Available online at www.eccomasproceedia.org
Eccomas Proceedia SMART (2023) 873-885

ECCOMAS
Proceedia

X ECCOMAS Thematic Conference on Smart Structures and Materials
SMART 2023
D.A. Saravanos, A. Benjeddou, N. Chrysochoidis and T. Theodosiou (Eds)

ADVANCED CLUSTERING OF ENVIRONMENTAL AND OPERATIONAL CONDITIONS OF AN OFFSHORE WIND TURBINE USING SELF-ORGANIZING MAPS

Lukas Bonekemper*, Marcel Wiemann*, Holger Huhn♦ and Peter Kraemer*

* University of Siegen, Chair of Mechanics – Structural Health Monitoring
♦ WindMW Service GmbH, Bremerhaven, Germany
lukas.bonekemper@uni-siegen.de; www.uni-siegen.de

Key words: Self-Organizing Map, Clustering, EOC Compensation, SCADA Data, Offshore Wind Turbines.

Abstract. With the ever-increasing number of distributed structures like wind energy plants (WEP) that require regular service and maintenance, the reliable detection of damages becomes more and more important. Analyzing different vibrational quantities is a widely used method for determining possible damages. One difficulty in this process is the compensation of varying environmental and operational conditions (EOC) and their impact on those parameters. In this paper a method using the self-organizing map (SOM) and a subsequent clustering of the SOM is presented and used to define different EOC. The SOM is trained using only SCADA-Data (Supervisory Control and Data Acquisition) of an offshore wind turbine to approximate the environmental conditions, like windspeed and temperature, and the resulting operational parameters of the wind turbine, like power production or pitch angle. Pre-Processing of raw SCADA data is used to create a training data set. This includes removing erroneous datapoints and calculating features from cleaned data. The training of the SOM results in an ordered representation of the training data revealing important relations between the input dimensions. A clustering algorithm should exploit this additional information provided by the SOM to yield a meaningful classification. The clustering uses the k-means algorithm, applied to each weight plane separately instead of the complete, multidimensional set of weights. It is shown that this method leads to an easily interpretable clustering result with grid-like distributed cluster centers, uses the information provided by the SOM more effectively and makes the comparison between different cluster centers more straight-forward.

1 INTRODUCTION

When using vibration-based parameters for damage detection in mechanical structures like wind turbines, a difficult part is to reliably detect if a change in those parameters is due to a damage in the structure or caused by varying environmental conditions and the corresponding reaction of the structure. What is necessary is the definition of a range of values for each vibration-based parameter that represents normal operation under any given EOC. From the

Peer-review under responsibility of the organizing committee of SMART 2023.
doi: 10.7712/150123.9840.450856

continuous range of EOCs, this paper aims to define a finite number of representative states by training a self-organizing map (SOM) with EOC data and a subsequent clustering. The EOC data used in this paper are measurements of SCADA data of an offshore wind turbine that will be used to train the self-organizing map. A SOM can be used to organize and represent high-dimensional data in a 2-dimensional map, which is much easier to analyze and to interpret. This aspect is what can be utilized to define EOC by analyzing the SOM. To define EOC means to find regions on the SOM, where all neurons are relatively constant when compared to neighboring areas. For this, a clustering algorithm is used to merge multiple similar neurons into clusters which in turn represent an EOC. This paper is organized as follows. First, a brief introduction into the algorithm behind the self-organizing map and the classical k-means algorithm is given. This is followed by a description of the necessary pre-processing steps to transform the raw SCADA data into a proper training data set for the SOM. From all channels of the original SCADA data a subset is chosen as relevant for the purpose of this paper. With the prepared training data set some exemplary SOMs are trained to identify training parameters that yield a SOM suited for clustering. The next step is the clustering of the SOM. Here the k-means algorithm is used on each weight plane separately and then superpositioned to the final clustering result. This way, the exploitation of information can be improved when compared to the standard k-means algorithm.

2 SELF-ORGANIZING MAP – ALGORITHM

The Self-Organizing Map is a type of vector quantization algorithm originally presented by Teuvo Kohonen [1]. A SOM is a fixed grid-wise arrangement of neurons, usually in rectangular or hexagonal topology, which constitutes neighborhood relations between those neurons. Each of the neurons has a weight vector defined in the input space given by the training data. The main goal of a SOM is to find a low dimensional representation of high dimensional training data. This is done by training the weights in a way that the weights of neighboring neurons are more similar than those of neurons farther apart. The result is a ordered representation of the training data. Figure 2 shows a simple visualization of the way the SOM maps datapoints onto the lattice. Before the training can start, the neuron weights w_i have to be initialized. The most simple way is to assign random values within the range of the training data to all neurons. It has been shown that the influence of initial neuron states diminishes as training progresses [2], so this can be a viable way. Training becomes easier and more reliable, if the initialization is done using the PCA of the training data. In case of a two-dimensional SOM the two largest principal components span a plane which can be used to initialize neuron weights w_i into an already ordered state [1]. The actual training goes as follows. The algorithm picks a random datapoint of the training data and looks for the best fitting weight. The according neuron is then called the Best-Matching-Unit (BMU). This will be called “presenting” a datapoint. By updating the weight of the BMU towards the datapoint, the algorithm tries to specialize neurons to represent certain datapoints better than others. The weight update can follow two methods, namely the Online-Update Equation (1) and the Batch-Update Equation (2).The Online-Update changes the neuron weights after every single datapoint x_j. In Equation (1) $\alpha(k)$ is the time-dependent learning rate, which expresses how

