Effects of the Pre-processing Algorithms in Fault Diagnosis of wind turbines

Pere Marti-Puig[a,*], Alejandro Blanco-M.[a,b,*], Juan José Cárdenas[b], Jordi Cusidó[b], Jordi Solé-Casals[a]

[a]*Data and Signal Processing Group, U Science Tech, University of Vic - Central University of Catalonia, c/ de la Laura 13, 08500 Vic, Catalonia, Spain*
[b]*Smartive-ITESTIT SL, Catalonia, Spain*

Abstract

The wind sector spends roughly 2200M€ in repair the wind turbines failures. These failures do not contribute to the goal of reducing greenhouse gases emissions. The 25-35% of the generation costs are operation and maintenance services. To reduce this amount, the wind turbine industry is backing on the Machine Learning techniques over SCADA data. This data can contain errors produced by missing entries, uncalibrated sensors or human errors. Each kind of error must be handled carefully because extreme values are not always produced by data reading errors or noise. This document evaluates the impact of removing extreme values (outliers) applying several widely used techniques like Quantile, Hampel and ESD with the recommended cut-off values. Experimental results on real data show that removing outliers systematically is not a good practice. The use of manually defined ranges (static and dynamic) could be a better filtering strategy.

Keywords: Wind Farms, SCADA data, Pre-processing, Outliers, Fault Diagnosis, Renewable Energy

*These authors contributed equally to this work
Phone: +34 938815519
Fax: +34 938814307
Email: pere.marti@uvic.cat

1. Introduction

The reduction of greenhouse gases emissions and the independence of the fossil fuels are main goals of many government policies (

The SCADA system collects data from different parts of the turbine, which are grouped into systems (

Raw data obtained from SCADA contains several kind of errors categorized as: missed data caused by communications failures, presence of extreme values due to sensors failures, data coming from poorly calibrated sensors or by replaced sensors which report outputs in a different range, errors in the SCADA system or even human errors (

This study shows the importance of outliers in the prognosis models for wind turbines. Removing outliers systematically, which are frequently considered noise or extreme values, is not a good strategy. This is also indicated in several general case studies (see for example Gibert et al. (2016) survey), but without taking into account that these values are the less frequent turbine's operation mode (failure states), representing 2% to 3% of the dataset according to Conroy et al. (2011). This information cannot be removed or replaced, otherwise the generated models will have high accuracy rates in the training step but low accuracy rates in the testing step, since the failures states and the generalization power of the model is reduced.

2. Materials and methods

This section covers the techniques that have been applied in the experiments in order to identify and remove outliers. For each case a description of the method and the data work-flow over the algorithm is shown. Each method will be applied separately over the same input dataset to demonstrate the effect of the method. An independent analysis is done for each case using all the available information in several wind turbines. Finally the results generated by the models are analyzed taking into account the technique used when removing outliers and comparing them with the results obtained with the original values (i.e.: without removing outliers).

Each dataset is split in two parts, *train* dataset and *test* dataset. It is important to respect the time arrangement because random sampling might introduce future patterns in the *train* dataset which will affect the model estimation and the final prognosis results.

Some studies point out the benefits of removing outliers to improve final results in a machine learning system. That is the case of (

2.1. Data background

The used SCADA data follows the IEC 61400-25 format IEC (2006) with a hierarchical structure of turbines (Logical devices) and physical systems (Logical nodes). Data was gathered via an OPC(OLE for Process Control) (OPC Fundation (2016)) with update periods of 5 to 10 minutes, producing several types of indicators. Only failure events and statistical indicators are kept. Each sensor usually provides *min*, *mean*, *max* and *standard deviation* values.

The dataset is stored in a local database which has been recording values from the SCADA over the years. The dataset has a structure of table with the entries at each time interval in rows with as many columns as different sensors readings. The failure events are stored in a different table since they are recorded in a different format. These failure events are categorized as *alarms* (failure states) and *warnings* (maintenance service, start or stop messages). An example of the data format generated is shown in table 1.

date_time	power	bearing_temp	gen_1_speed	temp_oil_mult
2014-12-08 06:20:00	1701.17	29.40625	1291.84	36.39
2014-12-08 06:30:00	1583.11	28.14462	1055.23	22.08
2014-12-08 06:40:00	1664.03	28.03261	1132.16	23.43
2014-12-08 06:50:00	1722.47	29.8721	1312.66	22.68
2014-12-08 07:00:00	1647.91	29.0121	1231.78	21.82

Table 1: Example of the data analyzed (part of a real table)

Turbine Model	Machines	Years	Rows/ year	Variables	Triggered alarms	Total registers evaluated
Fuhrlander fl2500	5	4	105.120	303	72.422	2.102.400
Vestas V90 'wf1'	7	4	52.560	194	9.681	1.471.680
Vestas V90 'wf2'	13	4	52.560	63	5.063	2.733.120
Siemens Izar 55/1300	26	1	52.560	24	369.218	1.366.560
Wfa H1	1	7	52.560	406	83.716	52.560
Total	52	20		992	540.100	7.726.320

Table 2: Data summary

2.2. Input data pre-selection

In order to study a specific type of failure, an expert have to choose from all possible variables, the most relevant to the physical system or subsystem to be analyzed. This variable is the output of the model (Vestas R+D

(2004)). These experiments are focus on the transmission system, and more specifically the

Based on the selected subset of events (by an expert for a system), a contrast of hypothesis is generated in order to identify the variables that are more related with the selected events. The null hypothesis H_0 defines that a variable presents

The most common value for the threshold t is $t = 3$, which means that all points that deviates 3σ from the mean value will be rejected, considering about 0.3% of the observed data as outliers. This method is very sensitive to distributions that contains many outliers and it will fail with data containing more than 10% of outliers (Pearson (2005)). The ESD algorithm is implemented as follows:

Algorithm 1 ESD outlier filter

```
procedure CLEANESD(variables)
    t ← 3
    for all variable,varID in variables[:,:] do :
        mean ← mean(variable[:])
        σ ← sd(variable[:])
        for all entry,entID in variable[:] do :
            if entry < mean − (t ∗ σ) or mean + (t ∗ σ) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

2.3. Quantile filter

Another commonly used method is based on the distance of the points being above of the third quartile or below of the first quartile. These quartile values determine the acceptable range of the values following equation 22:

$$(Q_1 - (c * IQR)) < x_i < (Q_3 + (c * IQR)), \tag{1}$$

where:

x_i : is the i entry from a single variable X
Q_1, Q_3 : are the $first$ and $third$ quartile of the current variable X
IQR : is the interquartile as in equation (23)
c : is the number of IQR

$$IQR = (Q_3 - Q_1). \tag{2}$$

A common value for c is $c = 1.5$. This method is less sensitive to outliers than the ESD and it is well suited for asymmetric distributions since it does not depend on the center of the data (Pearson (2005)), but it declares as outliers many nominal observations determined as non-outliers by a human expert. The simplified algorithm has been implemented as follows:

Algorithm 2 Quantile outlier filter

```
procedure CLEANQUANTILE(variables)
    c ← 1.5
    outlierList ← []                                   ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        Q1 ← quantile(variable[:], 25%)
        Q3 ← quantile(variable[:], 75%)
        IQR ← Q3 − Q1
        for all entry,entID in variable[:] do :
            if entry < (Q1 − c ∗ IQR) or (Q3 + c ∗ IQR) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

2.4. Hampel identifier

The Hampel identifier is based on two robust measures of location and scale, the median and the MAD (median of the absolute deviations), respectively. Observations too far from the median of the data with respect to their MAD are declared to be outliers (Christophe Leys (2013)). Again, a proportion factor k will modulate how to calculate that distance. In this case, this factor is derived by using the inverse of the Gaussian cumulative distribution function (Φ^{-1}) calculated on the 75% confidence interval which takes the area until the quantile Q_3:

$$k = 1/\left(\Phi^{-1}(3/4)\right) \approx 1.4826. \tag{3}$$

The accepted range for the detection procedure is calculated as follows:

$$(\hat{X} - (k * MAD)) < x_i < (\hat{X} + (k * MAD)), \tag{4}$$

where:

x_i : is the i entry from a single variable X
$\hat{X}$: is the median of single variable X
k : is the constant scale factor calculated as in equation (24)
MAD : is the median absolute deviation calculated as in equation (26)

and the MAD is calculated as follows::

$$MAD = median(|x_i - \hat{X}|). \tag{5}$$

The simplified algorithm has been implemented as follows:

Algorithm 3 Hampel outlier filter

```
procedure CLEANHAMPEL(variables)
    k ← 1.4826
    outlierList ← []                                   ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        median ← median(variable[:])
        MAD ← mad(variable[:])
        for all entry,entID in variable[:] do :
            if entry < (median − k ∗ MAD) or (median + k ∗ MAD) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is marked as outlier,value removed
            end if
        end for
    end for
end procedure
```

2.5. Evaluation

The evaluation of the methods will be done with the datasets of the wind farms in table 2. The filtering methods will be applied on the train datasets and the models will be tested on the (unknown) test dataset respecting the original time arrangement. All the experiments will be performed using the same target variable, which is the most important one that indicates the temperature of the wind turbine gearbox system. Modeling the relationship between the selected inputs and this target variable, the failures could be detected because a significant difference will exist between the real and the modeled result.

To quantify the effect of the filtering step, a set of indicators gathered from the results from the models are evaluated. One of the most effective method to evaluate the impact of such filters on machine learning algorithms is to implement a normality model based on Partial Least Squares (PLS) (Wold (2001)), which can be evaluated using the mean squared error (MSE). The model is computed using the same train dataset with and without outliers and then both models will be applied to the test dataset. Apart from the MSE, the scatter plots of the real and estimated values are used to compute the best regression line that fits to them. Ideally, if there is a perfect relation between the points, a line with a gradient of 45° is obtained.

3. Results

3.1. Results summary

In table 6 and figure 14, a summary of the experiments performed is presented. The considered parks and wind turbines are listed in table 2. For lack of space, the list only contains some wind turbines of each park. For the sake of clarity, an MSE ratio is calculated as the quotient of MSE values obtained by filtering and without filtering. Therefore the PLS model is generated and evaluated, the quotient will be >1 if the filtering strategy doesn't work appropriately. On the contrary, if the filtering strategy works as expected, the ratio will be <1 (these cases are indicated in italics in table 6).

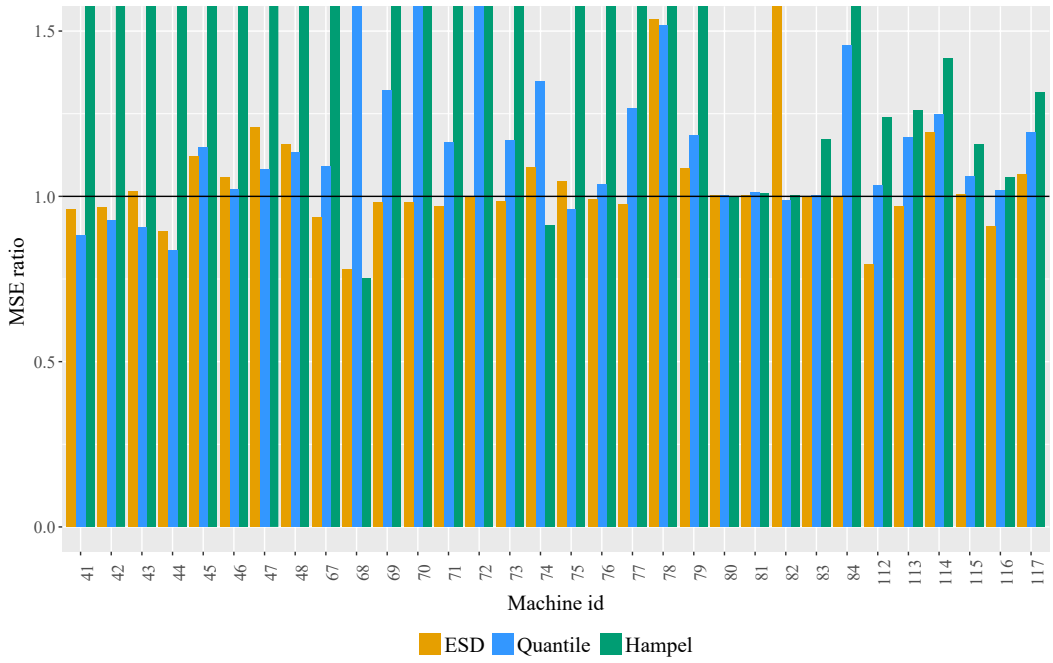

Figure 1: Result summary bar plot. A MSE ratio of one means no improvement.

As can be seen in table 6, values are usually >1. Note that the results are much worse using quantile or Hampel filtering than without filtering. (i.e.: MSE ratios are $\gg$1). Only the ESD filter seems to be interesting in some cases, but even in these cases, corresponding to the ratio <1, the difference in MSE is small.

