A Feature Engineering Approach

GEFCom 2012 – Wind Power Forecast

Lucas Silva
Software Engineer and Data Scientist
DTI Sistemas
Belo Horizonte, Brazil
lucas.eustaquio@gmail.com

***Abstract*— this paper describes detailed information about team Leustagos approach to the wind power forecast track of GEFCom 2012. The task was to predict 48-hour ahead hourly power generation at 7 wind farms. The problem was addressed by extracting time and weather related features. These features were used to build gradient boosted decision trees and linear regression models. This approach granted us 1st place both in the public and private leaderboards.**

Keywords— GEFCom, feature engineering, gradient boosted decision trees, linear regression, machine learning, time series

I. INTRODUCTION

The "GEFCom 2012 – Wind Forecast" competition [1] posed the challenge of forecasting the hourly wind power generation for 7 wind farms. A dataset containing historical power measurements for these wind farms, as well as meteorological forecasts of wind components at the level of those wind farms were provided. A detailed description of the dataset can be found on [2].

A big challenge in this task was dealing with the time-series nature of the dataset. Since no time-series specific model was used, it had to be kept constantly in mind.

Another difficult part was to create efficient features. Feature creation is one of the most important steps in solving a supervised learning problem. In order to do so, many features where derived from the dataset. The feature creation had two main guiding principles:

1. Model the wind power generation equation, based on constants, wind strength, direction and air density (surrogated). These features represent the windmill behavior.
2. Discover the the relationship between wind power generation of T and T±n. The objective here was to reflect the inertia of windmills as well as the time-series nature of the dataset.

In this paper, is described in detail used features and models. Section II briefly describes the dataset and how the data was split for model training. Section III outlines the algorithms used. Section IV shows the performance measurement. Section V is about the chosen approach to this task as it describes the training methods and features. Section VI presents results, section VII discusses some aspects of the solution, and section VIII presents some conclusions.

II. DATASET

The provided dataset [2] consisted of the following files:

1. "train": contained normalized hourly wind power generation for each farm. The period between 2009/7/1 and 2010/12/31 was for model identification and training period (~~full data available~~), while the remainder of the dataset, that is, from 2011/1/1 to 2012/6/28, was there for the leaderboard evaluation. It had many 48 hour missing periods at 36 hours intervals, that were the target of prediction.
2. "windforecasts_wf1", …, "windforecasts_wf7": Wind forecasts (strength and direction) for the 7 wind farms. Forecasts are issued every 12 hours with a forecast horizon of 48 hours and an hourly temporal resolution. The forecasts for the missing training periods were given such that only the forecasts available prior to the start of each period were present, mimicking operation conditions.

As can be seen, the data provided consisted only of a series of hourly wind power generation for each farm, and the forecasted wind strength and direction, issued every 12 hours. It was provided in this way to mimic the real operation conditions and didn't have all desired information, like forecasted temperature and air density that, maybe, could improve accuracy. So to make up for this missing information, many surrogate features were created, as it will be explained later. Also, an important step was to build a consistent validation set using only the training period. This validation set allowed to build a model that didn't overfit the training data.

A. *Validation Set*

The first step of the used approach was to build a consistent validation set. It was built trying to replicate the same structure found in the evaluation period between filled and missing spots. In order to replicate it, the training and evaluation periods were each split in 312 sets of 84 consecutive hours (36+48) and those sets were used to define the 5-fold validation splitting. Each training/validation fold assigned entire 84 hour buckets to either training or validation.

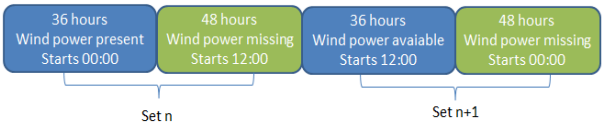


Figure 1 - Training/validation splitting

Figure 1 shows how the set assignment was done. One can notice that odd numbered sets starts at 00:00 and even numbered starts at 12:00. This also happens with the missing spots in the evaluation period. As for the 5-fold splitting, it was done like this:

- Validation folds: Fold 1 contained sets 1, 6, 11 and so on as validation set and the remaining as training set. Fold 2, contained 2, 7, 12, …, as validation, and analogous splitting were done for folds 3, 4 and 5.
- Test fold: Contained all available filled data and was used to build models to predict missing data.

In this scenario, each model needed to be trained six times. One for each validation folds to tune model parameters and also train ensembles, totalizing five and another one with full available data to predict the leaderboard. This procedure allowed us to produce consistent results most of the time, meaning that improvements found on available data almost always generalized to the missing unknown data.