much influence a new weight update Δw_i will have on the old weights. This factor usually decreases to near-zero at the end of the training. The factor $h_{j,i}(k)$ is the so-called neighborhood function. In both Equations, the index j is replaced by BMU since the BMU is used as a reference. The neighborhood function extends the influence of a datapoint to the neighboring neurons of the BMU and is crucial in creating the globally organized representation of the training data by a SOM. After each training step the width of the neighborhood function is decreased. Usually, a Gauss function is used, but other functions like the Mexican-Hat are also possible choices, see Figure 1, left. Figure 1, right, shows a very basic order of the main training steps.

$$_i(k+1) = \alpha(k) \cdot h_{BMU,i}(k) \cdot \left(x_j - \ _i(k)\right) \tag{1}$$

$$_i(k+1) = \frac{\sum_{j=1}^{M} n_j \cdot h_{BMU,i}(k) \cdot \overline{x_j}}{\sum_{j=1}^{M} h_{BMU,i}(k) \cdot \overline{x_j}} \tag{2}$$

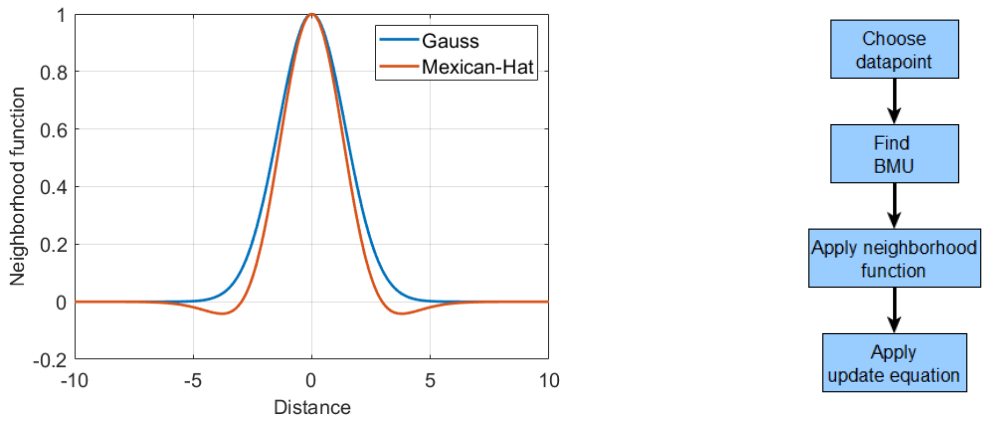

Figure 1: Gaussian and Mexican-Hat as neighborhood function (left) and general order of the SOM algorithm (right)

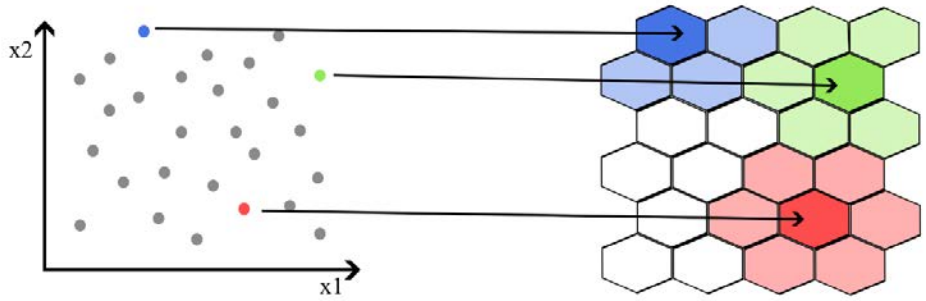

Figure 2: Principle of the main training step. The neighborhood function extends the influence of a training sample beyond the BMU

The Batch-Update in Equation (2) takes all datapoints into account before updating weights. The learning rate is no longer relevant, the factor n_j is the number of datapoints with neuron j

as BMU and $\overline{x_j}$ is their mean. The Batch-Update is also not dependent on the order in which datapoints are presented and has only the width of the neighborhood function as a free parameter. Using PCA-Initialization and Batch-Update, the result of the SOM becomes deterministic. For this reasons the SOMs presented in this paper use the Batch-Update. If all datapoints are presented once, this is called an epoch. Usually complete training consists of multiple epochs to reach convergence of the SOM.

