A Diagnostic and Predictive Framework for Wind Turbine Drive Train Monitoring

Bach-Andersen, Martin; Winther, Ole; Rømer-Odgaard, Bo

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

A Diagnostic and Predictive Framework for Wind Turbine Drive Train Monitoring

Martin Bach-Andersen

Kongens Lyngby 2017
PhD-2017-449

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

Vast amount of data are collected minute by minute from wind turbines around the world. This thesis represents a focused research effort into discovering new ways of processing these data streams in order to gain insights which can be used to lower the maintenance costs of wind turbines and increase the turbine availability.

First, it is demonstrated how simple sensor data streams can be leveraged based on a combination of non-linear predictive models and unsupervised fault detection to provide warnings of a critical bearing failure more than a month earlier compared to existing alarm systems.

Second, early fault identification based on analysis of complex vibration patterns which is a domain previously reserved for human experts, is shown to be solved with high accuracy using deep learning architectures trained in a fully supervised sense from the data collected in a large scale wind turbine monitoring platform.

The research shows a way towards a fully automatized data-driven wind turbine diagnostic processing system that is highly scalable and requires little or no feature engineering and system modeling.

Summary (Danish)

Store mængder data hentes hvert minut fra vindmøller verden over. Denne afhandling repræsenterer en fokuseret forskningsindsats i forhold til at opdage nye måder at behandle disse datastrømme for skabe indsigt der kan anvendes til at reducere omkostninger til vedligehold på møllerne samt forbedre deres oppetid.

Først vises det hvordan simple sensormålinger kan udnyttes, baseret på en kombination af ikke-lineære prediktive modeller og ikke-superviseret fejldetektion, til at give advarsler om kritiske lejefejl mere end en måned tidligere sammenlignet med de nuværende alarmsystemer.

Dernæst vises det at tidlig fejldetektion baseret på analyse af komplekse vibrationsmønstre, et område der tidligere var forbeholdt menneskelige eksperter, nu kan løses med høj nøjagtighed ved hjælp af dybe netværk trænet fuldt superviseret fra data indsamlet i et storskala mølleovervågningssystem.

Denne forskning viser en vej henimod et fuldautomatiseret data-dreven diagnostisk system til vindmøller, der er fuld skalerbart og kræver lidt eller ingen feature engineering og systemmodellering

Preface

This thesis was prepared at DTU Compute and Siemens Diagnostic Center in fulfillment of the requirements for acquiring a Ph.D. in Engineering.

The thesis consists of a summary report and three research papers.

Lyngby, 15-March-2017

Martin Bach-Andersen

Acknowledgements

First and foremost I would like to thank my supervisor Professor Ole Winther deeply for his encouragement, enthusiasm and guidance throughout this journey.

I am also deeply grateful for the support of my supervisor in Siemens, Bo Rømer-Odgaard who put faith in me and the vision I put forward for this research back in 2013, and of course Siemens and Innovationsfonden for funding this project.

A big thanks also goes out to all my colleagues at Siemens Diagnostic Center and Cognitive Systems at DTU Compute.

Last but not least, I wish to thank my beautiful wife Bodil for her love and support.

Contents

Chapter 1

Introduction

Data in one form or another is now ubiquitous in modern society. The advent of mobile and interconnected devices, low cost sensors and low cost computing and storage resources is now driving business model innovation at a fast pace. This development is also having an impact in the industrial arena where companies are now struggling to gain insight and more importantly, value, from the streams of data being collected from their critical assets. This ongoing effort serves as the back drop for the work presented in this thesis.

With the above perspective fresh in mind, the remainder of this thesis will now zoom in on a very particular industrial problem domain, namely condition monitoring applications for wind turbines. The scientific field of condition monitoring is a rich interdisciplinary field drawing from research in the disciplines of sensor technology, material sciences, signal processing and data science to name just a few. The research presented here will strictly touch upon the data modeling side of this equation to demonstrate how improved modeling can have a significant effect on the quality and timeliness of the outputs of existing wind turbine monitoring systems.

The common theme of the work presented in the following is a data-driven mindset where we put less emphasis on any preconceived models of reality and more emphasis on the large volume of data itself using highly flexible general models. The key benefit of such models is that they can alleviate the need of complex

hand tuned feature engineering, which can be both time consuming and suboptimal. Also, for physical systems of only moderate complexity such modeling is quickly becoming infeasible. However, data-driven means data hungry and the presented methods require a certain scale to be applicable.

The primary target audience of this work is researchers and practitioners within the field of wind turbine condition monitoring seeking a more data-driven perspective on this topic, but the work should also be of interest to readers working with industrial data analysis in general, as the methods presented here are broadly applicable in nature. Another reason why this work may spur interest here, is that the wind industry in many ways has been a precursor in terms of machine connectivity, industrial data management and predictive maintenance applications when comparing with other industries. This development has been a result of pure necessity. Wind farms are often located in remote areas and access may be restricted for extended periods of time, often counted in months, due to adverse weather conditions. This is particularly true for turbines placed offshore. It was quickly realized that the ability to operate the turbine remotely and to collect data at the same time could provide substantial benefits in terms of turbine availability and reduced maintenance costs. In the last decades, wind farms are therefore typically interfaced to a global control and data acquisition infrastructure. A rather unique combination here is the sheer scale, typically counted in thousands of turbines, and the data complexity drawing from hundreds of sensors mounted in each turbine. With the rapid growth in global wind energy installations the largest monitoring platforms are now collecting data continuously from tens of thousands of turbines simultaneously. The work in this thesis is specifically motivated by the challenges experienced when developing and operating such large scale monitoring platforms, and may provide insights relevant for other industries which are on a similar path of development in terms of scale and complexity.

As a final humble ambition, the work is also hoped to serve as an inspiration to machine learning researchers in order to facilitate new research into how machine intelligence can provide answers to some of the challenges experienced in wind energy systems.

1.1 Thesis outline

Given this diverse target audience chapter 2 will first be spent on providing a basic introduction to wind turbines and the current and future role of wind energy. Next, the basic concepts of statistical learning is introduced in chapter 3 which underpins the data-driven methods discussed in the following chapters.

The specific modeling efforts are then described in 4 for temperature data modeling and chapter 5 which contains a primer on vibration-based monitoring and a description of the learning algorithms employed for vibration-based, supervised learning. The work is concluded in 6. The research papers can be found in appendices A, B and C.

Chapter 2

Wind power

2.1 The modern wind turbine

To provide the necessary background for discussing the monitoring aspects of modern wind turbines, some basic wind turbine design will first be described here briefly. The aerodynamic design principles will be gracefully ignored, and the focus will be on the key physical characteristics of wind turbines. The illustration in figure 2.1 show a typical configuration of a geared wind turbine. The wind is converted into rotational energy by the turbine rotor in a three-blade configuration. The blades are attached to the rotor hub, and can be pitched in an out of the wind by rotating the blades using a hydraulic system. The pitch system ensures that the wind production is optimized, and enables accurate control of the performance of the turbine. Modern wind turbine designs are mostly variable-speed, where the rotational speed of the rotor will move up and down with the wind speed. This should be contrasted with older designs where the speed typically was locked to the electrical grid frequency.

The rotor hub is mounted to the main shaft which is supported by one or two main bearings. At the rated power, the rotor and main shaft will typically rotate at a speed of 12-15 rpm. To efficiently convert this mechanical energy to electrical energy, the speed is first increased. This is done using a gear box which contains one or more planetary and helical gear stages depending on the

design requirements. From the gear box the speed of the output shaft is now close to 1500 rpm, which is then mounted to a generator in the back of the turbine, typically of the induction type. Variable-speed designs also require power electronics in the form of frequency converters before the connection to the grid. The entire house, or nacelle, of the wind turbine is mounted on a large yaw bearing on top of the tower that allows the turbine turn towards the wind as the wind direction changes. The components in the turbine drive train, consisting of the rotor bearing(s), gear box and generator are critical for the operation of the turbine, and exchange and repairs of these components require significant resources. Combined with the often random nature of faults in the drive train, the drive train is therefore the primary target for advanced condition monitoring systems as discussed in the later chapters.

It must be noted that significant variations exist within the overall turbine designs, particularly in terms of the drive train layout. Geared wind turbine designs have historically proven most popular, but turbine manufacturers are pursuing other designs to push reliability up and costs down. For example, the gear box has been completely removed in some turbine designs by using a low-speed synchronous generator (so called direct-drive technology). Hybrid solutions also exist using medium speed gear boxes and synchronous generators.

So how much power can a wind turbine produce? Most wind turbines sold for on-shore locations range from 2.0-3.5 MW of rated power, and this power is typically reached at wind speeds of 12-13 m/s. For offshore turbines, the rated power and turbine size increase significantly. The newest off-shore turbines are behemoths with rotor diameters of more than 150 meters and power ratings of 8 MW, with each turbine generating electricity for 8000 European households on a yearly basis.

2.2 Global wind energy trends

To illustrate how wind energy is changing the modern electricity system, it is enlightening to observe some numbers from one of the countries which embraced wind energy at an early stage, namely Denmark. In all of 2015, a record breaking 42% of the total Danish demand for electricity was supplied by wind turbines [1]. Moreover, the wind energy production exceeded the actual demand in a total of 409 hours throughout the year, with a peak supply of 138% compared to the demand measured a July morning in 2015. These numbers demonstrate not only the potential of wind energy, but also highlight some the challenges faced in the energy system when adopting wind energy on a wider scale.

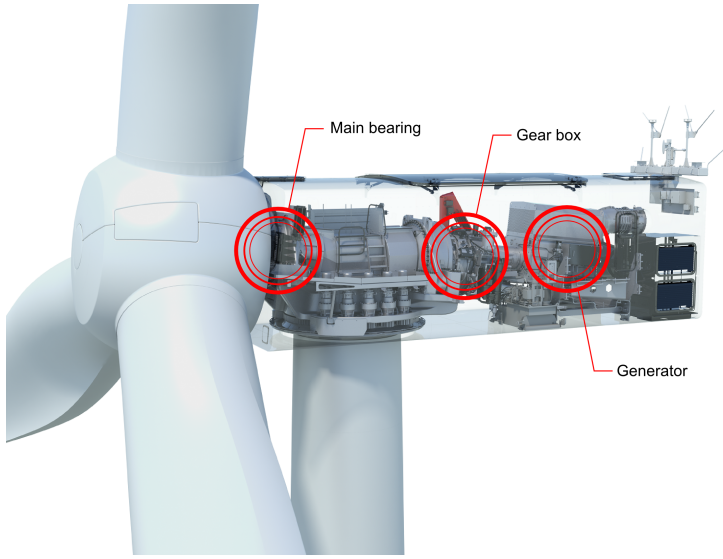


Figure 2.1: An inside view in a nacelle of a Siemens geared wind turbine.

At the global scale, the investments in wind energy have been growing steadily and the total cumulative wind production capacity is now 487 GW at the end of 2016 [2]. The global numbers are however quite different compared to the case of Denmark, as wind only accounts for approximately 4% of the global electricity supply. The cumulative capacity is summarized for each world region in figure 2.3. On the country level, China alone accounts for 169 GW of installed capacity, followed by the US (74 GW), Germany (45 GW), India (25 GW) and Spain (23 GW) measured in cumulative capacity. Is is noteworthy that wind energy is being deployed across all geographical regions, not only in developed countries.
Most wind energy has been deployed onshore, and offshore wind has historically been restricted to Northern Europe dominated by the UK (5.2 GW), Germany (4.1 GW), China (1.6 GW), Denmark (1.3 GW) and the Netherlands (1.1 GW). As of 2016 the total global off-shore capacity is 14.4 GW. The first offshore wind farm has recently been commissioned in the US, and the expansion of offshore wind is expected to grow in the next years globally.

To forecast the future deployment of wind energy, the Global Wind Energy Council operates with a number of different scenarios [2]. The Modest Scenario builds on the assumption that the agreements made at COP21 will be followed through. Under this scenario the global wind capacity is expected to grow to 1676 GW in 2030 and to 3984 GW in 2050. The most ambitious scenario (Advanced) projects a total installation of 5806 GW in 2050, while the most

Figure 2.2: Siemens 6 MW offshore turbines in operation. This is a direct-drive design where the drive train consists only of a single rotor bearing and a low-speed synchronous generator. Notice the heli-hoist platform mounted on each turbine which allows crew transfer by helicopter.

conservative scenario (NPS) projects 2870 GW in 2050. Under these scenarios, wind is expected to account for respectively 25%, 36% and 18% of the global electricity demand.

2.3 Cost reduction potentials

The cost of energy from a given source is often defined in terms of the Levelized Cost of Energy, LCOE, which factors in all projected costs related to the construction and operation of this particular energy source during its lifetime per produced unit of energy. These type of metrics are required in order to compare the cost of energy between different energy sources. All development in the last decades in the wind industry should be viewed as an ongoing effort to reduce the LCOE. This entails both the changes in the design of the turbines and their supporting structures, new production facilities, and innovations within logistic solutions and wind farm maintenance strategies. The motivation from the industry is to rely less on political decision processes and more on market powers offering energy at a price equal to or lower than competing energy sources. The large scale penetration of wind power is highly dependent on how the price point will move in the future (and also the price point of competing technologies of course). The most recent work on such estimates can be found in [3]. Based

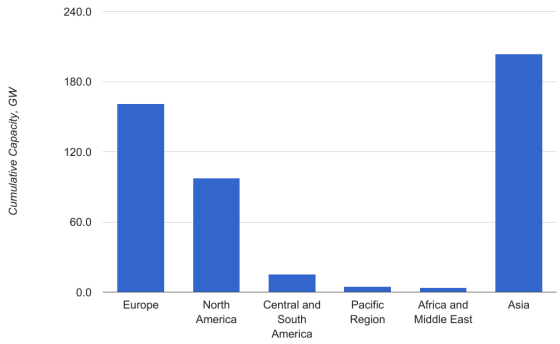


Figure 2.3: Cumulative wind power capacity by region as of 2016. China is the worlds biggest wind energy market with an installed capacity of 169 GW. Numbers are based on [2].

on a large expert elicitation study, the LCOE for onshore, offshore and floating offshore wind are projected throughout 2050. As of 2014, onshore wind was by far the lowest cost option with an LCOE of approx. 60 EUR/MWh compared with approx. 125 EUR/MWh for offshore wind. The prices are however expected to drop significantly as observed in figure 2.4 which show the projected LCOE for onshore wind, fixed offshore wind and floating offshore wind. Floating foundation offshore technology is not currently competitive, but the cost of floating solutions is expected to drop rapidly and converge to the price of fixed foundation offshore technology in 2050. In 2050, the price of offshore wind is expected to drop to approx. 80 EUR/MWh, but the price of offshore wind is still expected to be twice as expensive compared to onshore wind.

Five key factors have also been identified in the study across the different technological platforms that drive the costs down. These are

CAPEX is short for CAPital EXpenditure, and represents the upfront capital costs that are required to purchase, install and ready the equipment for energy production.

Capacity factor is a measure of the ratio between the average output power of the turbine, and the rated power of the turbine. Capacity factors typically range from 0.30-0.45, and capacity factors are typically higher for offshore wind farms due to the more steady wind conditions. Some design variations that aim to increase the capacity factor is higher turbine towers and larger rotors which are suitable for low wind conditions.

