# Outlier Detection in Wind Turbine Frequency Converters using Long-Term Sensor Data

**Nils Schwenzfeier, [1] Markus Heikamp, [1] Ole Meyer, [1] André Hönnscheidt, [1] Michael Steffes, [2] Volker Gruhn [1]**

[1] University of Duisburg-Essen, Essen, Germany
[2] ConverterTec Deutschland GmbH, Kempen, Germany
{firstname.lastname}@uni-due.de, andre.hoennscheidt@stud.uni-due.de, Michael.Steffes@convertertec.com

## Abstract

Wind energy is an important source of renewable and sustainable energy and therefore an elementary component of any future energy supply. However, the operation of large wind farms places high demands on reliability and is often impacted by high maintenance and repair costs in the event of a failure. A frequency converter is one of the most important components of each wind turbine, which ensures that the frequency of the generated energy synchronises with the grid frequency and thus enables the flow of energy into the power grid. The detection of anomalies in these devices is complex due to the high frequency and multidimensionality of different sensor information from the energy control units and requires fault patterns to be discovered and detected in large time series. In this paper, we show how state-of-the-art self-supervised-learning techniques, namely LSTM autoencoders, can be successfully applied to real-world data. We describe the extensions we have made to deal with the often very noisy sensors and describe the construction of the training data set. The trained system was first tested and evaluated on synthetic data and subsequently on a large real-world data set. In both cases, it was shown that outliers can be reliably identified using our presented approach.

## 1 Introduction

Operation & Maintenance (O&M) is an essential cost driver in the wind sector, accounting for about 25-30% of the total life cycle costs for offshore wind turbines (Röckmann, Lagerveld, and Stavenuiter 2017). The key contribution to reducing costs is early fault detection, as it not only prevents critical failures, but also makes maintenance schedules more effective, reduces downtime and increases operational reliability (Entezami et al. 2012). Predictive (condition-based) maintenance (PM) is commonly used to detect errors at an early stage. It combines data-driven reliability models with data from sensor streams aiming to improve O&M strategies (Hameed, Ahn, and Cho 2010). The economic benefit of predictive maintenance compared to other maintenance strategies has already been demonstrated (see e.g. Horenbeek et al.) In the preventive approach, both current sensor data and historical data are analysed to plan maintenance proactively. The collected data is used for analytics,

decision-making procedures, diagnosis and prognosis, and optimisation (Rinaldi, Thies, and Johanning 2021).

When analysing the data, the focus has to be on outliers in the data sets in order to detect untypical operations of the wind turbines. Finally, outlier detection is an important factor in uncovering erroneous behaviour. In our work, we examine outliers in data based on electric sensor measurements. This includes values for electric power, mechanical speed, voltage and current. Studying these values is of special significance, as they provide numerical information about the essential task of frequency conversion. During this task, the appropriate frequency to the grid and generator is established and a stable energy delivery is set up.

The goal of our work is to find anomalous behaviour of frequency converters without concrete faults having already occurred. To do this, we examine real-world data recorded by sensors from single converters, aiming to find unusual parts. As a further step, we try to detect untypical behaviour of multiple converters located in the same wind farm and make a comparison between them. In summary, we present and evaluate an outlier detection system based on a long-short-term memory (LSTM) autoencoder in detail so that it can be reused in similar use cases. We thus contribute to the field of predictive maintenance for wind turbines by:

- applying outlier detection methods on an extensive real-world dataset and describing the steps taken.
- suggesting a LSTM autoencoder architecture for the given task.
- providing full source code and documentation to understand the procedure and apply it to other projects.
- providing a working outlier detection system to find anomalous behaviour of wind turbines, which has the potential to ease future maintenance tasks.