Model	Machine id	ESD filter MSE Ratio	Quantile filter MSE Ratio	Hampel filter MSE Ratio
Fuhrlander FL2500	80	1,002	1,002	*0,998*
	81	1,002	1,011	1,008
	82	*0,999*	*0,987*	1,002
	83	1,000	1,002	1,171
	84	*0,996*	1,458	44,838
Vestas V90 wfa1	67	*0,935*	1,090	5,274
	68	*0,780*	4,640	*0,753*
	69	*0,983*	1,319	1,868
	70	*0,983*	1,604	8,317
	71	*0,971*	1,162	8,253
	72	*0,996*	1,851	12,410
	73	*0,985*	1,168	6,892
	74	1,088	1,347	*0,912*
	75	1,046	*0,959*	5,505
	76	*0,992*	1,037	4,813
	77	*0,975*	1,267	5,801
	78	1,536	1,518	8,010
	79	1,085	1,185	4,826
Siemens Izar 55/1300	41	*0,961*	*0,882*	210,940
	42	*0,966*	*0,928*	307,942
	43	1,015	*0,905*	250,313
	44	*0,895*	*0,835*	242,414
	45	1,121	1,147	172,567
	46	1,057	1,022	218,819
	47	1,208	1,080	280,106
	48	1,158	1,133	157,796
Vestas V90 wfa2	112	*0,795*	1,033	1,239
	113	*0,971*	1,179	1,260
	114	1,193	1,247	1,418
	115	1,007	1,060	1,156
	116	*0,908*	1,019	1,057
	117	1,065	1,193	1,315

Table 3: Result summary

Analyzing in detail all the cases reported in table 6, in 17 over 32 cases, the ESD filtering method is useful when testing the model representing 53% of the cases. Even if that seems a high number of cases, in all of them the quotient is ≈1 , indicating that the MSE is almost the same when using the filter compared to the original (non-filtered) case. For the quantile filter, only 6 over 32 cases reported a quotient smaller than one. It means that only about 19% of the cases improved results after filtering. Finally, for the Hampel filter only 3 cases over 32 reported a quotient higher than one, i.e.: 9% of the cases.

Computing all the filters analyzed, in 73% of the cases the filtering procedure increased the MSE. Thus, as a rule of thumb, filtering is not a good strategy, and only in very few cases could slightly improve the results by decreasing MSE in the test dataset. According to the experiments carried out, in the case of needing a filter, the best choice would be to use the ESD filter, since it is able to eliminate some outliers that are not relevant nor related to the alarms.

3.2. Detailed results for unfiltered data

In order to better understand how the filtering strategy works, a specific example is detailed in the following sections, first without filtering, to have a baseline reference, and then by introducing the analyzed filtering strategies. The first turbine (named *T13*) of the first plant, composed by Vestas V90 machines, is selected as an example. An expert determined that the target variable for the model of this turbine is *gear_oil_temp_avg*, which has the distribution shown in figure 15. The following list shows the input variables selected by the method detailed on subsection 2.2:

- gear_bearing_temp_avg: Temperature of bearing that holds the rotor with blades.
- power_avg: Average power generated
- wind_avg: Average wind speed
- hydraulic_oil_temp_avg: Temperature of the oil which cool the gearbox.
- blades_pitchangle_max: Angle of the wind turbine blades.
- blades_bladea_controlvoltage_min: Voltage of the motors which controls the angle of the blades.

In this particular example, the smallest p-value is for *gear_bearing_temp_avg*. This is somehow expected because the *target variable* and this variable are components that are physically closer and in contact by metal parts which transfer heat.

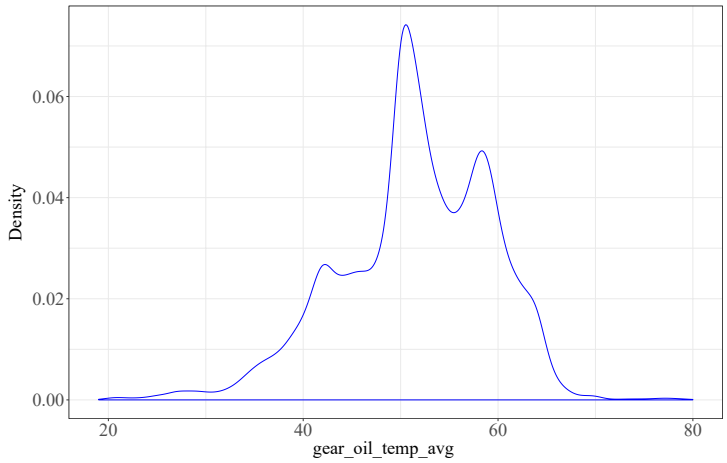

Figure 2: Histogram of target variable gear_oil_temp_avg

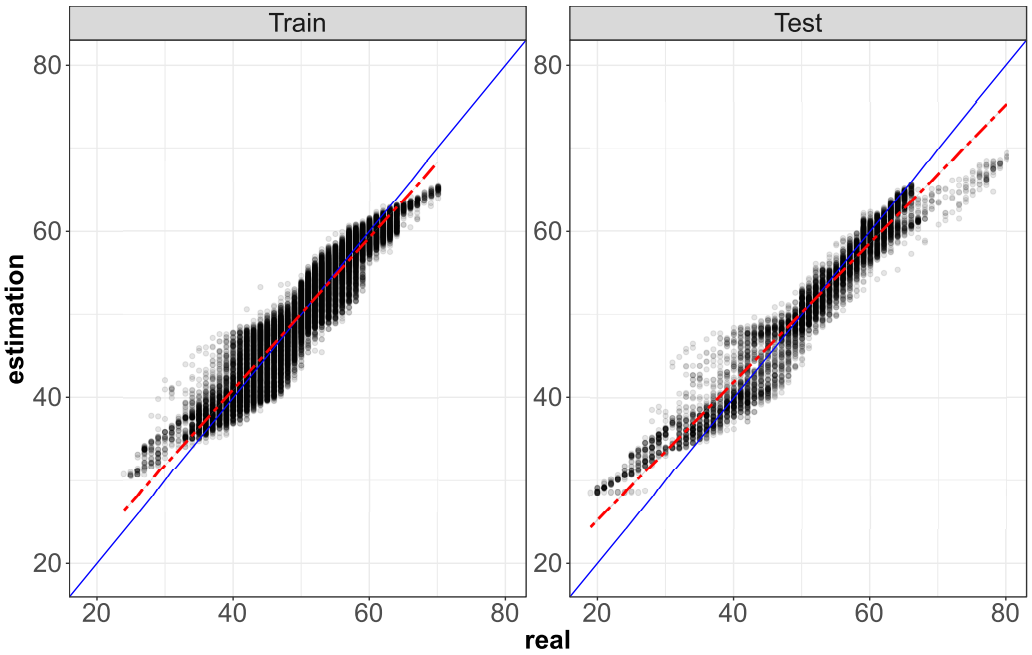

Figure 3: Estimation vs. real value of target variable in train and test. In horizontal axis the real values, while vertical axis the estimated values. The blue line would be the best prediction, the red one is the best fit line of the model prediction.

As a reference, the results of the model without filtering are shown in figure 16 for the train(left) and test(right) datasets. The X axis is the real

value of the target variable and the Y axis is the estimated value from the model. The best possible result is indicated by the 45°blue line and the red line indicates the best fit regression line for the current results, which is slightly leaned with respect to the reference. In this example the obtained gradient has a value of 42.4° with an MSE of 2.0768 for the training dataset, indicating that the model is not estimating all the values perfectly even on the same training dataset. On the test dataset the gradient is 40° with an MSE of 2.612 which is worse than the previous one. This is what it was expected as the model is now dealing with new (unknown) data.

3.3. Detailed results for the ESD filter

With the data being filtered by the ESD filter, many periods of alarm were identified as outliers, as can be seen in figures 17 and 18. Each figure corresponds to a different variable. In all these figures, outliers are in orange color. Violet color corresponds to the values which have been identified as outliers by the algorithms but at the same time are alarms reported by the wind turbine. Values with alarms are indicated in red color. Finally the remaining (non filtered data) are in green color. Two variables are detailed, corresponding to the variables that have the greatest number of alarms identified as outliers. This will reduce the number of alarms that feed the*no statistical relevance on the change of its mean on the day when the alarm/failure event is present. The alternative hypothesis* H_a *defines that a variable presents*

The most common value for the threshold t *is* $t = 3$*, which means that all points that deviates* 3σ *from the mean value will be rejected, considering about 0.3% of the observed data as outliers. This method is very sensitive to distributions that contains many outliers and it will fail with data containing more than 10% of outliers (Pearson (2005)). The ESD algorithm is implemented as follows:*

Algorithm 4 ESD outlier filter

```
procedure CLEANESD(variables)
    t ← 3
    for all variable,varID in variables[:,:] do :
        mean ← mean(variable[:])
        σ ← sd(variable[:])
        for all entry,entID in variable[:] do :
            if entry < mean − (t ∗ σ) or mean + (t ∗ σ) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

3.4. Quantile filter

Another commonly used method is based on the distance of the points being above of the third quartile or below of the first quartile. These quartile values determine the acceptable range of the values following equation 22:

$$(Q_1 - (c * IQR)) < x_i < (Q_3 + (c * IQR)), \tag{6}$$

where:

x_i *: is the* i *entry from a single variable* X
Q_1, Q_3 *: are the first and third quartile of the current variable* X
IQR *: is the interquartile as in equation (23)*
c *: is the number of IQR*

$$IQR = (Q_3 - Q_1). \tag{7}$$

A common value for c *is* $c = 1.5$*. This method is less sensitive to outliers than the ESD and it is well suited for asymmetric distributions since it does not depend on the center of the data (Pearson (2005)), but it declares as outliers many nominal observations determined as non-outliers by a human expert. The simplified algorithm has been implemented as follows:*

Algorithm 5 Quantile outlier filter

```
procedure cleanQuantile(variables)
    c ← 1.5
    outlierList ← []                                  ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        Q1 ← quantile(variable[:], 25%)
        Q3 ← quantile(variable[:], 75%)
        IQR ← Q3 − Q1
        for all entry,entID in variable[:] do :
            if entry < (Q1 − c ∗ IQR) or (Q3 + c ∗ IQR) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                  ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

3.5. Hampel identifier

The Hampel identifier is based on two robust measures of location and scale, the median and the MAD (median of the absolute deviations), respectively. Observations too far from the median of the data with respect to their MAD are declared to be outliers (Christophe Leys (2013)). Again, a proportion factor k will modulate how to calculate that distance. In this case, this factor is derived by using the inverse of the Gaussian cumulative distribution function (Φ^{-1})a statistically relevant difference in its mean value on the day when the alarm/failure event is present. The interval of confidence is defined at 95% which determines a p-value of 0.05. Any variable that has a p-value smaller than 0.05 is considered as a possible input variable for the model. All the considered candidates are sort from the lowest to the highest p-value, then the first six variables are selected to analyze them. In all the analyzed parks, using more than six variables does not significantly increase the model performance. On the contrary, computational time also increases when more than six variables are used. Therefore, the number of variables is set at six, which is a good trade-off between performance and computational time. As shown in other works (A. Zaher (2009), Meik Schlechtingen (2011), Michael Wilkinson (2014)) it is common to use the minimum number of variables in order to optimize the results while minimizing the complexity of the system. A diagram of the process is shown in figure **??**

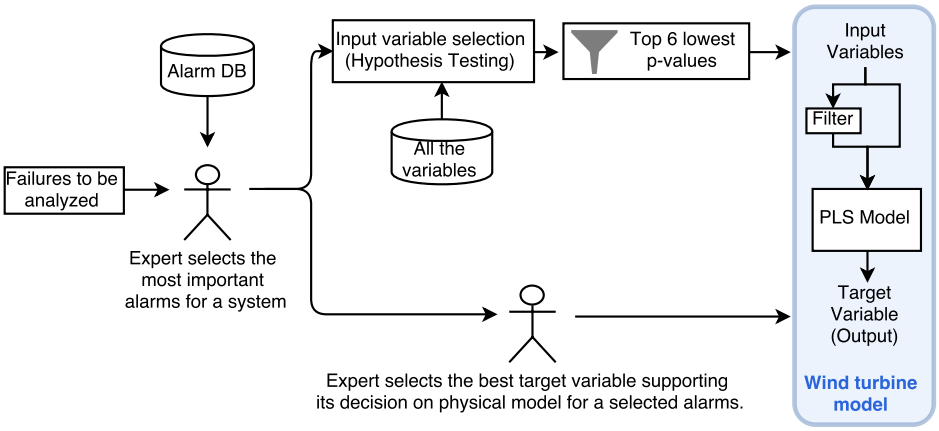

Figure 4: Flowchart of the process.