In this problem there is a time component in the data, so random cross-validation tends to overestimate the performance of models as it implicitly gives to the model more information of the surroundings of each 48 hour spot. This was constated in an early step benchmark.

III. ALGORITHMS

In this work we have used mainly three machine learning algorithms, all of them on the R statistical environment. Table 2 gives a brief description of how each algorithm was employed. The rest of this section outlines these algorithms in more detail.

Table 1 - Used Algorithms

Algorithm	Used for
GLM [3]	Wind strength and direction components influence on each farm. Ensemble. Post process to smooth (high frequency filter) the predictions.
K-MEANS [4]	Similarity model to detect farm and overall inertia components
GBM[5]	Models by farm. Models by time slot. Overall models

IV. EVALUATION METRIC

The evaluation metric was RMSE [6], rooted mean squared error. This metric goes down as the accuracy of the model goes up.

V. THE APPROACH

The basic framework of the approach is shown on **Figure 2**. First, in the preprocessing step, features were created. After that three types of models were trained using the processed data:

- Models that were trained for each farm;
- Models that were trained for each predicted time slot (1h-3h ahead, 4h-6h ahead, … 45h-48h ahead);
- Overall models trained without splitting the samples (except for the cross-validation).

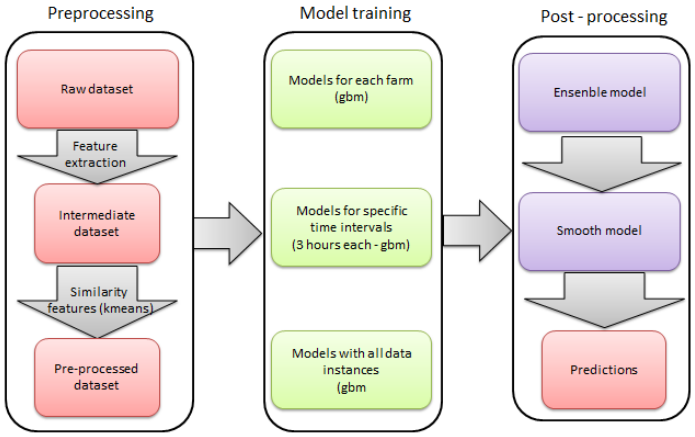


Figure 2 - Basic approach framework

A. *Features creation*

The first step of features creation was to create instances. The provided dataset did not contain these explicitly. To create these instances, the files "train.csv" and "benchmark.csv" were processed creating one entry for each date and each farm. Also the features hour, month and year were created at this point. Figure 3 how the produced data looked like and Table 2 explains each feature.

```
                    date farm    wp hour month year
     1: 2009-07-01 00:00:00    1 0.045   00    07 2009
     2: 2009-07-01 00:00:00    2 0.233   00    07 2009
     3: 2009-07-01 00:00:00    3 0.494   00    07 2009
     4: 2009-07-01 00:00:00    4 0.105   00    07 2009
     5: 2009-07-01 00:00:00    5 0.056   00    07 2009
    ---
183711: 2012-06-28 12:00:00    3    NA   12    06 2012
183712: 2012-06-28 12:00:00    4    NA   12    06 2012
183713: 2012-06-28 12:00:00    5    NA   12    06 2012
183714: 2012-06-28 12:00:00    6    NA   12    06 2012
183715: 2012-06-28 12:00:00    7    NA   12    06 2012
```

Figure 3 - Instance Creation Step 1

Table 2 – Features created in the first step

Name	Type	Description
Date	Date and time	This is the date and time of each wind power measurement. It is used mainly to join different tables of features
farm	Categorical (1-7)	Represents each farm
wp	Numeric	Wind power – value to be predicted
hour	Categorical (01-24)	Hour of each wind power measurement
month	Categorical (01-12)	Month of each wind power measurement
year	Categorical (2009-2012)	Year of each wind power measurement

The next step was to process the forecasts. Every forecast for each farm was used in this step. So for each date 4 distinct forecasts were obtained, except for the ones in the leaderboard set, which only one was used because only one was available. Also some other features were derived on this step. **Figure 4** and **Table 3** shows the data and the features meaning.