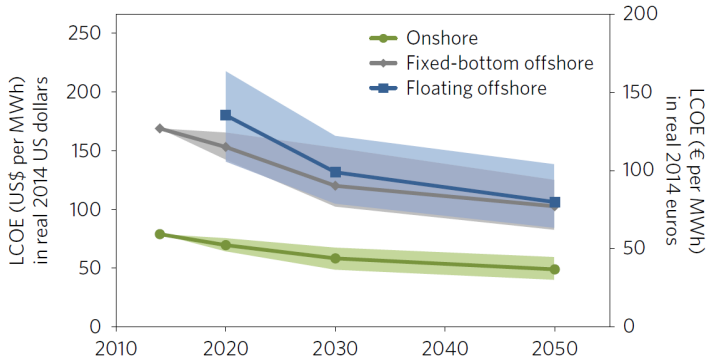


Figure 2.4: Estimates of trends in Levelized Cost of Energy (LCOE) for the three main technological platforms. Error bars indicate lower and upper quartiles of expert responses. The figure has been reused with permission from [3].

Financing includes the costs related to the financing of the project.

OPEX is short for OPerational EXpenditures, and includes costs related to the operation and maintenance of wind farms throughout their lifetime.

Project life is related to the expected life of a given project in which energy can be produced. Most wind farm projects sold today have projected lifetimes of 25 years.

When looking at the relative importance of these factors across the platforms in figure 2.5, some interesting insights can be extracted. For onshore wind we see that the key driver is improvements in capacity factor. More energy is simply expected to be generated from each turbine. For offshore wind, reductions in the capital expenditure is of more importance. Here, larger turbine designs with higher power ratings are a major factor in this development, with a projected average offshore turbine capacity of 11 MW in the year 2030. Interestingly, financial cost reductions seem to play a major role for lowering offshore wind LCOE. This indicates that investors will evaluate the risk (and interest rates) of offshore projects to be lower in the future as the maturity of this technology increases. The same effect does not apply for onshore wind, which seem to have bottomed out already now with respect to the perceived risk. Reductions in OPEX, which have the most direct link to the research presented in this thesis, is expected to account for 11% of the onshore relative cost reductions and 6% for offshore technology platforms.

Given the newest record breaking results from offshore tenders in Denmark, the question is if the median scenario considered in the above LCOE predictions

already is too pessimistic. The winning bids on the projects Vesterhav Syd & Nord (nearshore) and Kriegers Flak (offshore) reached an all-time-low price of approx. 56 EUR/MWh including the costs of grid connections [4]. Also, the newest data from the UK offshore projects [5] show that the average price has already dropped below 100 EUR/MWh, four years ahead of the previously set UK government targets.

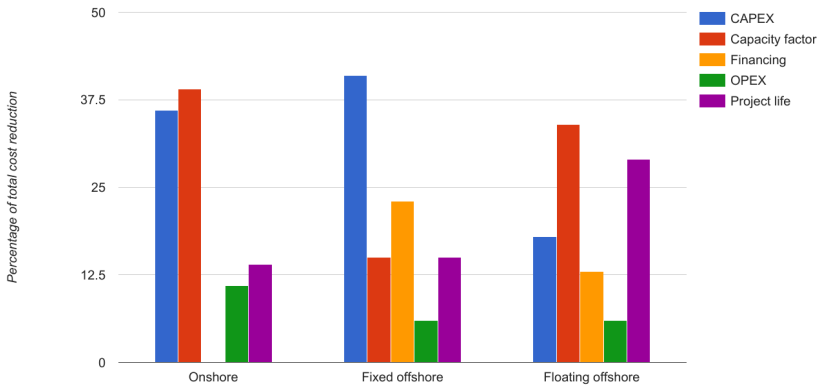


Figure 2.5: Breakdown of relative cost reduction potentials into key factors, split on the three major technology platforms. Numbers are based on [3].

2.4 Data-driven operation and maintenance

Modern wind farms are connected to a global data infrastructure which enable both remote control and remote diagnostics on the turbines. The often remote and intermittently inaccessible locations of wind turbines have pushed this development forward as discussed in the introduction. As also shown in the next chapters, a huge amount of operational data is being collected from each turbine, and some turbines also have advanced sensor systems installed specifically tailored to provide early fault detection on critical subcomponents.

One benefit of such an infrastructure is the potential for OPEX-reductions, mainly in terms of increased production from the turbines and maintenance-related cost reductions. For example, if the failure of a component can be predicted months ahead, the planning of the repair or replacement is greatly improved as spare parts, equipment and personnel resources can be ordered in due time, and the downtime of the turbine is kept at an minimum. It must

be remembered that repairs on wind turbines may require significant resources which are not readily available, for example in the case of rotor bearing or gear box replacements which require large cranes to be deployed. Adverse weather conditions may also prevent repairs for extended periods of time, and it may be advantageous to react to a an fault warning early to avoid long periods of downtime following a critical failure in a severe weather window. Finally, if faults are corrected at an early stage, secondary damage to equipment can be avoided. Variations of these points are often used as arguments favoring the use of predictive maintenance programs, but an often overlooked point is that such programs can provide valuable insight to the entire product engineering and improvement cycle. If faults are detected at an early stage, potential design problems or quality problems can more easily be identified which in the long-term may provide significant value which extends beyond the basic OPEX-related benefits.

The terms fault and failure have been used rather loosely, and perhaps it is worthwhile at this point to settle for a more terse definition of these terms. In this work, the definition from [6] is generally followed. To summarize, a fault is a condition of a machine that occurs when one of its components or assemblies degrades or exhibits abnormal behavior. A fault can in turn lead to a failure which is a termination of the ability of an item to perform a required function. Failure is an event as distinguished from fault, which is a state. A complete failure of the main capability of a machine is a catastrophic failure.

Chapter 3

Statistical learning

The next sections will provide the theoretical foundation on which the diagnostic learning algorithms are based on. As implied in the chapter title, this foundation is deduced from probabilistic theory and statistics, which provides a unified approach to formulating, training and evaluating such algorithms.

3.1 Learning paradigms

To further frame the approach taken in this work, we will first go through the three fundamental learning paradigms that exists within the field of machine learning:

Unsupervised learning: Given observations $\boldsymbol{x}$, unsupervised learning defines the process of finding structure in this data, or in probabilistic terms, estimating the probability density $p(\boldsymbol{x})$. The broad family of density estimation methods therefore fall under this category, including cluster analysis, principal component analysis (PCA) and related methods. A subset of these models are *generative* in the sense that new data points can be drawn from the model. A goal of generative modeling is to fit models such

that drawn data points will be statistically similar to the data the model was fitted on.

Supervised learning: In a supervised setting each input $\boldsymbol{x}$ is matched to a target t, and the goal is to learn the mapping from input to target. In probabilistic terms, we seek a model of the conditional distribution $p(t|\boldsymbol{x})$. With continuous target variables supervised learning is also known as *regression*. In case of discrete target variables (class labels), supervised learning is also known as *classification*. The major leaps taken in machine intelligence in the last decade has primarily been in the supervised domain, which requires an ample supply of ground truth target values to be trained effectively. In some scenarios, such values may be expensive or simply impossible to obtain which makes a supervised learning approach difficult.

Reinforcement learning: Given an agent, an environment in which the agent can perform actions and a reward function depending on the state of this environment, the goal of reinforcement learning is to find a behavior, also known as a policy, of this agent that will maximize the long-term reward. This could for example be an optimal control problem, where the agent is a control system acting on a physical system. Given the exploratory nature of this learning problem, one could argue that this is similar in nature to unsupervised learning, but the difference here is that the learning hinges entirely on a specified reward function. It also differs from supervised learning, since no optimal actions are presented to the agent during training.

Hybrids of these paradigms also exist. Semi-supervised learning is an attempt to bridge the strengths of the unsupervised and supervised learning paradigms to facilitate efficient learning on data where unlabeled data exist in abundance, but where labeled data are more scarce. Some learning systems also employ multi-step learning procedures, where different paradigms are employed sequentially, e.g. unsupervised learning followed by a discriminative supervised step or supervised learning followed by reinforcement learning [7]

As will be shown, the approach taken in chapter 4 is based on supervised learning using regression models, followed by an unsupervised fault detection step. In chapter 5 we will strictly use supervised learning in the discrete case (classification) where the models are trained using known input-target pairs. Reinforcement learning is mentioned here for the sake of completeness, but will not be discussed further.

3.2 Loss functions, maximum likelihood

The models considered here are all fitted using maximum likelihood parameter estimation. Formally, with observations $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \boldsymbol{x}_N]$ and $\boldsymbol{t} = [t_1, t_2, \dots t_N]$, we seek

$$\underset{\boldsymbol{w}}{\operatorname{argmax}}\, p\left(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}\right) \tag{3.1}$$

in the supervised setting where $p\left(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}\right)$ is a parameterization of the conditional likelihood with parameter vector $\boldsymbol{w}$. The learning problem is now cast as an optimization problem. It is customary to perform this optimization in the log-domain, as this prevents numerical underflow when working with small probabilities and has the property that multiplication of probabilities is transformed into sums. Also, a negative sign is typically prepended to the likelihood to transform the maximization problem into a minimization problem. We can now formulate the negative log-likelihood cost function, or error function, as a function of model parameters $\boldsymbol{w}$:

$$E\left(\boldsymbol{w}\right) = -\log p\left(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}\right) \tag{3.2}$$

$$= -\log \sum_{n=1}^{N} p\left(t_n|\boldsymbol{x}_n, \boldsymbol{w}\right), \tag{3.3}$$

using the basic assumption that the data points are independent and identically distributed (i.i.d.). The negative log-likelihood cost function forms the basis of all model training procedures in the following chapters.

Before diving deeper into actual model architectures and training algorithms, it is worthwhile to discuss commonalities across different architectures for tasks such as regression and classification in terms of the output parameterization and their related cost functions. In the case of regression, a common assumption is that the target values are corrupted by i.i.d noise terms, $t_n = g(\boldsymbol{x}_n) + \epsilon$, where $g(\boldsymbol{x})$ is the true mapping we wish to approximate and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the noise with variance σ^2. Under the Gaussian noise assumption, the problem reduces to a least-squares problem since

$$E\left(\boldsymbol{w}\right) \propto \sum_{n=1}^{N} \left(t_n - y\left(\boldsymbol{x}_n, \boldsymbol{w}\right)\right)^2, \tag{3.4}$$

where $y\left(\boldsymbol{x}, \boldsymbol{w}\right)$ is the model parameterization of $g(\boldsymbol{x})$. The cost in 3.4 is used directly in paper A for many of the regression models. For some model classes however, the noise variance is also assumed to depend on the input, and one has to revert to the full likelihood cost given in 3.3.

For two-class classification problems (denoted class A and B respectively), the target values, or ground truth labels in other words, are encoded as integers $t \in$

$\{0, 1\}$ where $t = 0$ indicates a class A affiliation, whereas a $t = 1$ indicates a class B affiliation. With $y(\boldsymbol{x}_n, \boldsymbol{w})$ as the parameterization of the class B probability (and $1 - y(\boldsymbol{x}_n, \boldsymbol{w})$ as the class A probability) the negative log-likelihood cost equals the cross-entropy cost function defined by

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N} \left(t_n \log y(\boldsymbol{x}_n, \boldsymbol{w}) + (1 - t_n) \log \left(1 - y(\boldsymbol{x}_n, \boldsymbol{w})\right)\right). \tag{3.5}$$

Data points with an incorrectly assigned low class probability are observed to be penalized heavily under this cost function. Examples of such two-class discriminative models can be found in paper B and C. In paper C the discriminative models are extended to scenarios where multiple discriminative tasks have to be solved by the models simultaneously (fault/no-fault on multiple physical components). Here, data from task-separate data sets are combined and presented to the the model as $(\boldsymbol{x}_n, t_n, k)$-tuples, where $k \in \{1, 2, \ldots K\}$ indicates which task each data point belong to. The output of the models is similarly extended to K outputs with the jth output denoted y^j. A masked cross-entropy cost function can now be defined as

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N} \sum_{j=1}^{K} \delta_{jk_n} \left(t_n \log y^j(\boldsymbol{x}_n, \boldsymbol{w}) + (1 - t_n) \log \left(1 - y^j(\boldsymbol{x}_n, \boldsymbol{w})\right)\right), \tag{3.6}$$

where δ_{jk_n} denotes a Kronecker delta.

3.3 Gradient-based learning

A number of distinct challenges present themselves when tackling the minimization problem governed by the various cost functions defined in section 3.2. First, analytical solutions only exist for the most simple class of models, and one must quickly resort to numerical optimization procedures. Second, convexity of the cost function can rarely be guaranteed and one cannot be certain that a found minimum is a global minimum. Third, in many cases both the number of model parameters and the number of data samples are very high, which has a direct influence on the choice of optimization strategy.

For all considered models in this work, the cost function is differentiable with respect to the model parameters which enables the use of gradient based optimization procedures. First-order gradient descent algorithms scale well to both complex model architectures and large data sets, and have been utilized heavily throughout this work. An overview of these methods is therefore presented here. In its simplest form, the parameter update steps in stochastic gradient descent

(SGD) are

$$\boldsymbol{w}_{\tau+1} = \boldsymbol{w}_\tau - \eta \nabla_{\boldsymbol{w}} \tilde{E}(\boldsymbol{w}_\tau) \ , \tag{3.7}$$

where η is the learning rate which controls the rate of descent on the cost surface, and $\nabla_{\boldsymbol{w}}$ is the vector differential operator w.r.t. parameters $\boldsymbol{w}$. Stochasticity is introduced by operating on an estimate $\tilde{E}(\boldsymbol{w}_\tau)$ of the true cost $E(\boldsymbol{w}_\tau)$. In one extreme, this estimate is simply based on individual data points. A full iteration of the dataset therefore provides N (noisy) updates in parameter space. In the other extreme, the estimate is based on the entire dataset and the procedure equals full batch gradient descent. In practice, the estimate is based on smaller minibatches drawn from the dataset, which provides a balance between the number of update steps per epoch and the variance of the estimated gradient. Here, an epoch is defined as a full iteration through the data set, and the samples are drawn randomly without substitution. Also, evaluation on a batch lend itself to fast parallelization on modern processors, which will speed up learning. The noise of the gradient estimate may intuitively be seen as a hindrance to efficient learning, but in complex models some noise is actually beneficial as it aides in escaping local minima and saddle points on the cost surface. SGD introduces some additional hyper-parameters into the learning problem, including the learning rate η and the batch size B.

In many situations the hyper parameters introduced here are adjusted *during* training. For example, the learning rate is often reduced as the learning progresses using either a gradual or stepwise reduction. Such schedules can in some cases improve learning. The gradient descent algorithm can also be extended. One popular extension to the basic SGD algorithm is stochastic gradient descent with momentum [8]:

$$\boldsymbol{v}_{\tau+1} = \boldsymbol{v}_\tau \gamma - \eta \nabla_{\boldsymbol{w}} \tilde{E}(\boldsymbol{w}_\tau) \tag{3.8}$$

$$\boldsymbol{w}_{\tau+1} = \boldsymbol{w}_\tau + \boldsymbol{v}_{\tau+1} \tag{3.9}$$

The gradient updates are here appended to the velocity term $\boldsymbol{v}$, which retains it momentum via the parameter $\gamma \in [0, 1]$. The velocity then provides the update to the parameters $\boldsymbol{w}$. Adaptive learning rate schemes such as ADAM [9] and AdaGrad [10] can also provide more efficient learning, particularly in deeper architectures. These methods are utilized in papers B and C respectively.