In the following sections, we present our approach in detail. We first provide background knowledge about the underlying frequency converter in section 2. Section 3 reviews related research regarding predictive maintenance for wind turbines and outlier detection in sensor streams. Section 4 introduces our outlier detection system, and section 5 describes its evaluation along with the experiments conducted. In section 6 we give a brief overview of what needs to be done to deploy our system in a wind farm, and conclude with an outlook on future work in section 7.
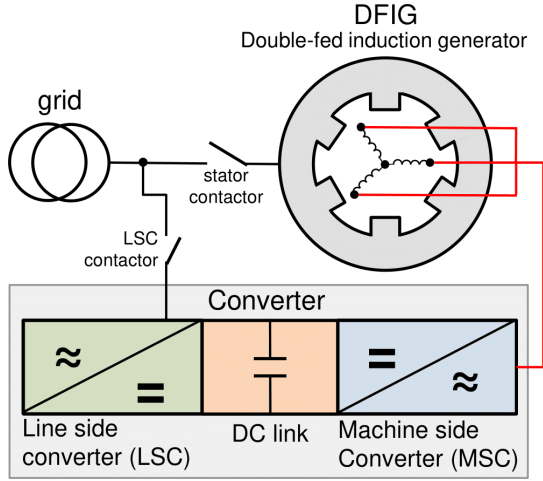
Figure 1: System Overview of the Frequency Converter.

| Sensor | Name | Range* |
|---|---|---|
| 2003 (rpm) | Generator Rotor Speed | 0 to 1300 |
| 2072 (kW) | Mains Power (Re) | 0 to 3000 |
| 2073 (kVAR) | Mains Power (Im) | -500 to 500 |
| 2101 (V) | DC Link Voltage | 0 to 1100 |

Table 1: Relevant sensors in data set. *Range of values* corresponds to the usual range in which the values vary and does not cover any theoretically possible values of the system.

## 2  Frequency Converter

In this section, we first describe the underlying converter system and its operating modes. We then explain the dataset used for our analyses with a focus on sensor data, as they are the main input for our outlier detection approach.

### 2.1  Conversion System

The main task of an electrical generator in a wind turbine is to convert rotational mechanical energy into electrical energy. As an essential component, the frequency converter processes the energy generated and adjusts the frequency to the power grid. The underlying frequency conversion system in our work is designed for the operation of a doubly-fed induction generator (see DFIG (Peña, Clare, and Asher 1996; Muller, Deicke, and De Doncker 2002)). In DFIGs the stator side is directly connected to the grid, while the rotor frequency, rotor speed and generator voltage are controlled by the converter. Our conversion system includes a line side (grid side) converter (LSC) and a machine side converter (MSC), both monitored by the control unit CSC4 (see Figure 1). A DC-link is in place as an important connection between the LSC and MSC.

### 2.2  Dataset

During the operation of a wind turbine, the control unit monitors a variety of sensors. The values of individual sensors are logged at intervals of about five seconds. While analysing the sensor values, we focused on rotor speed, mains power and DC link voltage (see Table 1). The sensor *Speed: Generator Rotor* does not represent the speed of the mechanical rotor blades, but the speed that results according to the gear ratio. For mains power, a real part and an imaginary part are stored. The value of the real part represents energetically usable active power and the value of the imaginary part represents the reactive power that is used to stabilise the voltage.

The sensors we are looking at provide us with tangible measurements during the switching of operating modes. Examining the sensor data is essential in order to find unusual behaviour of the wind turbines. To gain an understanding, the course of the sensor values in fault-free operation is first described. The value for *Speed: Generator Rotor* is rather low in status *Standstill*, but depends on the surrounding wind speed. At a threshold value of 750rpm according to the gear ratio, the status changes to *Charge*. The value for *Voltage: DC Link* increases in status *Charge* and remains at the specified rated value during the following statuses. After switching to previous statuses back to *Standstill*, the value decreases slowly. The value for *Power: Mains (Re)* equals 0 at first and rises after the change from *Standby* to *Synchronisation* and finally to *Mains Parallel*. This value indicates the usable energy output of the wind turbines.

Our dataset consists of the sensor streams of five onshore wind turbines, all located in the same wind farm. It contains data from August 2018 to December 2020. Thus, a data set is available that allows the observation of the behaviour of converters using historical data, since it is based on a period of about two and a half years.

## 3  Related Work

Predictive Maintenance for wind turbines focuses on several subsystems and components. According to a study, the highest amount of downtime is caused by the pitch and gearbox, followed by the converter and the generator (Li, Jing, and Zhang 2013).