	date	farm	start	dist	turn	set	ws	wd	wd_cut
1:	2009-07-02 13:00:00	1	2009-07-01 00:00:00	37	00	1	1.76	198.89	(180,210]
2:	2009-07-02 13:00:00	1	2009-07-01 12:00:00	25	12	1	1.46	232.75	(210,240]
3:	2009-07-02 13:00:00	1	2009-07-02 00:00:00	13	00	1	1.51	191.55	(180,210]
4:	2009-07-02 13:00:00	1	2009-07-02 12:00:00	01	12	1	2.22	169.75	(150,180]
5:	2009-07-02 13:00:00	2	2009-07-01 00:00:00	37	00	1	1.22	104.02	(90,120]

576572:	2012-06-28 12:00:00	3	2012-06-26 12:00:00	48	12	312	4.21	53.80	(30,60]
576573:	2012-06-28 12:00:00	4	2012-06-26 12:00:00	48	12	312	3.05	318.31	(300,330]
576574:	2012-06-28 12:00:00	5	2012-06-26 12:00:00	48	12	312	1.66	115.34	(90,120]
576575:	2012-06-28 12:00:00	6	2012-06-26 12:00:00	48	12	312	3.49	325.47	(300,330]
576576:	2012-06-28 12:00:00	7	2012-06-26 12:00:00	48	12	312	4.06	328.97	(300,330]

Figure 4 - Features created using forecast

Table 3 - Forecast features

Name	Type	Description
date	Date and time	Target forecast date
farm	Categorical (values 1-7)	Represents each farm
start	Date and time	Date when the forecast was issued.
dist	Categorical (values 01-48)	Difference in hours of start and date. Distance of the forecasting.
turn	Categorical (00 and 12)	Indicates the starting hour of forecasting.
set	Categorical (1-312)	Each set represents a period of 36h+48h. This variable was used to do cross-validation training.
ws	Numerical	Predicted wind strength
wd	Numerical	Predicted wind direction
wd_cut	Categorical ([0,30]...(330,360])	Categorical version of wind direction.

The next created features were historical ones. For each forecasting the six values before the start of the forecast were used. On the test set, many of those values were unknown, and those instances were discarded during some part of training. For validation purposes it would be useful to calculate those previous values for every single instance, even in training set, as in the real situation they would be known. The historical features are in Figure 5 and Table 4.

	start	farm	dist	wp_hn01	wp_hn02	wp_hn03	wp_hn04	wp_hn05	wp_hn06
1:	2009-07-03 00:00:00	1	01	0.201	0.080	0.025	0.080	0.010	0.010
2:	2009-07-03 00:00:00	1	02	0.201	0.080	0.025	0.080	0.010	0.010
3:	2009-07-03 00:00:00	1	03	0.201	0.080	0.025	0.080	0.010	0.010
4:	2009-07-03 00:00:00	1	04	0.201	0.080	0.025	0.080	0.010	0.010
5:	2009-07-03 00:00:00	1	05	0.201	0.080	0.025	0.080	0.010	0.010

575732:	2012-06-26 12:00:00	7	44	0.000	0.076	0.076	0.076	0.101	0.126
575733:	2012-06-26 12:00:00	7	45	0.000	0.076	0.076	0.076	0.101	0.126
575734:	2012-06-26 12:00:00	7	46	0.000	0.076	0.076	0.076	0.101	0.126
575735:	2012-06-26 12:00:00	7	47	0.000	0.076	0.076	0.076	0.101	0.126
575736:	2012-06-26 12:00:00	7	48	0.000	0.076	0.076	0.076	0.101	0.126

Figure 5- History features

Table 4 - History features

Name	Type	Description
start	Date and time	Date when the forecast was issued.
farm	Categorical (1-7)	Wind farm
dist	Categorical (01-48)	Difference in hours of start and date. Distance of the forecasting.
wp_hnXX (XX = [01,06])	Numerical	X previous known value. 01 is the value for the farm at start date and time, 02 is start – 1 hour and so on.

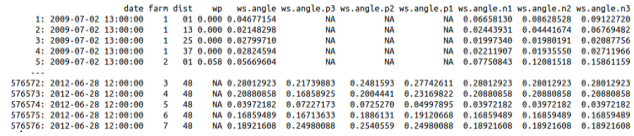
The next features were made to unify wind strength and wind direction on one single numeric variable. That unification was exploited later to make a similarity model and to correct predictions using moving averages.

For wind power generation, it is expected that not only strength of wind matters, but also direction. The windmills will have different efficiencies for different directions. As can be seen in [7] the wind power conversion is proportional to some constants and ws^3 (wind strength). It also depends on air density.