3 K-MEANS CLUSTERING

k-means clustering is a widely used algorithm to cluster data sets into a given number (k) of clusters. It was originally described in [**3**]. The k-means algorithm takes the input data set and positions the k cluster centers in such a way that a loss function given in Equation (3) is minimized. In this context $c_m$is the m-th cluster center and x_n is the n-th datapoint. Equation 3 therefore calculates the sum of squared distances between all cluster centers m and the N_m datapoints belonging to cluster m. A common difficulty of this method is the definition of a correct or reasonable value for k since the true number of clusters is normally unknown. Although there are some methods that aid in determining a value for k, the general problem remains. In case the SOM neighborhood function is set to include only the BMU itself and the Batch-Update of Equation (2) is used, the SOM and k-means algorithm are closely related [1].

$$J = \sum_{m=1}^{k} \sum_{n=1}^{N_m} \|c_m - x_n\|_2^2 \rightarrow \min_{c_m} \tag{3}$$

4 PRE-PROCESSING OF SCADA-DATA

The SCADA-data used in this paper was gathered on a 3.6 MW offshore wind turbine, that is part of a wind farm located in the German North Sea. The data was gathered between October 2021 and September 2022. The examined plant has a hub height of 89 m and a rotor diameter of 120 m. Some additional characteristics of the plants operation are given in Table 1.

Table 1: Some WEP parameters

Hub height	Rotor diameter	Cut-in windspeed	Cut-out windspeed	Average windspeed
89 m	120 m	~3 m/s	~28 m/s	9,5 m/s

The wind turbines SCADA system record several important operational parameters for monitoring and maintenance purposes. For this study, data about temperature, yaw-angle, windspeed, rotor and generator rpm, pitch-angle, blade pressure and power generation were available. Pitch-angles contain values for each blade as well as their mean. Data about power generation contains active, reactive and available power. Here, available power is a calculated quantity defined by the turbines operational characteristics and external parameters like windspeed. In contrast, active power is what is fed into the power grid. All SCADA parameters are sampled with a rate of 1 Hz. To take all basic parameters of wind turbine

operation into account, that are relevant to the purpose of this paper, six SCADA parameters were chosen. This set of variables was chosen to contain the most important environmental parameters (temperature, windspeed), internal parameters that describe the "operational state" (pitch angle, rotor RPM, generator RPM) and the resulting power as the output variable.

The main pre-processing steps are shown in Figure 3.

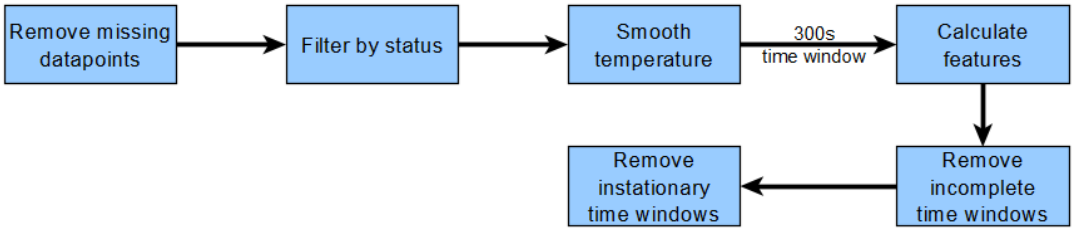

Figure 3: Pre-Processing steps to prepare training data

To deal with the possibility of missing datapoints in the raw data, whenever the value of one channel is missing, all other channels at this point in time are ignored. Additionally, the SCADA system records the "status information" of the wind turbines operation, like normal, too much/too little wind, etc. For this paper the datapoints are filtered by "status information" to use only datapoints associated with normal operation. The temperature is recorded with a quantization of 1 degree Celsius. A coarse quantization can be problematic for training a Self-Organizing Map, therefore this channel is filtered before further processing. The cleaned data set is then divided into non-overlapping time windows of 300s. Of all time windows only those not containing any missing datapoints are used. For the purpose of this paper only the mean values of the 300s-time windows will be used as features for training the Self-Organizing Map. A final step in pre-processing is removing all time windows that represent an instationary state of wind turbine operation. This can be, for example, an accelerating turbine caused by increasing windspeed. A stationary time window is defined by the standard deviation of yaw-angle, generator RPM and rotor RPM. If the standard deviation within one time window is above a given threshold, the respective time window is considered instationary.

5 SELF-ORGANIZING MAP - TRAINING EXAMPLES

Given the described training algorithm and training data set, multiple SOMs were trained in order to find parameters that yield a good training result. The parameters concerns the side length of the SOM's grid, the total number of neurons and training parameters like the number of training epochs. Also the width of the neighborhood function must be set according to the SOM. As already stated, setting these parameters is difficult. Kohonen provided some advice [1] that can be used as a starting point for reasonable adequate parameters. He states that

- the number of neurons should be proportional to the number of datapoints in the training data set. For *n* neurons he proposes $n = 5\sqrt{N}$ with a data set of *N* datapoints.
- For the sidelengths of a rectangular lattice he proposes to use the ratio of the two largest eigenvalues of the autocorrelation matrix of the training data.

This approach is somewhat similar to the PCA initialization. Shaping the lattice similar to the main characteristics of the training data should make the following training a bit easier. Imagine a long and narrow data set. A square-shaped SOM would yield a poor representation of the training data, a SOM of similar shape would be better suited to represent this data set. It should be mentioned that both points are onlysuggestions. They still have to be adapted to the particular situation for optimal results. For the data set described in the previous section the suggested parameters, as well as a set of adapted parameters, are given in Table 2.

