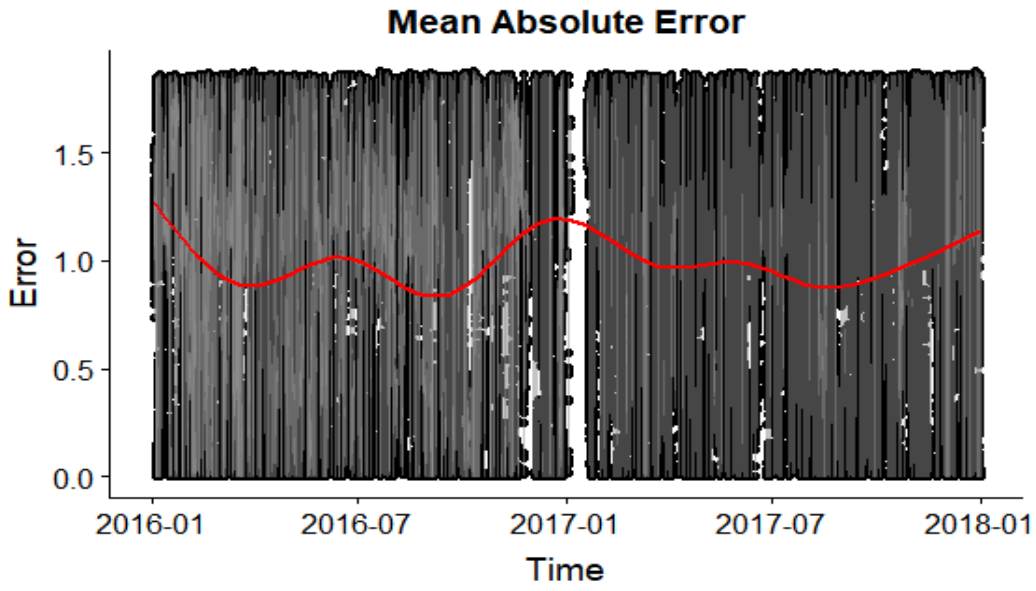




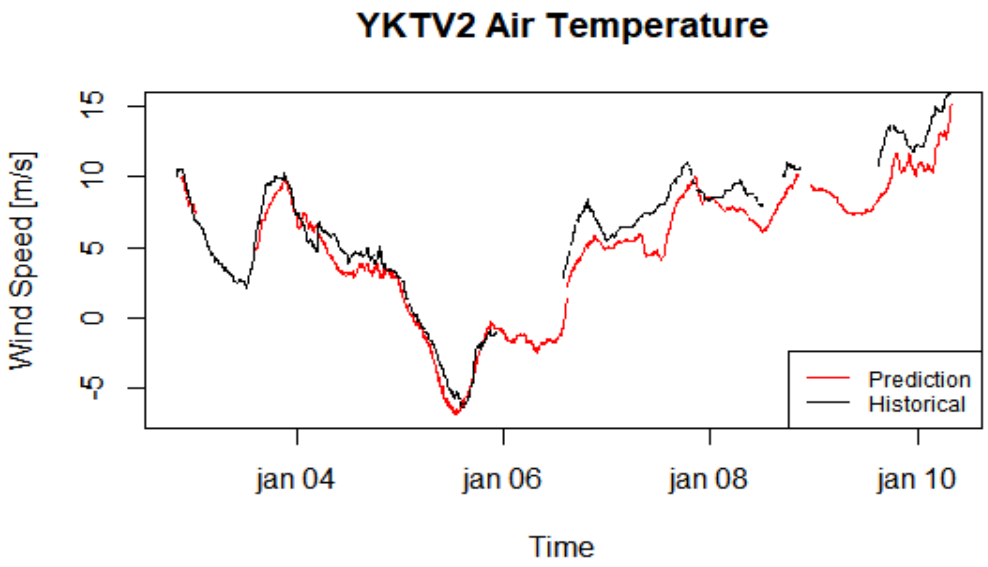


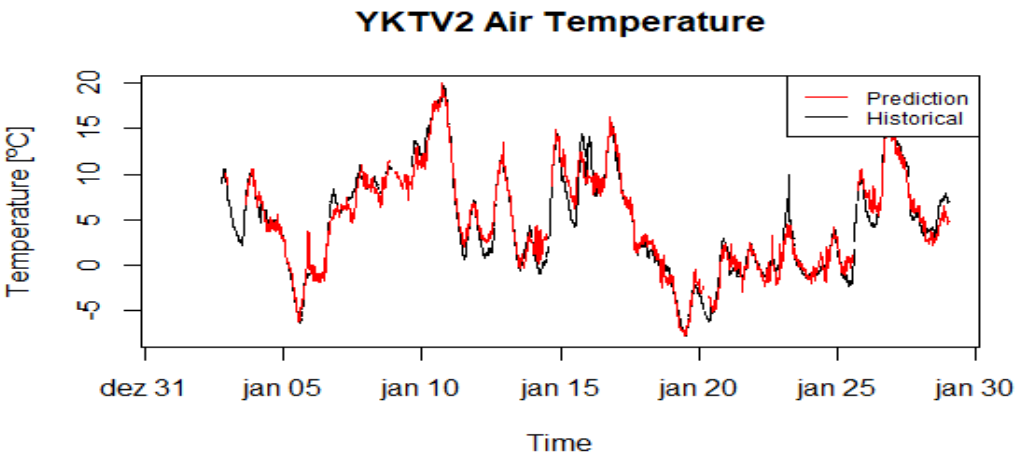
2 – Wind Direction (WDIR)

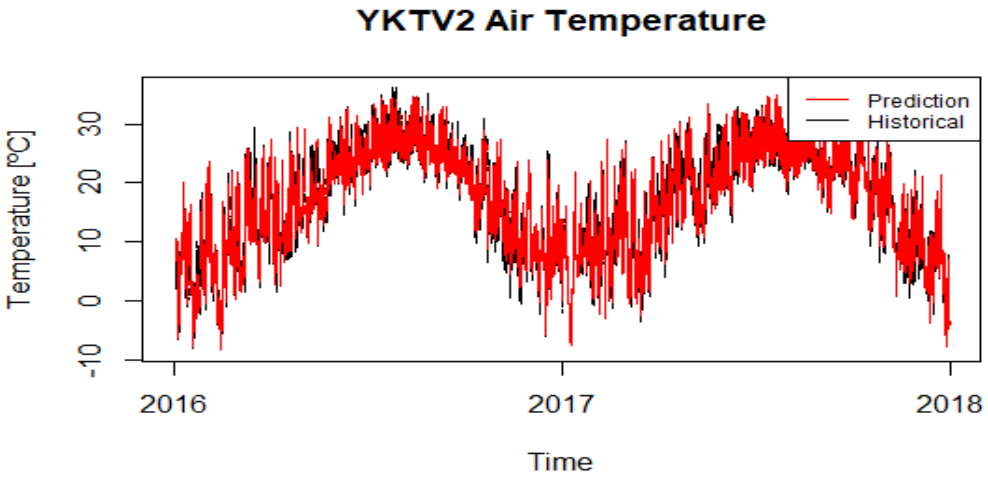


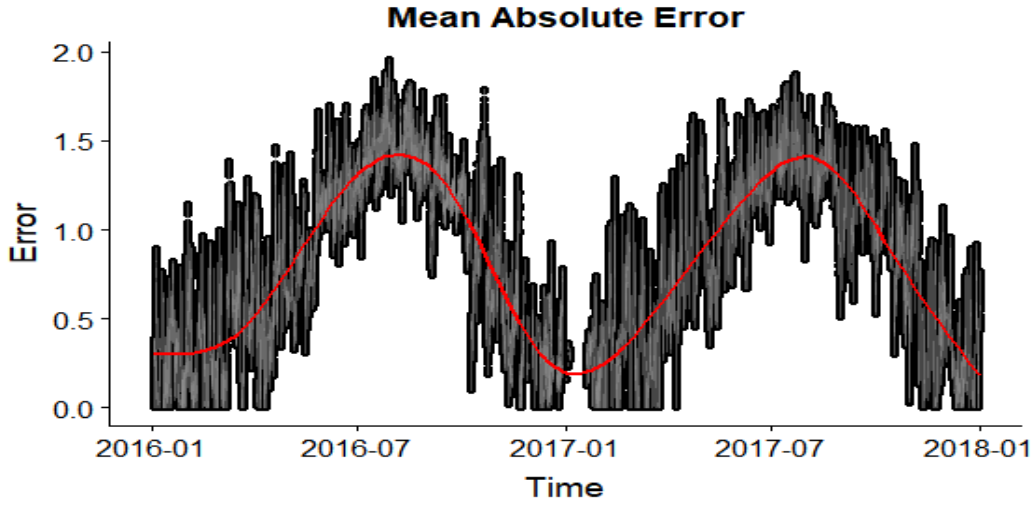


3 – Air Temperature (ATEMP)