The spectrum of different methods for predictive maintenance solutions covers a wide range of techniques. A rough distinction can be made between model-based and model-free approaches.

Model-based approaches use an underlying model of the system dynamics from which statements, expectations and ultimately also deviations can be predicted. Model-based methods are used in many domains, from wind energy (Habibi, Howard, and Simani 2019), to the control of gas networks (Syed et al. 2020), to space travel (Djebko, Puppe, and Kayal 2019). One possibility is to use a Kalman filter as an estimator for different system values. This can be used, for example, to estimate the blade pitch angle, which helps to successfully and effectively prevent possible failures in this area of the wind turbine, which can avert more expensive and catastrophic subsequent failures (Cho, Gao, and Moan 2018). Such an approach can be successfully used to monitor different subcomponents of a wind turbine and detect potential failures. For example, the use of SCADA data and information can help to better monitor the installed cooling

systems (Röckmann, Lagerveld, and Stavenuiter 2017).

Just like model-based approaches, model-free approaches are also widely used in industry. Model-free approaches are less based on known or modelled system dynamics, but often use learning systems to find indications of errors without necessarily having to know the complete dynamics of the underlying system. One way to implement this approach is to formulate the problem as a classification problem and then use known methods, such as gradient tree bossting, to solve the problem. This can happen, for example, by learning the system state from the recorded data, the estimation of which can then be used as a basis for further decisions (Mazzoleni, Maccarana, and Previdi 2017). Other examples that use the classification scheme but use other learning methods such as SVM can be found for example in (Santos et al. 2015). In addition to classification approaches, which belong to supervised learning, other approaches can be found in less supervised or unsupervised procedures, which, for example, find deviations between learned or determined data representations. In (Pozo and Vidal 2016), a principal component analysis (PCA) model of a non-damaged turbine is compared to the values of the turbine under investigation. A subsequent statistical hypothesis testing is carried out to make a decision on whether a fault is reported or not. In addition to mathematical decomposition methods, there are gradient-based learning systems that enable data representation to be learned. One of the most important representatives in this field are autoencoders, which offer another possibility for deviation detection: the reconstruction probability (An and Cho 2015). Autoencoders have attracted a lot of attention in recent years because they are able to process almost arbitrarily complex data streams. First applications can be found in wind turbines for the detection of icing (Yuan et al. 2019), damage in the blades (Yang and Zhang 2020) or faults in the gearbox (Jiang et al. 2017).

# 4 Outlier Detection System

We tackle the problem of identifying anomalous parts in a large time series by dividing the complete time series into smaller overlapping fragments of equal size. In the following, we will reference to these time slices as *windows*. Each window is treated as an individual short time series. Finding an anomalous part in the large time series is now equivalent to finding an anomalous sample in a database of multiple short time series.

To identify an anomalous window is still a challenging task, since multiple sensor values have complex interaction patterns over time. We tackle this task using a concept from the outlier detection domain that utilises the reconstruction error to identify outliers. The general idea is to model a lossy compression algorithm that is able to reconstruct a normal sample of a dataset. Due to the lossy compression, this algorithm removes details of the samples and instead focuses on reconstructing the general structure. Since outliers are by definition rare events, it is reasonable to assume that the compression algorithm cannot reconstruct these samples well. Therefore, the reconstruction error of a sample can be used as an anomaly score with high values indicating anomalous samples.

To find such a lossy compression algorithm, we train a long short-term memory (LSTM) autoencoder to learn a compact representation (embedding) for every window. The model is trained by measuring the difference between the input and the reconstruction and propagate the error back through the network. Since we can assume that the great majority of windows in our dataset show normal behaviour, we hypothesise that the LSTM autoencoder indeed focuses on reconstructing the normal patterns. Therefore, rare patterns in the data would lead to high reconstruction errors.