Table 2: Different configurations for SOM lattice

Configuration	# Neurons	Sidelength-Ratio	Dim X	Dim Y
(1)	784	16	7	112
(2)	1495	2,85	23	65

Training with ten epochs and an initial neighborhood width of 30 that is decreased to 1 at the end, yields a SOM shown in Figure 4a. Obviously, the size of the SOM is not adequate for this particular dataset. The main characteristic is the long and narrow shape of the wind-power curve, which defined this narrow SOM. Therefore the SOM is not able to represent the relatively wide temperature characteristic adequately.

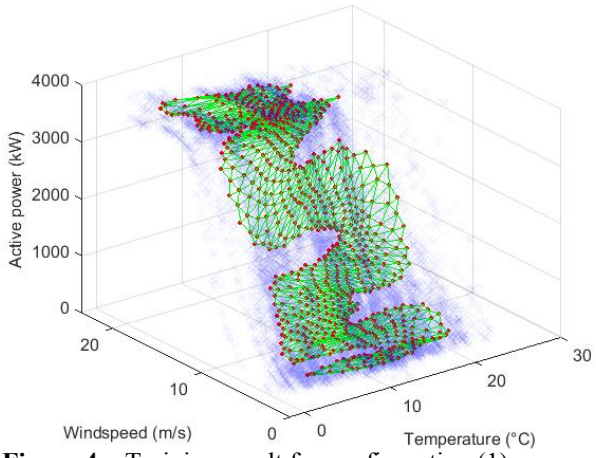

Figure 4a: Training result for configuration (1)

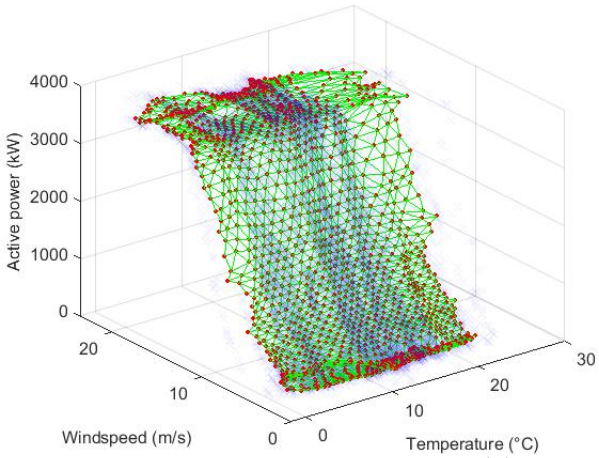

Figure 4b: Training result for configuration (2)

This leads to the curvy shape, where the SOM tries to approximate the complete temperature range. Since the temperature is an important parameter, the parameters were altered to a wider and shorter SOM given by Configuration (2). Additionally, in Figure 4a the well-known

border effect can be observed. The weights of border neurons are always affected by neurons inside of the lattice without any counter effect. This effect is well described by [1].