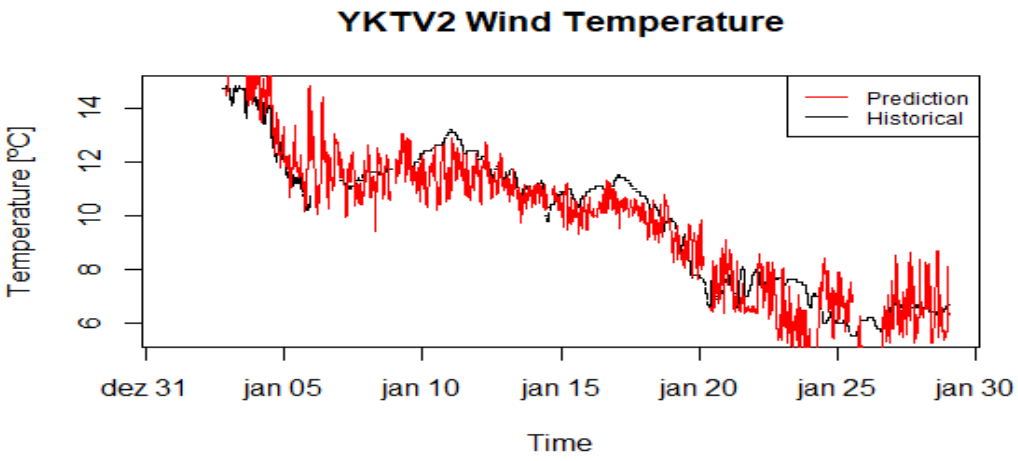


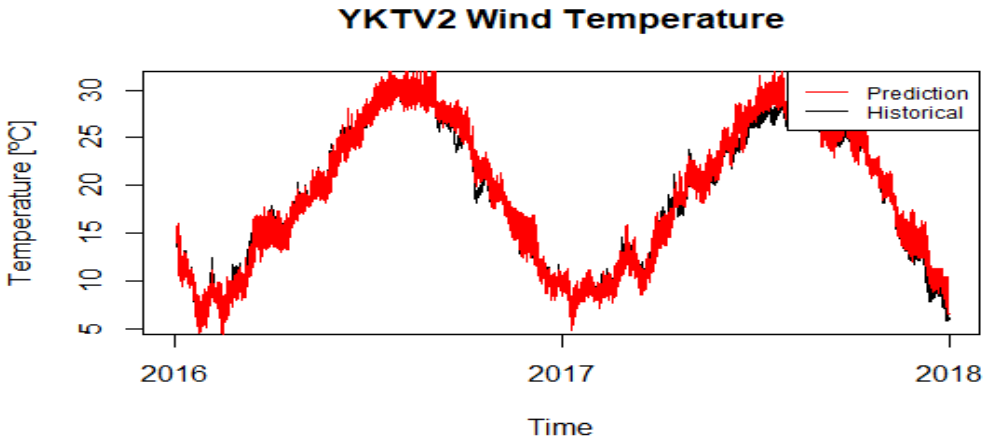

YKTV2 Air Temperature
Temperature [°C]
-5
0
5
10
15
20
Prediction
Historical
dez 31
jan 05
jan 10
jan 15
jan 20
jan 25
jan 30
Time

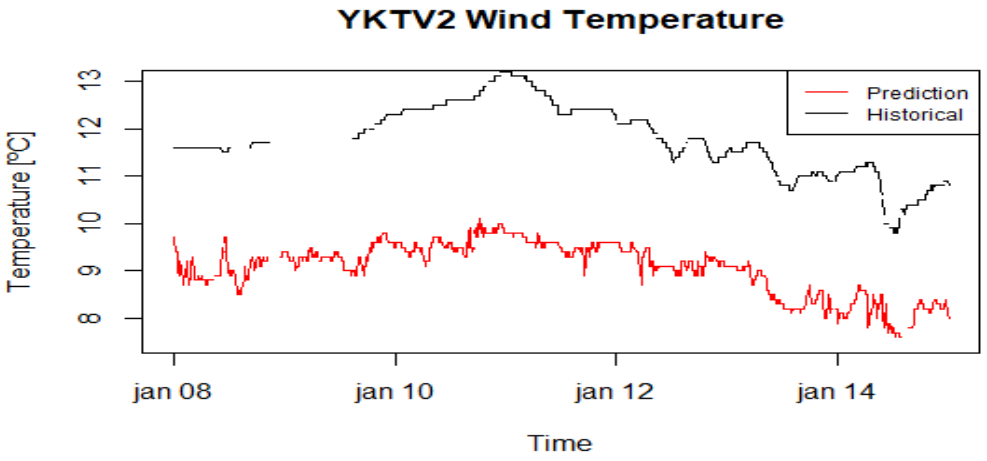

YKTV2 Air Temperature
Temperature [°C]
-10
0
10
20
30
Prediction
Historical
2016
2017
2018
Time


Mean Absolute Error
Error
0.0
0.5
1.0
1.5
2.0
2016-01
2016-07
2017-01
2017-07
2018-01
Time

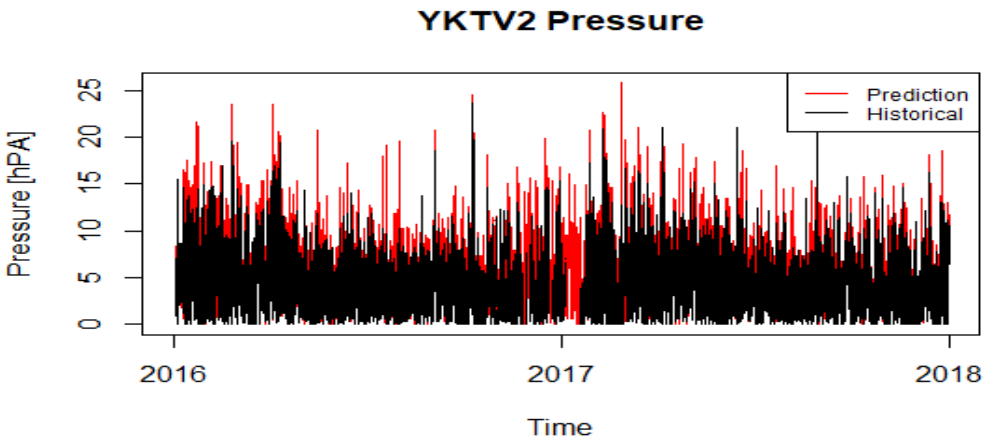
4 – Wave Temperature (WTMP)