3.4 Training, validation, evaluation

Care must be taken when evaluating the performance of a given model, particularly when using likelihood scores as an evaluation criteria for models fitted

using maximum likelihood. The fundamental problem is the finite size of the data set. By fitting a model on a finite number of samples from a given process, the model may be able to describe the observed samples very well while failing to describe the underlying process which we are trying to capture. The result is that the model is not able to explain new data points drawn from the process, since it has essentially just memorized the observed data during training. This phenomenon is also known as *overfitting*. Overfitting is a concern when the number of data points is small and/or the model contains a large number of free parameters. An example is given in figure 3.1, where an overly complex model will adapt to the noise in the process, leading to poor generalization properties in the regions between the training data points.

To control overfitting, the data is often split in separate training, validation and test data sets. The training data set contains the samples observed by the model during training, while the validation set is used to observe how well the model generalizes to unobserved data points. Overfitting can be monitored by observing the likelihood scores of the training and validation set. A higher likelihood on the training set compared to the validation likelihood is a clear indication of overfitting. Within the training/validation framework, the model hyper parameters are adjusted until a desired performance on the validation is achieved. The error measure on the validation set can however be overly optimistic with respect to the model performance following a number of optimization steps of the model hyper parameters. The final performance is therefore measured using a test set. Alternatively, cross validation techniques can be employed, where model performance is averaged across different data splits. This is particularly useful when the data set is small and a high proportion of the samples are required for training.

3.5 Performance metrics for diagnostic systems

The test likelihood score is a relevant metric for how well the distribution of the model agrees with the distribution of the data. But for diagnostic purposes, other metrics could also be of interest. In a medical context for example, a doctor reviewing blood sample tests from a patient may be asking questions like "How sure can I be that this test will find an illness?" or "If the test is positive, how sure can I be that the patient is actually sick?". The context is of course critical when evaluating such tests. If the test, if we stay within the domain of medical metaphors, is used to diagnose a possibly lethal disease the implications of not identifying a sick patient (a false negative) may be unacceptable. Similarly, the implications of misidentifying a healthy person as being sick (a false positive) may be unacceptable if the treatment entails a large risk for the continued well

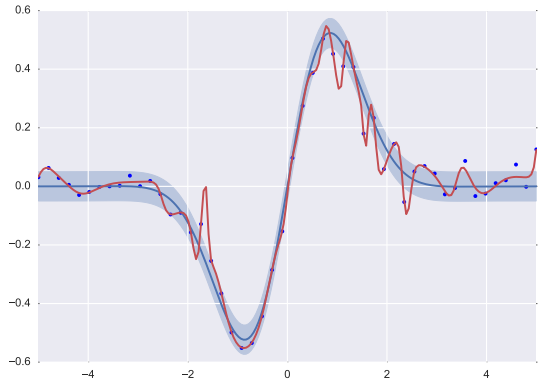


Figure 3.1: Highly flexible models or small sample sizes can lead to overfitting, as observed in this regression example where a neural network model (red) has been fitted to a set of data points (blue dots) drawn from a distribution (blue curve) which we seek to estimate. The shaded regions denote the noise standard deviations.

being of the patient. An overall accuracy metric is also problematic in the case of heavily skewed class distributions. Just by outputting negatives any diagnostic test can reach an accuracy of 99.999% if tested on the general population, if the purpose is to detect a rare disease that only appears in 1 out of 100,000 people.

The overall point here is that the same concerns appear in a large scale monitoring system for wind turbines. The financial or safety cost of not identifying some faults in due time may be extremely high. False positives may also bear financial costs, for example a shutdown of the turbine followed by downtime and loss of produced energy before the fault is handled and analyzed by a human operator (perhaps manageable), or the dispatchment of a full service crew to an offshore turbine (perhaps unacceptable). Also, the class fault distributions are highly skewed, and negative samples are (hopefully) much more abundant than positive samples.

This motivates to search for performance metrics that extend beyond the basic likelihood and accuracy scores to provide answers to these analytical challenges. Two related metrics will be presented here for this purpose, namely the Receiver-Operating-Characteristic (ROC) and the Area-Under-Curve score (AUC):

Receiver-Operating-Characteristic (ROC): The Receiver Operating Characteristic (the name is derived from its roots from radar detection systems) evaluates a given classifier in terms of its true positive and false positive

rates as functions of a decision threshold. The true positive rate is the ratio of true positives TP to the total number of positives P, tpr $= \frac{TP}{P}$, whereas the false positive rate is the ratio between the number of false positives and the total number of negatives N, fpr $= \frac{FP}{N}$. The plot of fpr vs. tpr is known as a ROC curve, and provides a valuable tool for determining suitable operating points for classifiers.

Area-Under-Curve The Area-Under-Curve, or AUC, is the *area* under the ROC curve which is a value between 0 and 1. The AUC provides an overall measure of the performance of a classifier, and can be interpreted as the probability that a classifier will rank a positive sample higher than a negative sample. An important feature of the AUC, and the ROC as well, is that the metric is invariant to class unbalance.

Figure 3.2 is used to illustrate these concepts further, as the ROC and AUC metrics have been used as principal comparison measures between diagnostic models in all papers. The figure is based on a two-class classification problem, with a positive indicating a fault state and a negative indicating a no-fault state. The two left figures visualize the distribution $p(x, C)$ for two different classifiers, where x is a continuous variable representing some input data transformation of the given classifier, and C is the class membership, with the red curve showing the distribution for positives, and the blue curve showing the distribution for negative samples. The value x could generally be viewed as a scalar transformation of some complex input space, and the distributions shown here would typically be viewed as the activation distribution in the last layer of a network model. The concept of a *score* is central in this context. Here we will use the value in x-space directly, but typically the score would be a an estimate of the posterior class probability $p(C|x)$ using an increasing function in x, e.g. an sigmoid, and the arguments still hold.

In our example case, the class distribution is somewhat skewed, with a 70%/30% distribution of negative/positive samples. By observing the distributions qualitatively, classifier B is observed to separate the two classes better compared to classifier A. We would therefore expect classifier B to outperform classifier A in this classification task.

In a detection setting, a detection threshold α must be set where an observation is categorized as a positive if $x > \alpha$. The choice of threshold is up to the system designer, and the ROC can be used to interpret the performance of the classifier based on this threshold setting. For example, the detection threshold which maximizes the overall accuracy is depicted by the green vertical line where $p(\text{Fault}|x) = p(\text{No fault}|x)$. This threshold value maps to a point in ROC-space, which is marked by a green cross on the individual classifier curves in the right most figure. Also, a mapping for threshold values under and above the maximum

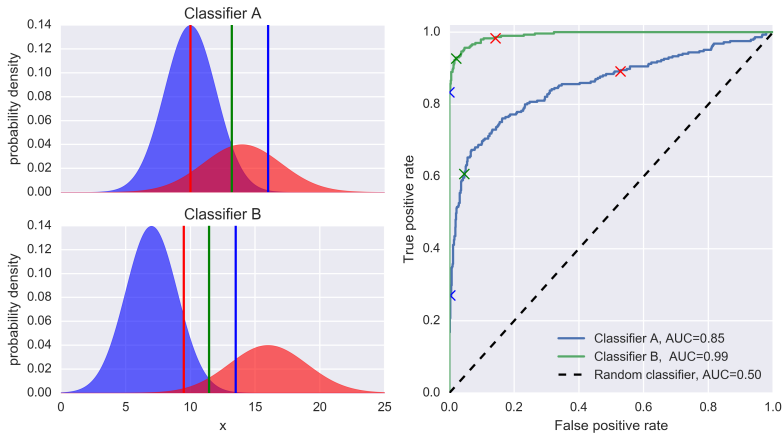


Figure 3.2: From decision thresholds to ROCs. The curves display some degree of randomness since they are based on a finite set of random samples.

accuracy threshold have been plotted for both classifiers, and their mappings in ROC-space can likewise be observed. As seen, a very sensitive classifier with a low detection threshold will ensure a high true positive rate. But, this will come at a cost since the false positive rate will also increase. Conversely, a classifier with low sensitivity will lower the false positive rate at the cost of a lower true positive rate. A classifier with a low false positive rate is said to very specific, since most detections will indicate an actual fault. One can however not be certain that all faults are detected.

By shifting the detection threshold continuously across the x-dimension, one obtains the ROC-curves for each of the classifiers. As mentioned, each point on the curve corresponds to an α-value. The curve for classifier B is observed to be closer to the upper left corner in all cases, meaning that a higher fpr or lower fpr always can be achieved by choosing classifier B over A. This is however not a general principle, and one often observes that different classifiers show different performance rankings in different regions of ROC-space. The AUC is employed here to characterize the performance of the classifier in overall terms as already mentioned. The area under curve B (AUC = 0.99) is observed to be higher than the area under curve A (AUC = 0.85). For comparison, a perfect classifier which is able to separate positives and negatives completely has a value of AUC = 1. A completely random classifier on the other hand, has a value of AUC = 0.5 with a diagonal line in ROC-space as also visualized in figure 3.2.

As a final note, the ROC and AUC metrics are made time dependent in the presented research. To view this in light of our simple example, one could view

the two classifiers as instantiations of the same classifier at different points in time. One at an early stage (classifier A) where the fault signal is still weak, and one at a later stage (classifier B) where the fault signal has increased and the classifier is more accurately able to separate fault from no-fault states.

3.6 Feed forward networks

With the basic concepts of model training and evaluation out of the way, the actual learning architectures can now be discussed in more detail. All models considered here belongs to the highly flexible class of deterministic feed-forward networks. These networks are deterministic since they provide a deterministic mapping from inputs to outputs with no internal/external stochasticity involved[1]. The models are feed-forward since information flows from input to output by a directed acyclic graph. This also means that these models are stateless, and only rely on the current input to provide an output. To not digress further, the basic architecture of a feed forward network is given below:

$$\begin{aligned}
\boldsymbol{h}_1 &= \boldsymbol{f}_1\left(\boldsymbol{W}_1^T\boldsymbol{x}\right)\\
\boldsymbol{h}_2 &= \boldsymbol{f}_2\left(\boldsymbol{W}_2^T\boldsymbol{h}_0\right)\\
&\vdots\\
\boldsymbol{h}_{L-1} &= \boldsymbol{f}_{L-1}\left(\boldsymbol{W}_{L-1}^T\boldsymbol{h}_{L-2}\right)\\
\boldsymbol{y} &= \boldsymbol{f}_L\left(\boldsymbol{W}_L^T\boldsymbol{h}_{L-1}\right),
\end{aligned} \tag{3.10}$$

where $\boldsymbol{x}$ is the input and $\boldsymbol{y}$ equals the output of the model as a vector in the general case. Each layer is constructed from a linear operation and an activation function $\boldsymbol{f}$ applied element wise. The weight matrix $\boldsymbol{W}$ defines the connections from layer to layer. The parameters of choice for each layer is then the shape of the weight matrix $\boldsymbol{W}$ and the activation function $\boldsymbol{f}$ which provides non-linearity if needed. The learnable parameters are the weights contained within $\boldsymbol{W_1}, \boldsymbol{W_2}, \ldots \boldsymbol{W_L}$, where an implicit bias term has is included by a constant input added to each layer in the above formulation. An overall architectural hyper parameter is of course the number of layers, L, which will be discussed again shortly.

The choice of final output transformation is determined by the learning problem. For regression problems, an identity function is typically applied providing a

[1]The training of these models can however be highly stochastic as already discussed in section 3.3

final linear transformation of the output

$$\boldsymbol{y} = \boldsymbol{W}_L^T \boldsymbol{h}_{L-1} \tag{3.11}$$

In order to parameterize both the mean and variance of a gaussian target variable, one may choose two output units as is the case for the heteroschedastic models considered in paper A. One unit is here kept linear, while an exponential transformation is applied to the second unit to ensure that $\sigma^2 > 0$.

For classification problems, the output is often used to parameterize a posterior class distribution. This is achieved by using a sigmoidal output unit given by

$$\boldsymbol{y} = \frac{1}{1 + \exp\left(-\boldsymbol{W}_L^T \boldsymbol{h}_{L-1}\right)} \tag{3.12}$$

which essentially squashes the output of each component of $\boldsymbol{y}$.

It will now be shown that familiar classes of models can be retrieved using the architecture described in eq. 3.10 using certain choice of parameters. First, by setting $\boldsymbol{y} = \boldsymbol{h}_1$ and using an identity output function, a linear regression model is obtained as

$$\boldsymbol{y} = \boldsymbol{W}_L^T \boldsymbol{x}. \tag{3.13}$$

By restricting the output to a scalar y in 3.13, and applying a sigmoidal output transformation for classification purposes, one obtains

$$y = p\left(\text{Fault}|\boldsymbol{x}\right) = \frac{1}{1 + \exp\left(-\boldsymbol{w}_L^T \boldsymbol{x}\right)}, \tag{3.14}$$

which is also known as a logistic regression model. Both of these models are amenable to optimization. In the linear regression case, an analytical solution exist if the usual squared cost function 3.4 is utilized. For logistic regression, the cross-entropy cost function 3.5 is guaranteed to be convex, and a minimum can be found efficiently using second-order methods[2].

If the models are now extended with additional non-linear layers we obtain the highly flexible model class of feed-forward neural networks. If we restrict ourselves to networks with only two layers, the mapping is parameterized as

$$\boldsymbol{y} = \boldsymbol{f}_2\left(\boldsymbol{W}_2^T \boldsymbol{f}_1\left(\boldsymbol{W}_1^T \boldsymbol{x}\right)\right), \tag{3.15}$$

where the choice of output activation function $\boldsymbol{f}_2$ again is problem specific. Non-linearity is injected by the activation function in the hidden layer, $\boldsymbol{f}_1$.

[2]For large data sets however, it may be necessary to revert to stochastic first-order optimization methods for both model classes.

The classical approach has been to apply squashing functions such as tanh in the hidden layer, and this has also been the choice for all two-layer neural network models in papers A, B and C. Compared to a linear regression model, a neural network is able to capture non-linearities in the data as for example visualized in figure 3.1. Similarly, the logistic regression model is only able to fit a linear decision boundary in the space of $\boldsymbol{x}$, whereas a neural network is able to accommodate non-linear decision boundaries if needed[3].

The flexibility of neural network models come at a cost. A basic hyper parameter is the number of hidden units, M. If the input dimensionality is denoted D, the size of $\boldsymbol{W}_1$ will scale as $D \times M$. Increased model capacity therefore comes at a cost of computational complexity, which is hardly a surprise. A bigger challenge is that the optimization properties of the linear models do generally *not* apply in the non-linear case. Convexity of the cost function can therefore no longer be guaranteed, and local minima and saddle points in parameter space makes learning more difficult.