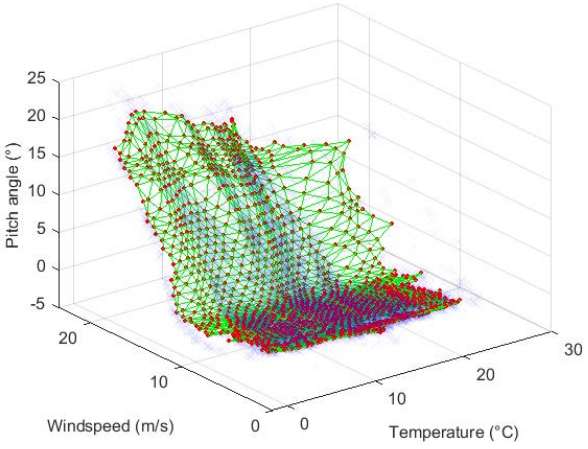

Figure 5a: Configuration (2). View of temperature – windspeed - pitch angle

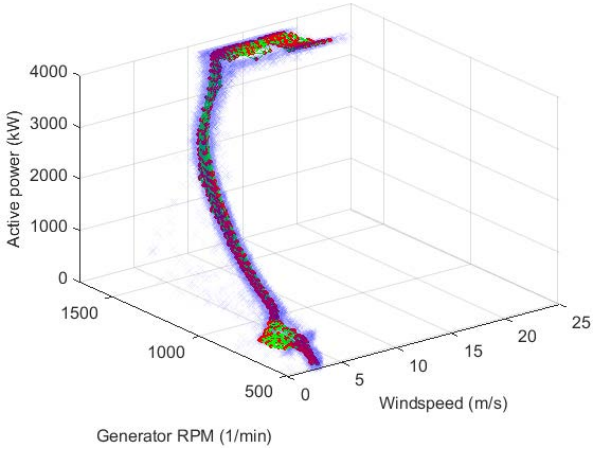

Figure 5b: Configuration (2). View of windspeed – generator rpm – active power

In the SOM of Configuration (2) in Figure 4b the 5% smallest and largest datapoints of temperature and windspeed were presented multiple times during training, thus increasing their relative effect on the neuron weights and improving their representation by the SOM. The overall coverage of datapoints at the border of the input space by the SOM is much improved due to the more frequent presentation of those datapoints. As can be seen by comparing both visualizations in Figure 5, this affected only the temperature and windspeed dimension. Plots including other dimensions like the plot Figure 5b still show the border effect, but to a much lesser degree. For further evaluation and clustering the SOM of Configuration (2) will be used. The weightplanes in Figure 6 allow some interesting insights and illustrate the motivation for the clustering in the following section. The temperature is mainly oriented vertically, while the other variables are more horizontally oriented. This is due to the fact that temperature is not causally related to the other variables, while power and RPM do depend on the windspeed. Especially in the weightplanes of rotor and generator rpm well separated regions are recognizable, but areas of constant weights are clearly present in each weightplane.

The next section explains the clustering approach in this paper in more detail and provides a comparison with the standard k-means-algorithm. For this goal, the structured representation of the data by the SOM gives a good starting point. In theory one could use single neurons as EOC

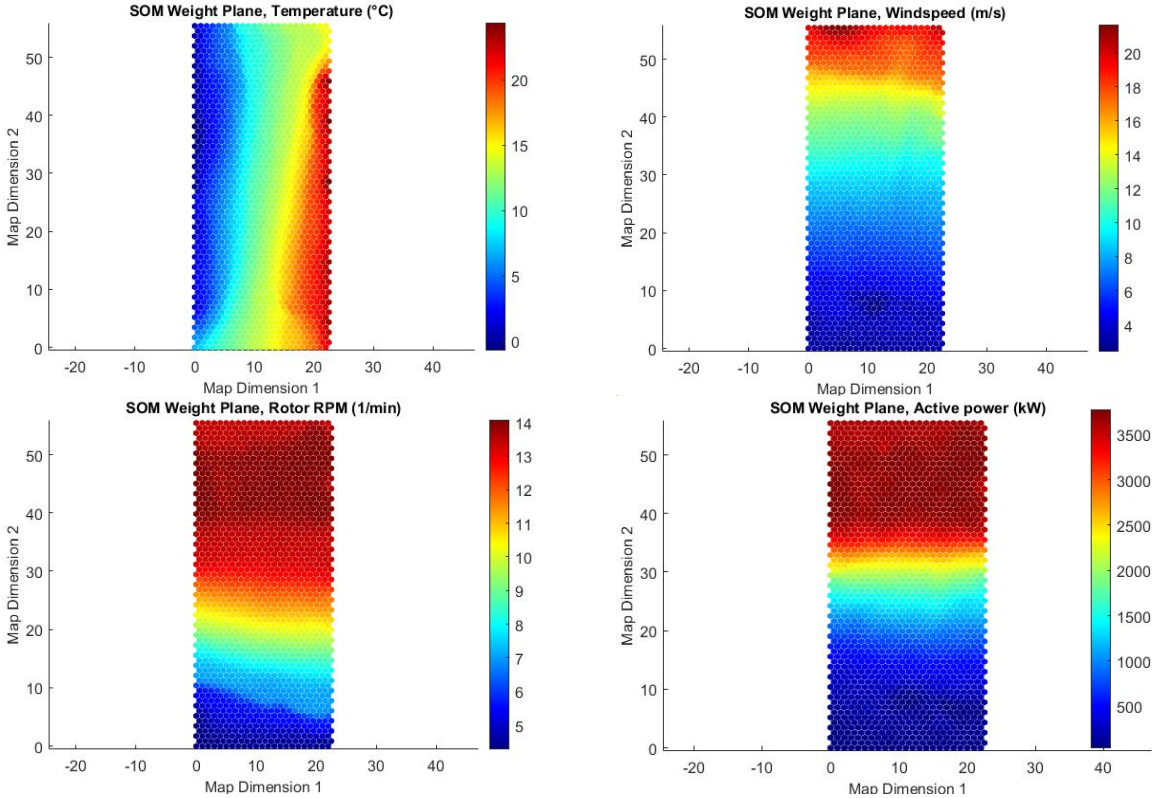

Figure 6: Weightplanes of SOM of Configuration (2)

6 CLUSTERING OF THE SELF-ORGANIZING MAP

One option for clustering the weights of a SOM is the k-means algorithm. There are several examples of the application of k-means for clustering the SOM with the goal of defining some kind of operational or environmental condition. [4,5] A short theoretical background was given in Section 3. In the following an alternative is described, that doesn't change the algorithm itself, but the way it is applied to the data. The alternative method of using the k-means algorithm described in this paper is motivated by the goal of exploiting the organized form of information contained in the SOM. A self-organizing map is not only a simple representation of the training data, but also of the relationships between each variable of the training data. During training the SOM organized its weights in a way that neighboring neurons are relatively similar to each other. Therefore defining EOC means finding regions of the SOM where multiple neurons have similar weights. A basic way to achieve such a clustering is by using the k-means algorithm. As described previously, the standard k-means minimizes the sum of squared distances between cluster centers and datapoints within each cluster. There are several drawbacks with this method. First, the value of k as the number of clusters must be set by the user and is difficult to define objectively. Although there are several methods to aid in defining k, often a trial-and-error approach is used to determine the

value of *k*. This becomes extremely tedious with high dimensional data sets without any prior knowledge regarding an appropriate number of clusters. Secondly, the k-means algorithm is an optimization problem, where the loss function in Equation 3 is minimized. As a result, not all characteristics of the SOM weight planes are preserved. Characteristics that appear in some or only one weightplane are likely to be neglected. Those bits of information are therefore lost in the resulting clustering. Instead of using k-means in the standard way, i.e. using the complete set of neuron weights as input and defining a global number of clusters, this paper uses a different approach. A schematic is given in Figure 7.