This feature generation consisted on building a simple regressive model for each farm. To avoid overfitting, 5-fold cross-validation were used. A very important issue of this cross validation is that it must use the set feature calculated before. Each set must be either include or excluded as a whole. Doing a instance based sampling would overfit, because the regression will infer some information related to specifics 36h+48h periods.

The R formula used in this training was "*wp ~ wd_cut*(ws + ws2 + ws3)*", *ws2* and *ws3* are ws^2 and ws^3. wd_cut isn't exactly the same as described previously. For this training it was a categorical value that split *wd* in intervals of 8 degrees (this splitting was found iteratively). The result of this training

was called *ws.angle*. Another feature map was built with these values. In this map it was also included the three previous and three next values of *ws.angles*. The result is shown on Figure 6 and Table 5.



	date	farm	dist	wp	ws.angle	ws.angle.p3	ws.angle.p2	ws.angle.p1	ws.angle.n1	ws.angle.n2	ws.angle.n3
1:	2009-07-02 13:00:00	1	01	0.000	0.04677154	NA	NA	NA	0.06658130	0.08628528	0.09122720
2:	2009-07-02 13:00:00	1	13	0.000	0.02148298	NA	NA	NA	0.02443931	0.04441674	0.06769482
3:	2009-07-02 13:00:00	1	25	0.000	0.02799710	NA	NA	NA	0.01997340	0.01980191	0.02087756
4:	2009-07-02 13:00:00	1	37	0.000	0.02824594	NA	NA	NA	0.02211907	0.01935550	0.02711966
5:	2009-07-02 13:00:00	2	01	0.058	0.05669604	NA	NA	NA	0.07750843	0.12081518	0.15861159

576572:	2012-06-28 12:00:00	3	48	NA	0.28012923	0.21739883	0.2481593	0.27742611	0.28012923	0.28012923	0.28012923
576573:	2012-06-28 12:00:00	4	48	NA	0.20880858	0.16858925	0.2004441	0.23169822	0.20880858	0.20880858	0.20880858
576574:	2012-06-28 12:00:00	5	48	NA	0.03972182	0.07227173	0.0725270	0.04997895	0.03972182	0.03972182	0.03972182
576575:	2012-06-28 12:00:00	6	48	NA	0.16859489	0.16713633	0.1886131	0.19120668	0.16859489	0.16859489	0.16859489
576576:	2012-06-28 12:00:00	7	48	NA	0.18921608	0.24980088	0.2540559	0.24980088	0.18921608	0.18921608	0.18921608

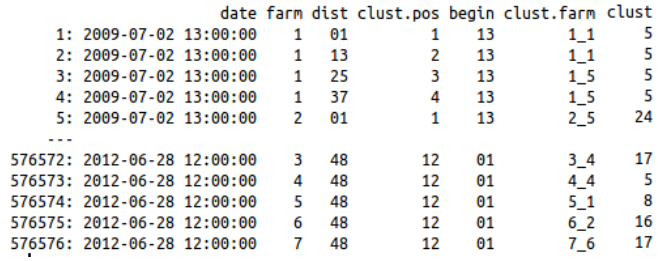
Figure 6 - Features unifying ws and angle

Table 5 - Features unifying ws and angle

Name	Type	Description
date	Date and time	Target forecast date
farm	Categorical (1-7)	Wind farm
dist	Categorical (01-48)	Difference in hours of start and date.
wp	Numeric	Wind power
ws.angle	Numeric	Influence of both ws and wd for date
ws.angle.p3	Numeric	ws.angle for date – 3 hours. If not found defaults nearest ws.angle.
ws.angle.p2	Numeric	ws.angle for date – 2 hours. If not found defaults nearest ws.angle
ws.angle.p1	Numeric	ws.angle for date – 1 hours. If not found defaults nearest ws.angle
ws.angle.n3	Numeric	ws.angle for date + 3 hours. If not found defaults nearest ws.angle
ws.angle.n2	Numeric	ws.angle for date + 2 hours. If not found defaults nearest ws.angle
ws.angle.n1	Numeric	ws.angle for date + 1 hours. If not found defaults nearest ws.angle

Finally, the last features were similarity ones. Since the predicted values are a time series, it is expected to have some inertia between predicted values. There shouldn't have any sudden transitions. One way to deal and let the model learn this was done in the last step. Including previous and next expected values will make the model learn some dependency between $T \pm n$ and T. Another way used to do this is similarity. Extract the farm behavior for a set of predictions.