3.7 Deep architectures

Although we still remain in the domain of feed-forward networks, the topic of deep learning deserves to be highlighted in its own section as deep learning has proven to be one of the major, if not the biggest, leap forward in the last decade within the field of machine learning. Deep learning has its roots in classical neural networks which we have already covered. The difference is that the number of layers in the architectures has increased significantly, i.e. the networks have become deeper compared to the shallow architectures discussed in section 3.6. The primary motivations for exploring deeper networks are

Biological, since certain biological systems of interest are observed to have similarities with deep feed forward neural networks, e.g. the visual cortex [11],

Theoretical, since deep networks can map certain functions more compactly compared to shallow networks [11], and finally

Empirical, because deep learning has been shown to work in complex processing domains, where existing methods have failed in reaching human like performance [12, 13, 14, 7].

[3] If the data has linear structure, the neural network should also ideally adapt to this linear structure

The recent success of deep learning can be credited to a number of factors, but a primary driver is raw computing power, typically provided by modern Graphics Processing Units (GPUs), that has made it feasible to train complex models within a reasonable time frame. Another driver is the large data sets which are now collected everywhere, since these architectures are data hungry. This thesis is also a testament to this development.

A type of architecture known as deep convolutional neural networks (CNNs) has been found to be particularly useful for vision related learning tasks [15] as these networks naturally exploit spatial structures in the data. As will be shown in later chapters, some of the data collected from wind turbines also have spatial characteristics, and CNNs are shown to provide excellent fault detection performance for a number of different drive train faults. As convolutional layers are the backbone of the architectures showcased in papers B and C, the basic concepts are covered in some detail for readers not familiar with vision-related machine learning systems.

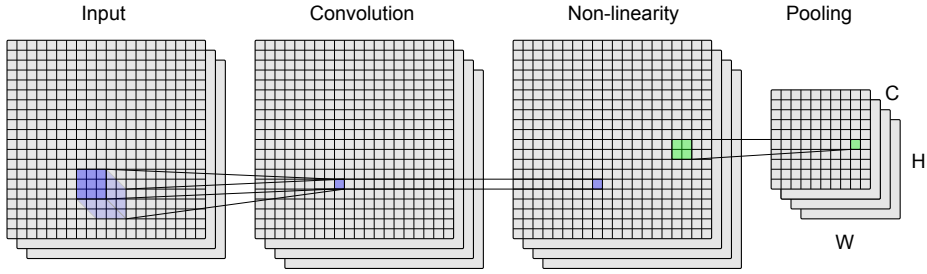


Figure 3.3: The basic building blocks in convolutional neural networks. The convolution is shown with a 3×3 kernel, using strides of 1 pixel in each direction. For the pooling operation, a 2×2 receptivity field is employed with strides of 2 pixels in each direction.

The building blocks of convolutional neural networks are illustrated in figure 3.3. The input layer is a 2D feature map with a number of distinct channels, in this case three. Each value here is denoted as a pixel. An example of such an input could be an image with three RGB-channels. A kernel spanning only small rectangular area in the spatial dimensions and spanning the full depth of the input provides a linear mapping to a single scalar value. By sliding the kernel along the width and height one obtains a new spatial feature map, processed by this specific kernel. If more kernels are utilized, one obtains several new feature maps one for each kernel. This step is denoted convolution in the diagram. Here, the kernel size is 3×3 and a single mapping for one kernel is illustrated in the figure. The number of pixels the kernel is moved between each processing step is known as the stride of the kernel which have been set to 1

pixel here. The number of distinct kernels is in this case four signified by the four output layers. Zero-padding has been implied at the borders of the input map to preserve the spatial dimensions of the output following the convolution. The kernels can be regarded as spatial filters, which the network can utilize to extract features in the input image. Following the convolution, a non-linearity is applied element-wise which is shown in the non-linearity step. The size of the map is here preserved. The non-linearity could be a squashing function as typically employed for classical neural networks, but a popular approach is to use a rectified linear unit [16] $\text{ReLu}(x) = \max(0, x)$ in deeper networks, as this type of non-linearity can mitigate vanishing gradient problems and speed up learning of the lower level layers. In the final step, a pooling layer is used to down-sample the feature map. Here, the operation is again defined by a receptive field in the spatial dimensions on which the down-sampling is performed, typically via a max-operator. This layer reduces the number of features and the complexity of the network, and provides some invariance to translations in the input. Here, the size of the max-pooling operation is observed to be 2×2, with a stride of 2 providing a down-sampling factor of 2 in both height and width dimensions.

When comparing this network structure with the classical feed-forward architectures described in the previous section, it is perhaps realized that CNNs can be described as a classical neural network with only a local connectivity between the layers with shared weights across the layer units. As a result, the number of free parameters is reduced significantly compared to a fully connected neural network, and the number of weights only scales as $k_n^2 C_{n-1} C_n + C_n$ where n denotes the layer index, k denotes the size of a uniformly sized kernel and C_n denotes the number of channels for a given layer. The primary hyper-parameters of CNNs are of course the size of the kernels for each layer, the number of kernels, the strides and choices of activation function and pooling operator.

In an actual deep network, the convolutional layers are typically followed by one or more fully connected layers to provide the last classification processing. Examples of this can be found in both paper B and C. It must also be noted that each convolution operation is not always followed by a pooling operation and stacks of several subsequent convolutions are often employed as for example in [17].

Chapter 4

Predictive modeling and unsupervised fault detection

With our brief review of both wind turbine technology and statistical learning concepts in the preceding chapter, the stage has now been set to investigate how data-driven methods can be utilized for wind turbine fault detection tasks. In this journey, methods based on modeling of wind farm SCADA-data[1] will first be explored based on the research presented in paper A. The basic concept is to extract more detailed knowledge of the state of an industrial system or process by applying a further modeling layer to data streams collected for supervisory control purposes. This idea is not new and early efforts to use the data in a fault detection context includes [18] for general industrial purposes and [19] for wind farm specific purposes.

This chapter will only discuss the research motivations and overall data characteristics and modeling concepts. For details, we refer to paper A and the provided references as the topic of SCADA-data modeling is a mature research field to the wide availability of industrial operational data. The data consists of sensor values samples equidistantly in time, and readers familiar with basic time series modeling should quickly be able to pick up on the methods presented in the following sections.

[1] Short for Supervisory Control And Data Acquisition

4.1 Problem domain and methodology

For wind turbines, the modeling scope can range from be the overall turbine performance, e.g. in terms of power curve modeling down to detailed analysis of the health state of specific subcomponents such as the rotor bearing which is the subject here. The methods presented are however quite general in nature and should be applicable to a wide range of domains without much alteration. Research into wind turbine bearing temperature-based data modeling already exist [20], also for rotor bearings [21, 22]. The motivation behind the work presented here is to extend the existing research with 1) an exploration of richer data models that capture the longer range linear and non-linear dynamics within the operational data, and 2) a detailed analysis of model performance when applied to a both a sizable turbine fleet and a sizable amount of actual turbine faults. The last item is particularly important, as existing work on SCADA-based fault detection in wind turbines has only included very few faults, hereby making it difficult to establish more general properties of the presented models in an actual monitoring setting.

The use case for accurate fault detection for the rotor bearing(s) has already been motivated in chapter 2. The faults included in this study are spalling damages to the bearing raceways. As micro-fractures in the raceway surface converge, material is released from the surface creating a crater in the surface. Once this failure mechanism has initiated, the process will typically accelerate until large parts of the bearing raceways are damaged [23]. Progressed damage will lead to increased friction, which in turn will generate heat in the bearing. Temperature sensors are therefore often employed as part of the monitoring system in wind turbines [24] in order to detect bearing overheating a result of mechanical faults, or insufficient lubrication. As will be shown in chapter 5, vibration-based monitoring is the method of choice for detection of small localized damage in drive train bearings, but temperature-based monitoring can provide valuable diagnostic input as a complement to vibration-based monitoring as demonstrated in the following chapters.

The basic challenge with wind turbine diagnostic applications is the inherently stochastic state of the wind turbine and its environment, an issue which also rears its head when working with temperature data. Consider the temperature trend from a rotor bearing temperature sensor depicted in 4.1, spanning several years. This trend has been obtained from a rotor bearing which eventually failed due to overheating at the end of the depicted time interval. Some striking features of this trend are 1) the large seasonal variation in the mean and 2) the high variance around the mean which depends both on the day/night temperature cycle and the operating state of the turbine. From a human operators' perspective, the temperature variation makes it difficult to evaluate when the

bearing temperature is within an acceptable range. Similarly, the limitations of a simple temperature-threshold warning system become immediately obvious, and one would necessarily expect such a system to provide a late warning in order to limit the number of false positives given the signal variance for the individual turbine as well across wind farms in different geographical regions. Clearly, a monitoring system could benefit from a more adaptive approach.

The approach taken in the following is a two-step procedure:

1. **No-fault state modeling:** Based on certain basic assumptions, a data-driven model is established for each individual turbine representing a *fault-free state*.
2. **Outlier detection:** New data points are now evaluated in terms of how likely they are under the established no-fault state model. Outliers are interpreted as potential faults.

Each of these steps will now be discussed from a slightly more general perspective compared to the discussion in paper A.

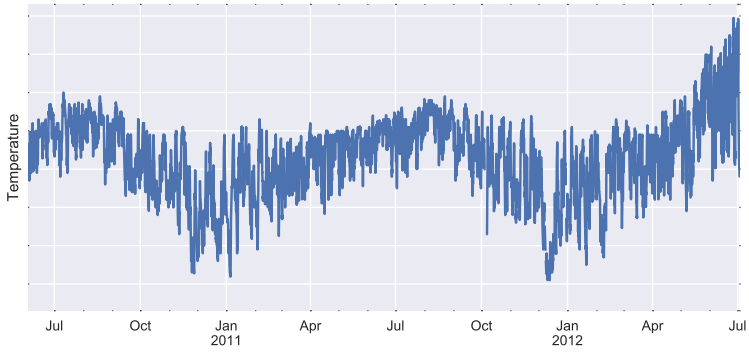


Figure 4.1: Rotor bearing temperature

4.2 No-fault state modeling

Consider a number of sensor readings denoted by the vector $\boldsymbol{x}$. The readings are sampled in time, but this dependence will be ignored initially. In some cases it is possible to simplify the problem by exploiting prior knowledge of the of the

underlying physical system to establish an functional relation. If $\boldsymbol{x}$ is split into input $\boldsymbol{u}$ denoted as *exogenous* variables and output $\boldsymbol{t}$, we then seek a model of the type

$$\boldsymbol{t} = \mu(\boldsymbol{u}) + \epsilon \tag{4.1}$$

in the simplest case with i.i.d. noise terms ϵ. One could attempt to formulate the mapping $\mu(\boldsymbol{u})$ from first principles, but even for simple systems this may prove to be difficult. If enough data samples have been collected from the system, a data-driven approach can instead be adopted where flexible models are fitted to the data collected from the system. From a statistical learning perspective, equation 4.1 is recognized as a regression problem.

The bearing temperature models provides a good example of this practice. Based on a simple thermal flow analysis of the drive train, a number of operational values are chosen which are indicative of the overall thermal state of the system. Given these values as input $\boldsymbol{u}$ we now seek a predictive model for the bearing temperature t, and this model is fitted using the data collected during a fault-free period. If the temperature start to deviate from the predicted temperature, this might indicate an impeding bearing failure. One could also consider the model formulation to be biased towards a specific fault state based on knowledge about the physical properties of the underlying system.

There is some finer points to the conditional modeling aspect as also discussed in paper A when using the conditional models for outlier detection. One must generally require that given the transition to a number of fault states of interest, the support of the input density $p(\boldsymbol{u})$ does not change since the conditional distribution $p(\boldsymbol{t}|\boldsymbol{u})$ is not strictly defined outside this region. For example, given a number of mild assumptions, the support stationarity can expected to hold for the exogenous variables used for the rotor bearing temperature models. If past values of the bearing temperature is included as input as in the case of the the auto-regressive (AR) models in paper A the support stationarity cannot be expected to hold, and the use of the model prediction in a outlier detection context become highly questionable. This has also been demonstrated in the article, where highly accurate AR predictive models can be demonstrated, but where the fault detection capability is restricted compared to conditional models that operate on exogenous inputs only. If the support stationarity cannot be established for a given system, a more prudent approach would perhaps be to model $p(\boldsymbol{x})$ directly instead. This approach is however not without difficulties, as density estimation is a difficult task for high-dimensional data and interpretation of outliers may also prove difficult when employed later for fault detection.

At the end of this section is is worthwhile to discuss the overall merits, and pitfalls, of no-fault state modeling. A major benefit of the presented methods is that no actual fault data are required which makes the methods valuable in any

domain where faults are a rare occurrence and/or the nature of these faults are generally unknown. This could for example be in small turbine monitoring setup with only a few turbines, or a structural health monitoring system installed on a bridge. The downside of this methodology is that the systems are not adapted to specific faults, which can make them sensitive to noise and other state changes in the system which are not fault-related. Of course, one must also posses knowledge of when the system is actually fault-free. In the presented paper, it is simply assumed that the rotor bearing is fault free in the first year of operation, and the models are then fitted to this data period. In more complex cases, this clear no-fault state distinction is not possible, and one may use unsupervised methods to separate the different states of the system. Finally, the no-fault state methodology enables specific diagnostic measures that may prove highly valuable in a diagnostic and prognostic setting. In the case of the rotor bearing temperature modeling, this measure is the temperature residual which is straightforward to interpret as the residual stemming from the fault mechanism itself.

4.3 Modeling of sequential data

Until now, the sequential nature of the data has largely been ignored. Ideally, one would like to include both short-and long-term dynamics in the predictive model. A straight-forward way to include this in any model architecture is by expanding the input space to a data window spanning L time steps. By denoting the index in time as subscript n, the predictive model now takes the form

$$p\left(t_n \mid \boldsymbol{u}_{n-L+1}^T, \boldsymbol{u}_{n-L+2}^T \dots \boldsymbol{u}_n^T\right), \tag{4.2}$$

which is also known as a *sliding window* approach. This methodology has been used extensively throughout this work. The window width L then determines the maximum time scale from which the models are able to capture the signal dynamics. This author advocates the general use of rich models and long window lengths in contrast to the methods in most existing wind turbine literature which rarely employ more than one time step in the predictive step. This advice comes with the caveat that the data set size should be able to support such rich models possibly combined with one form of regularization to avoid overfitting.

The advantage of using the sliding-window approach is that the feed-forward nature of the network models employed in this work is kept intact. One could also consider sequential models such as recurrent neural networks. The sliding window methods were found to provide an excellent performance across the different data sets in this thesis, but recurrent neural network in the form of Long Short-Term Memory (LSTM) architectures [25] were tested on the temperature

data set to investigate if further predictive performance could be gained. This was found not to be the case. As also argued in paper A based on [26], recurrent architectures are not expected to perform well in a problem domain with fixed range dynamics, a case much better suited for sliding window approaches. Readers working similar data sets are therefore advised to first test models based on sliding windows, and to only explore more complex architectures if these initial efforts fail.