3.6. ESD filter

Extreme Studentized Deviate test (ESD) is a statistical test to detect outliers in an univariate dataset that have a normally distributed population. ESD defines that any point further from t standard deviations of the mean is an outlier. As shown in equation 21, any value falling outside the interval is considered an outlier:

$$(\mu - (t * \sigma)) < x_i < (\mu + (t * \sigma)), \tag{8}$$

where:

x_i *: is the* i *entry from a single variable* X
μ *: is the mean of the current variable* X
t *: is the number of standard deviations*
σ *: is the standard deviation of a single variable* X

The most common value for the threshold t *is* $t = 3$*, which means that all points that deviates* 3σ *from the mean value will be rejected, considering about 0.3% of the observed data as outliers. This method is very sensitive to distributions that contains many outliers and it will fail with data containing more than 10% of outliers (Pearson (2005)). The ESD algorithm is implemented as follows:*

Algorithm 6 ESD outlier filter

```
procedure cleanEsd(variables)
    t ← 3
    for all variable,varID in variables[:,:] do :
        mean ← mean(variable[:])
        σ ← sd(variable[:])
        for all entry,entID in variable[:] do :
            if entry < mean − (t ∗ σ) or mean + (t ∗ σ) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

3.7. Quantile filter

Another commonly used method is based on the distance of the points being above of the third quartile or below of the first quartile. These quartile values determine the acceptable range of the values following equation 22:

$$(Q_1 - (c * IQR)) < x_i < (Q_3 + (c * IQR)), \tag{9}$$

where:

x_i *: is the* i *entry from a single variable* X
Q_1, Q_3 *: are the first and third quartile of the current variable* X
IQR *: is the interquartile as in equation (23)*
c *: is the number of IQR*

$$IQR = (Q_3 - Q_1). \tag{10}$$

A common value for c *is* $c = 1.5$*. This method is less sensitive to outliers than the ESD and it is well suited for asymmetric distributions since it does not depend on the center of the data (Pearson (2005)), but it declares as outliers many nominal observations determined as non-outliers by a human expert. The simplified algorithm has been implemented as follows:*

Algorithm 7 Quantile outlier filter

```
procedure cleanQuantile(variables)
    c ← 1.5
    outlierList ← []                                  ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        Q1 ← quantile(variable[:], 25%)
        Q3 ← quantile(variable[:], 75%)
        IQR ← Q3 − Q1
        for all entry,entID in variable[:] do :
            if entry < (Q1 − c ∗ IQR) or (Q3 + c ∗ IQR) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                  ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

3.8. Hampel identifier

The Hampel identifier is based on two robust measures of location and scale, the median and the MAD (median of the absolute deviations), respectively. Observations too far from the median of the data with respect to their MAD are declared to be outliers (Christophe Leys (2013)). Again, a proportion factor k will modulate how to calculate that distance. In this case, this factor is derived by using the inverse of the Gaussian cumulative distribution function (Φ^{-1}) calculated on the 75% confidence interval which takes the area until the quantile Q_3:

$$k = 1/\left(\Phi^{-1}(3/4)\right) \approx 1.4826. \tag{11}$$

The accepted range for the detection procedure is calculated as follows:

$$(\hat{X} - (k * MAD)) < x_i < (\hat{X} + (k * MAD)), \tag{12}$$

where:

x_i *: is the i entry from a single variable X*
$\hat{X}$ *: is the median of single variable X*
k *: is the constant scale factor calculated as in equation (24)*
MAD *: is the median absolute deviation calculated as in equation (26)*

and the MAD is calculated as follows::

$$MAD = median(|x_i - \hat{X}|). \tag{13}$$

The simplified algorithm has been implemented as follows:

Algorithm 8 Hampel outlier filter

```
procedure cleanHampel(variables)
    k ← 1.4826
    outlierList ← []                                   ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        median ← median(variable[:])
        MAD ← mad(variable[:])
        for all entry,entID in variable[:] do :
            if entry < (median − k ∗ MAD) or (median + k ∗ MAD) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is marked as outlier,value removed
            end if
        end for
    end for
end procedure
```

3.9. Evaluation

The evaluation of the methods will be done with the datasets of the wind farms in table 2. The filtering methods will be applied on the train datasets and the models will be tested on the (unknown) test dataset respecting the original time arrangement. All the experiments will be performed using the same target variable, which is the most important one that indicates the temperature of the wind turbine gearbox system. Modeling the relationship between the selected inputs and this target variable, the failures could be detected because a significant difference will exist between the real and the modeled result.

To quantify the effect of the filtering step, a set of indicators gathered from the results from the models are evaluated. One of the most effective method to evaluate the impact of such filters on machine learning algorithms is to implement a normality model based on Partial Least Squares (PLS) (Wold (2001)), which can be evaluated using the mean squared error (MSE). The model is computed using the same train dataset with and without outliers and then both models will be applied to the test dataset. Apart from the MSE, the scatter plots of the real and estimated values are used to compute the best regression line that fits to them. Ideally, if there is a perfect relation between the points, a line with a gradient of 45° is obtained.

4. Results

4.1. Results summary

In table 6 and figure 14, a summary of the experiments performed is presented. The considered parks and wind turbines are listed in table 2. For lack

of space, the list only contains some wind turbines of each park. For the sake of clarity, an MSE ratio is calculated as the quotient of MSE values obtained by filtering and without filtering. Therefore the PLS model is generated and evaluated, the quotient will be >1 if the filtering strategy doesn't work appropriately. On the contrary, if the filtering strategy works as expected, the ratio will be <1 (these cases are indicated in italics in table 6).

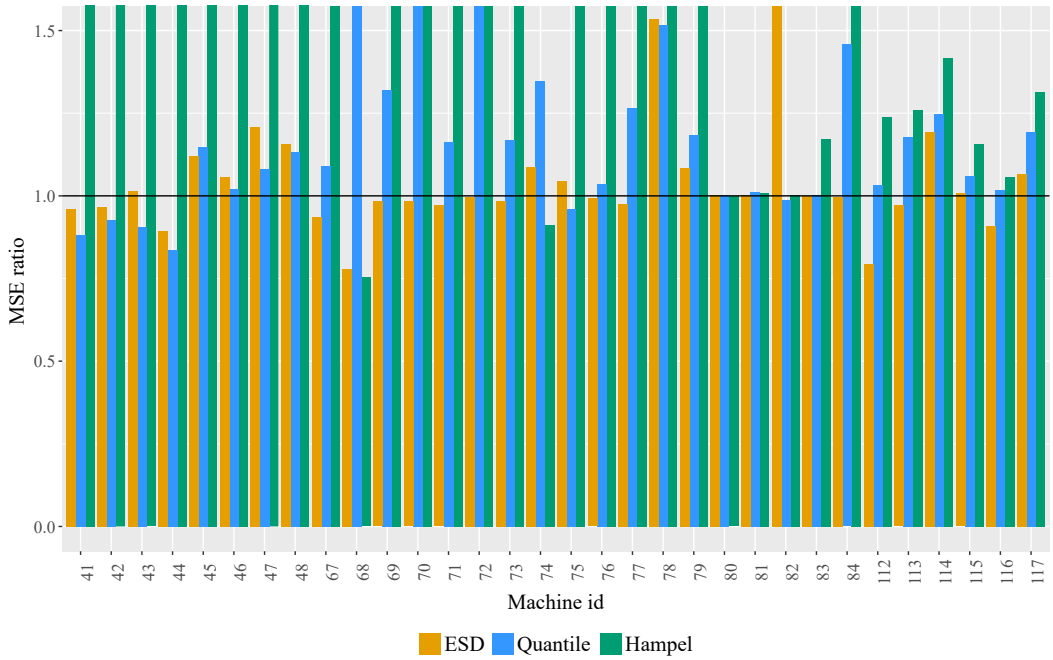

Figure 5: Result summary bar plot. A MSE ratio of one means no improvement.

As can be seen in table 6, values are usually >1. Note that the results are much worse using quantile or Hampel filtering than without filtering. (i.e.: MSE ratios are ≫1). Only the ESD filter seems to be interesting in some cases, but even in these cases, corresponding to the ratio <1, the difference in MSE is small.

Analyzing in detail all the cases reported in table 6, in 17 over 32 cases, the ESD filtering method is useful when testing the model representing 53% of the cases. Even if that seems a high number of cases, in all of them the quotient is ≈1 , indicating that the MSE is almost the same when using the filter compared to the original (non-filtered) case. For the quantile filter, only 6 over 32 cases reported a quotient smaller than one. It means that

Model	Machine id	ESD filter MSE Ratio	Quantile filter MSE Ratio	Hampel filter MSE Ratio
Fuhrlander FL2500	80	1,002	1,002	*0,998*
	81	1,002	1,011	1,008
	82	*0,999*	*0,987*	1,002
	83	1,000	1,002	1,171
	84	*0,996*	1,458	44,838
Vestas V90 wfa1	67	*0,935*	1,090	5,274
	68	*0,780*	4,640	*0,753*
	69	*0,983*	1,319	1,868
	70	*0,983*	1,604	8,317
	71	*0,971*	1,162	8,253
	72	*0,996*	1,851	12,410
	73	*0,985*	1,168	6,892
	74	1,088	1,347	*0,912*
	75	1,046	*0,959*	5,505
	76	*0,992*	1,037	4,813
	77	*0,975*	1,267	5,801
	78	1,536	1,518	8,010
	79	1,085	1,185	4,826
Siemens Izar 55/1300	41	*0,961*	*0,882*	210,940
	42	*0,966*	*0,928*	307,942
	43	1,015	*0,905*	250,313
	44	*0,895*	*0,835*	242,414
	45	1,121	1,147	172,567
	46	1,057	1,022	218,819
	47	1,208	1,080	280,106
	48	1,158	1,133	157,796
Vestas V90 wfa2	112	*0,795*	1,033	1,239
	113	*0,971*	1,179	1,260
	114	1,193	1,247	1,418
	115	1,007	1,060	1,156
	116	*0,908*	1,019	1,057
	117	1,065	1,193	1,315

Table 4: Result summary

only about 19% of the cases improved results after filtering. Finally, for the Hampel filter only 3 cases over 32 reported a quotient higher than one, i.e.: 9% of the cases.

Computing all the filters analyzed, in 73% of the cases the filtering procedure increased the MSE. Thus, as a rule of thumb, filtering is not a good strategy, and only in very few cases could slightly improve the results by decreasing MSE in the test dataset. According to the experiments carried out, in the case of needing a filter, the best choice would be to use the ESD filter, since it is able to eliminate some outliers that are not relevant nor related to the alarms.

4.2. Detailed results for unfiltered data

In order to better understand how the filtering strategy works, a specific example is detailed in the following sections, first without filtering, to have a baseline reference, and then by introducing the analyzed filtering strategies. The first turbine (named T13) of the first plant, composed by Vestas V90 machines, is selected as an example. An expert determined that the target variable for the model of this turbine is gear_oil_temp_avg, which has the distribution shown in figure 15. The following list shows the input variables selected by the method detailed on subsection 2.2:

- *gear_bearing_temp_avg: Temperature of bearing that holds the rotor with blades.*
- *power_avg: Average power generated*
- *wind_avg: Average wind speed*
- *hydraulic_oil_temp_avg: Temperature of the oil which cool the gearbox.*
- *blades_pitchangle_max: Angle of the wind turbine blades.*
- *blades_bladea_controlvoltage_min: Voltage of the motors which controls the angle of the blades.*

In this particular example, the smallest p-value is for gear_bearing_temp_avg. This is somehow expected because the target variable and this variable are components that are physically closer and in contact by metal parts which transfer heat.

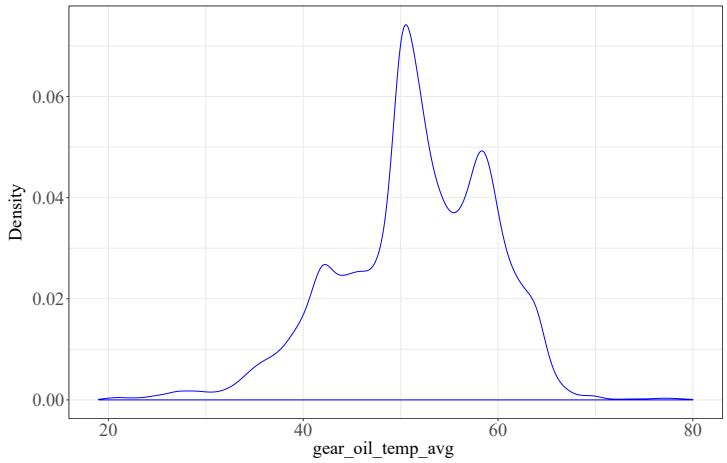

Figure 6: Histogram of target variable gear_oil_temp_avg

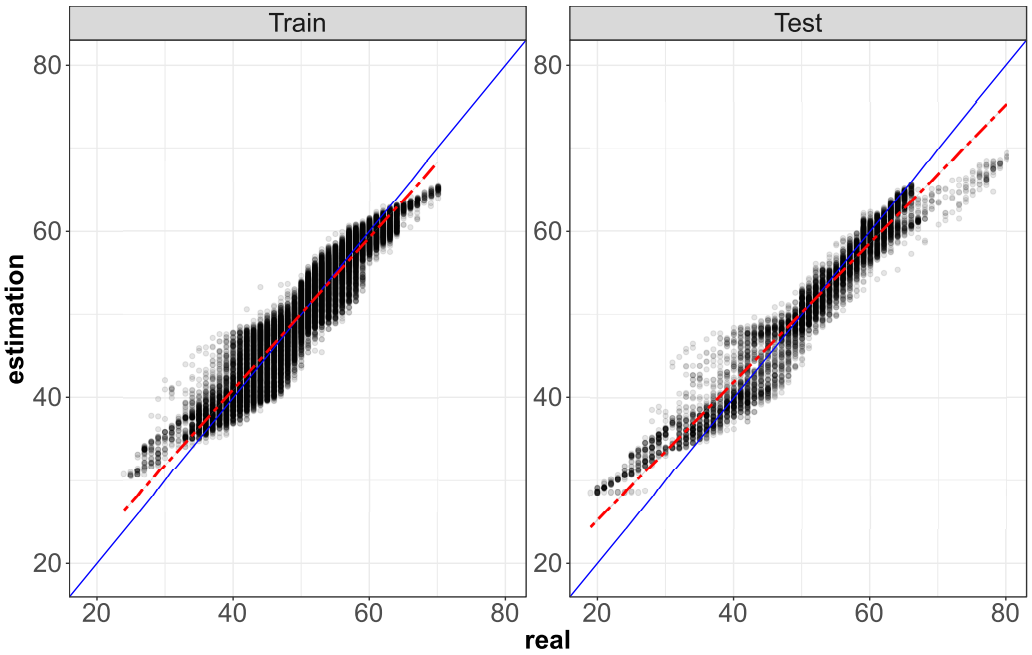

Figure 7: Estimation vs. real value of target variable in train and test. In horizontal axis the real values, while vertical axis the estimated values. The blue line would be the best prediction, the red one is the best fit line of the model prediction.

As a reference, the results of the model without filtering are shown in figure 16 for the train(left) and test(right) datasets. The X axis is the real value of

the target variable and the Y axis is the estimated value from the model. The best possible result is indicated by the 45°blue line and the red line indicates the best fit regression line for the current results, which is slightly leaned with respect to the reference. In this example the obtained gradient has a value of 42.4° with an MSE of 2.0768 for the training dataset, indicating that the model is not estimating all the values perfectly even on the same training dataset. On the test dataset the gradient is 40° with an MSE of 2.612 which is worse than the previous one. This is what it was expected as the model is now dealing with new (unknown) data.