The similarity features were calculated using k-means. For each 12 hour period of forecasting ([1h-12h],[13h-24h],[25h-36h],[37h-48h]) a instance was built. For each farm, six clusters were calculated, and for a dataset including all farms, twenty four clusters were calculated. See Figure 7 Table 6 for the results.



	date	farm	dist	clust.pos	begin	clust.farm	clust
1:	2009-07-02 13:00:00	1	01	1	13	1_1	5
2:	2009-07-02 13:00:00	1	13	2	13	1_1	5
3:	2009-07-02 13:00:00	1	25	3	13	1_5	5
4:	2009-07-02 13:00:00	1	37	4	13	1_5	5
5:	2009-07-02 13:00:00	2	01	1	13	2_5	24

576572:	2012-06-28 12:00:00	3	48	12	01	3_4	17
576573:	2012-06-28 12:00:00	4	48	12	01	4_4	5
576574:	2012-06-28 12:00:00	5	48	12	01	5_1	8
576575:	2012-06-28 12:00:00	6	48	12	01	6_2	16
576576:	2012-06-28 12:00:00	7	48	12	01	7_6	17

Figure 7 - Similarity cluster

Table 6 - Similarity features

Name	Type	Description
date	Date and time	Target forecast date
farm	Categorical (1-7)	Wind farm
dist	Categorical (01-48)	Difference in hours of start and date. Distance of the forecasting.
clust.pos	Categorical (1-12)	Position of forecast inside the 12 sequential values. The same as the remainder of the division of (dist-1) by 12.
begin	Categorical	Hour of the first value of the 12 hour sequence.
clust.farm	Categorical (42 = 7 x 6)	Cluster that the 12 sequential hour got assigned.
clust	Categorical (1-24)	General cluster

The final features were obtained by joining all produced features. To join the features simply match the common columns of each using a cartesian product.

B. Modeling techniques

All models were trained using the same 5-fold validation scheme. A very important characteristic of this training is that the validation sets were built to either include or remove the 36h+48h hours as a whole, as repeatedly said before. Instance by instance sampling wouldn't reflect neither the real situation nor the test set one. By doing this, a consistent validation set was obtained. Significant improvements almost always reflected improvements on the leaderboard.

The models were trained aiming two things, as already mentioned before:

- Model the wind power generation equation, based on constants, wind strength, direction and air density. The last one was surrogated by seasonal features, like time, hour and year. Express the windmill behavior.

- Discover the relationship between wind power generation of T and T±n. The objective here was to reflect the inertia of windmills.

To achieve this, the following types of models were trained:

1. GBM with gaussian distribution and formula *wp ~ ws + farm + dist + ws.angle + ws.angle.p1 + ws.angle.p2 + ws.angle.p3 + ws.angle.n1 + ws.angle.n2 + ws.angle.n3 + hour + month*: The aim of this model was to learn inertial behavior and something about air density and other occasional environment influences. The inertial behavior was learned through *ws.angle(T±3)*. Occasional influences were inferred using *hour*, *month* and *year* categorical variables. Maybe using temperature and or humidity measurements could achieve it better on real world situations. Hour and month is related to night/day and seasons relationship. The *farm* value will deal with any farm specifics and *dist* with errors due to the distance of forecast (the longer the distance, greater the error)

2. GBM with gaussian distribution and formula *wp ~ ws + farm + ws.angle + ws.angle.p1 + ws.angle.p2 + ws.angle.p3 + ws.angle.n1 + ws.angle.n2 + ws.angle.n3 + hour + month + year + dist + wp_hn01*: Same as before including the latest known value of wind power. This value can provide some valuable context information and a starting point for the time series. The reason why both models exist is because many instances of the testing period were discarded to train this one (they lack wp_hn01). In the real situation it wouldn't be necessary, because it would be always available.

3. GBM with gaussian distribution and formula *wp ~ farm + dist + wp_hn02 + wp_hn03 + wp_hn04 + hour + month + clust.farm + clust + begin + clust.pos*: This model was built to discover some environment influences. Mixing these features with the previous ones didn't show improvement. Doing separated trainings did.

All models mentioned before were trained in tree flavors: per farm, per clustered distance ((0,3],(3,6]...(45,48]), and one without distinction (Figure 2).

The final model was a linear ensemble of the previously mentioned (9 models). It was trained using the cross-validated responses for each one. It also used *farm* and *dist* as features, because they were used to do the splitting, and had a relationship with each models weakness and strengths.