4.4 Fault detection and evaluation strategy

For probabilistic models, a suitable metric for evaluating a new data point $\boldsymbol{u}_n, \boldsymbol{t}_n$, would simply be the likelihood $p(\boldsymbol{t}_n|\boldsymbol{u}_n)$ or $p(\boldsymbol{t}_n, \boldsymbol{u}_n)$ depending on the modeling approach. An outlier detection system could then be specified via a likelihood warning threshold. This approach was tested in paper A, but the likelihood measure was not found to be robust for long-term monitoring due to instationarities in the underlying turbine systems such as wear and tear. The simple temperature residual $r_n = t_n - \mu(\boldsymbol{u}_n)$ was found to be a better choice for outlier detection. This rudimentary detection strategy is easily extendable, and the temperature residual lends itself to further processing in more advanced detection schemes. The residual can here be interpreted as a distance feature from which the variation due to changing operational conditions are stripped away. Further processing of the residuals has been taken up in [27] where a Bayesian state prediction scheme is applied to the output of the bearing temperature models presented here.

The remainder of this chapter is now spent on the basic concepts of the fault detection procedure. With slight variations, the core parts of this procedure will be used throughout most of the work in this thesis. The reader is first reminded of the basic concepts of Receiver-Operating-Characteristics and Area-Under-Curve metrics illustrated in chapter 3. The underlying scores for the purpose of temperature-based fault detection are simply the residuals as discussed earlier. By observing figure 4.2 showing the development of a temperature residual for a faulty rotor bearing, the mean of the residual is observed to increase almost monotonically before the bearing fails at the rightmost point in time of the plot. The end-of-life for the bearings are defined as the point in time when the critical temperature of the bearing is reached and the turbine is shut down automatically for safety reasons. The time of failure is also used as a reference point for each turbine, and the residual is now measured in terms of Δt, which is the time until bearing failure. This definition aligns the residuals across turbines, and the classifier performance is now measured as a function of Δt. As the residuals increases towards the point of failure, the ability of the detection system to

separate fault states from no-fault states is expected to also increase.

The evaluation procedure is as follows: For a given value of Δt, illustrated in the figure as vertical blue line, residuals are collected across the population of turbines with recorded bearing faults. Similarly, residuals are collected from turbines without bearing faults. Operating points on the ROC-curve can now be generated based on the collected residuals by adjusting the detection threshold. For a sensitive system (green horizontal line), the residuals for the negatives and positive samples will largely overlap and the number of false positive is expected to be high. For a less sensitive system (red horizontal line) the number of false positives are expected to drop, but a detection of a fault is not guaranteed as is the case for the turbine shown here. Remember, the graph here only show the residual for one turbine but the performance is evaluated across the entire population of faulty turbines aligned in Δt and the negatives included from fault-free turbines. The value of Δt is now adjusted, and a new ROC-curve, and AUC-value, can be generated. As a result, time-dependent performance metrics can be generated for each predictive model architecture.

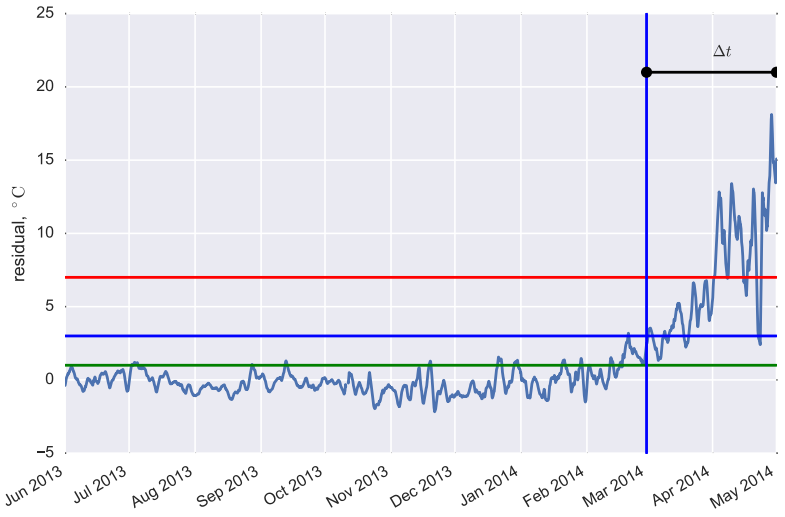


Figure 4.2: Rotor bearing temperature residual plotted for a bearing with a confirmed fault. The residual is observed to increase gradually over several months before the bearing fails.

4.5 Summary

A two-step procedure for data-driven fault detection based on SCADA-like data was outlined in this chapter. The first step entails a predictive model for a specific subsystem based on general flexible models that are fitted to data collected from the system. Once deployed, the residual between the model and the actual sensor readings can be used as a fault indicator. The reader is now urged to review paper A to get a clearer picture of the modeling details and the potential of this methodology when applied to turbine drive train sensor data. Here, the results show that the modeling outlined here is able to push the warning lead time ahead 45 days on average, compared to the hard-threshold warning system.

This section ends the journey into temperature-based fault detection models, and the following section will explore how vibration-based detection models can be utilized for not only rotor bearing faults, but also gear box bearing faults. The results from the temperature models will however be revisited in 5.3 when the results from the various methods will be combined for direct comparison.

Chapter 5

Supervised fault detection

From a loosely defined fault model, it was demonstrated in the previous chapter how valuable diagnostic information could be gained by framing the problem as an outlier detection problem by fitting predictive models to simple data streams from assumed no-fault periods. We will now consider a second class of data streams, namely vibration data collected from a number of sensors mounted on the primary components of the turbine drive train. Such sensors are typically installed with the specific task of early fault detection in mind, and vibration based monitoring will in general provide earlier warnings of mechanical faults in the drive train compared to e.g. purely temperature-based methods [28].

It is trivial to extend the approach of no-fault state modeling to vibration signals, and this approach form the basis of most practical implementations of vibration-based monitoring systems, particularly in wind turbines. Vibration monitoring systems can potentially provide extremely rich signals and a number of preprocessing steps are typically employed to reduce the data to a limited number of scalar values indicative of different failure modes. The alarm thresholds are either fixed a priori or adapted during a run-in period. In the adaptive case, the data model is often limited to fitting and evaluation by simple parametric distributions on each scalar stream. This methodology present a number of closely related challenges:

1. The system relies heavily on **feature engineering** which may not capture

the full complexity and noise of the signals relevant for condition monitoring purposes. This leads to reduced sensitivity and high false positive rates for the system.

2. To mitigate the drawbacks mentioned above, the warnings from these systems are often further handled by human experts in order to separate true positives from false positives. This leads to **poor scalability** and reaction times, and a constantly updated view of the health state of the mechanical components is generally not possible.

3. No structured approach exist for sensors and system data **fusion**.

By first recognizing that human experts are able to detect faults early and accurately from analysis of the data collected in the Siemens large scale monitoring platform which form the basis of this thesis, one might speculate if a learning system could be devised which learns from the actions from these experts to provide a fully automated virtual expert-like system. This is the primary research question which will be investigated in this chapter.

The chapter starts with a primer on vibration-based condition monitoring to give readers who are unfamiliar with this field an understanding of the basic techniques employed for fault detection. The machine learning approach to he problem domain is then described in section 5.2, based on the research papers found in papers B and C.

5.1 A primer on vibration-based turbine diagnostics

In this section the basic methods of vibration-based monitoring is discussed with an emphasis on wind turbine monitoring applications, particularly bearing diagnostics. From both a practical and theoretical perspective this topic is very rich, and the brief overview given here is by no means intended as full tutorial in applied vibration-based condition monitoring. The intention is to give the reader a basic understanding of why this technology is the method-of-choice for rotating machinery monitoring applications and to provide a description of the analysis involved in such applications, aspects which have mostly been left out in the presented research articles for the sake of brevity. For more details on the general subject of vibration measurement and analysis, readers are referred to [29]. For details on vibration-based condition monitoring, [30] provides a comprehensive review of the field including many of the signal modeling details which is only briefly touched upon here.

The vibration pattern in a wind turbine is complex. At the lowest frequencies below a few Herz, the vibrations are dominated by forcing components from the rotor-wind interaction and the basic response from the overall turbine structure. In the case of geared turbine designs, the medium frequency range extending up to 3 Khz is largely dominated by vibrations from the gear box when we restrict our attention to the sensors mounted near the drive train components. In the case of direct drive designs, this frequency range is dominated by generator and bearing dynamics. For the higher frequency range the signals become highly complex as local structural dynamics mix with excitation from high to intermediate speed bearings, the generator and power electronics.

The vibration signals discussed here will be in units of acceleration. Industrial accelerometers of the piezoelectric type are cost-effective, robust and provide an adequate frequency response and sensitivity for the applications discussed here, and are therefore the primary choice for wind turbine drive train monitoring.

5.1.1 Gear stage fault detection

First, we will consider the dynamic forcing from the gear box. The overall gear ratio is determined by the teeth ratios throughout the different gear stages. On typical wind turbine gear box designs the first stage is planetary, possibly followed by a second planetary stage[1]. After the planetary stage(s), one or two simple helical stages provide the final speed-up of the output shaft.

The gear mesh interactions during operation will produce distinct vibration frequencies. For a simple gear, the fundamental gear mesh frequency is simply determined by

$$f_g = f_r N_g, \tag{5.1}$$

where f_r is the rotational frequency of the gear shaft and N_g is the number of gear teeth. The gear signal consists of this fundamental gear mesh frequency and its higher harmonics. The amplitude of these coefficients will depend non-trivially on the gear geometry, loading and wear and tear. In a gear box with multiple stages, the overall signal can be expected to be a combination of various harmonic families, one for each stage. This is exemplified in figure 5.1 showing an acceleration amplitude spectrum obtained from a gear box sensor. Multiple harmonics are observed from a gear stage with a lower fundamental mesh frequency combined with a single peak which is the fundamental frequency of a secondary gear stage. Some simple gear fault signal models can easily be derived. For a gear wheel, a fault state would include gear tooth surface degradation or material fractures. Consider the case where a single tooth on a gear

[1] see https://www.youtube.com/watch?v=tfT7Vrv71J0 for an animation of planetary gears

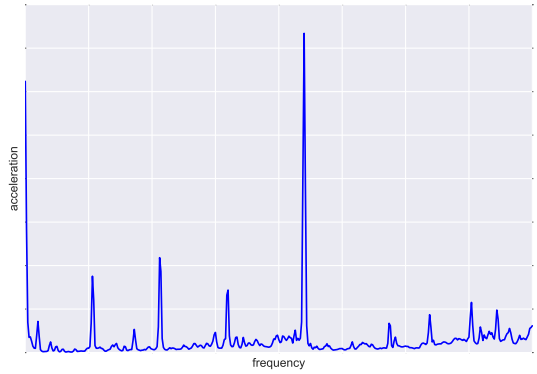


Figure 5.1: Spectrum of acceleration signal measured on a gear box. Discrete harmonics of the gear mesh frequency can be observed from different gear stages.

wheel develop a fracture. This changes the stiffness of the tooth, and the meshing interaction then changes every time this particular tooth is in interaction with the opposing wheel. The effect is that the forcing function is modulated by a signal with a fundamental frequency of f_r, the rotational frequency of the gear shaft. As the modulation is a multiplication in the time domain, it is quickly realized that the fault will appear as f_r-spaced sidebands to the mesh frequencies, as the signals are convolved in the frequency domain.

The above analysis on the gear box highlights why basic frequency analysis is such a fundamental tool for monitoring rotating equipment. In the frequency domain, the signals from different gear stages can be easily separated and identified, given that we know the mechanical configuration of the system. For example, if increasing sidebands are observed around a given frequency peak, the location of this peak will tell us in which stage a fault may be present, and the separation between the sidebands will identify the specific wheel. Moreover, the signals propagate more or less easily throughout the gear box structure, and using just a few sensors mounted on the gear box casing one is able to identify faults on the tens of different gear wheels and bearings within the gear box.

The example of gear wheel fractures is more than just an illustrative example of basic vibration analysis. The loss of a tooth is here defined as a gear failure since continued operation may result in serious damage to the entire drive train, particularly if the failure happens in the high-torque section of the gear box. Gear wheel failures are actually one of the primary failure modes targeted by vibration based monitoring systems in wind turbines. Such faults can progress from a small defect to a full surface-to-surface crack in hours depending on the location of the defect. In other cases, the fault may develop for weeks before

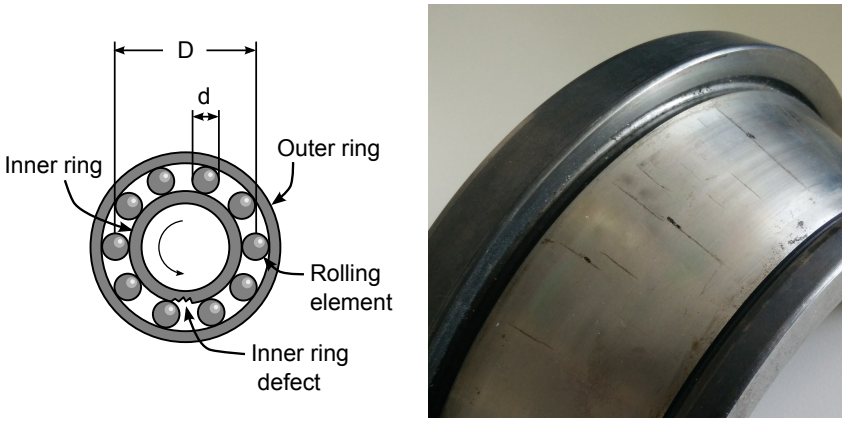
they result in gear failure.

Gear wheel faults will not be given much attention in the remainder of this work, as the work strictly considers bearing fault detection. Some of the basic principles for gear wheel fault detection does however transfer to bearing fault detection also as shown in the next section.

5.1.2 Bearing fault detection

Tens of different bearings are located throughout a geared wind turbine drive train, including the large rotor bearing(s), the different gear stages and the generator. Mechanical faults in drive train bearings are the second major target of advanced vibration monitoring systems in wind turbines. Compared to gear wheel faults, bearing faults generally progress at a slower rate and the turbine can continue operation for an extended period of time before a replacement is needed.

Most wind turbine bearings are of the rolling element type. This includes ball bearings, and various designs of cylindrical, tapered and spherical roller bearings. The typical faults encountered in wind applications include fractures, spalling and pitting of the raceway surfaces. A diagram of a simple rolling element bearing can be observed in figure 5.2a including the inner and outer rings and the rolling elements. Typically one of the rings is held stationary, while the other rotates (in this case the inner ring is rotating). Also, a small inner ring surface defect has been illustrated where material is released from the raceway surface. An example of actual raceway damage is depicted in 5.2b. The bearing inner raceway displays surface fractures in the axial direction and small spalling areas have begun to appear at several points near the fracture lines. As was the case for gears, bearing faults manifest themselves with distinct frequency domain characteristics. Consider the case of a localized defect on the inner ring raceway as depicted in figure 5.2b. A small impact would be expected each time a rolling element passes this defect. The impacts are repeated at the frequency in which the rolling elements are passing the defect. This frequency is known as the Ball Pass Frequency of the Inner race (BPFI, or RPFI for roller bearings). Similarly, distinct frequencies can be derived for outer raceway faults (BPFO, or RSFO for rollers) and rolling element faults (BSF, or RSF for rollers). The



(a) Rolling element bearing. **(b)** Inner raceway damage.