4.3. Detailed results for the ESD filter

With the data being filtered by the ESD filter, many periods of alarm were identified as outliers, as can be seen in figures 17 and 18. Each figure corresponds to a different variable. In all these figures, outliers are in orange color. Violet color corresponds to the values which have been identified as outliers by the algorithms but at the same time are alarms reported by the wind turbine. Values with alarms are indicated in red color. Finally the remaining (non filtered data) are in green color. Two variables are detailed, corresponding to the variables that have the greatest number of alarms identified as outliers. This will reduce the number of alarms that feed the machine learning model and therefore will reduce its prediction capability. The outliers detected by this algorithm represents the 2.1% of the training data.

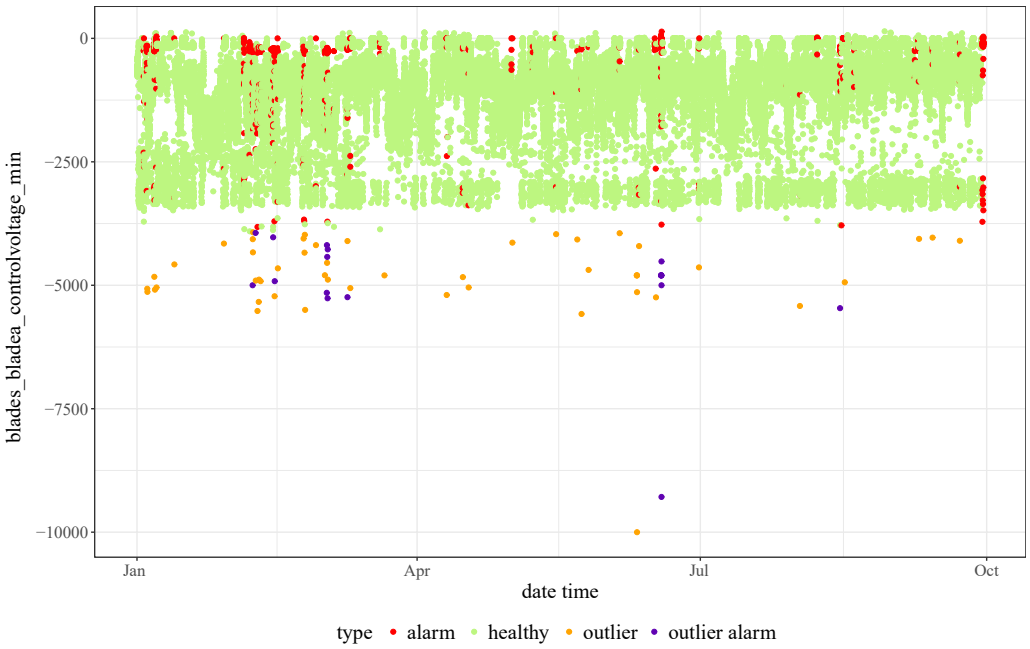

Figure 8: Labeling of points (variable *blade_control_voltage*) on filtered dataset

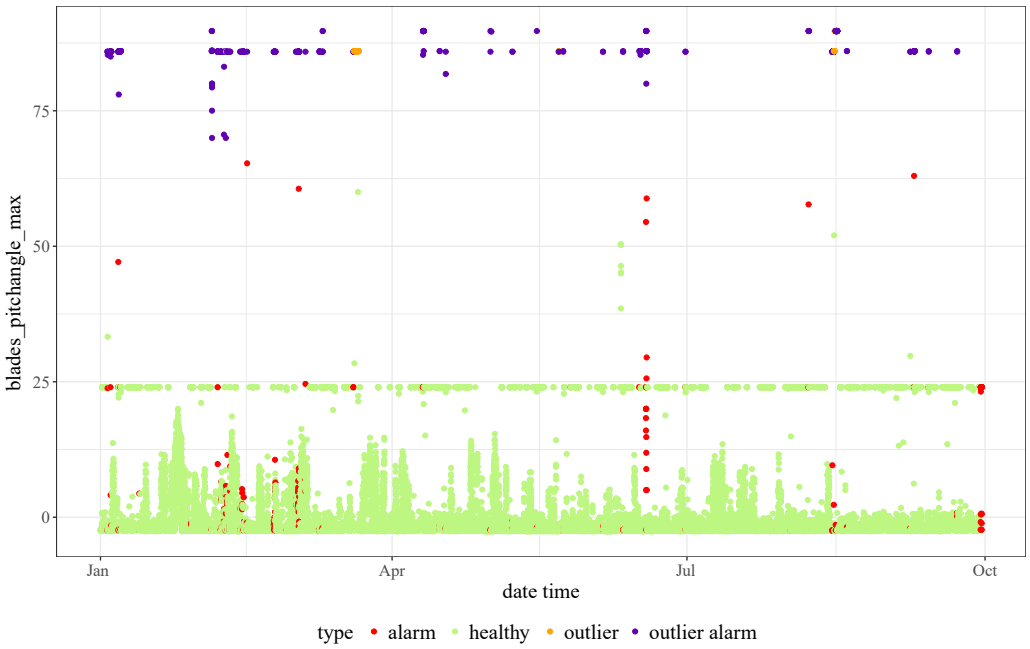

Figure 9: Labeling of points (variable *blade_pitch_angle_max*) on filtered dataset

The impact on the model results is presentedMain Bearing subsystem which supports the rotor of the wind turbine, the origin of many alarms and with the longest downtime (Tavner (2009)).

Based on the selected subset of events (by an expert for a system), a contrast of hypothesis is generated in order to identify the variables that are more related with the selected events. The null hypothesis H_0 defines that a variable presents

The most common value for the threshold t is $t = 3$, which means that all points that deviates 3σ from the mean value will be rejected, considering about 0.3% of the observed data as outliers. This method is very sensitive to distributions that contains many outliers and it will fail with data containing more than 10% of outliers (Pearson (2005)). The ESD algorithm is implemented as follows:

Algorithm 9 ESD outlier filter

```
procedure CLEANESD(variables)
    t ← 3
    for all variable,varID in variables[:,:] do :
        mean ← mean(variable[:])
        σ ← sd(variable[:])
        for all entry,entID in variable[:] do :
            if entry < mean − (t ∗ σ) or mean + (t ∗ σ) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

4.4. Quantile filter

Another commonly used method is based on the distance of the points being above of the third quartile or below of the first quartile. These quartile values determine the acceptable range of the values following equation 22:

$$(Q_1 - (c * IQR)) < x_i < (Q_3 + (c * IQR)), \tag{14}$$

where:

x_i *: is the i entry from a single variable X*
Q_1, Q_3 *: are the first and third quartile of the current variable X*
IQR *: is the interquartile as in equation (23)*
c *: is the number of IQR*

$$IQR = (Q_3 - Q_1). \tag{15}$$

A common value for c is c = 1.5. This method is less sensitive to outliers than the ESD and it is well suited for asymmetric distributions since it does not depend on the center of the data (Pearson (2005)), but it declares as outliers many nominal observations determined as non-outliers by a human expert. The simplified algorithm has been implemented as follows:

Algorithm 10 Quantile outlier filter

```
procedure CLEANQUANTILE(variables)
    c ← 1.5
    outlierList ← []                                   ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        Q1 ← quantile(variable[:], 25%)
        Q3 ← quantile(variable[:], 75%)
        IQR ← Q3 − Q1
        for all entry,entID in variable[:] do :
            if entry < (Q1 − c ∗ IQR) or (Q3 + c ∗ IQR) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

4.5. Hampel identifier

The Hampel identifier is based on two robust measures of location and scale, the median and the MAD (median of the absolute deviations), respectively. Observations too far from the median of the data with respect to their MAD are declared to be outliers (Christophe Leys (2013)). Again, a proportion factor k will modulate how to calculate that distance. In this case, this factor is derived by using the inverse of the Gaussian cumulative distribution function (Φ^{-1}) calculated on the 75% confidence interval which takes the area until the quantile Q_3:

$$k = 1/\left(\Phi^{-1}(3/4)\right) \approx 1.4826. \tag{16}$$

The accepted range for the detection procedure is calculated as follows:

$$(\hat{X} - (k * MAD)) < x_i < (\hat{X} + (k * MAD)), \tag{17}$$

where:

x_i *: is the i entry from a single variable X*
$\hat{X}$ *: is the median of single variable X*
k *: is the constant scale factor calculated as in equation (24)*
MAD *: is the median absolute deviation calculated as in equation (26)*

and the MAD is calculated as follows::

$$MAD = median(|x_i - \hat{X}|). \tag{18}$$

The simplified algorithm has been implemented as follows:

Algorithm 11 Hampel outlier filter

```
procedure CLEANHAMPEL(variables)
    k ← 1.4826
    outlierList ← []                                   ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        median ← median(variable[:])
        MAD ← mad(variable[:])
        for all entry,entID in variable[:] do :
            if entry < (median − k ∗ MAD) or (median + k ∗ MAD) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is marked as outlier,value removed
            end if
        end for
    end for
end procedure
```

4.6. Evaluation

The evaluation of the methods will be done with the datasets of the wind farms in table 2. The filtering methods will be applied on the train datasets and the models will be tested on the (unknown) test dataset respecting the original time arrangement. All the experiments will be performed using the same target variable, which is the most important one that indicates the temperature of the wind turbine gearbox system. Modeling the relationship between the selected inputs and this target variable, the failures could be detected because a significant difference will exist between the real and the modeled result.

To quantify the effect of the filtering step, a set of indicators gathered from the results from the models are evaluated. One of the most effective method to evaluate the impact of such filters on machine learning algorithms is to implement a normality model based on Partial Least Squares (PLS) (Wold (2001)), which can be evaluated using the mean squared error (MSE). The model is computed using the same train dataset with and without outliers and then both models will be applied to the test dataset. Apart from the MSE, the scatter plots of the real and estimated values are used to compute the best regression line that fits to them. Ideally, if there is a perfect relation between the points, a line with a gradient of 45° is obtained.

5. Results

5.1. Results summary

In table 6 and figure 14, a summary of the experiments performed is presented. The considered parks and wind turbines are listed in table 2. For lack of space, the list only contains some wind turbines of each park. For the sake of clarity, an MSE ratio is calculated as the quotient of MSE values obtained by filtering and without filtering. Therefore the PLS model is generated and evaluated, the quotient will be >1 if the filtering strategy doesn't work appropriately. On the contrary, if the filtering strategy works as expected, the ratio will be <1 (these cases are indicated in italics in table 6).

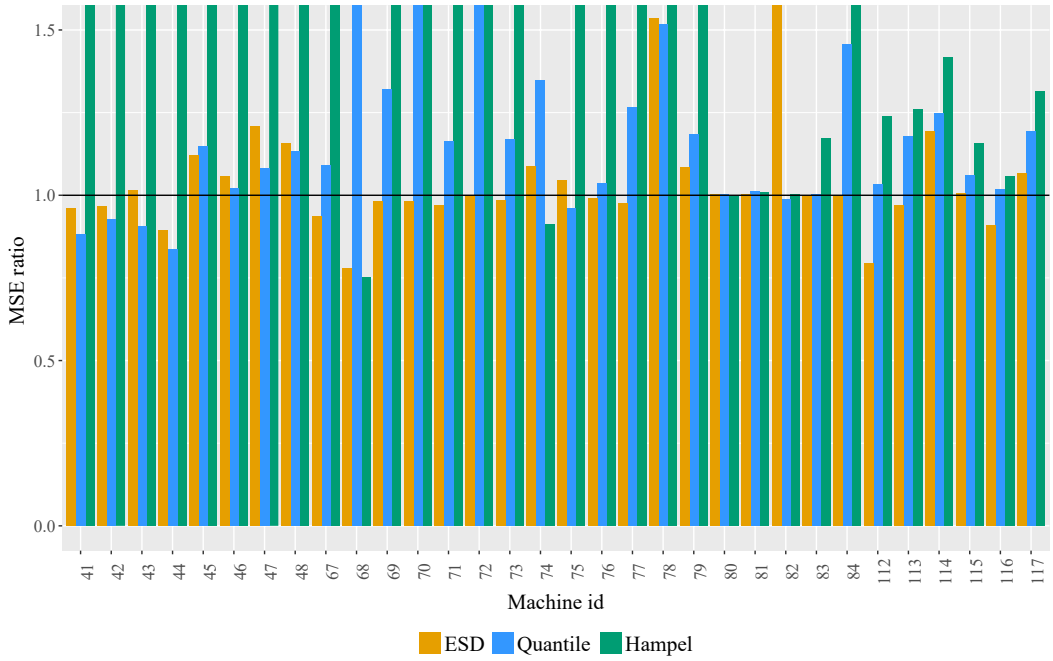

Figure 10: Result summary bar plot. A MSE ratio of one means no improvement.

As can be seen in table 6, values are usually >1. Note that the results are much worse using quantile or Hampel filtering than without filtering. (i.e.: MSE ratios are ≫1). Only the ESD filter seems to be interesting in some cases, but even in these cases, corresponding to the ratio <1, the difference in MSE is small.