## 4.1 Preprocessing

The different sensor series are resampled to a unified time axis with a constant frequency of one value every five seconds. After that, any missing values are filled using nearest neighbour interpolation with an interpolation limit of three consecutive values. Consequently, no gaps larger than 15 seconds are interpolated. The time series is subdivided into overlapping windows with a length of 180 values each (i.e. 15min). Choosing the right overlapping to extract the windows is a tradeoff. On the one hand, we do not want to miss any anomalous event in the time series, which favors a large overlapping (small shift). On the other hand, a small shift leads to many windows that are almost identical. This can violate the assumption that outliers are rare events and make the autoencoder also learn how to reconstruct these rare patterns. In our case the additional computational costs of a small shift can be neglected during this consideration, since our models can already be trained on several hundreds of thousands of samples in just a few minutes. We choose an offset of 60 values (i.e. 5min) as a compromise to guarantee that each moment in time is only present in a maximum of three windows. After the extraction, all windows still containing missing values are discarded. Finally, all sensors are normalized, i.e. scaled to the interval $[0, 1]$.

## 4.2 Model architecture

We use two different model architectures for the single-turbine and multi-turbine scenario. Both models are derived from the same architectural design.

The encoder module of our LSTM autoencoder architecture consists of a block of one or more 1D convolutional layers followed by a stacked lstm layer. The first convolutional layers are intended to be a fully learnable preprocessing step for the input sensors. In particular, this eliminates the need to smooth sensor values during preprocessing, since this can be learned in the convolutional layer. Without the first convolution block, the model was not trainable. This is most likely due to the complex structure of the cost function, whose minimum is even more difficult to find due to highly fluctuating input sensors. After the convolutional layers, lstm layers are used to encode the temporal sensor information into a fixed-size embedding. This is where most of the compression is realised.

From the last lstm layer of the encoder, data goes to the decoder, which repeats the embedding of the encoder to reach the window length. The repeated embedding is processed by another stacked lstm layer to unfold compressed

information on the time axis. A linear layer is used to expand the result to the same dimension that is returned by the last convolutional layer in the encoder. Then, a 1D transpose convolution is performed to undo the original convolution to obtain a result that has the same shape as the input. Our single-turbine model architecture is shown in figure 2.

The single-turbine and multi-turbine model differ primarily in the amount of layers and layer sizes in convolution and lstm blocks. While the single-turbine model contains a single convolutional and lstm layer in the encoder, the multi-turbine model uses two consecutive convolutional layers followed by two stacked lstm layers. These modifications were found to be necessary due to the larger feature space for multiple wind turbines.

In our single-turbine model, a convolutional layer with 8 filters and an embedding size (lstm output) of 32 dimensions is used. We use convolution filter with a kernel size of 7 and a stride of 2. The multi-turbine model uses 35 (conv-1) and 50 (conv-2) filters and an embedding size of 128. The stride is adapted for the multi-turbine model to 3 in the first and 1 in the second convolutional layer.

## 4.3   Training

During the training, the *mean squared error (MSE)* of the input $x$ and its reconstruction $x'$ (model output) is calculated and the error is backpropagated through the network. We train our single-turbine models for 10 epochs on over 218,000 samples (windows) with a batch size of 128 and a fixed learning rate of 0.001 using the Adam optimizer. The multi-turbine model is trained for 20 epochs on over 103,000 samples (windows) with the same batch size and learning rate, since the dataset is much smaller compared to a single wind turbine. The reduction of the dataset size is caused by our constraint that windows containing any missing values after interpolation are not considered in our analysis, which is obviously more difficult to achieve the more wind turbines are considered.

# 5   Evaluation

We evaluate our outlier detection system in various settings. First, the single-turbine scenario is evaluated. In this scenario, we try to identify anomalous windows in the sensor data of a single wind turbine. To do this, we use both synthetically generated outliers and a large real-world dataset (s. 2.2). Second, we observe the behavior of all five wind turbines to analyse the behavior for inconsistencies between different wind turbines of the same wind farm. We call this the multi-turbine scenario.

Synthetic data is used to verify that our model indeed finds windows that show rare situations. All work done on the real-world dataset is performed in a completely unsupervised setting, i.e. we do not have any information about the expert interpretation of the wind turbine's behaviour upfront. The dataset has been analysed during a research project with the aim to search for anomalous behaviour of a wind turbine with a focus on its frequency converter.

We implemented our models based on the framework PyTorch (Paszke et al. 2017). We provide full access to our source code as well as to all data samples given to the subject matter experts.[1] Each of our models was trained on a single NVIDIA RTX A6000 GPU within a few minutes.