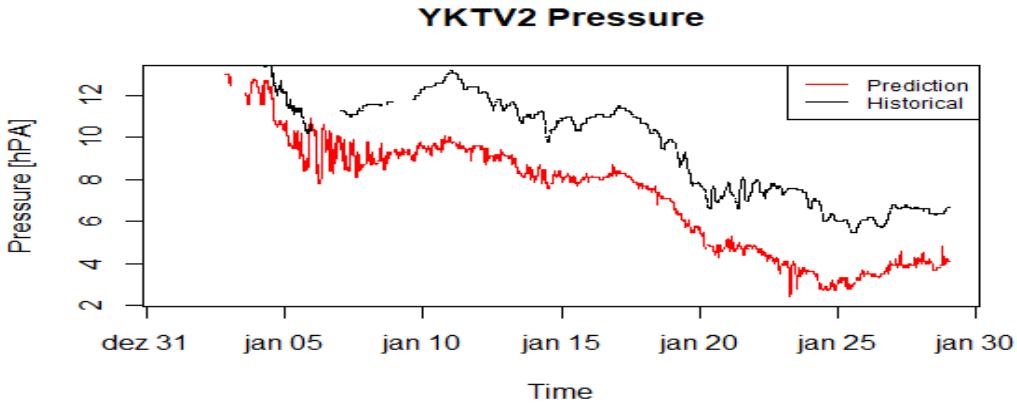


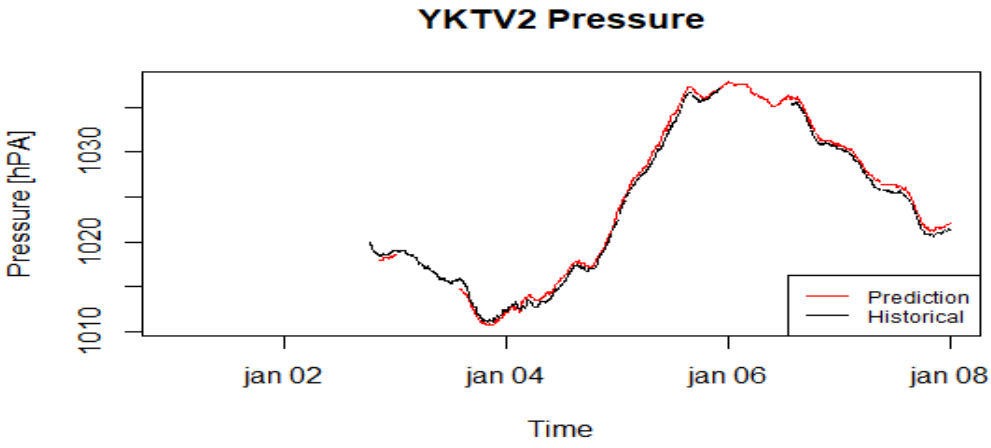




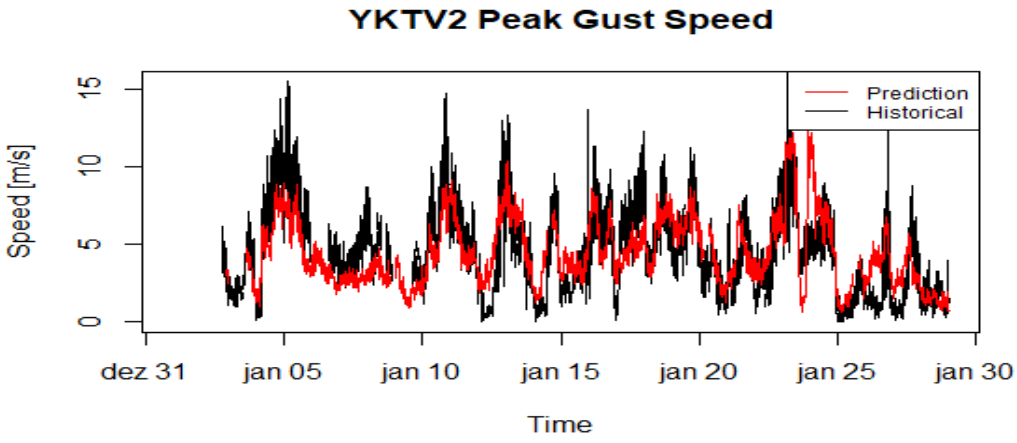
5 – Pressure (PRES)