Figure 5.2: Diagram of rolling element bearing and actual bearing damage.

frequencies are given by the following geometric relations [30]:

$$\text{BPFO} = \frac{N_r f_r}{2}\left(1 - \frac{d}{D}\cos\phi\right) \tag{5.2}$$

$$\text{BPFI} = \frac{N_r f_r}{2}\left(1 + \frac{d}{D}\cos\phi\right) \tag{5.3}$$

$$\text{BSF} = \frac{D f_r}{2d}\left(1 - \left(\frac{d}{D}\cos\phi\right)^2\right) \tag{5.4}$$

where N_r is the number of rolling elements, D is the pitch diameter, d is the rolling element diameter and ϕ is the contact angle between raceways and the rolling element. Each bearing therefore has a unique frequency fingerprint which can be calculated if the basic properties of the bearings are known. The appearance of a specific frequency peak can then be attributed to not only a singular bearing, but also to the individual parts of this bearing such as the inner ring, outer ring or rollers. It must be noted that the relations are based on the important assumption of no-slip conditions between rollers and raceways which rarely holds in practice. As opposed to the deterministic gear signals, the bearing signal is somewhat random and the fault frequencies may deviate from the values calculated from 5.2-5.4.

For small localized faults, the dynamic forcing is often impulse-like which provides a wide band excitation of the bearing housing and surrounding structures. For early fault detection, the signal may be masked by other sources such as gear noise. Here, a useful processing step known as envelope analysis [30] may help to increase the bearing fault signal-to-noise ratio. The overall processing steps

in envelope analysis have been visualized in figure 5.3. As shown in the leftmost figure, the low to mid frequency domain is often dominated by gear meshing harmonics and stochastic gear noise. But, due to the wide band nature of the bearing fault signal, structural resonances located in the higher frequency domain is excited by the fault impulses. By carefully choosing a suitable band pass filter near these resonances, one can obtain a time signal resembling the lower figure. By demodulating the signal which extracts the lower frequency information in the amplitude signal followed by a frequency domain transformation, the harmonics of the bearing fault frequency can be retrieved as illustrated in the rightmost figure. An elegant way to extract the envelope is by forming the analytic signal $y_a(t)$ from the band passed signal $y_b(t)$ and the Hilbert transform $\mathcal{H}$ [30]:

$$y_a(t) = y_b(t) + j\left(\frac{1}{\pi t} * y_b(t)\right) \tag{5.5}$$

$$= y_b(t) + j\mathcal{H}\left(y_b(t)\right) \tag{5.6}$$

where $*$ denotes convolution. Now the low frequency modulation signal, highlighted in green in figure 5.3, can be extracted as the modulus of the analytic signal

$$a(t) = |y_a(t)|. \tag{5.7}$$

Envelope analysis requires a suitable choice of bandpass filter from which the fault signal is extracted. This choice could be made on the basis of structural testing or simulations, or adaptive methods could be utilized which seek to maximize the impulsiveness of the filtered signal. In practice, a number of fixed filters are employed which may be adjusted as actual fault data start to be collected from the fleet.

Additional factors can influence the signals from bearing faults. For example, if the bearing is loaded radially and a fault develops on the rotating inner raceway, the damaged area will enter and leave the vertical load zone with a frequency that equal the rotational frequency f_r. This will in turn modulate the excitation signal from the damage region, and f_r-spaced sidebands to the inner raceway harmonics will appear in the frequency domain. An example of this can be observed in figure 5.4 where the development of an envelope spectrum over time is depicted for an inner raceway bearing fault on a roller bearing. The RPFI has been marked with red, and multiple sidebands can be observed on either side of this frequency peak. Also, harmonics of the fundamental rotational frequency appear in the low frequency region of the envelope spectrum.

The highly specific frequency features discussed until now, are of course not the only assessment tool for bearing diagnostics. Simple diagnostic indicators such

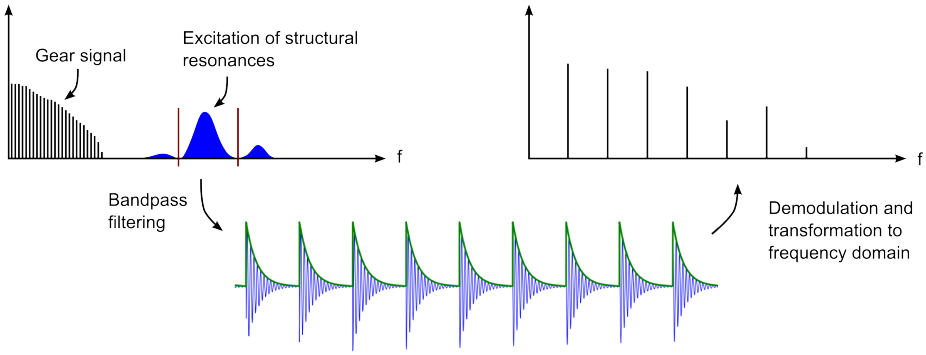


Figure 5.3: Envelope analysis can amplify weak bearing impulse-like signals by demodulating the excitation signals of higher order structural resonances. Hereby the low-frequency bearing signal (green) is recovered, and the usual frequency analysis tool box can be employed for further signal and fault identification.

as overall vibration energy, perhaps in certain frequency bands, are an integrated part of practical monitoring systems. Single measures of impulsiveness, the signal kurtosis for example, is also a popular choice for bearing monitoring applications. It must also be noted that the above analysis is based on an idealized setting suitable for early fault detection. Early is here defined as detection of the initial fault state where the defect is highly localized. In more progressed damage scenarios, for example where most of the raceway show signs of surface spalling, the excitation signal can be dominated by random noise components and the specific fault-indicative frequency features may not be visible at all.

5.1.3 Dynamic sampling

The signal characteristics discussed in the previous sections all scale with one common factor, namely the rotation speed. For machines operating in a stationary regime this does not pose any challenges from an analysis point of view, but a modern wind turbine is quite different as it operates in a highly non-stationary environment and the speed fluctuates with the wind conditions. This have a direct impact on the vibration signals which are collected throughout the turbine. For example, if a frequency spectrum is averaged over a 1-minute period, the rotational speed may fluctuate significantly within the recording window. As a result, the speed-dependent peaks are *smeared* in the spectrum.

An effective method for mitigating this problem is dynamic sampling[2] Using

[2]in practice, this is often implemented as dynamic *resampling* of a fixed high sample-rate

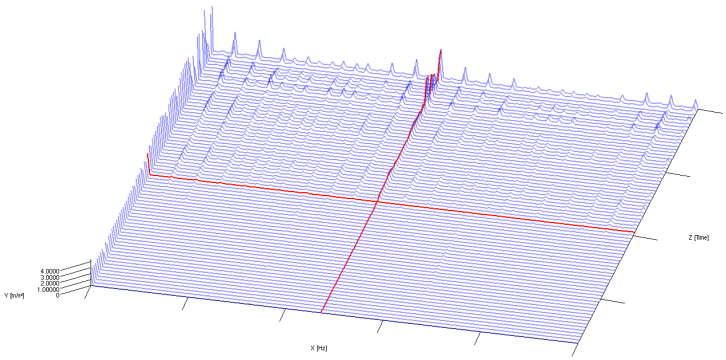


Figure 5.4: Envelope spectra recorded from a gear box with a faulty roller bearing. The individual spectra are plotted along the horizontal axis, the vertical axis denotes time. The most recent measurement is observed at the very top. A peak identified as an inner raceway fault frequency grows in amplitude, and modulation with the rotational frequency shows up as sidebands due to load zone modulation. Harmonics of the rotational frequency can also be observed in the envelope spectrum. The time span of this plot is several months showing the slow progressive nature of typical bearing damage.

an accurate reading of the instantaneous speed of the drive train, the sampling frequency is adjusted dynamically with the rotational speed. The effect is that in the dynamically sampled time domain, the signals look as if the turbine was operating at a fixed speed. This is analogue to the concept of order analysis. Dynamical sampling has actually been enabled when calculating the gear box spectrum shown in 5.1. A spectrum from the same sensor is displayed in figure 5.5 with dynamical sampling *disabled*. The peaks in the high frequency region are now completely indiscernible due to smearing. Frequency smearing therefore poses a significant challenge when attempting to characterize the signals in terms of the fault fingerprints which we discussed earlier. For early and accurate fault detection in wind turbine drive train, speed-dependent sampling methods are a key component in the signal processing pipeline. The instantaneous speed is typically estimated using a tacho-probe which outputs one or more pulses for every rotation of a drive train shaft. Such measurements also provides the means for separating the deterministic part of the signal from the stochastic part. Time-synchronous averaging is such a technique which by averaging segments spanning a full rotation whereby the estimate of the deterministic part is gradually improved [30].

recording.

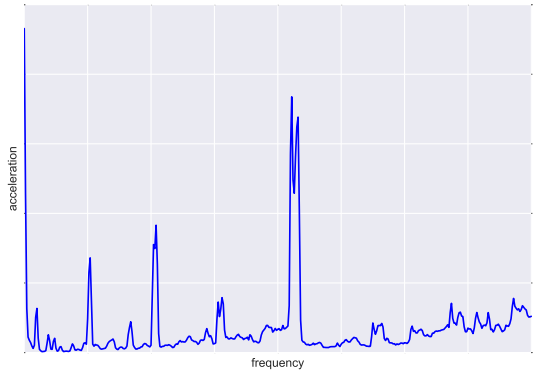


Figure 5.5: An example of frequency smearing caused by variations in the turbine speed. This figure should be compared with figure 5.1 where speed-dependent sampling has been enabled.

5.1.4 Data recording, processing and evaluation

With the basic vibration monitoring and recording concepts now in place, the more practical details of vibration-based monitoring will be discussed specifically in terms of how data are collected and evaluated. The description is based on the vibration-based monitoring architecture which has been deployed at Siemens for more than a decade. The monitored fleet size now exceeds 10,000 turbines which is the worlds largest monitoring platform for advanced wind turbine vibration diagnostic services.

A diagram of a data flow in this platform is illustrated in figure 5.6. The Gram & Juhl M-system installed in all Siemens wind turbines is a state-of-the-art data acquisition and processing system tailored for vibration-based monitoring applications. The system provides 41 kHz sampling rate synchronously on 24 channels and has built-in support for speed-dependent dynamic sampling. In the turbines from which data has been gathered for this thesis, the system is connected to six accelerometers distributed across the drivetrain as illustrated in figure 5.6. One sensor is mounted on the rotor bearing housing, three sensors are located on the gear box, and two sensors are mounted on the generator front and rear.

The system does not record and process data continuously. Instead the signals are processed in segments, with each segment spanning two minutes. From this recording a number of complex features are generated, for example frequency spectra and envelope spectra tailored to diagnostics of specific physical com-

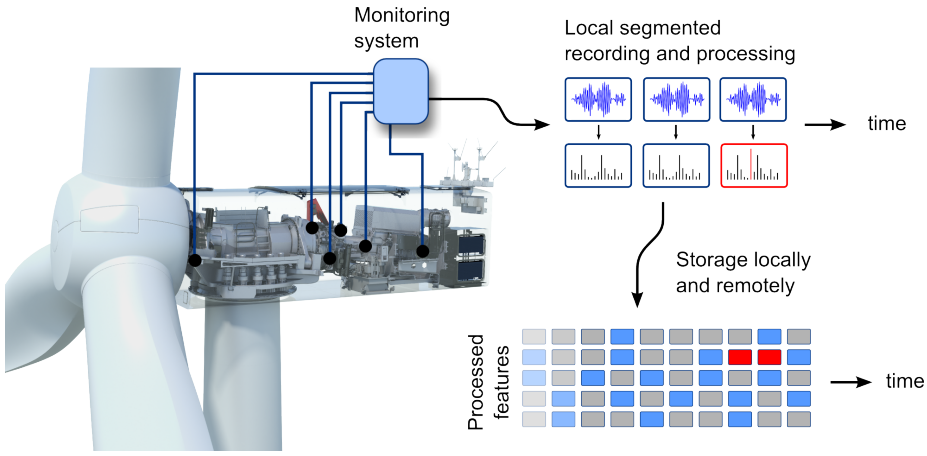


Figure 5.6: Vibration data flow in the Siemens monitoring platform. Data is processed in non-overlapping segments from which a number of frequency and time domain features are extracted. Alarm threshold are then defined for some of these features which may trigger a warning in the system. Only a subset of the data points are stored as illustrated in the lower right corner.

ponents. More simple diagnostic indicators can also be extracted, like overall vibration levels and values of the specific frequency indicators which have been described earlier. This cycle is then repeated, segment by segment. A number of warning thresholds may have been implemented at this stage to detect abnormalities in the data. If a potential problem is detected, the system may have the possibility to shut down the turbine in case of critical faults. This option is typically used for gear wheel faults. The fault indicators will generally depend on the operating conditions of the turbine which must be taken into account by the detection system. As the measurements are very sensitive in nature, the readings differ significantly even between neighboring turbines. Here, the detection system is made adaptive by adjusting the warning thresholds during in the initial operation of each turbine. This establishes a no-fault state fingerprint of the individual turbine.

Not all data is saved following the processing and evaluation stage. This is indicated in the lower part of the figure where only the blue segments are flagged for storage and data transport. As frequency domain data are heavily utilized in these setups, such data are often stored at a higher rate compared to e.g. raw time recordings. If warnings have been triggered, the measurements may be flagged for storage in all cases as denoted by the red segments in the diagram to make sure that abnormal data always is available for further analysis.

Data and alarm events will relayed to a local server at the wind farm, from where

they are further relayed to a central diagnostic center. Each case will then be handled by trained experts that evaluate the data and makes a notification in case a problem is detected. The monitoring infrastructure can therefore be viewed as a massively distributed outlier detection system, from which events are collected and evaluated by human operators.

5.2 Learning systems for automated vibration-based fault detection

With the basic concepts of vibration-based monitoring of wind turbines taken care of, the path towards a more automated fault-detection strategy can now be explored. From the initial description of the problem domain, readers may ask themselves if this problem is not easily solved given how simple frequency analysis already provides some highly distinct fault characteristics across the fault types of interest. The simplicity conveyed before is however deceiving as already hinted at the beginning of this chapter. The influence of the non-stationary operating conditions of the wind turbine has already been discussed at length in this and preceding chapters. Surely dynamic sampling may ameliorate the effect of frequency smearing, but effects from changing load regimes, wind turbulence and advanced turbine control will have a influence on the vibration signals both during fault-free states and with-fault states which is hard, if not impossible, to predict with any certainty from physics-based models. Various noise phenomena, both physical in terms of vibration transmitted from components distributed within the turbine nacelle and electrical in terms of interference and even sensor and cable faults, can also affect the signals used for diagnostic purposes. Further complexity arises from the fact that multiple faults can be present at the same time, and some faults may mask other faults while other interact from a signal perspective. A basic assumption in the fault signal models in section 5.1 is also that the exact mechanical configuration of the drive train must be known. This information is not always known, and considerable variations in the configuration (e.g. bearing configurations) can be expected across turbines and across time, as components are replaced during maintenance.

To summarize, the practice of drive train vibration-based monitoring is a highly empirical discipline despite the deceivingly simple frequency-characteristics of the important failure modes, and classical systems ultimately relies on the long time experience of the human operators. This is also encouraging with respect to the research direction presented here, as it suggests that a data-driven approach presents a more sensible way forward compared to the classical approach of heavy feature engineering.