Analyzing in detail all the cases reported in table 6, in 17 over 32 cases, the ESD filtering method is useful when testing the model representing 53%

Model	Machine id	ESD filter MSE Ratio	Quantile filter MSE Ratio	Hampel filter MSE Ratio
Fuhrlander FL2500	80	1,002	1,002	*0,998*
	81	1,002	1,011	1,008
	82	*0,999*	*0,987*	1,002
	83	1,000	1,002	1,171
	84	*0,996*	1,458	44,838
Vestas V90 wfa1	67	*0,935*	1,090	5,274
	68	*0,780*	4,640	*0,753*
	69	*0,983*	1,319	1,868
	70	*0,983*	1,604	8,317
	71	*0,971*	1,162	8,253
	72	*0,996*	1,851	12,410
	73	*0,985*	1,168	6,892
	74	1,088	1,347	*0,912*
	75	1,046	*0,959*	5,505
	76	*0,992*	1,037	4,813
	77	*0,975*	1,267	5,801
	78	1,536	1,518	8,010
	79	1,085	1,185	4,826
Siemens Izar 55/1300	41	*0,961*	*0,882*	210,940
	42	*0,966*	*0,928*	307,942
	43	1,015	*0,905*	250,313
	44	*0,895*	*0,835*	242,414
	45	1,121	1,147	172,567
	46	1,057	1,022	218,819
	47	1,208	1,080	280,106
	48	1,158	1,133	157,796
Vestas V90 wfa2	112	*0,795*	1,033	1,239
	113	*0,971*	1,179	1,260
	114	1,193	1,247	1,418
	115	1,007	1,060	1,156
	116	*0,908*	1,019	1,057
	117	1,065	1,193	1,315

Table 5: Result summary

of the cases. Even if that seems a high number of cases, in all of them the quotient is ≈1 , indicating that the MSE is almost the same when using the filter compared to the original (non-filtered) case. For the quantile filter, only 6 over 32 cases reported a quotient smaller than one. It means that only about 19% of the cases improved results after filtering. Finally, for the Hampel filter only 3 cases over 32 reported a quotient higher than one, i.e.: 9% of the cases.

Computing all the filters analyzed, in 73% of the cases the filtering procedure increased the MSE. Thus, as a rule of thumb, filtering is not a good strategy, and only in very few cases could slightly improve the results by decreasing MSE in the test dataset. According to the experiments carried out, in the case of needing a filter, the best choice would be to use the ESD filter, since it is able to eliminate some outliers that are not relevant nor related to the alarms.

5.2. Detailed results for unfiltered data

In order to better understand how the filtering strategy works, a specific example is detailed in the following sections, first without filtering, to have a baseline reference, and then by introducing the analyzed filtering strategies. The first turbine (named T13) of the first plant, composed by Vestas V90 machines, is selected as an example. An expert determined that the target variable for the model of this turbine is gear_oil_temp_avg, which has the distribution shown in figure 15. The following list shows the input variables selected by the method detailed on subsection 2.2:

- *gear_bearing_temp_avg: Temperature of bearing that holds the rotor with blades.*
- *power_avg: Average power generated*
- *wind_avg: Average wind speed*
- *hydraulic_oil_temp_avg: Temperature of the oil which cool the gearbox.*
- *blades_pitchangle_max: Angle of the wind turbine blades.*
- *blades_bladea_controlvoltage_min: Voltage of the motors which controls the angle of the blades.*

In this particular example, the smallest p-value is for gear_bearing_temp_avg. This is somehow expected because the target variable and this variable are components that are physically closer and in contact by metal parts which transfer heat.

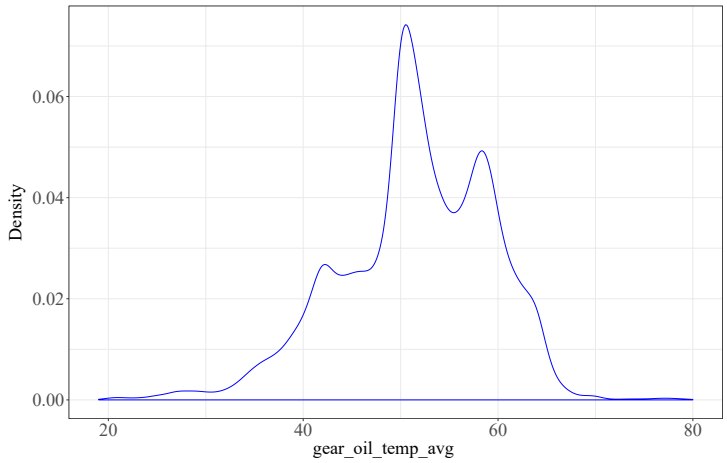

Figure 11: Histogram of target variable gear_oil_temp_avg

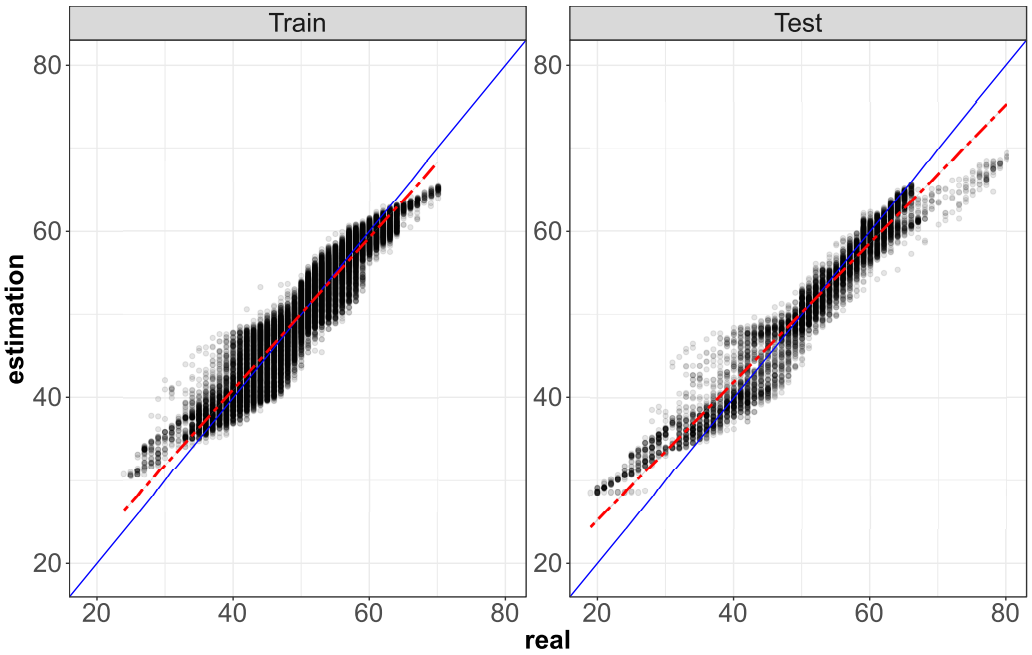

Figure 12: Estimation vs. real value of target variable in train and test. In horizontal axis the real values, while vertical axis the estimated values. The blue line would be the best prediction, the red one is the best fit line of the model prediction.

As a reference, the results of the model without filtering are shown in figure 16 for the train(left) and test(right) datasets. The X axis is the real value of

the target variable and the Y axis is the estimated value from the model. The best possible result is indicated by the 45°blue line and the red line indicates the best fit regression line for the current results, which is slightly leaned with respect to the reference. In this example the obtained gradient has a value of 42.4° with an MSE of 2.0768 for the training dataset, indicating that the model is not estimating all the values perfectly even on the same training dataset. On the test dataset the gradient is 40° with an MSE of 2.612 which is worse than the previous one. This is what it was expected as the model is now dealing with new (unknown) data.

5.3. Detailed results for the ESD filter

With the data being filtered by the ESD filter, many periods of alarm were identified as outliers, as can be seen in figures 17 and 18. Each figure corresponds to a different variable. In all these figures, outliers are in orange color. Violet color corresponds to the values which have been identified as outliers by the algorithms but at the same time are alarms reported by the wind turbine. Values with alarms are indicated in red color. Finally the remaining (non filtered data) are in green color. Two variables are detailed, corresponding to the variables that have the greatest number of alarms identified as outliers. This will reduce the number of alarms that feed theno statistical relevance on the change of its mean on the day when the alarm/failure event is present. The alternative hypothesis H_a defines that a variable presents

The most common value for the threshold t is $t = 3$, which means that all points that deviates 3σ from the mean value will be rejected, considering about 0.3% of the observed data as outliers. This method is very sensitive to distributions that contains many outliers and it will fail with data containing more than 10% of outliers (Pearson (2005)). The ESD algorithm is implemented as follows:

Algorithm 12 ESD outlier filter

```
procedure cleanEsd(variables)
    t ← 3
    for all variable,varID in variables[:,:] do :
        mean ← mean(variable[:])
        σ ← sd(variable[:])
        for all entry,entID in variable[:] do :
            if entry < mean − (t ∗ σ) or mean + (t ∗ σ) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

5.4. Quantile filter

Another commonly used method is based on the distance of the points being above of the third quartile or below of the first quartile. These quartile values determine the acceptable range of the values following equation 22:

$$(Q_1 - (c * IQR)) < x_i < (Q_3 + (c * IQR)), \tag{19}$$

where:

x_i *: is the* i *entry from a single variable* X
Q_1, Q_3 *: are the first and third quartile of the current variable* X
IQR *: is the interquartile as in equation (23)*
c *: is the number of IQR*

$$IQR = (Q_3 - Q_1). \tag{20}$$

A common value for c *is* $c = 1.5$*. This method is less sensitive to outliers than the ESD and it is well suited for asymmetric distributions since it does not depend on the center of the data (Pearson (2005)), but it declares as outliers many nominal observations determined as non-outliers by a human expert. The simplified algorithm has been implemented as follows:*

Algorithm 13 Quantile outlier filter

```
procedure cleanQuantile(variables)
    c ← 1.5
    outlierList ← []                                  ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        Q1 ← quantile(variable[:], 25%)
        Q3 ← quantile(variable[:], 75%)
        IQR ← Q3 − Q1
        for all entry,entID in variable[:] do :
            if entry < (Q1 − c ∗ IQR) or (Q3 + c ∗ IQR) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                  ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

5.5. Hampel identifier

The Hampel identifier is based on two robust measures of location and scale, the median and the MAD (median of the absolute deviations), respectively. Observations too far from the median of the data with respect to their MAD are declared to be outliers (Christophe Leys (2013)). Again, a proportion factor k will modulate how to calculate that distance. In this case, this factor is derived by using the inverse of the Gaussian cumulative distribution function (Φ^{-1})a statistically relevant difference in its mean value on the day when the alarm/failure event is present. The interval of confidence is defined at 95% which determines a p-value of 0.05. Any variable that has a p-value smaller than 0.05 is considered as a possible input variable for the model. All the considered candidates are sort from the lowest to the highest p-value, then the first six variables are selected to analyze them. In all the analyzed parks, using more than six variables does not significantly increase the model performance. On the contrary, computational time also increases when more than six variables are used. Therefore, the number of variables is set at six, which is a good trade-off between performance and computational time. As shown in other works (A. Zaher (2009), Meik Schlechtingen (2011), Michael Wilkinson (2014)) it is common to use the minimum number of variables in order to optimize the results while minimizing the complexity of the system. A diagram of the process is shown in figure **??**

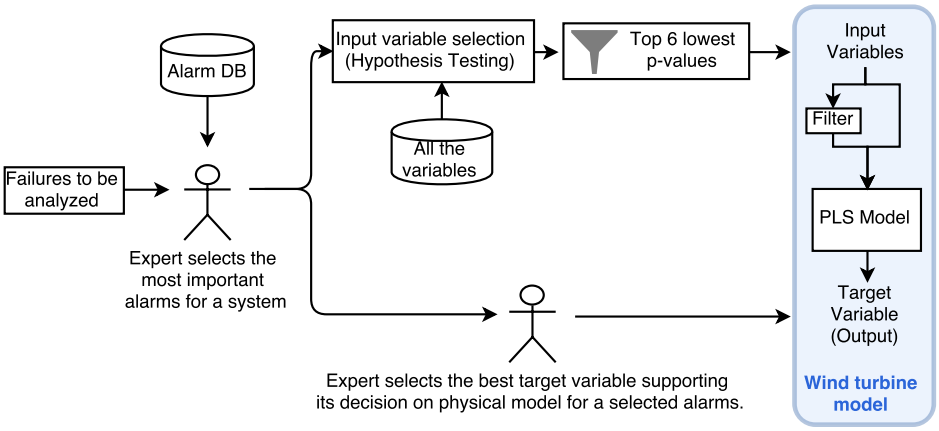

Figure 13: Flowchart of the process.