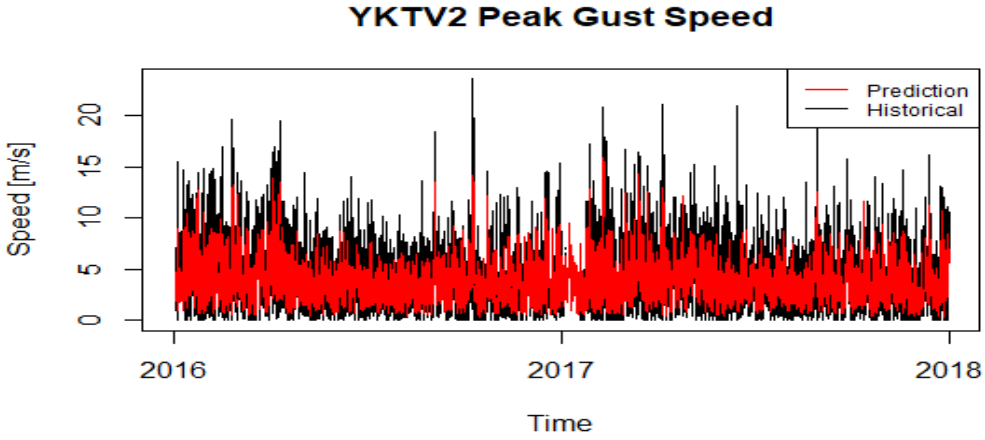


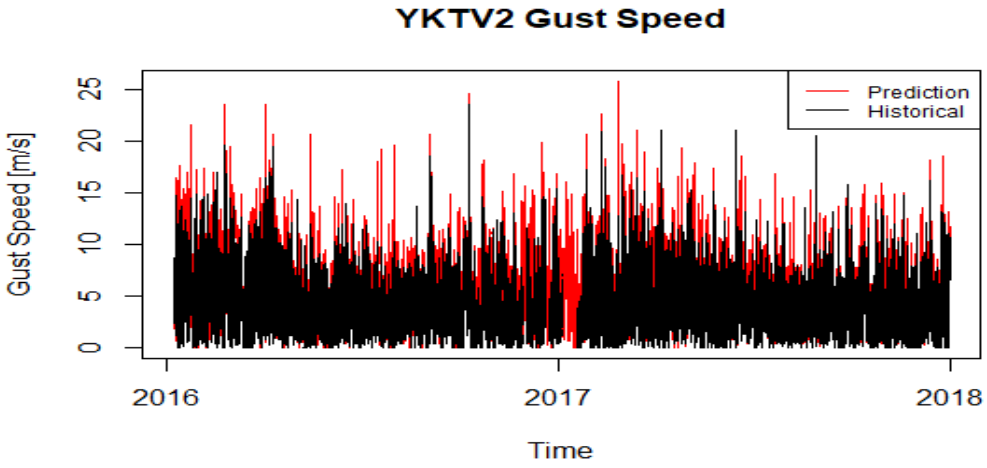


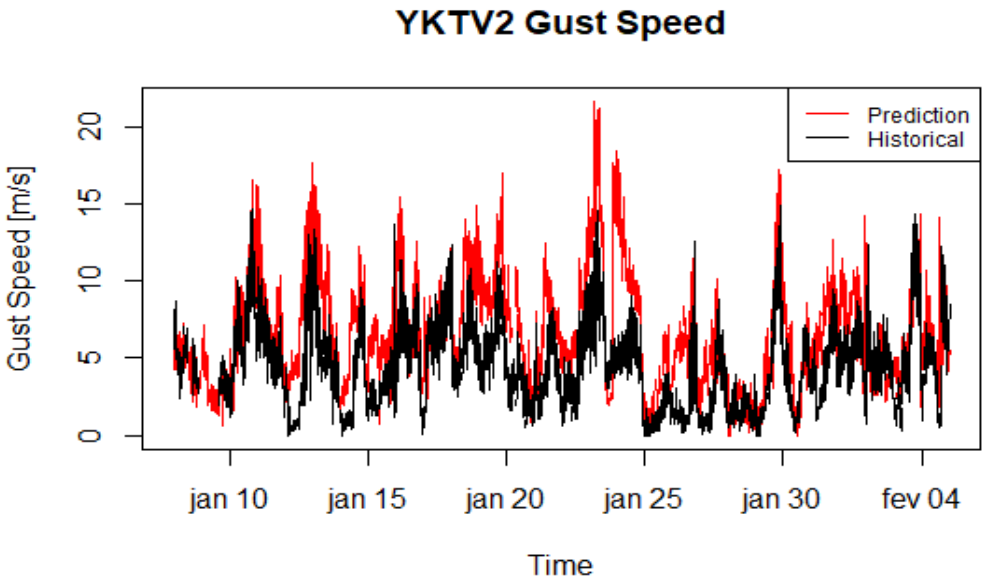


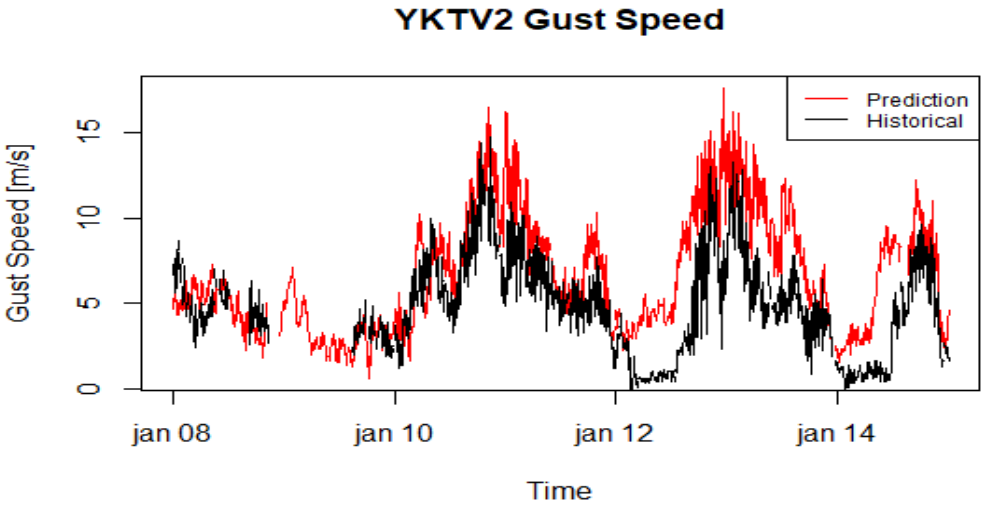
6 – Gust Speed (GST)








YKTV2 Gust Speed
Prediction
Historical
Gust Speed [m/s]
0
5
10
15
20
jan 10
jan 15
jan 20
jan 25
jan 30
fev 04
Time


YKTV2 Gust Speed
Prediction
Historical
Gust Speed [m/s]
0
5
10
15
jan 08
jan 10
jan 12
jan 14
Time