Figure 5.7: A Siemens monitoring room from which thousands of turbines are controlled and monitored globally.

5.2.1 Supervised learning

The efforts to pursue a supervised learning regime for vibration-based diagnostic applications in wind turbines are documented in detail in papers B and C and the reader is left to the actual papers for most of the implementation details. The purpose of the paragraphs below is only to highlight the main contributions in each of those papers.

In paper B, entitled "Scalable systems for early fault detection in wind turbines: A data driven approach", the ground work for a suitable learning framework for expert-driven vibration-based supervised learning is established, together with some key findings in terms of deep model architectures for vibration-based fault detection. The scope of this paper is only a single type of fault, namely rotor bearing spalling faults which was also covered in paper A from a temperature-based diagnostic perspective. For the supervised approach, the fault state was reduced to a fault/no-fault indicator, whereby the progressive severity of the faults is ignored. As for the ground truth target values, a number distinct choices had to be made for the learning framework. First, the actual state of a bearing at certain point in time is not known and the damage is only confirmed by visual means during an inspection by a technician. The best estimate of the fault initiation is the change-point estimate determined from inspection of historical vibration data by a human diagnostic expert. The end of the fault date is then set to the date of repair or component replacement which can be extracted from the service records. These two end-points then designate the with-fault period in time. An important point here is that the diagnosis on which the models are trained is made by a *particularly knowledgeable* expert since the person has 1) prior knowledge of the presence and type of fault and 2) access to the entire historical data both before and after the fault onset, making the output of the expert non-causal. All diagnostic models are conversely tested in a completely on-line manner. The reference point in time for ROC and AUC-evaluations is in this case set to the fault initiation determined by the expert. All performance metrics are then measured relative to the human detection.

From the preceding section, an example of a frequency-time feature map has already been displayed in 5.4. Such frequency-time maps is particularly useful for turbine diagnostics, as the development in time is an important diagnostic indicator. A suitable choice as input to the diagnostic architectures was therefore sliding windows in time, with each window containing a time $\times$ frequency map. From the discussion related to convolutional neural networks earlier, it becomes clear that the architecture could benefit from convolutional processing steps since the data had significant spatial structure in the time $\times$ frequency domain. This also turned out to be the case, since a deep convolutional architecture was shown to provide excellent detection performance across the test turbines, in

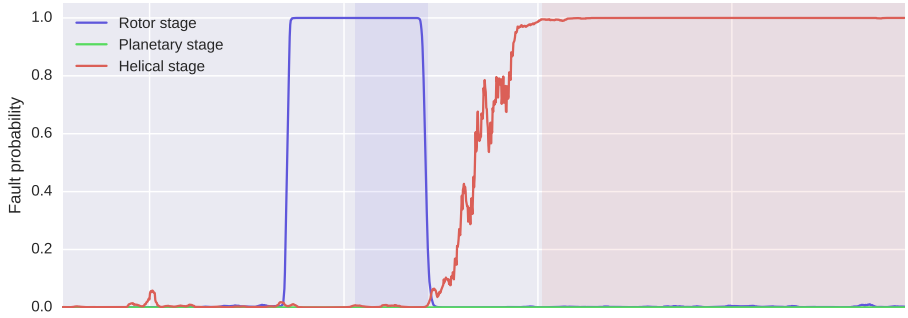


Figure 5.8: Deep convolutional neural network fault probability output as a function of time for a turbine with two distinct faults, one rotor bearing fault and one helical stage bearing fault. The vertical lines denotes years in time. The shaded regions display the individual fault windows specified by a human expert.

some cases months earlier than the human non-causal expert.

The overall key finding from paper B was that one is able to train deep convolutional neural networks on the basis of complex vibration patterns in a fully supervised fashion using an amount of training cases that is feasible within existing turbine monitoring platforms. This finding led to paper C, titled "Deep learning for automated drive train fault detection", in which the work is extended to multi-sensor, multi-fault scenarios using an extended deep convolutional neural network model. Here, faults on planetary stage and helical stage bearings are included in the analysis to investigate the broader applicability of deep convolutional neural networks in drive train diagnostics. The results are very encouraging as the deep networks are able to accurately detect faults across the drive train at a very early stage without showing sensitivity to noise and other phenomena that might interfere with the detection. An example of the model output from paper C is observed in figure 5.8 to highlight the quality of the model outputs on a turbine with multiple faults recorded on the rotor and helical stage bearing.

5.3 Model fusion

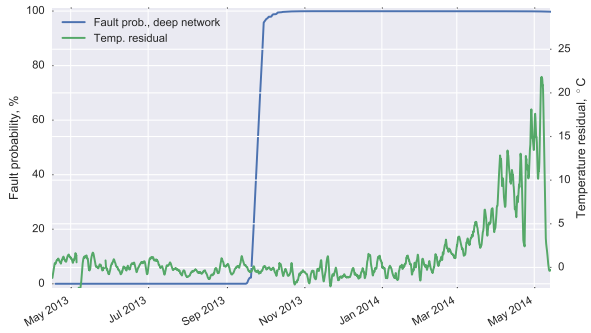
To illustrate how the supervised and unsupervised modeling paradigms demonstrated so far complement each other, combined outputs from temperature-based and vibration-based diagnostic models will now be merged for rotor bearing diagnostics. Examples of three rotor bearing faults are displayed in figure

5.9 with an overlay of the temperature residual evaluated by the heteroschedastic neural network model from paper A, and the rotor bearing fault probability output from the single-task deep network from paper C. Starting from figure 5.9a, the deep network is observed to trigger on a fault signal at an early stage. After an operational period of about 6 months, the temperature residual also starts to increase over a period of 3 months before the bearing reaches a critical temperature. A similar picture can be observed in figures 5.9b and 5.9c, but with smaller lead times between the vibration-based model and temperature-based model. The fault progression is obviously different between these three cases, with the last plot depicted in 5.9c showing a fast temperature rise spanning a little more than a month.

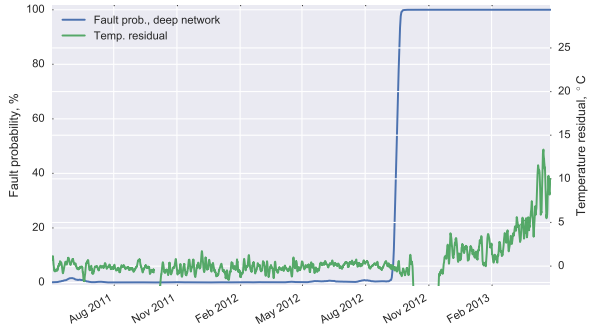
From an operational and maintenance perspective, these plots are highly interesting. The supervised models are able to provide some very early warnings of an impeding failure and preparations for a replacement in terms of spare parts, tools and cranes can be undertaken at an early stage possibly taking into account other planned service works in the area and long term weather forecasts. The temperature residuals then provide important feedback in terms of the fault progression. As soon as the residual begins to rise, the bearing can be expected to be in a progressed fault state and the replacement plans may have to be accelerated. The planning can possibly supported by estimates on the time to failure based on residual projections. To conclude, a combination of various state-of-the-art data modeling practices can provide valuable actionable insights for a wind farm O&M organization.

5.4 Summary

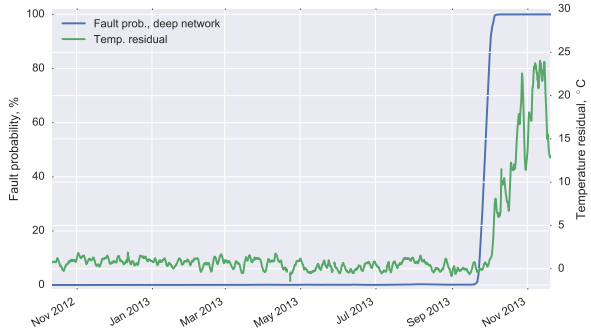
To provide a foundation for the analysis of data-driven methods for vibration-based fault detection, a primer on the basic principles of vibration-based monitoring was first given. From here, a learning framework was outlined from which the supervised models could be trained on followed by a discussion of suitable network architectures for complex vibration data processing. The reader is here strongly advised to review paper C, as this work provides the most comprehensive view of the effectiveness of the proposed deep learning architectures. Finally the chapter was ended by comparing and combining the outputs from both the supervised models and the unsupervised temperature models, which is a new contribution compared to the work in the attached papers.



(a)



(b)



(c)

Figure 5.9: The combination of the temperature-based models and the vibration-based models displayed for three different cases of rotor bearing faults. Early fault detection is obtained via the vibration-based model, while the temperature-based models provide details on the fault severity during fault progression

Chapter 6

Conclusion and outlook

In this thesis data-driven modeling have been applied to two important data streams collected from wind turbines with the aim to extract further knowledge about the state of of the turbine to facilitate more efficient operation and maintenance on both a wind turbine and wind farm level. Despite the overall data-driven commonalities, different modeling paradigms have been used for each data stream. For the SCADA data streams non-linear regression techniques and unsupervised fault detection was combined to demonstrate

1. Accurate data-driven predictive models for rotor bearing temperature readings with long-term stability in predictive accuracy.
2. An extensive study on the performance of the predictive models for fault detection based on the model temperature residual. Faults were shown to be detected with high accuracy 45 days before a critical rotor bearing failure.

The last item is particularly interesting since it showcases how existing data sources which may not be engineered for fault detection can be improved considerably by applying more advanced data modeling techniques.

The work on operational wind turbine data streams serves as an extension to existing research by providing more complex model architectures and a more

structured approach to model evaluation in a diagnostic context using a large number of actual turbine faults. The work presented in paper B & C based on vibration data modeling however departs considerably from existing research on this topic. Specifically, the key findings here are

1. Deep learning architectures are found to learn complex diagnostic processing tasks from multi-sensor signals collected from a large scale monitoring platform based on fully supervised learning from the outputs of human experts. Deep learning architectures significantly outperforms shallow architectures in this domain, and rivals the performance of human non-causal experts with prior knowledge of the fault state when tested on-line.
2. The architectures are able to operate on a minimum of feature engineering. Input features have only been selected on a high level, and the data driven methods provide highly accurate and early detection with no knowledge of the detailed mechanical configuration of the individual drive trains.

These results are quite remarkable given that interpretation of vibration data is one of the most challenging tasks related to remote diagnostics on wind turbines. The methods have here demonstrated production-ready capabilities for turbine monitoring platforms.

The merits of the different modeling approaches have further been highlighted by combining the outputs from the temperature residuals and the deep networks. Early automatized fault detection is enabled by the proposed models which also have the added benefit of providing a probabilistic output which is straightforward to process in added decision and planning layers. The temperature residuals are not suited for early fault detection, but instead they provide a continuous measure of the fault progression. The combination of these different modeling approaches provide a powerful diagnostic tool.

6.1 Outlook

Considering only the immediate methodological foundation of this thesis, ample opportunities exist for broadening the scope in terms of:

Generalized bearing diagnostics: The combined processing of temperature and vibration data presents itself as a *general* tool for bearing diagnostics, and only a subset of the rolling element bearings in a geared turbine drive train have been considered so far. Given the wide coverage of the drive

train with both temperature and vibration sensors, it is feasible to target other bearing locations also.

Detailed bearing diagnostics: Based on the supervised learning approach, opportunities also exists for providing even more detailed fault analysis, e.g. in terms of inner, outer and rolling element faults.

Gear wheel faults: There is not many qualitative differences between gear wheel faults and bearing faults from a high-level signal feature perspective, with the major difference being in terms of the time scales involved. It would be an interesting extension to apply the supervised vibration based models to gear wheel faults also.

In a broader perspective, some interesting research directions also start to emerge from the presented work. These include:

Data-driven prognostics: In the plots in figure 5.9, one observes a gap in time between the early fault detection from the supervised methods, and the fault severity estimation from the temperature residual. Ideally, one would like to reduce this gap and obtain not only a fault severity estimation from the immediate onset of the fault, but also an accurate prediction on the remaining useful lifetime of the component. Unsupervised and semi-supervised methods possibly provide a way forward in developing capable data-driven systems for such tasks.

End-to-end learning: The vibration-based models have all been based on inputs tailored for human interpretation. By adopting a true end-to-end learning approach using raw time series input, the fault detection capabilities of a data-driven system may be improved. The challenge is of course the high sample rate signals which are employed for vibration-based diagnostics, and from a more practical perspective, the bias of current vibration data platforms towards human-based analysis.

Finally, it is also worth to consider the presented work from a system-level perspective. What have been demonstrated here is a focused effort to employ a data-driven methodology to an admittedly small corner of the overall wind energy system, namely diagnostic applications for wind turbines. The impact of moving towards algorithmic complex reasoning with probabilistic outputs as shown here is however significant. Imagine how the outputs of such diagnostic systems can now be directly fed to the turbine control system which then is able to adjust the operation of the turbine accordingly. The control system itself may be a reinforcement learning system that feeds on intricate sensor systems within the turbine and radar systems that provide detailed volumetric mapping of the

air flow, to optimize the power production and long term turbine reliability. Similarly, the data can flow to maintenance decision support systems that will plan ahead in terms of available tools, equipment and other resources, possibly including predictions on not only the weather but also the expected electricity demand and prices in the decision process. In this vision, machine intelligence is foreseen to take a prominent role in the future renewable energy system.

Appendix A

Article A

Martin Bach-Andersen, Bo Rømer-Odgaard, and Ole Winther. Flexible non-linear predictive models for large-scale wind turbine diagnostics. *Wind Energy*, 2016, doi: 10.1002/we.2057.