5.6. ESD filter

Extreme Studentized Deviate test (ESD) is a statistical test to detect outliers in an univariate dataset that have a normally distributed population. ESD defines that any point further from t standard deviations of the mean is an outlier. As shown in equation 21, any value falling outside the interval is considered an outlier:

$$(\mu - (t * \sigma)) < x_i < (\mu + (t * \sigma)), \tag{21}$$

where:

x_i *: is the* i *entry from a single variable* X
μ *: is the mean of the current variable* X
t *: is the number of standard deviations*
σ *: is the standard deviation of a single variable* X

The most common value for the threshold t *is* $t = 3$*, which means that all points that deviates* 3σ *from the mean value will be rejected, considering about 0.3% of the observed data as outliers. This method is very sensitive to distributions that contains many outliers and it will fail with data containing more than 10% of outliers (Pearson (2005)). The ESD algorithm is implemented as follows:*

Algorithm 14 ESD outlier filter

```
procedure cleanEsd(variables)
    t ← 3
    for all variable,varID in variables[:,:] do :
        mean ← mean(variable[:])
        σ ← sd(variable[:])
        for all entry,entID in variable[:] do :
            if entry < mean − (t ∗ σ) or mean + (t ∗ σ) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

5.7. Quantile filter

Another commonly used method is based on the distance of the points being above of the third quartile or below of the first quartile. These quartile values determine the acceptable range of the values following equation 22:

$$(Q_1 - (c * IQR)) < x_i < (Q_3 + (c * IQR)), \tag{22}$$

where:

x_i *: is the* i *entry from a single variable* X
Q_1, Q_3 *: are the first and third quartile of the current variable* X
IQR *: is the interquartile as in equation (23)*
c *: is the number of IQR*

$$IQR = (Q_3 - Q_1). \tag{23}$$

A common value for c *is* $c = 1.5$*. This method is less sensitive to outliers than the ESD and it is well suited for asymmetric distributions since it does not depend on the center of the data (Pearson (2005)), but it declares as outliers many nominal observations determined as non-outliers by a human expert. The simplified algorithm has been implemented as follows:*

Algorithm 15 Quantile outlier filter

```
procedure cleanQuantile(variables)
    c ← 1.5
    outlierList ← []                                  ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        Q1 ← quantile(variable[:], 25%)
        Q3 ← quantile(variable[:], 75%)
        IQR ← Q3 − Q1
        for all entry,entID in variable[:] do :
            if entry < (Q1 − c ∗ IQR) or (Q3 + c ∗ IQR) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                  ▷ is labeled as an outlier, value removed
            end if
        end for
    end for
end procedure
```

5.8. Hampel identifier

The Hampel identifier is based on two robust measures of location and scale, the median and the MAD (median of the absolute deviations), respectively. Observations too far from the median of the data with respect to their MAD are declared to be outliers (Christophe Leys (2013)). Again, a proportion factor k will modulate how to calculate that distance. In this case, this factor is derived by using the inverse of the Gaussian cumulative distribution function (Φ^{-1}) calculated on the 75% confidence interval which takes the area until the quantile Q_3:

$$k = 1/\left(\Phi^{-1}(3/4)\right) \approx 1.4826. \tag{24}$$

The accepted range for the detection procedure is calculated as follows:

$$(\hat{X} - (k * MAD)) < x_i < (\hat{X} + (k * MAD)), \tag{25}$$

where:

x_i *: is the i entry from a single variable X*
$\hat{X}$ *: is the median of single variable X*
k *: is the constant scale factor calculated as in equation (24)*
MAD *: is the median absolute deviation calculated as in equation (26)*

and the MAD is calculated as follows::

$$MAD = median(|x_i - \hat{X}|). \tag{26}$$

The simplified algorithm has been implemented as follows:

Algorithm 16 Hampel outlier filter