Appendix 2

Python Code for Preparing Datasets

This section presents the code that was used to prepare the data collected from the NDBC web page and to create the netCDF4 files.

```
#!/usr/bin/env python
# vim: set fileencoding=utf-8 fileformat=unix :
# -*- coding: utf-8 -*-
# vim: set ts=8 et sw=4 sts=4 sta :
import os
import gzip
from math import *
import numpy as np
import scipy as sp
import matplotlib as mpl
import matplotlib.pyplot as plt
import scipy.stats
import netCDF4 as netcdf
## Exemplo
# #YY MM DD hh mm WDIR WSPD GST WVHT DPD APD MWD PRES ATMP WTMP DEWP
VIS TIDE
# #yr mo dy hr mn degT m/s m/s m sec sec degT hPa degC degC degC
mi ft
# 2012 01 01 00 00 344 1.9 3.4 0.08 2.00 99.00 999 1020.2 10.7 8.9 999.0
99.0 99.00
## Significado dos campos: http://www.ndbc.noaa.gov/measdes.shtml
## Define limiting dates
# datetime(ano, mes, dia, hora, min, seg)
dstart = netcdf.datetime(2011, 1, 1, 0, 0, 0)
dfinal = netcdf.datetime(2017, 12, 31, 23, 59, 0)
## Station name
station = 'chyv2h'
#station = 'yktv2h'
#station = 'domv2h'
#station = 'kptv2h'
#station = 'mnpv2h'
#station = 'wdsv2h'
#station = 'ykrv2h'
#station = 'cryv2h'
#station = 'swpv2h'
#station = 'chbv2h'
kind = 'wind'
#kind = 'wave'
location = 'USA Sewells Point VA'
lon, lat = [-76, 0, 26], [36, 55, 35]
#lon, lat = [-76, 20, 33], [37, 15, 5]
#lon, lat = [-76, 25, 27], [36, 57, 44]
#lon, lat = [-75, 59, 18], [37, 9, 55]
#lon, lat = [-76, 18, 6], [36, 46, 41]
#lon, lat = [-76, 18, 56], [36, 58, 38]
```

```
#lon, lat = [-76, 20, 33], [37, 15, 5]
#lon, lat = [-76, 20, 18], [36, 53, 18]
#lon, lat = [-76, 19, 43], [36, 56, 34]
#lon, lat = [-76, 4, 59], [37, 1, 55]
## Original files
names = [\
station + '2016.txt.gz',\
station + '2017.txt.gz',\
station + '2011.txt.gz',\
station + '2012.txt.gz',\
station + '2013.txt.gz',\
station + '2014.txt.gz',\
station + '2015.txt.gz',\
]
## Invalid values
## Which values are invalid? Visual inspection...
# WDIR WSPD GST WVHT DPD APD MWD PRES ATMP WTMP DEWP VIS TIDE
# degT m/s m/s m sec sec degT hPa degC degC degC mi ft
# 999 99.0 99.0 0.12 4.10 99.00 999 1032.4 999.0 2.5 999.0 99.0 99.00
# 240 1.5 2.3 99.00 99.00 99.00 999 9999.0 6.2 999.0 999.0 99.0 99.00
invalid = {\
'WDIR' : 999,\
'WSPD' : 99.0,\
'GST' : 99.0,\
'WVHT' : 99.00,\
'DPD' : 99.00,\
'APD' : 99.00,\
'MWD' : 999,\
'PRES' : 9999.0,\
'ATMP' : 999.0,\
'WTMP' : 999.0,\
'DEWP' : 999.0,\
'VIS' : 99.0,\
'TIDE' : 99.00,\
}
##################################
##
## MAIN
##
##################################
def isfloat (s):
try:
float(s)
except ValueError:
return False
else:
return True
def allfloat (s):
for ss in s:
if not isfloat (ss):
return False
return True
def read_asciigz (name):
fid = gzip.open('/home/a40928/Desktop/Files/orig/' + name,'r') # Open file
      ## Treat header
      keys = None
      unit = None
      nvar = None
      for line in fid:
      if len(line) >= 2 and '#' == line[0]:
```

```
106.        ## Header line
107.        if None is keys:
108.        keys = line[1:].split()
109.        nvar = len(keys)
110.        elif None is unit:
111.        unit = line[1:].split()
112.        if len(unit) != len(keys):
113.        raise RuntimeError("Headers in %s have incorrect sizes?" %
114.        name)
115.        else:
116.        break
117.        #
118.        ## Prepare holder for data
119.        data = {}
120.        for k in keys:
121.        data[k] = []
122.        #
123.        ## Prepare holder for units
124.        unitaux = unit
125.        unit = {}
126.        for (k, u,) in zip(keys, unitaux):
127.        unit[k] = u
128.        #
129.        ## Read data
130.        fid.rewind()
131.        for line in fid:
132.        l = line.split()
133.        if len(l) >= nvar and allfloat(l[:nvar]):
134.        for n in xrange(nvar):
135.        data[keys[n]].append(l[n])
136.        #
137.        fid.close()
138.        #
139.        ## Convert data to numpy array
140.        for k in data.keys():
141.        if k in ('YY', 'MM', 'DD', 'hh', 'mm',):
142.        data[k] = np.array(data[k], np.object)
143.        else:
144.        data[k] = np.array(data[k], float)
145.        #
146.        return data, unit
147.        def join_data (new, old):
148.        knew = np.sort(np.array(new.keys()))
149.        kold = np.sort(np.array(old.keys()))
150.        if not (knew == kold).all():
151.        raise RuntimeError("knew and kold have different fields/keys!!!")
152.        #
153.        for k in kold:
154.        old[k] = np.concatenate([old[k], new[k]])
155.        return old
156.        ## RUN READING
157.        data = None
158.        for n in names:
159.        new, unit = read_asciigz (n)
160.        if None is data:
161.        data = new
162.        else:
163.        data = join_data (new, data)
164.        del(new)
165.        ################################
166.        ##
```

```
167.        ## Datetime stamps
168.        ##
169.        ################################
170.        d = []
171.        for n in xrange(data['YY'].size):
172.        d.append(netcdf.datetime(\
173.        year = int(data['YY'][n]),\
174.        month = int(data['MM'][n]),\
175.        day = int(data['DD'][n]),\
176.        hour = int(data['hh'][n]),\
177.        minute = int(data['mm'][n])))
178.        d = np.array(d) # array datetime stamp
179.        ## Define epoch (default is UNIX/POSIX)
180.        epoch = netcdf.datetime.utcfromtimestamp(0)
181.        tunitm = "minutes since %s" % epoch.isoformat()
182.        tunits = "seconds since %s" % epoch.isoformat()
183.        tm = netcdf.date2num(d, tunitm) # array minutes since epoch
184.        tm = tm.astype(int)
185.        ## Sort datetime and data
186.        n_sorted_tm = np.argsort(tm)
187.        tm = tm[n_sorted_tm]
188.        d = d[n_sorted_tm]
189.        for k in data.keys():
190.        data[k] = data[k][n_sorted_tm]
191.        ## Compute deltat
192.        deltat = tm[1:] - tm[:-1]
193.        dt = np.asscalar(sp.stats.mode(deltat).mode)
194.        ## Start-end dates for NetCDF
195.        tstartm = int(netcdf.date2num(dstart, tunitm))
196.        tfinalm = int(netcdf.date2num(dfinal, tunitm))
197.        tstartm -= tstartm % dt
198.        tfinalm -= tfinalm % dt
199.        dstart = netcdf.num2date(tstartm, tunitm)
200.        dfinal = netcdf.num2date(tfinalm, tunitm)
201.        print "\ndstart %s\ndfinal %s" % (dstart.isoformat(), dfinal.isoformat(
    ))
202.        ################################
203.        ##
204.        ## NetCDF
205.        ##
206.        ################################
207.        ## Create time arrays
208.        NT = (tfinalm - tstartm) / dt + 1
209.        TM = np.arange(NT) * dt + tstartm
210.        D = netcdf.num2date(TM, tunitm)
211.        ## Name of NetCDF file
212.        oname = station + "_" + dstart.strftime("%Y") \
213.        + "_" + dfinal.strftime("%Y") + ".nc"
214.        ## Create output directory
215.        odir = "netcdf_v3"
216.        if not os.path.exists(odir):
217.        os.makedirs(odir)
218.        ## Open NetCDF file
219.        ds = netcdf.Dataset(odir + "/" + oname, 'w')
220.        ## Global attributes
221.        ds.id = station + "_" + kind + "_data_" \
222.        + dstart.strftime("%Y") + "_" + dfinal.strftime("%Y")
223.        ds.summary = "Dataset " + location + ", station " + station \
224.        + ", " + kind
225.        ds.title = ds.summary
```

```
226.        ds.time_coverage_start = dstart.isoformat()
227.        ds.time_coverage_final = dfinal.isoformat()
228.        ds.station = station
229.        def dms2dd (dms):
230.        dd = dms[2]/3600. + dms[1]/60. + abs(dms[0])
231.        return -dd if dms[0] < 0 else dd
232.        ds.longitude = dms2dd(lon)
233.        ds.latitude = dms2dd(lat)
234.        ## Create dimensions
235.        ds.createDimension('time', TM.size)
236.        ## Create variables
237.        ds.createVariable('time', 'i4', ('time',))
238.        ds.variables['time'].long_name = "Time"
239.        ds.variables['time'].units = tunitm
240.        ds.variables['time'].time_origin = netcdf.num2date(min(TM), tunitm).iso
    format()
241.        ds.variables['time'][:] = TM[:]
242.        ## Indexes
243.        btin = (tm >= min(TM)) & (tm <= max(TM))
244.        bt = np.in1d(tm, TM) # bool i with index of tm[i] == TM[j]
245.        if np.all(bt[btin]):
246.        print "Datatime stamps tm matches with TM..."
247.        else:
248.        print "Datatime stamps in tm out-of-phase with TM!"
249.        # raise RuntimeError("Need to review this! Aborting...")
250.        BT = np.in1d(TM, tm) # bool j with index of tm[i] == TM[j]
251.        ## Write time-dependent variables
252.        k = 'WSPD'
253.        print "writing variable %s" % k
254.        ds.createVariable(k, 'f4', ('time',))
255.        ds.variables[k].long_name = "Wind speed"
256.        ds.variables[k].description = "Mean wind speed"
257.        ds.variables[k].units = unit[k]
258.        U = ds.variables[k][:]
259.        U[BT] = data[k][bt]
260.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
261.        U[i] = U.fill_value
262.        U.mask[i] = True
263.        ds.variables[k][:] = U[:]
264.        del(U)
265.        k = 'WDIR'