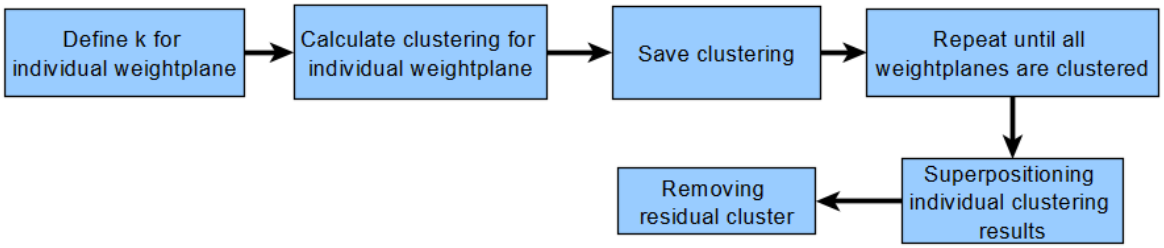

Figure 7: Schematic of the modified approach to the SOM clustering

For a SOM with N dimensions in the input space, each weightplane is clustered individually. A separate value of k for each of them must be defined, but since this affects only one weightplane and they are already ordered, defining a meaningful value should be relatively easy. Even if no reasonable value is obvious, each individual k can be interpreted as the clustering resolution in this dimension and set to a desired value. This process is repeated for each weightplane. All N results are then superpositioned to a resulting clustering of the complete SOM.

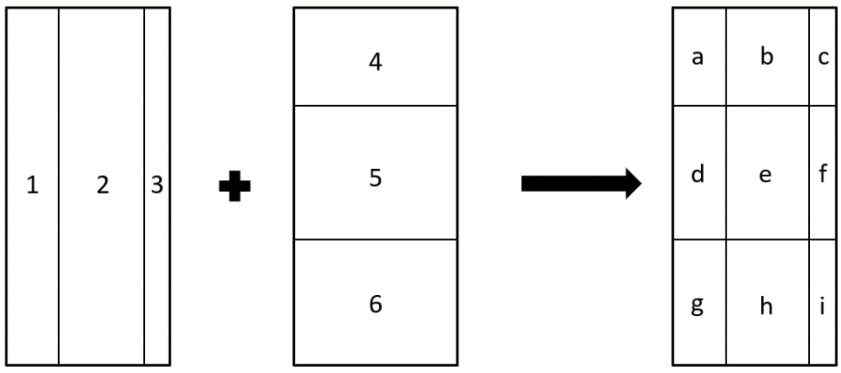

Figure 8: Principle of superpositioning individual k-means results

Superpositioning these clusters results in some very small clusters that are only due to the slight misalignment of original clusters. They are removed by adding their neurons to the most similar neighboring cluster. This way the cluster boundaries of each weightplane and therefore the characteristics of each weightplane are preserved. The principle of this

superposition is shown in Figure 8. All neurons belonging to clusters 1 and 4 form the new cluster a, neurons of clusters 2 and 5 form the new cluster e and so on. The final step of Figure 7 is the removal of residual clusters. This takes into account that clusters in different weightplanes do not align perfectly. For example, Rotor RPM and Generator RPM are mostly identical, though it is likely that two generally identical clusters do not contain the exact same neurons.

$$K = \begin{pmatrix} 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 & 4 & 6 \\ 4 & 4 & 4 & 4 & 4 & 8 \\ 4 & 4 & 4 & 4 & 4 & 10 \\ 4 & 4 & 4 & 4 & 6 & 4 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \tag{4}$$

Regarding the number of clusters for each weightplane all possible combinations from a given set of values will be calculated, see Equation 4 with each row being one combination and each column representing the value of k for one weightplane. This allows the comparison of various combinations and to find a combination that yields a good clustering. From a set of values $k_i = [4\ 6\ 8\ 10]$ all combinations for six weightplanes are formed, resulting in 4096 combinations. For a comparison between different results the number of resulting clusters after removing all residual clusters and a variant of root-mean-square-error (RMSE) as given in Equation 5 are used. With these two quantities the results for each combination can be compared qualitatively. For the SOM described in Figure 4b from the previous section the resulting values are given in Figure 9.

$$RMSE = \frac{1}{K}\left(\sum_{i=1}^{K}\sqrt{\frac{1}{N}\left(\sum_{j=1}^{N}\left(\|c_i - x_j\|^2\right)\right)}\right) \tag{5}$$

In both plots several regions can be clearly distinguished. The largest regions belong to those combinations with constant k for the temperature and each region is further divided into regions where other values are constant. Using these plots the influence of the choice of value of individual k on the resulting clustering becomes visible. The main improvement is achieved by choosing a sufficiently high k for the temperature weightplane. A value of 6 or 8 seems plausible, a value of 10 does not result in significantly better results. Since the other weightplanes are oriented mostly horizontally, their individual influence on the resulting clustering is smaller. Choosing a particular clustering result is ultimately dependent on the specific application, e.g. regarding the necessary clustering resolution. For this purpose the plots in Figure 9 can be very helpful.