## 5.1   Single-turbine

The goal of the single-turbine scenario is to find anomalous windows in a wind turbine's sensor data. That is, patterns should be selected that are rare for the wind turbine at hand.

**Synthetic data**   We expect that our synthetically generated windows will result in a high reconstruction error so that they can be clearly identified as outliers.

We use three different mechanisms to generate our artificial data. First, we choose random time windows that cover a period during which the wind turbine fed electricity into the grid. We set the rotor speed of the wind turbine to 0 for the entire window. Even if a rotor speed of 0 occurs many times in the data set during a complete window, this window should be uniquely identifiable, since this specific constellation cannot occur in practice. Our second-generation method swaps the values of two different sensors in a given window. In our last generation method, we simulate a defect in the wind turbine. Specifically, we model the occurrence of a short circuit in the intermediate circuit of the frequency converter. We again use windows during which the wind turbine fed electricity into the grid and vary the point of time of the defect.

We evaluate the performance of our model using the different generation methods one after the other. First, five windows are generated and injected into the dataset. After that, the trained model is used to calculate the reconstruction loss for all windows. We measure the recall in the top-n, i.e. the n windows with the highest reconstruction errors. We set n equal to 50. Compared to the size of the dataset, this is a relatively small sample. Our model is able to clearly separate all generated samples from the normal data leading to a recall of 1.0 for all generator methods in the top-50. The separation is made so clearly that even in the top-10 the recall is greater or equal than 0.8 for all methods. Figure 3, left subplot, shows the change in the reconstruction loss for the five generated windows before and after the swap of two sensors. This shows that the model has learned the ordinary relationships of the dataset and recognizes alienated data points without effort.

**Real-world data**   Knowing that the model basically detects exceptional time windows, we now examine the real-world dataset for exceptional sections. We handed over the top-20 time windows, i.e. the 20 windows with the highest reconstruction errors, for all five wind turbines to the technical experts. In addition, we delivered five time windows with a significantly lower reconstruction error for all five wind turbines, i.e. time window with the 10001-10005 highest reconstruction error, to provide a reference point. Figure 4 shows an example of such a "normal" window. A total of 125 samples were handed over. We assume that there is a significant difference between windows with high reconstruction error and those with a lower reconstruction error,

---

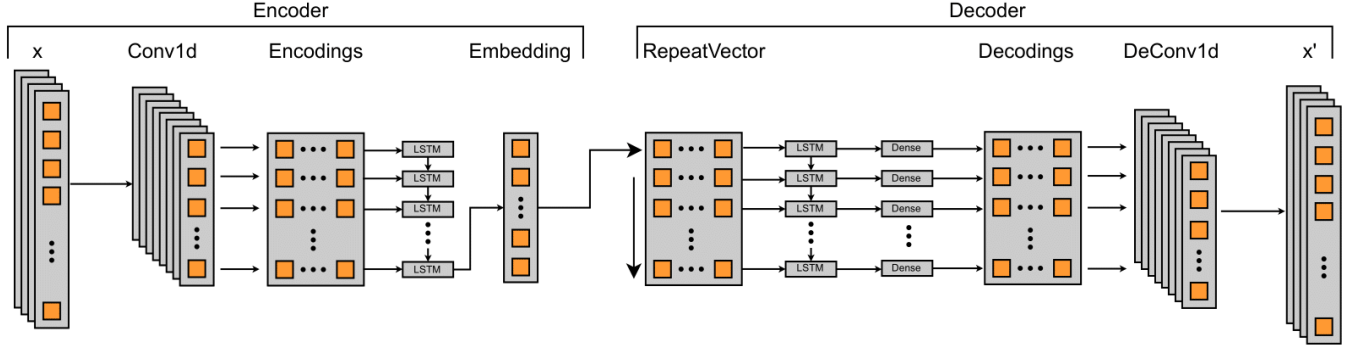[1]https://github.com/n-1-l-s/cobamas-sensor-od