266.        print "writing variable %s" % k
267.        ds.createVariable(k, 'f4', ('time',))
268.        ds.variables[k].long_name = "Wind direction"
269.        ds.variables[k].description = "Mean wind direction"
270.        ds.variables[k].units = unit[k]
271.        U = ds.variables[k][:]
272.        U[BT] = data[k][bt]
273.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
274.        U[i] = U.fill_value
275.        U.mask[i] = True
276.        ds.variables[k][:] = U[:]
277.        del(U)
278.        k = 'GST'
279.        print "writing variable %s" % k
280.        ds.createVariable(k, 'f4', ('time',))
281.        ds.variables[k].long_name = "Wind gust"
282.        ds.variables[k].description = "Maximum wind speed in integration time"

283.        ds.variables[k].units = unit[k]
284.        U = ds.variables[k][:]
```

```
285.        U[BT] = data[k][bt]
286.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
287.        U[i] = U.fill_value
288.        U.mask[i] = True
289.        ds.variables[k][:] = U[:]
290.        del(U)
291.        k = 'WVHT'
292.        print "writing variable %s" % k
293.        ds.createVariable(k, 'f4', ('time',))
294.        ds.variables[k].long_name = "Wave height"
295.        ds.variables[k].description = "Significant wave height is calculated as
     the
296.        average of the highest one-
    third of all of the wave heights during a 20-minute
297.        sampling period"
298.        ds.variables[k].units = unit[k]
299.        U = ds.variables[k][:]
300.        U[BT] = data[k][bt]
301.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
302.        U[i] = U.fill_value
303.        U.mask[i] = True
304.        ds.variables[k][:] = U[:]
305.        del(U)
306.        k = 'DPD'
307.        print "writing variable %s" % k
308.        ds.createVariable(k, 'f4', ('time',))
309.        ds.variables[k].long_name = "Dominant wave period"
310.        ds.variables[k].description = "Dominant wave period is the period with
    the
311.        maximum wave energy"
312.        ds.variables[k].units = unit[k]
313.        U = ds.variables[k][:]
314.        U[BT] = data[k][bt]
315.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
316.        U[i] = U.fill_value
317.        U.mask[i] = True
318.        ds.variables[k][:] = U[:]
319.        del(U)
320.        k = 'APD'
321.        print "writing variable %s" % k
322.        ds.createVariable(k, 'f4', ('time',))
323.        ds.variables[k].long_name = "Average wave period"
324.        ds.variables[k].description = "Average wave period of all waves during
    a 20-
325.        minute period"
326.        ds.variables[k].units = unit[k]
327.        U = ds.variables[k][:]
328.        U[BT] = data[k][bt]
329.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
330.        U[i] = U.fill_value
331.        U.mask[i] = True
332.        ds.variables[k][:] = U[:]
333.        del(U)
334.        k = 'MWD'
335.        print "writing variable %s" % k
336.        ds.createVariable(k, 'f4', ('time',))
337.        ds.variables[k].long_name = "Direction of dominant wave"
338.        ds.variables[k].description = "The direction from which the waves at th
    e
339.        dominant period (DPD) are coming"
```

```
340.        ds.variables[k].units = unit[k]
341.        U = ds.variables[k][:]
342.        U[BT] = data[k][bt]
343.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
344.        U[i] = U.fill_value#!/usr/bin/env Rscript
345.        #
346.        rm(list = ls())
347.        save(erro, prev, h, p, file="WDIR_ykt_mnp.RData")
348.        Est2=Sys.time()
349.        TotalEst=Est2-Est1
350.        U.mask[i] = True
351.        ds.variables[k][:] = U[:]
352.        del(U)
353.        k = 'PRES'
354.        print "writing variable %s" % k
355.        ds.createVariable(k, 'f4', ('time',))
356.        ds.variables[k].long_name = "Pressure"
357.        ds.variables[k].description = "Sea level pressure"
358.        ds.variables[k].units = unit[k]
359.        U = ds.variables[k][:]
360.        U[BT] = data[k][bt]
361.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
362.        U[i] = U.fill_value
363.        U.mask[i] = True
364.        ds.variables[k][:] = U[:]
365.        del(U)
366.        k = 'ATMP'
367.        print "writing variable %s" % k
368.        ds.createVariable(k, 'f4', ('time',))
369.        ds.variables[k].long_name = "Tair"
370.        ds.variables[k].description = "Air temperature"
371.        ds.variables[k].units = unit[k]
372.        U = ds.variables[k][:]
373.        U[BT] = data[k][bt]
374.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
375.        U[i] = U.fill_value
376.        U.mask[i] = True
377.        ds.variables[k][:] = U[:]
378.        del(U)
379.        k = 'WTMP'
380.        print "writing variable %s" % k
381.        ds.createVariable(k, 'f4', ('time',))
382.        ds.variables[k].long_name = "Tsea"
383.        ds.variables[k].description = "Sea surface temperature, for buoys the d
    epth is
384.        referenced to the hull's waterline, for fixed platforms it varies with
    tide, but
385.        is referenced to, or near Mean Lower Low Water"
386.        ds.variables[k].units = unit[k]
387.        U = ds.variables[k][:]
388.        U[BT] = data[k][bt]
389.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
390.        U[i] = U.fill_value
391.        U.mask[i] = True
392.        ds.variables[k][:] = U[:]
393.        del(U)
394.        k = 'DEWP'
395.        print "writing variable %s" % k
396.        ds.createVariable(k, 'f4', ('time',))
397.        ds.variables[k].long_name = "Tdew"
398.        ds.variables[k].description = "Dewpoint temperature"
```

```
399.        ds.variables[k].units = unit[k]
400.        U = ds.variables[k][:]
401.        U[BT] = data[k][bt]
402.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
403.        U[i] = U.fill_value
404.        U.mask[i] = True
405.        ds.variables[k][:] = U[:]
406.        del(U)
407.        k = 'VIS'
408.        print "writing variable %s" % k
409.        ds.createVariable(k, 'f4', ('time',))
410.        ds.variables[k].long_name = "Station visibility"
411.        ds.variables[k].description = "Station visibility (nautical miles), not
    e that
412.        buoy stations are limited to reports from 0 to 1.6 nmi"
413.        ds.variables[k].units = unit[k]
414.        U = ds.variables[k][:]
415.        U[BT] = data[k][bt]
416.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
417.        U[i] = U.fill_value
418.        U.mask[i] = True
419.        ds.variables[k][:] = U[:]
420.        del(U)
421.        k = 'TIDE'
422.        print "writing variable %s" % k
423.        ds.createVariable(k, 'f4', ('time',))
424.        ds.variables[k].long_name = "Water level height"
425.        ds.variables[k].description = "The water level in feet above or below M
    ean Lower
426.        Low Water"
427.        ds.variables[k].units = unit[k]
428.        U = ds.variables[k][:]
429.        U[BT] = data[k][bt]
430.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
431.        U[i] = U.fill_value
432.        U.mask[i] = True
433.        ds.variables[k][:] = U[:]
434.        del(U)
435.        ds.close()
436.        ##############################
437.        '''''
438.        variavel='WDIR'
439.        g1=d['WDIR']
440.        g2=g1.size
441.        a=d[variavel]
442.        a=variavel
443.        g=a[~np.isnan(a)]
444.        kk=g.size/g2
445.        ##################################3
446.        ## Sort datetime and data
447.        n_sorted_tm = np.argsort(tm)
448.        tm = tm[n_sorted_tm]
449.        d = d[n_sorted_tm]
450.        for k in data.keys():
451.        data[k] = data[k][n_sorted_tm]
452.        ## Compute deltat
453.        deltat = tm[1:] - tm[:-1]
454.        dt = np.asscalar(sp.stats.mode(deltat).mode)
455.        ## Start-end dates for NetCDF
456.        tstartm = int(netcdf.date2num(dstart, tunitm))
```

```
457.        tfinalm = int(netcdf.date2num(dfinal, tunitm))
458.        tstartm -= tstartm % dt
459.        tfinalm -= tfinalm % dt
460.        dstart = netcdf.num2date(tstartm, tunitm)
461.        dfinal = netcdf.num2date(tfinalm, tunitm)
462.        print "\ndstart %s\ndfinal %s" % (dstart.isoformat(), dfinal.isoformat(
    ))
463.        ################################
464.        ##
465.        ## NetCDF
466.        ##
467.        ################################
468.        ## Create time arrays
469.        NT = (tfinalm - tstartm) / dt + 1
470.        TM = np.arange(NT) * dt + tstartm
471.        D = netcdf.num2date(TM, tunitm)
472.        ## Name of NetCDF file
473.        oname = station + "_" + dstart.strftime("%Y") \
474.        + "_" + dfinal.strftime("%Y") + ".nc"
475.        ## Create output directory
476.        odir = "netcdf_v3"
477.        if not os.path.exists(odir):
478.        os.makedirs(odir)
479.        ## Open NetCDF file
480.        ds = netcdf.Dataset(odir + "/" + oname, 'w')
481.        ## Global attributes
482.        ds.id = station + "_" + kind + "_data_" \
483.        + dstart.strftime("%Y") + "_" + dfinal.strftime("%Y")
484.        ds.summary = "Dataset " + location + ", station " + station \