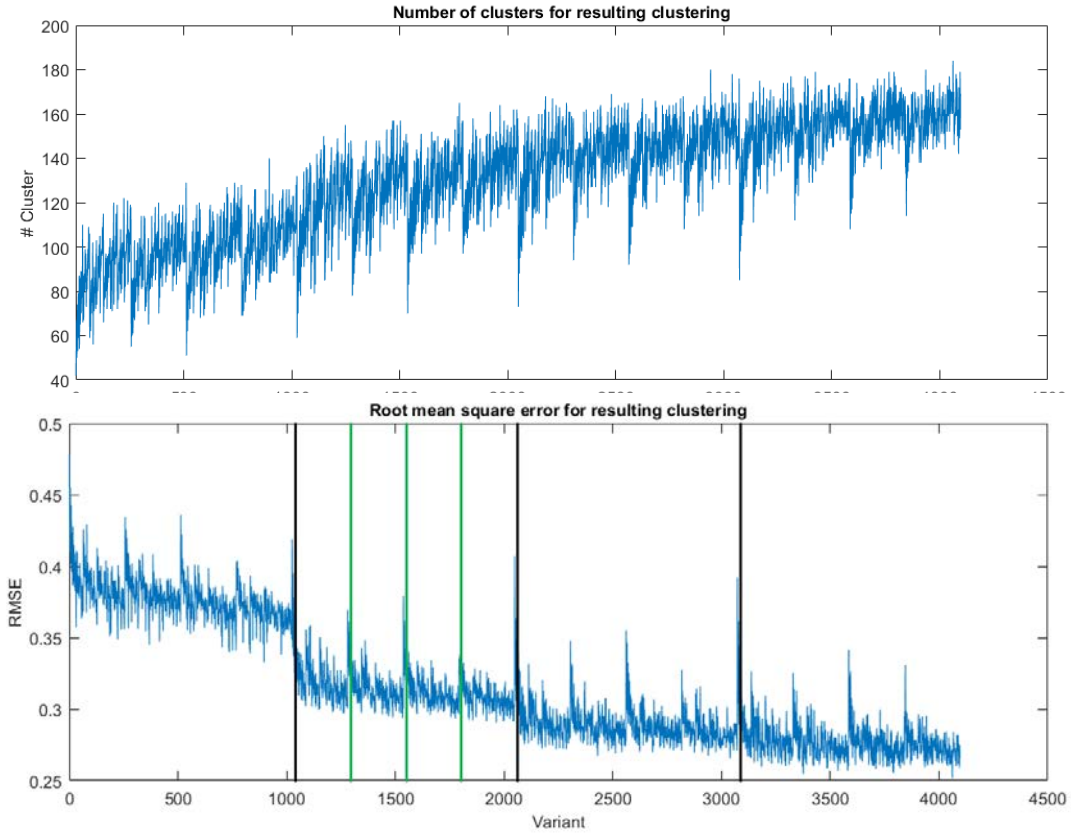

Figure 9: Number of clusters and RMSE per combination of k for clustering the SOM of configuration (2)

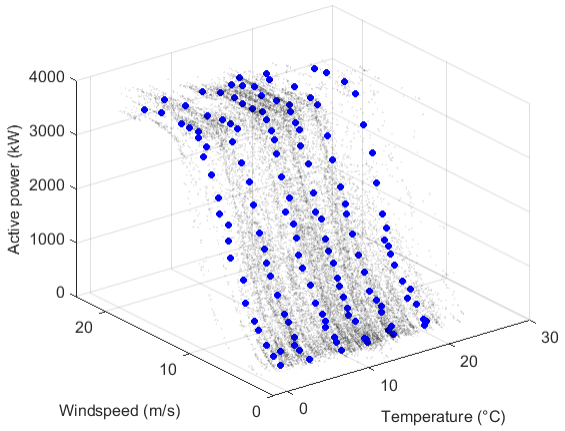

Figure 10a: Clustering result with combination #1750

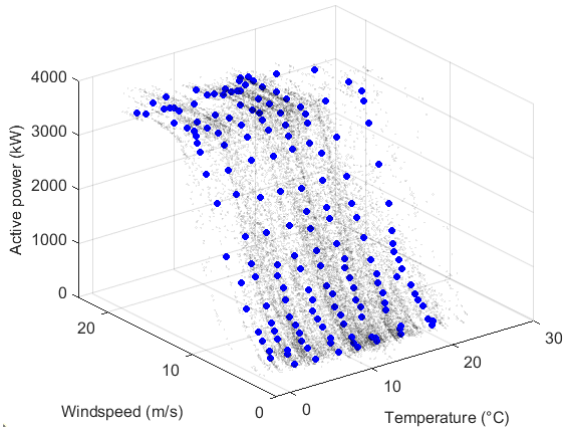

Figure 10b: Clustering result with combination #2683

The following Figures 10a and 10b show two exemplary results. Their respective parameter combination for each *k* is given in Table 3.