```
procedure cleanHampel(variables)
    k ← 1.4826
    outlierList ← []                                   ▷ The outlier list is initialized
    for all variable,varID in variables[:,:] do :
        median ← median(variable[:])
        MAD ← mad(variable[:])
        for all entry,entID in variable[:] do :
            if entry < (median − k ∗ MAD) or (median + k ∗ MAD) < entry then
                outlierList[varID, entID] ← entry          ▷ save the outlier for analysis
                entry ← NULL/NAN                           ▷ is marked as outlier,value removed
            end if
        end for
    end for
end procedure
```

5.9. Evaluation

The evaluation of the methods will be done with the datasets of the wind farms in table 2. The filtering methods will be applied on the train datasets and the models will be tested on the (unknown) test dataset respecting the original time arrangement. All the experiments will be performed using the same target variable, which is the most important one that indicates the temperature of the wind turbine gearbox system. Modeling the relationship between the selected inputs and this target variable, the failures could be detected because a significant difference will exist between the real and the modeled result.

To quantify the effect of the filtering step, a set of indicators gathered from the results from the models are evaluated. One of the most effective method to evaluate the impact of such filters on machine learning algorithms is to implement a normality model based on Partial Least Squares (PLS) (Wold (2001)), which can be evaluated using the mean squared error (MSE). The model is computed using the same train dataset with and without outliers and then both models will be applied to the test dataset. Apart from the MSE, the scatter plots of the real and estimated values are used to compute the best regression line that fits to them. Ideally, if there is a perfect relation between the points, a line with a gradient of 45° is obtained.

6. Results

6.1. Results summary

In table 6 and figure 14, a summary of the experiments performed is presented. The considered parks and wind turbines are listed in table 2. For lack

of space, the list only contains some wind turbines of each park. For the sake of clarity, an MSE ratio is calculated as the quotient of MSE values obtained by filtering and without filtering. Therefore the PLS model is generated and evaluated, the quotient will be >1 if the filtering strategy doesn't work appropriately. On the contrary, if the filtering strategy works as expected, the ratio will be <1 (these cases are indicated in italics in table 6).

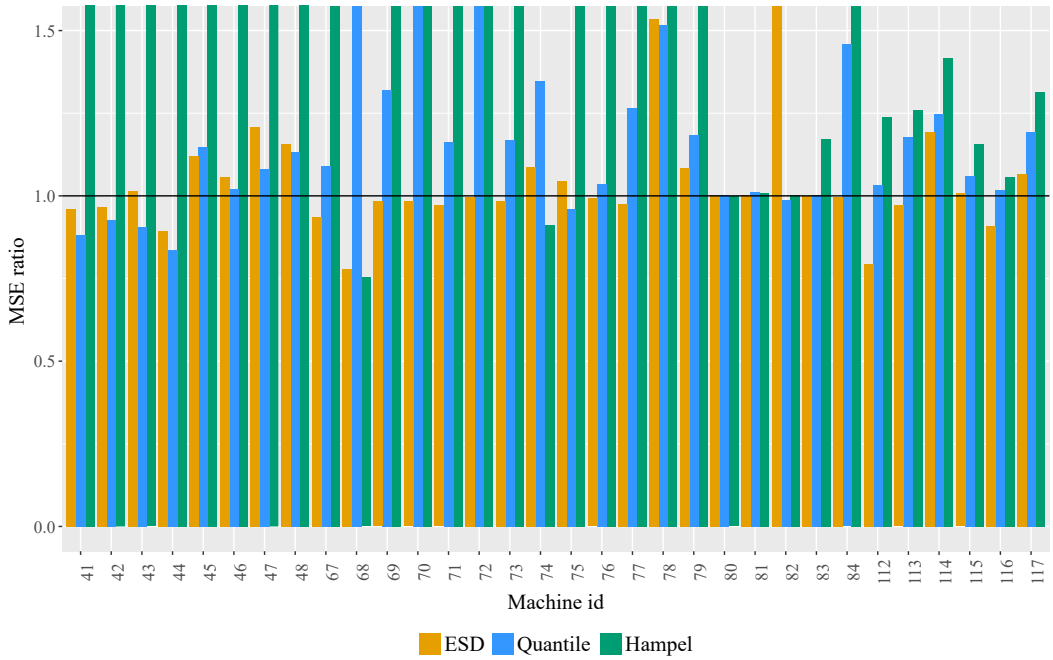

Figure 14: Result summary bar plot. A MSE ratio of one means no improvement.

As can be seen in table 6, values are usually >1. Note that the results are much worse using quantile or Hampel filtering than without filtering. (i.e.: MSE ratios are ≫1). Only the ESD filter seems to be interesting in some cases, but even in these cases, corresponding to the ratio <1, the difference in MSE is small.

Analyzing in detail all the cases reported in table 6, in 17 over 32 cases, the ESD filtering method is useful when testing the model representing 53% of the cases. Even if that seems a high number of cases, in all of them the quotient is ≈1 , indicating that the MSE is almost the same when using the filter compared to the original (non-filtered) case. For the quantile filter, only 6 over 32 cases reported a quotient smaller than one. It means that

Model	Machine id	ESD filter MSE Ratio	Quantile filter MSE Ratio	Hampel filter MSE Ratio
Fuhrlander FL2500	80	1,002	1,002	*0,998*
	81	1,002	1,011	1,008
	82	*0,999*	*0,987*	1,002
	83	1,000	1,002	1,171
	84	*0,996*	1,458	44,838
Vestas V90 wfa1	67	*0,935*	1,090	5,274
	68	*0,780*	4,640	*0,753*
	69	*0,983*	1,319	1,868
	70	*0,983*	1,604	8,317
	71	*0,971*	1,162	8,253
	72	*0,996*	1,851	12,410
	73	*0,985*	1,168	6,892
	74	1,088	1,347	*0,912*
	75	1,046	*0,959*	5,505
	76	*0,992*	1,037	4,813
	77	*0,975*	1,267	5,801
	78	1,536	1,518	8,010
	79	1,085	1,185	4,826
Siemens Izar 55/1300	41	*0,961*	*0,882*	210,940
	42	*0,966*	*0,928*	307,942
	43	1,015	*0,905*	250,313
	44	*0,895*	*0,835*	242,414
	45	1,121	1,147	172,567
	46	1,057	1,022	218,819
	47	1,208	1,080	280,106
	48	1,158	1,133	157,796
Vestas V90 wfa2	112	*0,795*	1,033	1,239
	113	*0,971*	1,179	1,260
	114	1,193	1,247	1,418
	115	1,007	1,060	1,156
	116	*0,908*	1,019	1,057
	117	1,065	1,193	1,315

Table 6: Result summary

only about 19% of the cases improved results after filtering. Finally, for the Hampel filter only 3 cases over 32 reported a quotient higher than one, i.e.: 9% of the cases.

Computing all the filters analyzed, in 73% of the cases the filtering procedure increased the MSE. Thus, as a rule of thumb, filtering is not a good strategy, and only in very few cases could slightly improve the results by decreasing MSE in the test dataset. According to the experiments carried out, in the case of needing a filter, the best choice would be to use the ESD filter, since it is able to eliminate some outliers that are not relevant nor related to the alarms.

6.2. Detailed results for unfiltered data

In order to better understand how the filtering strategy works, a specific example is detailed in the following sections, first without filtering, to have a baseline reference, and then by introducing the analyzed filtering strategies. The first turbine (named T13) of the first plant, composed by Vestas V90 machines, is selected as an example. An expert determined that the target variable for the model of this turbine is gear_oil_temp_avg, which has the distribution shown in figure 15. The following list shows the input variables selected by the method detailed on subsection 2.2:

- *gear_bearing_temp_avg: Temperature of bearing that holds the rotor with blades.*
- *power_avg: Average power generated*
- *wind_avg: Average wind speed*
- *hydraulic_oil_temp_avg: Temperature of the oil which cool the gearbox.*
- *blades_pitchangle_max: Angle of the wind turbine blades.*
- *blades_bladea_controlvoltage_min: Voltage of the motors which controls the angle of the blades.*

In this particular example, the smallest p-value is for gear_bearing_temp_avg. This is somehow expected because the target variable and this variable are components that are physically closer and in contact by metal parts which transfer heat.

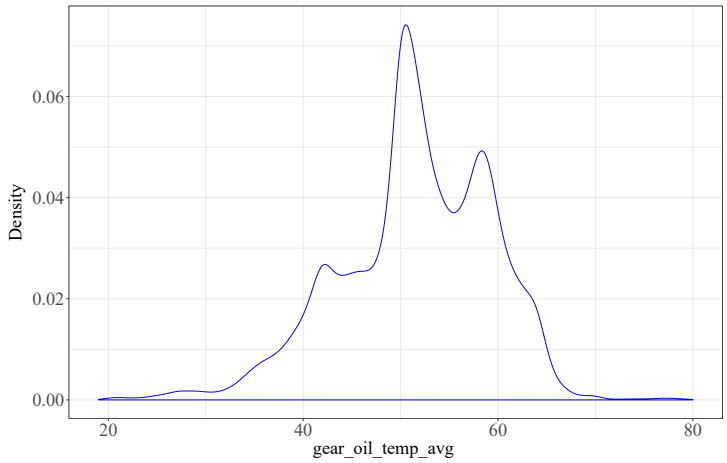

Figure 15: Histogram of target variable gear_oil_temp_avg

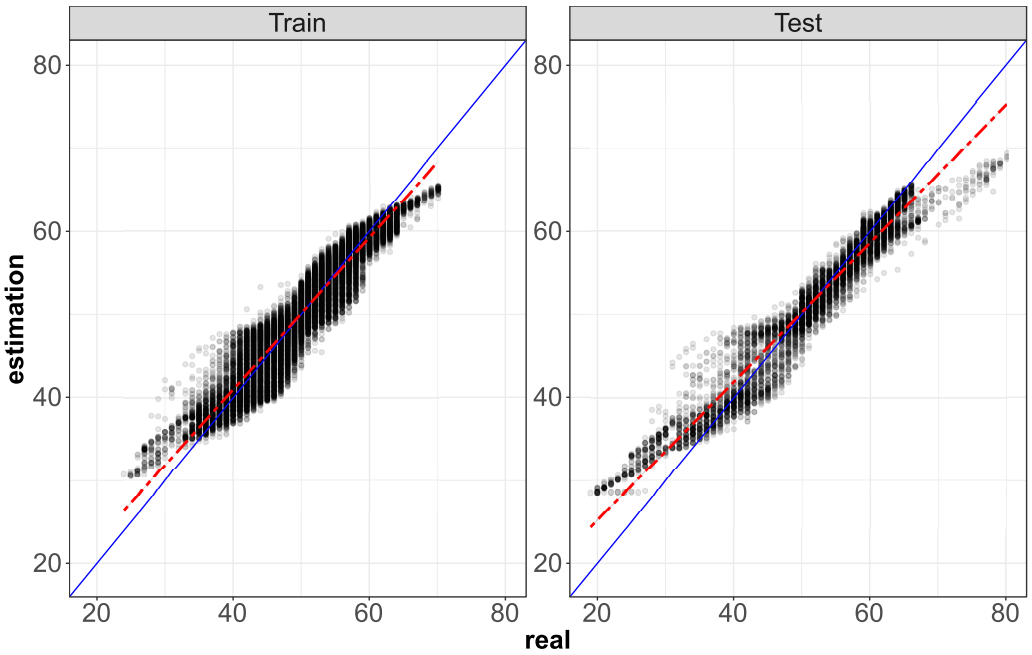

Figure 16: Estimation vs. real value of target variable in train and test. In horizontal axis the real values, while vertical axis the estimated values. The blue line would be the best prediction, the red one is the best fit line of the model prediction.

As a reference, the results of the model without filtering are shown in figure 16 for the train(left) and test(right) datasets. The X axis is the real value of

the target variable and the Y axis is the estimated value from the model. The best possible result is indicated by the 45°blue line and the red line indicates the best fit regression line for the current results, which is slightly leaned with respect to the reference. In this example the obtained gradient has a value of 42.4° with an MSE of 2.0768 for the training dataset, indicating that the model is not estimating all the values perfectly even on the same training dataset. On the test dataset the gradient is 40° with an MSE of 2.612 which is worse than the previous one. This is what it was expected as the model is now dealing with new (unknown) data.

6.3. Detailed results for the ESD filter

With the data being filtered by the ESD filter, many periods of alarm were identified as outliers, as can be seen in figures 17 and 18. Each figure corresponds to a different variable. In all these figures, outliers are in orange color. Violet color corresponds to the values which have been identified as outliers by the algorithms but at the same time are alarms reported by the wind turbine. Values with alarms are indicated in red color. Finally the remaining (non filtered data) are in green color. Two variables are detailed, corresponding to the variables that have the greatest number of alarms identified as outliers. This will reduce the number of alarms that feed the machine learning model and therefore will reduce its prediction capability. The outliers detected by this algorithm represents the 2.1% of the training data.

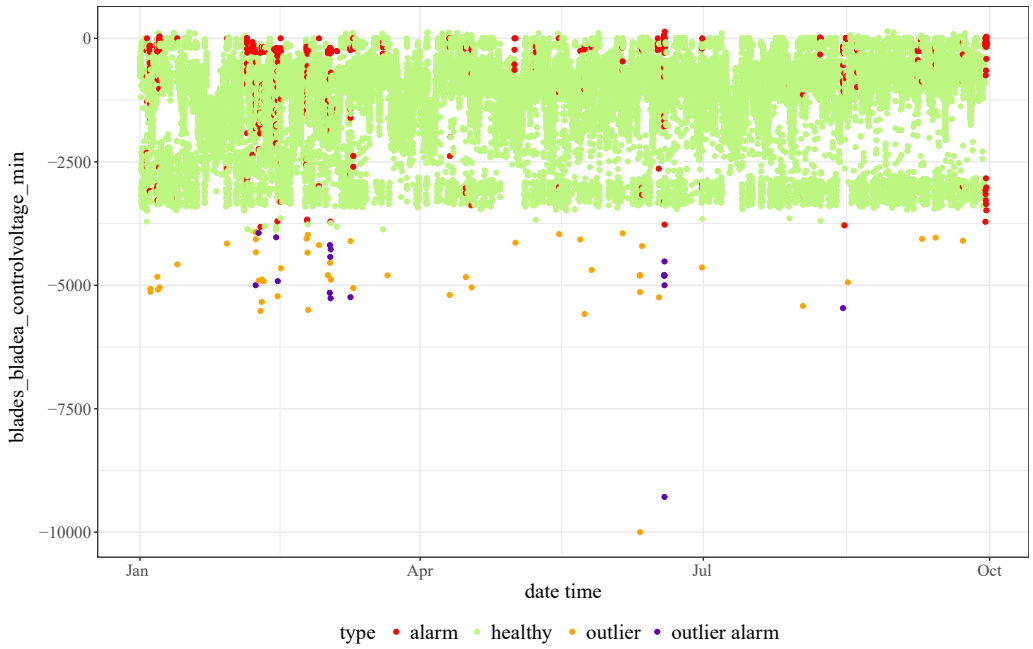

Figure 17: Labeling of points (variable *blade_control_voltage*) on filtered dataset

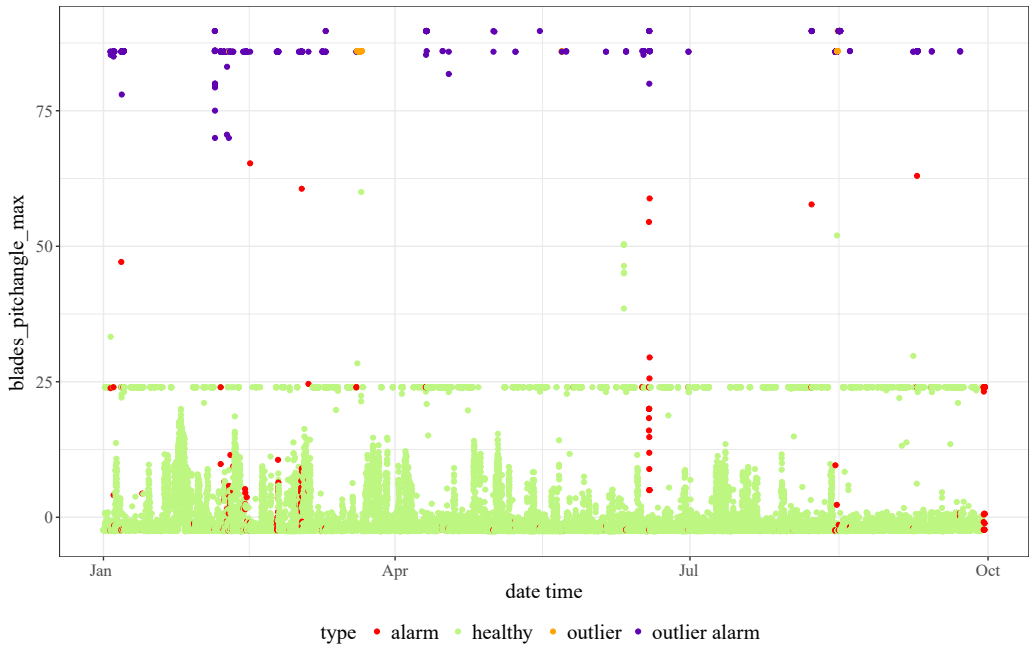