485.        + ", " + kind
486.        ds.title = ds.summary
487.        ds.time_coverage_start = dstart.isoformat()
488.        ds.time_coverage_final = dfinal.isoformat()
489.        ds.station = station
490.        def dms2dd (dms):
491.        dd = dms[2]/3600. + dms[1]/60. + abs(dms[0])
492.        return -dd if dms[0] < 0 else dd
493.        ds.longitude = dms2dd(lon)
494.        ds.latitude = dms2dd(lat)
495.        ## Create dimensions
496.        ds.createDimension('time', TM.size)
497.        ## Create variables
498.        ds.createVariable('time', 'i4', ('time',))
499.        ds.variables['time'].long_name = "Time"
500.        ds.variables['time'].units = tunitm
501.        ds.variables['time'].time_origin = netcdf.num2date(min(TM), tunitm).iso
    format()
502.        ds.variables['time'][:] = TM[:]
503.        ## Indexes
504.        btin = (tm >= min(TM)) & (tm <= max(TM))
505.        bt = np.in1d(tm, TM) # bool i with index of tm[i] == TM[j]
506.        if np.all(bt[btin]):
507.        print "Datatime stamps tm matches with TM..."
508.        else:
509.        print "Datatime stamps in tm out-of-phase with TM!"
510.        # raise RuntimeError("Need to review this! Aborting...")
511.        BT = np.in1d(TM, tm) # bool j with index of tm[i] == TM[j]
512.        ## Write time-dependent variables
513.        k = 'WSPD'
514.        print "writing variable %s" % k
515.        ds.createVariable(k, 'f4', ('time',))
```

```
516.        ds.variables[k].long_name = "Wind speed"
517.        ds.variables[k].description = "Mean wind speed"
518.        ds.variables[k].units = unit[k]
519.        U = ds.variables[k][:]
520.        U[BT] = data[k][bt]
521.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
522.        U[i] = U.fill_value
523.        U.mask[i] = True
524.        ds.variables[k][:] = U[:]
525.        del(U)
526.        k = 'WDIR'
527.        print "writing variable %s" % k
528.        ds.createVariable(k, 'f4', ('time',))
529.        ds.variables[k].long_name = "Wind direction"
530.        ds.variables[k].description = "Mean wind direction"
531.        ds.variables[k].units = unit[k]
532.        U = ds.variables[k][:]
533.        U[BT] = data[k][bt]
534.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
535.        U[i] = U.fill_value
536.        U.mask[i] = True
537.        ds.variables[k][:] = U[:]
538.        del(U)
539.        k = 'GST'
540.        print "writing variable %s" % k
541.        ds.createVariable(k, 'f4', ('time',))
542.        ds.variables[k].long_name = "Wind gust"
543.        ds.variables[k].description = "Maximum wind speed in integration time"
544.        ds.variables[k].units = unit[k]
545.        U = ds.variables[k][:]
546.        U[BT] = data[k][bt]
547.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
548.        U[i] = U.fill_value
549.        U.mask[i] = True
550.        ds.variables[k][:] = U[:]
551.        del(U)
552.        k = 'WVHT'
553.        print "writing variable %s" % k
554.        ds.createVariable(k, 'f4', ('time',))
555.        ds.variables[k].long_name = "Wave height"
556.        ds.variables[k].description = "Significant wave height is calculated as
     the
557.        average of the highest one-
    third of all of the wave heights during a 20-minute
558.        sampling period"
559.        ds.variables[k].units = unit[k]
560.        U = ds.variables[k][:]
561.        U[BT] = data[k][bt]
562.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
563.        U[i] = U.fill_value
564.        U.mask[i] = True
565.        ds.variables[k][:] = U[:]
566.        del(U)
567.        k = 'DPD'
568.        print "writing variable %s" % k
569.        ds.createVariable(k, 'f4', ('time',))
570.        ds.variables[k].long_name = "Dominant wave period"
571.        ds.variables[k].description = "Dominant wave period is the period with
    the
572.        maximum wave energy"
```

```
573.        ds.variables[k].units = unit[k]
574.        U = ds.variables[k][:]
575.        U[BT] = data[k][bt]
576.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
577.        U[i] = U.fill_value
578.        U.mask[i] = True
579.        ds.variables[k][:] = U[:]
580.        del(U)
581.        k = 'APD'
582.        print "writing variable %s" % k
583.        ds.createVariable(k, 'f4', ('time',))
584.        ds.variables[k].long_name = "Average wave period"
585.        ds.variables[k].description = "Average wave period of all waves during
    a 20-
586.        minute period"
587.        ds.variables[k].units = unit[k]
588.        U = ds.variables[k][:]
589.        U[BT] = data[k][bt]
590.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
591.        U[i] = U.fill_value
592.        U.mask[i] = True
593.        ds.variables[k][:] = U[:]
594.        del(U)
595.        k = 'MWD'
596.        print "writing variable %s" % k
597.        ds.createVariable(k, 'f4', ('time',))
598.        ds.variables[k].long_name = "Direction of dominant wave"
599.        ds.variables[k].description = "The direction from which the waves at th
    e
600.        dominant period (DPD) are coming"
601.        ds.variables[k].units = unit[k]
602.        U = ds.variables[k][:]
603.        U[BT] = data[k][bt]
604.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
605.        U[i] = U.fill_value
606.        U.mask[i] = True
607.        ds.variables[k][:] = U[:]
608.        del(U)
609.        k = 'PRES'
610.        print "writing variable %s" % k
611.        ds.createVariable(k, 'f4', ('time',))
612.        ds.variables[k].long_name = "Pressure"
613.        ds.variables[k].description = "Sea level pressure"
614.        ds.variables[k].units = unit[k]
615.        U = ds.variables[k][:]
616.        U[BT] = data[k][bt]
617.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
618.        U[i] = U.fill_value
619.        U.mask[i] = True
620.        ds.variables[k][:] = U[:]
621.        del(U)
622.        k = 'ATMP'
623.        print "writing variable %s" % k
624.        ds.createVariable(k, 'f4', ('time',))
625.        ds.variables[k].long_name = "Tair"
626.        ds.variables[k].description = "Air temperature"
627.        ds.variables[k].units = unit[k]
628.        U = ds.variables[k][:]
629.        U[BT] = data[k][bt]
630.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
631.        U[i] = U.fill_value
```

```
632.        U.mask[i] = True
633.        ds.variables[k][:] = U[:]
634.        del(U)
635.        k = 'WTMP'
636.        print "writing variable %s" % k
637.        ds.createVariable(k, 'f4', ('time',))
638.        ds.variables[k].long_name = "Tsea"
639.        ds.variables[k].description = "Sea surface temperature, for buoys the d
    epth is
640.        referenced to the hull's waterline, for fixed platforms it varies with
    tide, but
641.        is referenced to, or near Mean Lower Low Water"
642.        ds.variables[k].units = unit[k]
643.        U = ds.variables[k][:]
644.        U[BT] = data[k][bt]
645.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
646.        U[i] = U.fill_value
647.        U.mask[i] = True
648.        ds.variables[k][:] = U[:]
649.        del(U)
650.        k = 'DEWP'
651.        print "writing variable %s" % k
652.        ds.createVariable(k, 'f4', ('time',))
653.        ds.variables[k].long_name = "Tdew"
654.        ds.variables[k].description = "Dewpoint temperature"
655.        ds.variables[k].units = unit[k]
656.        U = ds.variables[k][:]
657.        U[BT] = data[k][bt]
658.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
659.        U[i] = U.fill_value
660.        U.mask[i] = True
661.        ds.variables[k][:] = U[:]
662.        del(U)
663.        k = 'VIS'
664.        print "writing variable %s" % k
665.        ds.createVariable(k, 'f4', ('time',))
666.        ds.variables[k].long_name = "Station visibility"
667.        ds.variables[k].description = "Station visibility (nautical miles), not
    e that
668.        buoy stations are limited to reports from 0 to 1.6 nmi"
669.        ds.variables[k].units = unit[k]
670.        U = ds.variables[k][:]
671.        U[BT] = data[k][bt]
672.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
673.        U[i] = U.fill_value
674.        U.mask[i] = True
675.        ds.variables[k][:] = U[:]
676.        del(U)
677.        k = 'TIDE'
678.        print "writing variable %s" % k
679.        ds.createVariable(k, 'f4', ('time',))
680.        ds.variables[k].long_name = "Water level height"
681.        ds.variables[k].description = "The water level in feet above or below M
    ean Lower
682.        Low Water"
683.        ds.variables[k].units = unit[k]
684.        U = ds.variables[k][:]
685.        U[BT] = data[k][bt]
686.        i = (-U.mask) & (U.data >= invalid[k] - 1E-8)
687.        U[i] = U.fill_value
```

```
688.        U.mask[i] = True
689.        ds.variables[k][:] = U[:]
690.        del(U)
691.        ds.close()
692.        ##############################
693.        '''
694.        variavel='WDIR'
695.        g1=d['WDIR']
696.        g2=g1.size
697.        a=d[variavel]
698.        a=variavel
699.        g=a[~np.isnan(a)]
700.        kk=g.size/g2
701.        ###################################3
```