After that yet another model was trained. A smoothing one. It took the result of the ensemble and did a T±n interpolation of the predictions. This model became almost useless when *ws.angle(T±n)* was included in the previous ones. But it did improve the prediction a bit, which was useful for a competition.

VI. RESULTS

Figure 8 shows the score improvement of the milestones in this solution. As can be seen, the first model, containing only forecasts and previous known values scored 0.1685. Then by adding some seasonal features (hour, month and year) it went to 0.16393. Next were included in the set the intervals of 36h that were available on the test period and it got better again. After the cross-validation scheme was changed to do the validation splitting using whole 36+48 the score jumped to 0.15315. Then adding the overall model and the smoothing model of final predictions, took it to 0.15103. The last big leap on the score was the inclusion of all combinations of dates and forecasts, with 4 forecasts for each known instance, scoring 0.14779. And finally the unsupervised cluster features were included to get the final and best score of 0.14567.

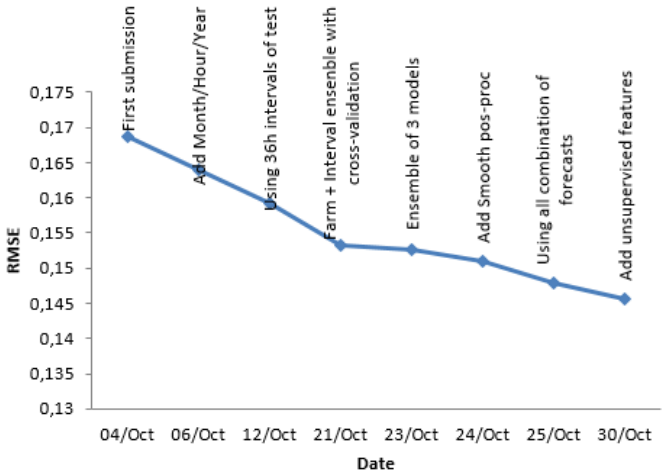


Figure 8 - Score evolution

Looking at the final results on Table 7, we can see that the whole procedure generalizes well, getting consistent results both on public and private leaderboards.

Table 7 - Final Standings

Rank	Team	Public Score	Private Score
1	Leustagos	0.14574	0.14567
2	DuckTile	0.14872	0.14720
3	Gilberto Titericz Jr	0.14792	0.14822
4	Stefan Henß	0.14684	0.14854
5	Mz	0.14804	0.14916

VII. DISCUSSION

In this work was provided an efficient approach to predict wind power generation on seven distinct wind farms. The solution was composed by two parts: extracting features and applying distinct aggregation schemes on the data to create models. In the end this approach used many models, more than needed in an actual situation, but that happened because this approach was done to win a competition, when every bit of performance must be considered, disregarding the computation expenses to build it. Also some data that would be known in

real operation, were hidden, adding complexity to the training process. After the competition, some tests with the models were performed and a simpler version of it was obtained, a version that performed almost as good, but much less complicated. In that approach, only the two overall gbm regressions were used and the features built to merge wind strength and wind direction were discarded. Maybe it would be possible to achieve even better accuracy by including more features, like temperature and humidity forecasts.

VIII. CONCLUSION

We have shown that with some clever combination of well-known algorithms (gbm, linear regression and k-means) it is possible to achieve a high precision for wind power prediction. The most important steps were choosing a reliable scheme of validation for training and creating good features. The key for the result was to never forget the time-series nature of the dataset. Many of the features, some post processing steps and the validation scheme used this characteristic to be designed, and they were the source of this work success. The proposed solution was somewhat complex but accurate. Some tests showed that it could be greatly simplified, with only marginal performance degradation.

REFERENCES

[1] Global Energy Forecasting Competition 2012 - Wind Forecasting. http://www.kaggle.com/c/GEF2012-wind-forecasting, 2013.

[2] Tao Hong, Pierre Pinson and Shu Fan, "Global Energy Forecasting Competition 2012", GEFCom 2013.

[3] Charles J. Geyer . Generalized Linear Models. http://www.stat.umn.edu/geyer/5931/mle/glm.pdf

[4] R Kmeans package, http://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html, 2013

[5] Greg Ridgeway. gbm: Generalized Boosted Regression Models. http://cran.r-project.org/web/packages/gbm/index.html, 2013

[6] Rooted Mean Square Deviation, RMSE, http://en.wikipedia.org/wiki/Root-mean-square_deviation, 2013

[7] Wind power generation. http://en.wikipedia.org/wiki/Wind_power