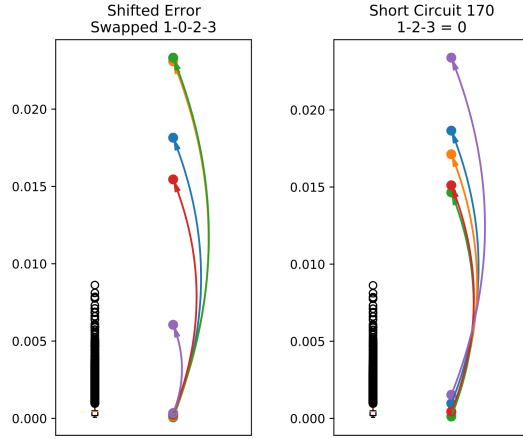Figure 2: Model architecture of our single-turbine LSTM autoencoder.



Figure 3: Change of the reconstruction error for five different windows when swapping sensors 2003 and 2101 (left) or simulating a short circuit after 170 timesteps, i.e. 14min (right).

| Reconstruction-Error | Wind Turbine | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|---|
| high | $T_1$ | 20 | 0 | 0 | 0 |
| | $T_2$ | 16 | 4 | 0 | 0 |
| | $T_3$ | 20 | 0 | 0 | 0 |
| | $T_4$ | 20 | 0 | 0 | 0 |
| | $T_5$ | 19 | 0 | 1 | 0 |
| low | $T_1$ | 0 | 5 | 0 | 0 |
| | $T_2$ | 0 | 4 | 0 | 1 |
| | $T_3$ | 0 | 5 | 0 | 0 |
| | $T_4$ | 0 | 5 | 0 | 0 |
| | $T_5$ | 0 | 5 | 0 | 0 |

Table 2: Results of the single-turbine evaluation. For each turbine 20 windows with the highest reconstruction errors and 5 windows with a low reconstruction error have been classified by the experts.

not only from the technical point of view, but also from the professional's point of view.

The technical experts evaluated all the examples and classified their feedback into four different categories:

$C_1$) Shutdown at nominal load. Unintentional stop, possibly due to overtemperature, overspeed or similar monitoring outside the frequency converter.

$C_2$) Normal shutdown due to low wind. Probably gusty wind or unfavorable wind conditions.

$C_3$) Jump in DC link voltage at standstill. Light error on DC link voltage meter (fiber optic interruption).

$C_4$) Manual stop.

Categories $C_1$ and $C_3$ represent clearly unintended behaviour of the wind turbine. Category $C_2$ represents a not ideal but still normal situation. Category $C_4$ describes a more rare event that is typically performed for maintenance tasks or to reset a wind turbine.

Table 2 summarises the given feedback. From that, one can see that the unintended behaviour classes $C_1$ and $C_3$

dominate the windows with the highest reconstruction errors. High reconstruction errors were only assigned to four windows that show normal behavior from a professional point of view. We consider these four windows as false positives. In the reference windows, that have a much lower reconstruction error, $C_2$ dominates. It turns out that normal wind turbine behavior is indeed correlated with lower reconstruction errors of our model.