Figure 18: Labeling of points (variable *blade_pitch_angle_max*) on filtered dataset

The impact on the model results is presented in figure 19 which reveals an increase of the performance on the train dataset (left) filtering the outliers: MSE error decreases from 2.0768 to 1.963 and the slope increases from 42.4° to 42.5° . But on the other side, when the model is tested with the test dataset (right), the MSE error increases from 2.612 to 2.836 which means a worse prediction capability. Concerning the slope of the regression line, even if the gradient is almost the same, there is a new small region of points far from the diagonal line indicating that the model is behaving worse.

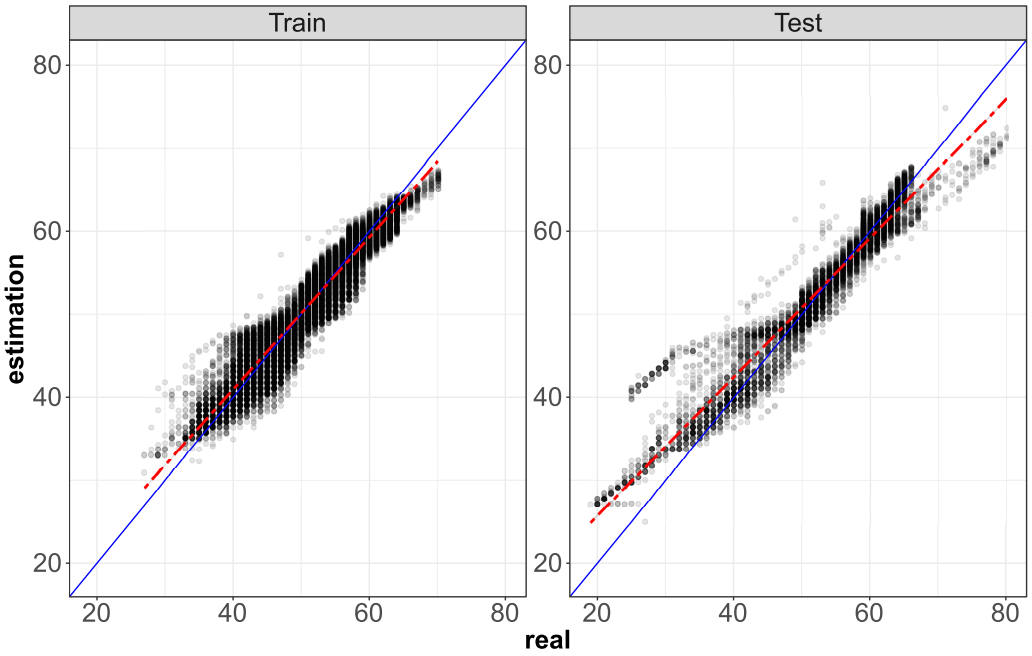

Figure 19: Estimation vs. real value of target variable in train and test. In horizontal axis the real values, while vertical axis the estimated values. The blue line would be the best prediction, the red one is the best fit line of the model prediction.

6.4. Detailed Results for quantile filter

Using the same procedure as in previous filtering strategy, the effects of the quantile filter is tested on the data with the highest alarm periods labeled as outliers see figures 20 and 21. Following the same color coding as in the ESD case, filtering will reduce the number of alarms that feed the model and therefore it will reduce its prediction capability. The outliers detected by this algorithm represent the 20.8% of the training data.

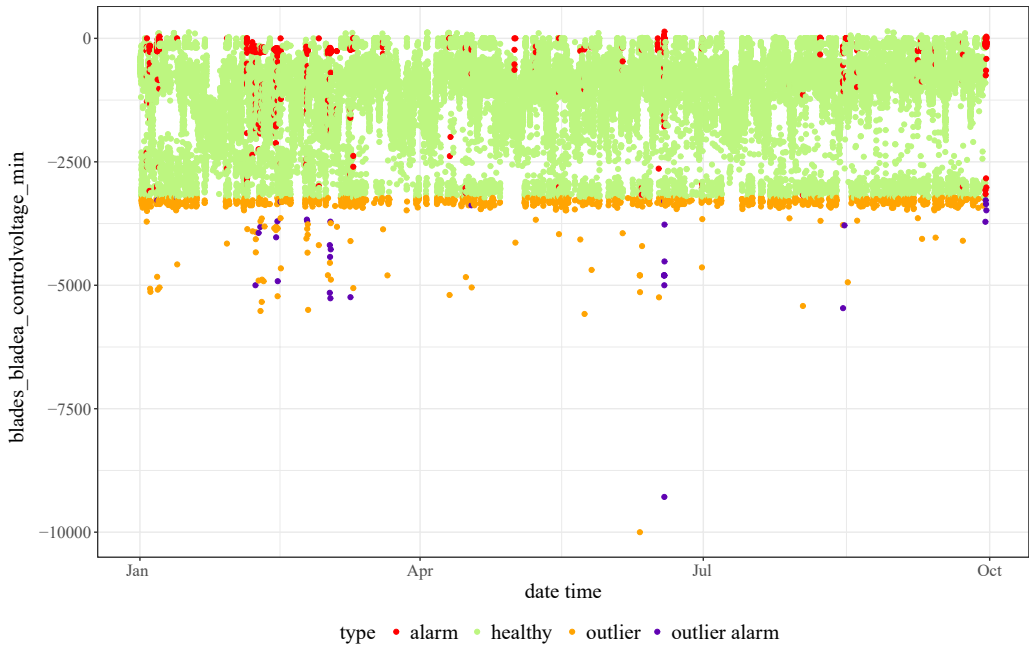

Figure 20: Labeling of points (variable *blade_control_voltage*) on filtered dataset

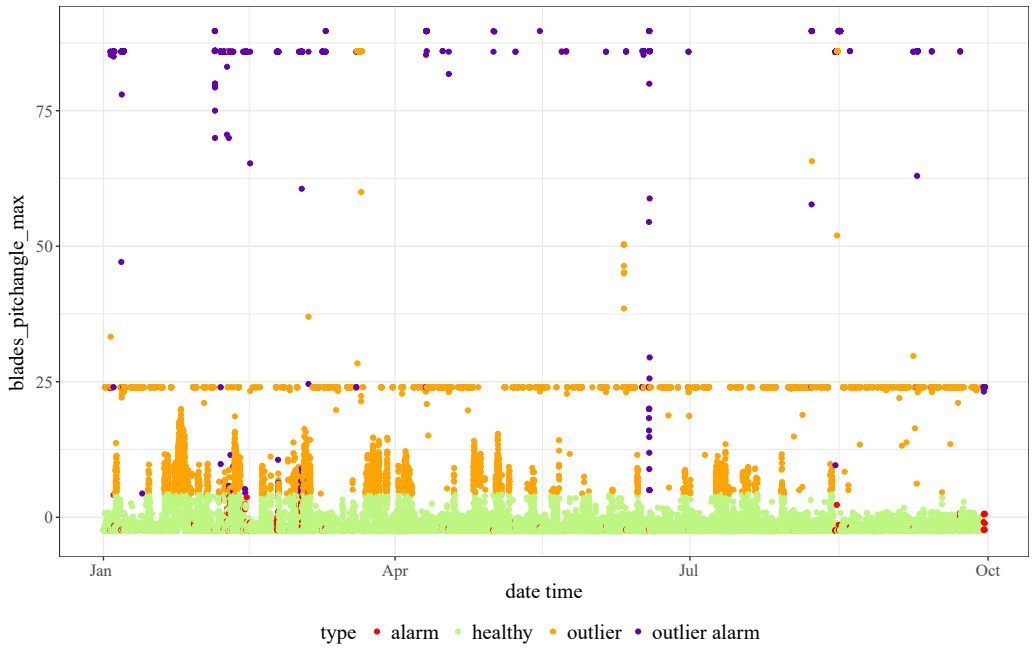

Figure 21: Labeling of points (variable *blade_pitch_angle_max*) on filtered dataset

Results in figure 22 show an increase of performance in the training dataset (right) with MSE decreasing from 2.0768 to 1.893. This value is smaller than the one obtained with the ESD filter due to robustness of the quartile to the outliers. On the contrary, results on the test dataset (right) reveal a higher increase of the MSE from 2.612 to 3.096 and the plot of estimation vs. real values indicates a decrease in the angle of the linear regression from 40° to 39.7°, which means that the model generalization performance is worse than the ESD. Some holes on the region between 50-60 can be observed due to the removal of values considered as outliers in the input variables.

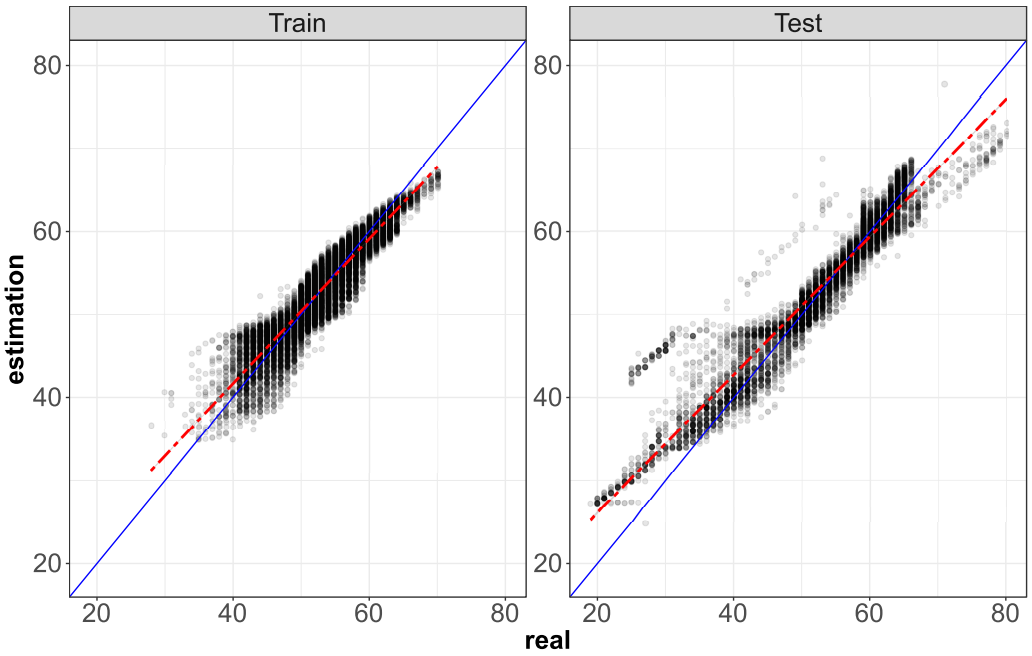

Figure 22: Estimation vs. real value of target variable in train and test. In horizontal axis the real values, while vertical axis the estimated values. The blue line would be the best prediction, the red one is the best fit line of the model prediction.

6.5. Detailed Results for Hampel identifier

Finally the third filtering system is analyzed in the same way as the previous ones. Figures with the results, using the same kind of representations, are shown in figures 23 and 24 for each variable. Again, the two variables which present the highest number of alarms identified as outliers are depicted. The outliers detected by this algorithm represents the 32.2% of the training data, taking into account all the variables.

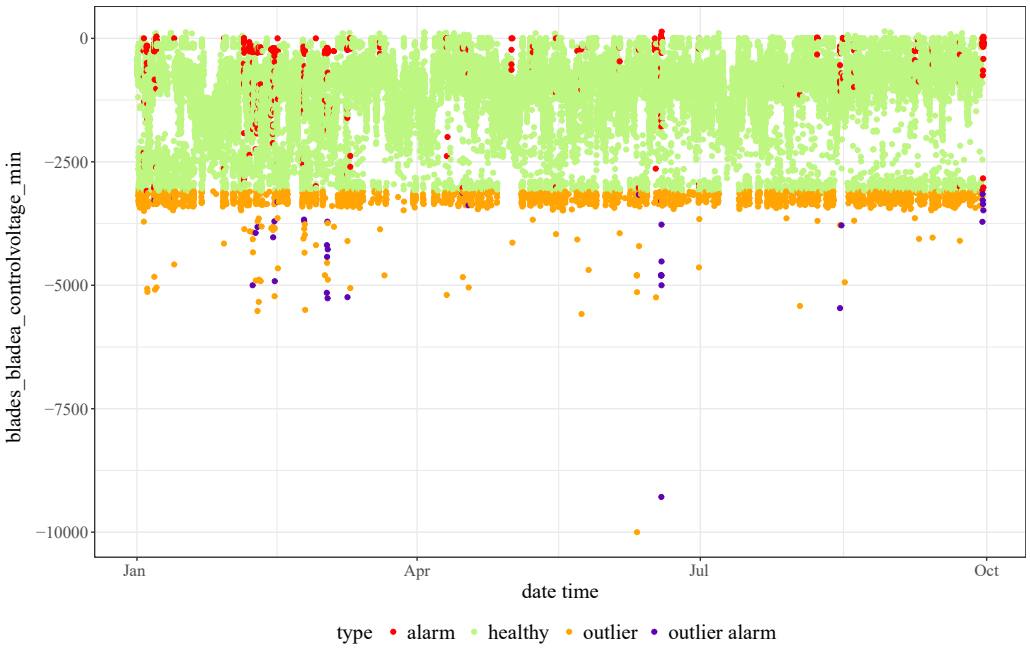

Figure 23: Labeling of points (variable *blade_control_voltage*) on filtered dataset

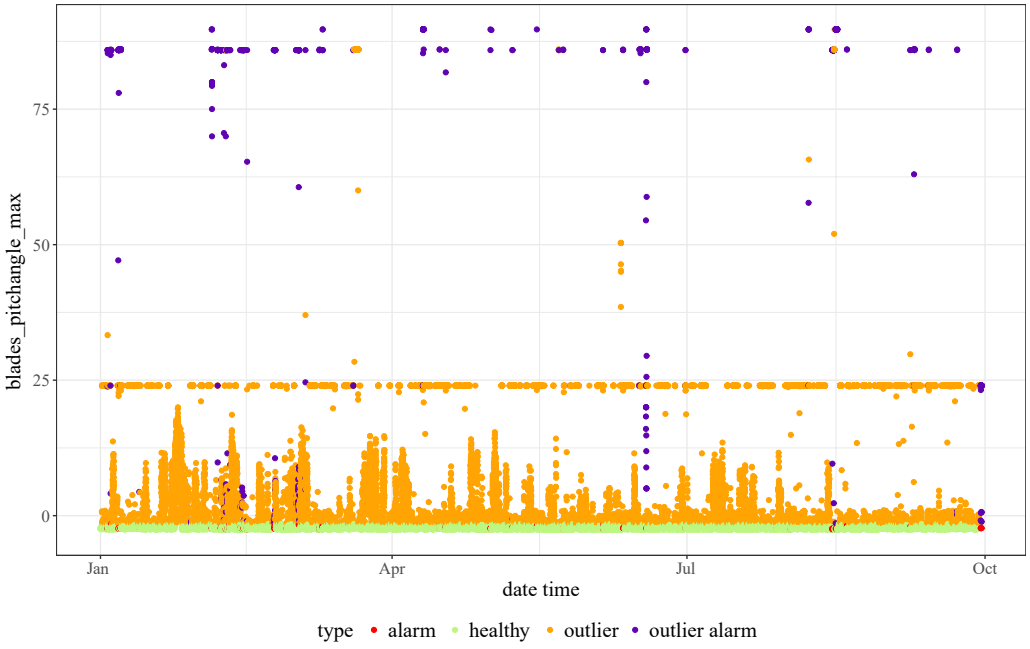

Figure 24: Labeling of points (variable *blade_pitch_angle_max*) on filtered dataset

Figure 25 shows an increase on performance using the training dataset (left). The results reveals an even higher decrease of the MSE error from 2.0768 to 1.816 and 42.4° to 40.3° which is a better regression line for estimation vs real value. But the analysis of the test dataset (right) reveals it as the worst of all the filtering methods with a MSE of 12.6°. The plot of the results shows clear regions of values that were removed by the filter and therefore these points are missing from the input variable when estimating the target variable. The angle of the linear regression line is about 17° which is clearly far from the theoretical one.

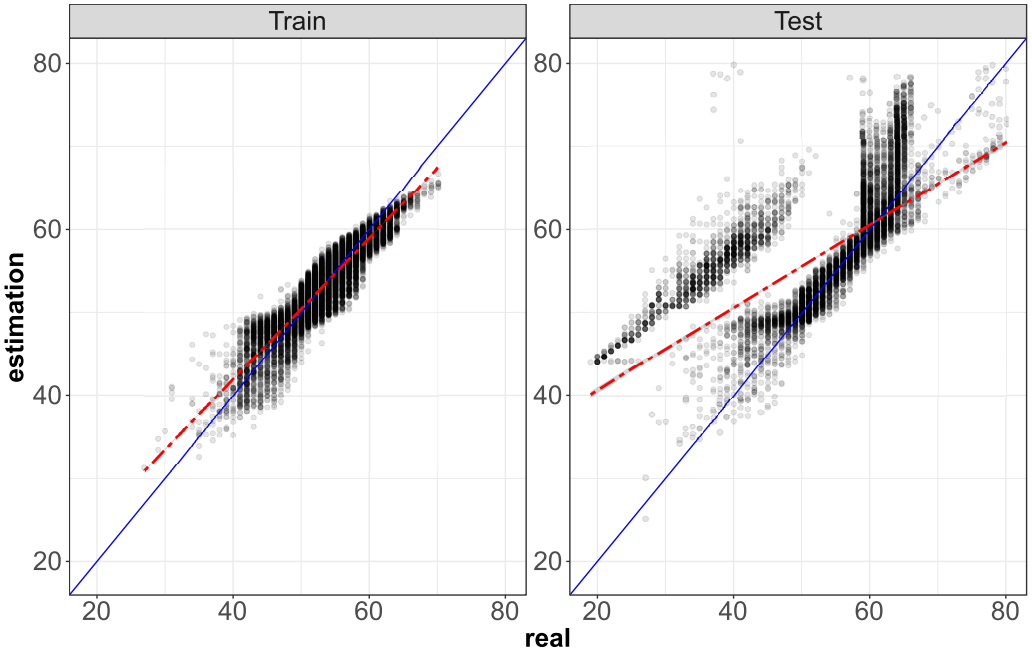

Figure 25: Estimation vs. real value of target variable in train and test. In horizontal axis, the real values while vertical axis, the estimated values. A blue line will be the best prediction, the red one is the best fit line of the model prediction.

7. Conclusions

This paper has explored several methods for outliers detection and compared their performance against the non-filtered data.

Experimental results demonstrate that using filters to eliminate outliers can decrease error in the train data set but unfortunately increases the error in the test data set. This is because many outliers were failure states of

the wind turbine, which are less frequent operation modes, as indicated in section 6 with the variables affected by each filtering method. Filtered data performance could generate good results with cross-validation on the same train dataset, which is already filtered and the values of each variable are closer, hence easier to model, but the performance is reduced using new data in all the cases due to the poor generalization capability by removing failure patterns that are present on the future datasets. In this case the performance of prognosis models over SCADA data performs best on new data with non-filtered train datasets. The effect of removing points labeled as outliers but that in fact contributes to identifying alarm states can be observed in figures 17, 18, 20, 21, 23 and 24 (red points).

In the light of these results, systematically filtering outliers with the methods described before has to be reconsidered to derive better prognosis models. The proposed strategy for the filtering step is to manually define ranges (absolute and relative) for the values of the variables. This is also stated in Gibert et al. (2016), for example, but our strategy differs in the way on how the ranges are defined. In order to define them, each variable has its own operation range specified by the manufacturer or by a human expert taking into account that the absolute range is broad enough to contain values of healthy and damaged wind turbines, excluding real outliers. The relative range is generated at each register entry by computing the range in reference to another variable with which it is closely related to the system. For example, the temperatures of bearings must be between +-30°C from the current ambient temperature assigned at the same register entry. These relatively defined ranges require knowledge of the wind turbineâĂŹs system structure. This filtering strategy removes less real alarms events and therefore the models contain more information about failure patterns. This results in lower management and maintenance costs and will allow us to increase the economic competitiveness of the wind energy with respect to fossil fuels and accelerate the transition towards ecologically sustainable systems.

Acknowledgement

Financial support by the Agency for Management of University and Research Grants (AGAUR) of the Catalan Government to Alejandro Blanco-M. is gratefully acknowledged.

References

A. Zaher, S. M., 2009. Online wind turbine fault detection through automated scada data analysis. Wind Energy 12, 574–593.

Christophe Leys, Olivier Klein, P. B., mar 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median.

Conroy, N., Deane, J. P., O Gallachoir, B. P., 2011. Wind turbine availability: Should it be time or energy based? - A case study in Ireland. Renewable Energy 36 (11), 2967–2971.

Gibert, K., Sànchez-Marrè, M., Izquierdo, J., dec 2016. A survey on preprocessing techniques: Relevant issues in the context of environmental data mining. AI Communications 29 (6), 627–663.

IEC, dec 2006. International standard iec 61400-25-1.

Meik Schlechtingen, I. F. S., 2011. Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. Mechanical Systems and Signal Processing 25, 1849–1875.

Michael Wilkinson, B. D., 2014. Comparison of methods for wind turbine condition monitoring with scada data. Renewable Power Generation, IET 8, 390–397.

OPC Fundation, oct 2016. Opc is the interoperability standard for the secure and reliable exchange of data in the industrial automation space and in other industries.

Pearson, R. K., apr 2005. Mining Imperfect Data: Dealing with Contamination and Incomplete Records. SIAM: Society for Industrial and Applied Mathematics.

Tavner, F. S. P., 2009. Reliability of wind turbine subassemblies. Renewable Power Generation, IET 3, 387–401.

Vestas R+D, 2004. General specification vestas v90 3.0mw. Tech. rep., Vestas Wind Systems.

Wold, S., oct 2001. Pls-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems 58, 109–130.