Appendix 3

R Code for Post-processing Forecasts

This section presents the code that was used to forecast the time series with single variables in this study. To expand its use for multiple variables, the main function in the code needs to be adapted to receive all the variables required.

```
#!/usr/bin/env Rscript
#
rm(list = ls())
## AnEn nomenclature
## input:
## p, the predictor timeseries, where both
## training and forecast periods are known,
## which will be used to compute the analogs
## h, the historical timeseries, from which the
## forecasts will be produced throuhg the indexes
## of the best analogs
## output:
## f, the forecasted/hindcasted timeseries
## method:
## 1. from predictor timeseries get the analogs through a metric
## 2. choose the best analogs and get their indexes
## 3. apply the indexes on the historical timeseries and
## get several possible values for the forecast
## 4. produce the forecast from the distribution of possible values
library(ncdf4)
library(parallel)
Est1=Sys.time()
namep = 'mnpv2h_2011_2017.nc' # file name of the predictor data
nameh = 'yktv2h_2011_2017.nc' # file name of the historical data
d0 <- as.POSIXct("2016-01-
01 00:00:00", tz="UTC") # start of the forecast period
v = 'D' # variable to predict
Na = 25 # size of analogs
M = 20 # half-window size of the window to compute analogs (equiv 1 hour)
## For netcdf and R, check
## http://geog.uoregon.edu/bartlein/courses/geog490/week04-netCDF.html
ds <- nc_open(namep, write=FALSE)
#print(ds)
p <- list()
p$tunit <- ncatt_get(ds, "time", "units")$value
p$epoch <- sub('T', ' ', sapply(strsplit(p$tunit, " "), tail, 1))
p$epoch <- as.POSIXct(p$epoch, tz="UTC")
p$N <- length(ncvar_get(ds, "time"))
p$t <- ncvar_get(ds, "time") # ncvar_get(ds, "time")[1:p$N]
p$stamp <- as.POSIXct(p$t * 60, origin=p$epoch, tz="UTC")
p$U <- ncvar_get(ds, "WSPD")
if (ncatt_get(ds, "WSPD", "_FillValue")$hasatt) {
```

```
p$U[p$U == ncatt_get(ds, "WSPD", "_FillValue")$value] <- NA
} else {
p$U[p$U > 9e+36] <- NA # missing value
}
p$D <- ncvar_get(ds, "WDIR")
if (ncatt_get(ds, "WDIR", "_FillValue")$hasatt) {
p$D[p$D == ncatt_get(ds, "WDIR", "_FillValue")$value] <- NA
} else {
p$D[p$D > 9e+36] <- NA # missing value
}
p$T <- ncvar_get(ds, "ATMP")
if (ncatt_get(ds, "ATMP", "_FillValue")$hasatt) {
p$T[p$T == ncatt_get(ds, "ATMP", "_FillValue")$value] <- NA
} else {
p$T[p$T > 9e+36] <- NA # missing value
}
nc_close(ds)
p$t0 <- (as.numeric(as.POSIXct(d0, tz="UTC")) - as.numeric(p$epoch)) / 60
p$n0 <- match(d0, p$stamp) # starting index for forecast period
ds <- nc_open(nameh, write=FALSE)
#print(ds)
h <- list()
h$tunit <- ncatt_get(ds, "time", "units")$value
h$epoch <- sub('T', ' ', sapply(strsplit(h$tunit, " "), tail, 1))
h$epoch <- as.POSIXct(h$epoch, tz="UTC")
h$N <- length(ncvar_get(ds, "time"))
h$t <- ncvar_get(ds, "time") # ncvar_get(ds, "time")[1:h$N]
h$stamp <- as.POSIXct(h$t * 60, origin=h$epoch, tz="UTC")
h$U <- ncvar_get(ds, "WSPD")
if (ncatt_get(ds, "WSPD", "_FillValue")$hasatt) {
h$U[h$U == ncatt_get(ds, "WSPD", "_FillValue")$value] <- NA
} else {
h$U[h$U > 9e+36] <- NA # missing value
}
h$D <- ncvar_get(ds, "WDIR")
if (ncatt_get(ds, "WDIR", "_FillValue")$hasatt) {
h$D[h$D == ncatt_get(ds, "WDIR", "_FillValue")$value] <- NA
} else {
h$D[h$D > 9e+36] <- NA # missing value
}
h$T <- ncvar_get(ds, "ATMP")
if (ncatt_get(ds, "ATMP", "_FillValue")$hasatt) {
h$T[h$T == ncatt_get(ds, "ATMP", "_FillValue")$value] <- NA
} else {
h$T[h$T > 9e+36] <- NA # missing value
}
nc_close(ds)
h$t0 <- (as.numeric(as.POSIXct(d0, tz="UTC")) - as.numeric(h$epoch)) / 60
h$n0 <- match(d0, h$stamp) # starting index for forecast period
print("Are the p and h arrays coincident in terms of time/indexes?")
if (all.equal(p$t, h$t)) {
print("True")
} else {
print("No! Code not prepared, correct this!")
}
## DEBUG START
#p$n0 <- match(d0, p$stamp) + 700 # debug
#h$n0 <- match(d0, h$stamp) + 700 # debug
      #p$N <- p$n0 + 100 # debug
      #h$N <- h$n0 + 100 # debug
      ## DEBUG END
```

```
103.        ## Algorithm
104.        n = which(p$t < p$t0)
105.        stdp = sd(p[[v]][n], na.rm=TRUE)
106.        #stdp = sqrt(mean(p[[v]][n]**2, na.rm=TRUE) - mean(p[[v]][n], na.rm=TRU
    E)**2)
107.        ## slow
108.        # Y <- p[[v]][1:(p$n0-M+1)]
109.        # for (i in 2:(2*M+1)) {
110.        # Y <- rbind(Y, p[[v]][i:(p$n0-M+i)])
111.        # }
112.        ## fast using array
113.        # Y = array(data=NA, dim=c(2*M+1, p$n0-M+1))
114.        # for (i in 1:(2*M+1)) {
115.        # Y[i,1:(p$n0-M+1)] <- p[[v]][i:(p$n0-M+i)]
116.        # }
117.        ## fastest
118.        Y <- list()
119.        for (i in 1:(2*M+1)) {
120.        Y[[i]] <- p[[v]][i:(p$n0-M+i-2)]
121.        }
122.        Y = do.call(rbind, Y)
123.        Ynan <- colSums(is.na(Y)) >= .5*M # mark periods whose 25% are nan's
124.        main <- function(n){
125.        mi <- max(n-M, 1)
126.        me <- min(n+M, p$N)
127.        y <- array(data=NA, dim=c(2*M+1))
128.        y[(M+mi-n+1):(M+me-n+1)] <- p[[v]][mi:me]
129.        if (sum(is.na(y)) < .5*M) { # guarantees window has 50% of valid values

130.        #A <- sweep(Y, 1, y, "*") # covariance
131.        #A <- abs(sweep(Y, 1, y, "-")) # abs diff
132.        A <- (sweep(Y, 1, y, "-"))**2 # quad error
133.        ## Method 1 - treat nan's as nan's...
134.        # Metric <- sqrt(colSums (A, na.rm=FALSE)) / stdp
135.        # Metric <- colSums (A, na.rm=FALSE)
136.        # Metric[is.na(Metric)] <- Inf
137.        ## Method 2 - use some periods with nan's
138.        # Metric <- sqrt(colSums (A, na.rm=TRUE)) / stdp
139.        Metric <- colSums (A, na.rm=TRUE)
140.        Metric[Ynan] <- Inf
141.        ## Get index for analogs
142.        na <- order(Metric, decreasing=FALSE)[1:Na] + M
143.        ## Store analogs
144.        #f[1:Na,n-p$n0+1] <<- h[[v]][na] # ordering f(best_analogs, time)
145.        result <- h[[v]][na]
146.        } else {
147.        # print("Too many NaN's in the predictor window... skipping forecast")

148.        # print(sprintf("forecast %d of %d", n-p$n0+1, p$N-p$n0))
149.        #f[1:Na,n-
    p$n0+1] <<- array(data=NA, dim=c(Na)) # ordering f(best_analogs,
150.        time)
151.        result <- array(data=NA, dim=c(Na))
152.        }
153.        return(result)
154.        }
155.        ################################
156.        ##
157.        ## LOOP START
158.        ##
```

```
159.        ################################
160.        ## Measure time of main loop
161.        cstart <- Sys.time()
162.        ## method 1 - for loop
163.        # f <- array(data=NA, dim=c(Na, h$N-h$n0+1)) # init forecast array
164.        # for (n in p$n0:p$N) {
165.        # f[1:Na,n-p$n0+1] <- main(n)
166.        # }
167.        ## method 2 - sapply
168.        # #f <- sapply(p$n0:p$N, main) # slower
169.        # f <- array(data=NA, dim=c(Na, h$N-h$n0+1)) # init forecast array
170.        # f[1:Na,1:(p$N-p$n0+1)] <- sapply(p$n0:p$N, main)
171.        ## method 3 - parSapply
172.        ncores <- detectCores()
173.        cluster <- makeCluster(ncores-1) # leave 1 core out for the system
174.        print(sprintf("Parallel parSapply loop using %d cores", ncores-1))
175.        #cluster <- makeCluster(ncores, type="FORK") # takes time
176.        clusterExport(cluster, c("Y", "Ynan", "M", "p", "h", "v", "Na")) # expo
    rt
177.        shared variables
178.        #f <- parSapply(cluster, p$n0:p$N, main) # slower
179.        f <- array(data=NA, dim=c(Na, h$N-h$n0+1)) # init forecast array
180.        f[1:Na,1:(p$N-p$n0+1)] <- parSapply(cluster, p$n0:p$N, main)
181.        stopCluster(cluster)
182.        ## Print time elapsed
183.        cfinal <- Sys.time()
184.        ctime = cfinal - cstart
185.        print(sprintf("%d forecasts, elapsed time %g s, time per forecast %g s"
    ,
186.        p$N-p$n0+1, ctime, ctime/(p$N-p$n0+1.0)))
187.        ################################
188.        ##
189.        ## LOOP END
190.        ##
191.        ################################
192.        prev <- colMeans(f, na.rm=TRUE)
193.        erro = abs(prev - h[[v]][h$n0:h$N]) / abs(mean(h[[v]][h$n0:h$N], na.rm=
    TRUE))
194.        plot(erro)
195.        save(erro, prev, h, p, file="WDIR_ykt_mnp.RData")
196.        Est2=Sys.time()
197.        TotalEst=Est2-Est1
```