## 5.2 Multi-turbine

In the multi-turbine scenario, time windows are searched for when the behavior of the various wind turbines is inconsistent. Therefore, time windows are identified that show unexpected behavior of one or more wind turbines in the same wind farm. In practice, wind turbines in a wind farm do not behave identically all the time. Due to their positioning, the wind turbine's behavior can differ, since the wind conditions vary at different locations. Even inside the area of a single wind farm the differences in wind conditions for each wind turbine are so large, that some wind turbines might be running while others do not have sufficient wind conditions to operate. We assume that our model is able to capture these
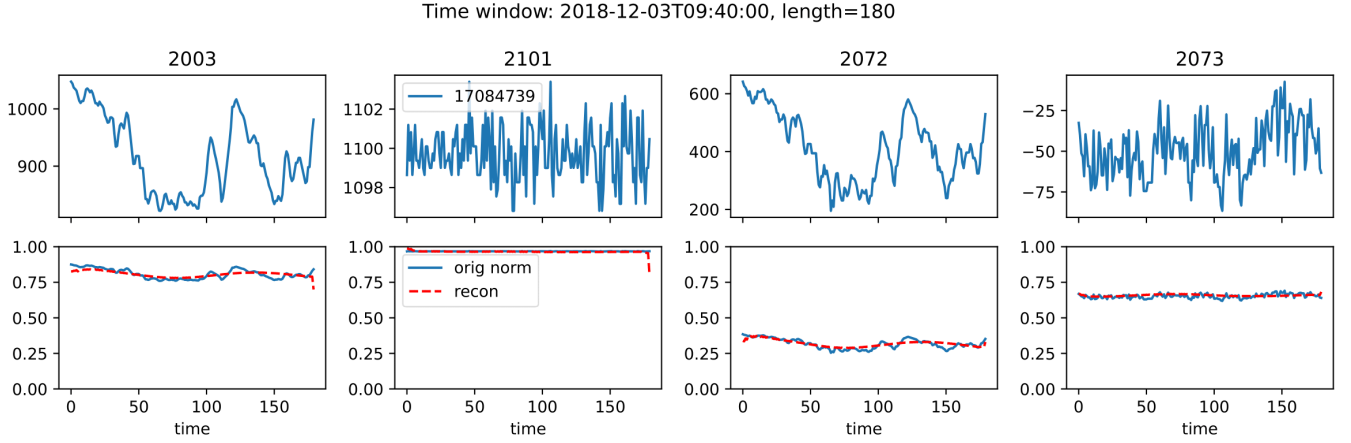
Figure 4: Example of a window with a low reconstruction error. The top row shows the raw values of the four sensors. Below, the normalized sensor values (orig norm) are displayed along with their reconstructions (recon) generated by our model. More examples can be found in the code repository.

differences between multiple wind turbines in a wind farm as normal variation. Consequently, large reconstruction errors should only occur when the wind turbines diverge in an unexpected way. To be able to capture the dependencies between multiple wind turbines, we extend the capacity of our model as described in section 4.2.

**Synthetic data** Analogous to the single-turbine scenario, we are first testing our system with synthetic data. To generate artificial outliers, we select two different windows. Sensor data is taken from window $w_1$ for four out of five wind turbines. The sensor data of the last wind turbine is taken from window $w_2$. After that, the newly constructed sample is injected into the real-world dataset. We repeat this process multiple times to insert a total of five different artificial outliers into the dataset. All generated samples are found in the ten windows with the highest reconstruction errors (top-10). Therefore, they are easily identified as outliers by our model.

**Real-world dataset** We handed over the ten windows with the highest reconstruction errors to the experts for their assessment. The experts rated 8/10 examples as useful for assessing the condition of the wind turbine. The top-10 windows show a noticeable reactive power at a wind turbine, the unexpected shutdown of a wind turbine at full capacity, a full breaking of a wind turbine with high wear of the mechanical components, half-charged DC link over a long period of time as well as short shutdowns from nominal range. The full braking situation is included two times in two different windows in the top-10. This is an artifact caused by the sliding window mechanism as described in section 4.1. Two samples, rated as normal behaviour, show a light wind situation and a rare situation in which, due to an oversupply of electricity in the power grid, the wind turbines absorb power. Images of the sensor waveforms of the top-10 examples, including the sensor reconstructions by our model, are included in our code repository.

## 6 Deployment

In section 5, we showed that our approach is able to detect anomalous situations for the behavior of a single wind turbine as well as for a group of wind turbines. We also showed that it is possible to use a pre-trained model for a new wind turbine in the same wind farm. Thus, our approach can be applied to new wind farms by equipping all wind turbines with a data acquisition unit. A model can be trained on historic data of a single wind turbine or a small set of wind turbines. After that, the trained model can be used to score the data series for all wind turbines inside the wind farm. Thereby, the monitoring processes for the wind turbines can be eased by providing a summary of anomalous situations for all wind turbines. Technical experts can use this information to assess the condition of each wind turbine and initiate maintenance before a fault occurs.

## 7 Conclusion

We proposed an outlier detection system, based on the reconstruction loss of sensor data, for wind turbines. Our system is based on a LSTM autoencoder and has been evaluated using synthetic data and a large real-world dataset. The system has proven that it is capable of reliably detecting anomalies in the behavior of a wind turbine, even though it has only a very limited view of the entire wind turbine (i.e., the frequency converter only). For future work, it is desirable to include data sources from other components to obtain a holistic view of the wind turbine. We evaluated our system on five wind turbines of a single wind farm. In the future, we will investigate whether a pre-trained model can be used in another wind farm.