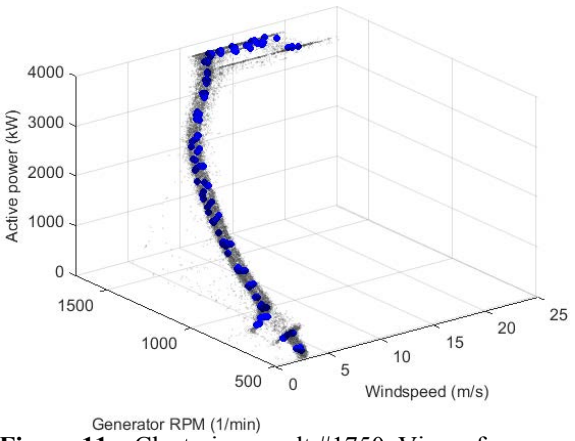

Figure 11a: Clustering result #1750. View of windspeed - generator rpm – active power

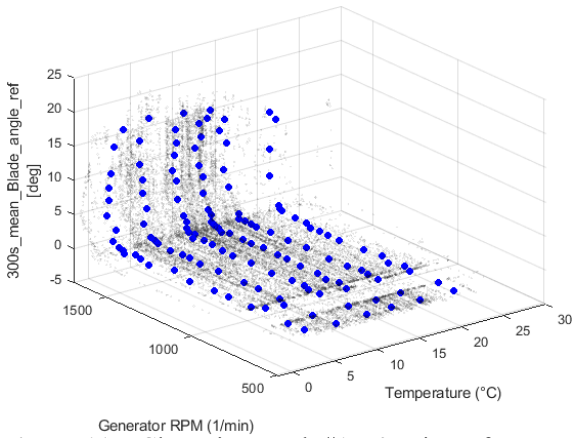

Figure 11b: Clustering result #1750. View of temperature – generator rpm – pitch angle

Although the combinations are quite different, the resulting clusterings look relatively similar. Superpositioning results in a grid-like distribution of cluster centers. Especially for subsequent examination and comparison of clusters with each other, this point is an advantage.

Table 3: Values of individual k per combination

Combination	k_{Temp}	k_{Wind}	k_{RotRPM}	k_{GenRPM}	k_{TPitch}	k_{Power}
1750	6	8	10	6	6	6
2683	8	8	6	10	8	8

With this clustering it is possible to choose two clusters with, e.g., the same windspeed at different temperatures to eliminate the influence of one variable. For combination #1750 Figures 11a and 11b show additional views of other dimensions. Well recognizable is the 'gap' at low generator rpm in Figure 11b, where no cluster centers are located. This gap is due to the plant control to prevent the excitation of the first eigenfrequency of the tower.

CONCLUSIONS

The goal of this work was to train a SOM using only SCADA data and to find a clustering method that facilitates the complex task of clustering high dimensional data. The motivation of clustering the SOM was to use the additional information provided by the organized representation of the training data to define EOC that can be used in further projects to compensate environmental effects on parameters like eigenfrequencies. For this purpose, an EOC must represent a relatively constant state when compared to neighboring EOC and the resulting partition of the SOM should take the characteristics of each weight plane into account. For this purpose, an alternative application of the k-means algorithm was used for the clustering of neuron weights. By clustering each weight plane separately , the structures

that resulted from the training in each weight plane are preserved and contribute to the final clustering. The grid-like clustering provides clusters that can be easily interpreted and compared to other clusters. Taking the number of resulting clusters and the RMSE into account facilitates the definition of an appropriate clustering result depending on the specific task. In further work this clustering approach will be applied to further parameters, e.g. eigenfrequencies and further dynamical WEP features.

ACKNOWLEGDEMENTS

The authors gratefully acknowledge the financial support of this research by the German Federal Ministry for Economic Affairs and Climate Action (Grant number 03EE3023B, Project In-Situ Wind; Grant number 03EE3074B, Project WEA-produktiv).

REFERENCES

[1] T. Kohonen, *Self-Organizing Maps,* 3rd ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.

[2] H. Yin and N. M. Allinson, "On the distribution and convergence of feature space in self-organizing maps," *Neural Computation*, vol. 7, no. 6, pp. 1178–1187, 1995, doi: 10.1162/neco.1995.7.6.1178.

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*: University of California Press, 1967, pp. 281–298.

[4] H. Chen, C. Xie, J. Dai, E. Cen, and J. Li, "SCADA Data-Based Working Condition Classification for Condition Assessment of Wind Turbine Main System," *Energies*, vol. 14, no. 21, p. 7043, 2021, doi: 10.3390/en14217043.

[5] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, 2000, doi: 10.1109/72.846731.