## Acknowledgements

# References

An, J.; and Cho, S. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1): 1–18.

Cho, S.; Gao, Z.; and Moan, T. 2018. Model-based fault detection, fault isolation and fault-tolerant control of a blade pitch system in floating wind turbines. *Renewable energy*, 120: 306–321.

Djebko, K.; Puppe, F.; and Kayal, H. 2019. Model-based fault detection and diagnosis for spacecraft with an application for the SONATE triple cube nano-satellite. *Aerospace*, 6(10): 105.

Entezami, M.; Hillmansen, S.; Weston, P.; and Papaelias, M. 2012. Fault detection and diagnosis within a wind turbine mechanical braking system using condition monitoring. *Renewable Energy*, 47: 175–182.

Habibi, H.; Howard, I.; and Simani, S. 2019. Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review. *Renewable energy*, 135: 877–896.

Hameed, Z.; Ahn, S.; and Cho, Y. 2010. Practical aspects of a condition monitoring system for a wind turbine with emphasis on its design, system architecture, testing and installation. *Renewable Energy*, 35(5): 879–894.

Horenbeek, A.; Van Ostaeyen, J.; Duflou, J.; and Pintelon, L. 2013. Quantifying the added value of an imperfectly performing condition monitoring systemApplication to a wind turbine gearbox. *Reliability Engineering & System Safety*, 111: 45–57.

Jiang, G.; He, H.; Xie, P.; and Tang, Y. 2017. Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 66(9): 2391–2402.

Li, H.; Jing, T.; and Zhang, H. 2013. Arena-Based Modeling of the Maintenance Operation for a Wind Farm. *Applied Mechanics and Materials*, 401-403: 2205–2208.

Mazzoleni, M.; Maccarana, Y.; and Previdi, F. 2017. A comparison of data-driven fault detection methods with application to aerospace electro-mechanical actuators. *IFAC-PapersOnLine*, 50(1): 12797–12802.

Muller, S.; Deicke, M.; and De Doncker, R. 2002. Doubly fed induction generator systems for wind turbines. *IEEE Industry Applications Magazine*, 8(3): 26–33.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch.

Peña, R.; Clare, J.; and Asher, G. 1996. Doubly fed induction generator using back-to-back PWM converters and its application to variable-speed wind-energy generation.

Pozo, F.; and Vidal, Y. 2016. Wind Turbine Fault Detection through Principal Component Analysis and Statistical Hypothesis Testing. *Energies*, 9(1).

Rinaldi, G.; Thies, P.; and Johanning, L. 2021. Current Status and Future Trends in the Operation and Maintenance of Offshore Wind Turbines: A Review. *Energies*, 14: 2484.

Röckmann, C.; Lagerveld, S.; and Stavenuiter, J. 2017. *Operation and Maintenance Costs of Offshore Wind Farms and Potential Multi-use Platforms in the Dutch North Sea*. Cham: Springer International Publishing. ISBN 978-3-319-51159-7.

Santos, P.; Villa, L. F.; Reones, A.; Bustillo, A.; and Maudes, J. 2015. An SVM-Based Solution for Fault Detection in Wind Turbines. *Sensors*, 15(3): 5627–5648.

Syed, M. M.; Lemma, T. A.; Vandrangi, S. K.; and Ofei, T. N. 2020. Recent developments in model-based fault detection and diagnostics of gas pipelines under transient conditions. *Journal of Natural Gas Science and Engineering*, 103550.

Yang, L.; and Zhang, Z. 2020. A Conditional Convolutional Autoencoder-Based Method for Monitoring Wind Turbine Blade Breakages. *IEEE Transactions on Industrial Informatics*, 17(9): 6390–6398.

Yuan, B.; Wang, C.; Luo, C.; Jiang, F.; Long, M.; Yu, P. S.; and Liu, Y. 2019. WaveletAE: A wavelet-enhanced autoencoder for wind turbine blade icing detection. *arXiv preprint arXiv:1902.05625*.