Wind Energy

WIND ENERGY
Wind Energ. (2016)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/we.2057

RESEARCH ARTICLE

Flexible non-linear predictive models for large-scale wind turbine diagnostics

Martin Bach-Andersen[1,2], Bo Rømer-Odgaard[1] and Ole Winther[2]

[1] Siemens Diagnostic Center, Brande, Denmark
[2] Cognitive Systems, Technical University of Denmark, Kongens Lyngby, Denmark

ABSTRACT

We demonstrate how flexible non-linear models can provide accurate and robust predictions on turbine component temperature sensor data using data-driven principles and only a minimum of system modeling. The merits of different model architectures are evaluated using data from a large set of turbines operating under diverse conditions. We then go on to test the predictive models in a diagnostic setting, where the output of the models are used to detect mechanical faults in rotor bearings. Using retrospective data from 22 actual rotor bearing failures, the fault detection performance of the models are quantified using a structured framework that provides the metrics required for evaluating the performance in a fleet wide monitoring setup. It is demonstrated that faults are identified with high accuracy up to 45 days before a warning from the hard-threshold warning system. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS
condition monitoring; fault-detection; data analysis

Correspondence
Martin Bach-Andersen, Siemens Diagnostic Center, Brande, Denmark.
E-mail: martin.bach-andersen@siemens.com

Received 2 May 2016; Revised 19 August 2016; Accepted 2 September 2016

1. INTRODUCTION

Significant efforts are invested by the wind industry to reduce the costs related to the operation and maintenance of wind farms in the quest to reduce the overall cost of wind energy. Proactive maintenance/predictive turbine maintenance program is one of the tools employed to realize these cost reductions. A pivotal element in these programs is the ability of wind farm operators and service providers to extract accurate information about turbine faults and operational state from the data streams collected from modern wind turbines. The potential benefit of advanced remote diagnostic capabilities is increasing dramatically for wind farms off-shore and, in the near future, far-shore. We will not reiterate overall methods and technologies related to wind turbine condition monitoring here, but instead refer to overviews given in the literature.[1,2]

The overall motivation of this study is to investigate how further diagnostic insight can be gained from existing data sources. The work is based on no-fault state models, where we seek accurate predictive models for turbine sensor readings during a no-fault state. This concept is not new, and early demonstrations of how such modeling can be used for fault detection in wind turbines can be found in the literature.[3,4] The predictive models have intrinsic value as they can provide the foundation for many tasks related to further analysis of wind turbine faults and operation. A model for a given physical system could be established using first principles. For systems of even moderate complexity, such an approach may however prove unfeasible. If enough sensor data from the system have been collected, one could dispense with the physical model and use flexible, *general* models, which are trained on collected data. As the cost of data and computing power now decreases, data-driven methods for turbine monitoring has gained significant interest from both industry and the research community.[5–9] Such a data-driven approach will also be used in the remainder of this work. We will restrict ourselves to the task of predicting the temperature of the rotor bearing assembly. Then, using data from a large number of actual bearing failures in both on-shore and off-shore wind turbines, it is investigated to what degree the output from such predictive models can be utilized in a fault detection scheme. Examples of work on rotor-bearing temperature modeling can be found in the literature.[10,11] Our novel contributions are model formulations with more complex input and output spaces and more actual failures to quantify the diagnostic performance of these models. The work presented is structured as follows:

1. The predictive performance of various learning architectures in a long-term setting is evaluated across a number of different turbines. Focus is on long-term stability of the predictions.
2. A structured methodology for evaluating the diagnostic performance is established. Models are trained on no-fault data, and 22 actual main bearing failures are used for testing the diagnostic ability of the trained models.

The main body of research in this area, and also the work presented here, is unsupervised in the sense that the models are fitted to no-fault state data, and fault detection is then framed as an outlier detection problem. The clear advantage of this approach is that no fault data required. If enough fault data are available, data-driven fault detection algorithms could also be trained in a supervised sense, hereby learning to discriminate directly between fault and no-fault states. The remainder of the paper is organized as follows. Section 2 describes the data, Section 3 the models, Section 4 the temperature prediction results and Section 5 the rotor bearing fault detection results. We conclude in Section 6 and discusses future research directions in Section 7.

2. DATA SET

To evaluate the models both in terms of their predictive accuracy and their usability in a fault detection application, sensor data were collected from 22 turbines with confirmed rotor bearing mechanical faults. To test the no-fault state predictive performance of the models, an additional set of 45 turbines was selected from the same set of wind farms with the requirement that no faults had been reported on the rotor bearing during the life span of these turbines. No other criteria were used for the selection. The best performing models from the no-fault predictive task were then tested in a fault detection context using the with-fault data set.

The set represents a diverse collection of turbines operating in different environmental conditions, including both on-shore and off-shore turbines, to ensure that conclusions drawn from this study are general in nature and not specific to any particular set of wind farm conditions.

2.1. Sensor inputs

The overall choice of inputs for the predictive models to be considered will be based on simple thermal energy flow considerations for the bearing assembly. The main heat source to be considered is the gear box. Heat will flow from the gear box through the main shaft and the bearing assembly before being dissipated through the hub to the external environment. Heat transfer will likewise occur between the bearing and the internal nacelle environment. Also, a conversion of mechanical to thermal energy will take place in the bearing because of various loss mechanisms. Based on these simple considerations, sensor readings can be selected that enable *a predictive model of the bearing temperature* to capture the thermal dynamics of the system. The choices of inputs, and their motivation, are given in the following:

Active power output: The active power output serves as an overall indicator of the operational state of the turbine and will also serve as a more stable proxy for the wind speed compared with, e.g., a wind vane measurement.

Generator speed: The frictional losses in the bearing assembly and the rest of the drive train will be dependent on the rotational speed.

Gear box oil temperature: The oil temperature provides an estimate of the mean gear box temperature.

Ambient temperature: The heat dissipation to the external environment will scale with the ambient temperature and wind speed.

Nacelle temperature: Heat transfer between the bearing assembly and internal environment will scale with the internal nacelle temperature.

To reiterate, the goal is not to derive a physical model of the functional dependencies between the bearing temperature and these variables. Instead, we learn these dependencies from historic data collected during turbine operation.

2.2. Data processing

The raw data used in this study are mean sensor readings sampled in 10 min intervals. Testing was performed on the raw data directly, but as the bearing temperature only varies slowly during normal operation because of the large thermal mass of the system, a lower time resolution proved to be sufficient for both the prediction and detection tasks. A 1 h re-sampling of all data was found to provide the best compromise between computational speed and accuracy. This also allows models to capture dynamics of a longer time scale while keeping the model complexity bounded.

All channels were also normalized according to the following scheme before further processing:

Temperature readings: All temperature readings are normalized as $\hat{u}_i = \frac{u_i - m_i}{s_i}$, where m_i, s_i for the individual channels are population wide mean and standard deviations inferred from a data set collected from 400 turbines. A

population-based normalization was chosen out of convenience as it provides a fixed transformation across all data sets, and testing showed that using per-turbine normalization did not improve convergence of the models.

Power output and generator speed: These parameters are operating within deterministic envelopes defined by the turbine controller, and a Gaussian-based normalization is not suitable for these parameters. Instead, we assume these channels to be uniformly distributed and normalize as $\hat{u}_j = \frac{u_j - m_j}{s_j}$, where $m_j = \frac{1}{2}\left(\max(u_j) + \min(u_j)\right)$ and $s_j = \frac{1}{2}\left(\max(u_j) - \min(u_j)\right)$, where j is the channel index.

3. ROTOR-BEARING PREDICTIVE MODELING

The predictive task is solved using a sliding window approach. The vector of input sensor readings is denoted by $\boldsymbol{u}_n$, and the simultaneous bearing temperatures is denoted by t_n, where n is the index in time. We will consider two basic classes of models, namely, models using strictly exogenous inputs and models also including auto-regressive terms. From a probabilistic perspective, we seek to model

$$\text{Exogenous only:} \quad p\left(t_n | \boldsymbol{u}_{n-L+1}^T, \boldsymbol{u}_{n-L+2}^T \ldots \boldsymbol{u}_n^T\right) \tag{1}$$

$$\text{Exogenous and auto-regressive:} \quad p\left(t_n | \boldsymbol{u}_{n-L+1}^T, \boldsymbol{u}_{n-L+2}^T \ldots \boldsymbol{u}_n^T, t_{n-L+1}^T, t_{n-L+2}^T \ldots t_{n-1}^T\right) \tag{2}$$

By using the entire window as input, the model is able to capture longer-term dynamics of the system relevant for the predictive task. The optimal value of length L will depend on the properties of the physical system. Here, L is treated as any other hyper parameter to be optimized using a validation set.

In general, the formulation in equation (2) can be expected to provide more accurate predictions compared with equation (1) as the past target values provide valuable input when predicting the current target value. However, in a diagnostic setting, the former model may prove more relevant as, under mild assumptions, a mechanical fault in the rotor bearing will not influence the parameters listed in 2.1. The residual between the purely exogenous regressive model and the measured temperature is therefore a direct measure of the temperature residual caused by the failure mechanism.

In the auto-regressive case, a change to a fault-state will also change the input distribution. The residual will then be a measure of how well the model extrapolates to this new input domain, or in other words, how well the no-fault state model is able to describe the thermal dynamics of the system during fault state conditions and a monotonic relationship between the fault temperature residual and the model residual can not be guaranteed. From a diagnostic perspective, an accurate model of type (1) is therefore desirable. We investigate this hypothesis empirically by testing both types of models in fault detection.

3.1. Linear models

To establish a baseline for the predictive performance, we first consider a linear model of the type

$$p\left(t_n | \boldsymbol{x}_n\right) = \mathcal{N}\left(t_n | \boldsymbol{w}^T \boldsymbol{x}_n, \sigma^2\right) \tag{3}$$

where $\mathcal{N}(\cdot)$ denotes the normal (Gaussian) probability distribution, $\boldsymbol{w}$ is the $D+1$ dimensional weight vector to be inferred from data, x_n is the input data window in vectorized form (with a zeroth input variable added to account for the bias term) and σ^2 is the noise variance. The regularized maximum likelihood solution to w is given by

$$\boldsymbol{w} = \left(\lambda \boldsymbol{I} + \boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{t} \tag{4}$$

controlled by the weight decay parameter λ, with an input data matrix $\boldsymbol{X}$ of size $N \times D + 1$.

We furthermore consider a heteroschedastic model where the log noise variance is parameterized using a linear function of the input. A model that is able to learn the noise distribution may prove valuable in a fault detection context as it conveys information about the uncertainty of the model predictions. The likelihood for individual data points is then

$$p\left(t_n | \boldsymbol{x}_n\right) = \mathcal{N}\left(t_n | \mu(\boldsymbol{x}_n, \boldsymbol{w}), \sigma^2(\boldsymbol{x}_n, \boldsymbol{w})\right) \tag{5}$$

with $\mu(\boldsymbol{x}_n, \boldsymbol{w}) = \boldsymbol{w}_\mu^T \boldsymbol{x}_n$ and $\sigma^2(\boldsymbol{x}_n, \boldsymbol{w}) = \exp(\boldsymbol{w}_\sigma^T \boldsymbol{x}_n)$. This leads to the following regularized maximum likelihood error function:

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} \frac{(t_n - \mu(\boldsymbol{x}_n, \boldsymbol{w}))^2}{\sigma^2(\boldsymbol{x}_n, \boldsymbol{w})} + \frac{1}{2} \sum_{n=1}^{N} \log \sigma^2(\boldsymbol{x}_n, \boldsymbol{w}) + \frac{1}{2} \sum_j \lambda_j w_j^2 \tag{6}$$

where λ_j is a weight decay parameter defined, in the general case, individually for each weight. This error function is minimized using numerical techniques as described in Section 3.3. The hyper parameter of the linear models is the weight decay λ, which is optimized using a validation set and weight initialization for models that are solved iteratively.

3.2. Non-linear models

If the system on which we apply the model exhibits non-linear dynamics, the linear model in equation (3) may prove to be too restrictive to predict the bearing temperature accurately. We will therefore also consider the non-linear two-layer neural network model[12] :

$$a(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}_2^T \boldsymbol{f}(\boldsymbol{W}_1^T \boldsymbol{x}) \tag{7}$$

where $\boldsymbol{W}_1$ is weight matrix for the first layer, f is the hidden layer non-linearity (activation function) applied element wise and $\boldsymbol{w}_2$ is the weight vector for the second layer. Like for the linear model, we add one constant dimension to both the input and hidden layer in order to include bias terms. For the homoschedastic model, we have the likelihood

$$p(t_n|\boldsymbol{x}_n) = \mathcal{N}\left(t_n|\mu(\boldsymbol{x}_n, \boldsymbol{w}), \sigma^2\right) \tag{8}$$

with the mean function $\mu(\boldsymbol{x}_n, \boldsymbol{w}) = a(\boldsymbol{x}_n, \boldsymbol{w})$. With a constant noise term, the regularized maximum likelihood solution is obtained by minimizing the error function

$$E(\boldsymbol{w}) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (t_n - \mu(\boldsymbol{x}_n, \boldsymbol{w}))^2 + \frac{N}{2} \log \sigma^2 + \frac{1}{2} \sum_j \lambda_j w_j^2 \tag{9}$$

These non-linear models fall under the general category of NARMAX models.[13]

We will also consider a heteroschedastic model as we did in the linear case. We extend the neural network to have two output units: $a_\mu = \boldsymbol{w}_{2\mu}^T \boldsymbol{f}(\boldsymbol{W}_1^T \boldsymbol{x})$ and $a_{\sigma^2} = \boldsymbol{w}_{2\sigma^2}^T \boldsymbol{f}(\boldsymbol{W}_1^T \boldsymbol{x})$ and set: $\mu(\boldsymbol{x}_n, \boldsymbol{w}) = a_\mu$ and $\sigma^2(\boldsymbol{x}_n, \boldsymbol{w}) = \exp a_{\sigma^2}$. The heteroschedastic error function in equation (6) also applies in the non-linear case. The hyper parameters of the neural-network-based models are the number of hidden units, weight decay parameters, choice of activation functions and weight initialization.

3.3. Optimization

In all but the simplest case of the linear model with a constant noise variance, the error function can not be minimized analytically, and one must resort to numerical optimization algorithms to obtain a solution in parameter space for a possibly local error minimum. We strictly used a momentum-based stochastic gradient descent procedure[14] for training the models:

$$\boldsymbol{v}_{\tau+1} = \boldsymbol{v}_\tau \gamma - \eta \nabla_{\boldsymbol{w}} \tilde{E}(\boldsymbol{w}_\tau) \tag{10}$$

$$\boldsymbol{w}_{\tau+1} = \boldsymbol{w}_\tau + \boldsymbol{v}_{\tau+1} \tag{11}$$

where γ is the momentum parameter, η is the learning rate and $\tilde{E}(\boldsymbol{w}_\tau)$ is the estimate of the true gradient of the error in function evaluated on a sub-set (mini-batch) drawn randomly from the training set. The velocity is initialized to $\boldsymbol{v}_0 = 0$. Weights are initialized randomly using a uniform distribution $U(-\alpha, \alpha)$. The training algorithm hyper parameters are also tuned using validation data.

4. TEMPERATURE PREDICTION

In this section, we present the results for the predictive modeling of the rotor-bearing temperature. First, we discuss the overall test design and the error measures of interest, and then we benchmark the different models on the temperature prediction on the non-fault test condition.

4.1. Predictive scoring

The choice of predictive error measures when comparing model performance must be adapted to the task at hand. Here, we focus on measuring long-term deviations, so once trained on no-fault state data, the models must provide accurate predictions days, weeks and even years ahead in time. The experiment is therefore designed as follows:

Training and validation: For all tests, the models are fitted to assumed no-fault state data from the first year of operation for each turbine. As explained in Section 4.2, the models are fitted to each turbine individually. To ensure that a full seasonal variation is captured within the training period, the actual year 1 data window is extended to 14 months. Also, the first month of data has been removed from the data set to avoid pollution from commissioning tests that may not be indicative of normal turbine operation. From the training data window, 20 random 72 h segments are held out as validation data.

Primary error measures: As for the validation set, another 20 random segments are held out from the training period to test the final performance of the models. By this design, we ensure that the models are trained, validated and evaluated on data that share the input and turbine state distributions. The error measures on this test set are therefore used as the primary measures of the predictive performance of the models.

Long-term error measures: To measure the long-term robustness of the predictive performance, additional test sets are extracted from the following two years of operation, denoted by year 2 and year 3, respectively. As for the primary test set, error measures are reported on each of these sets.

Two score metrics will be used on the various test sets, namely, the root mean square error (RMSE) and the average negative log-likelihood evaluated on the combined test data from all turbines:

$$\mathrm{RMSE} = \left(\frac{1}{N} \sum_{n=1}^{N} (\mu(\boldsymbol{x}_n, \boldsymbol{w}_n) - t_n)^2 \right)^{1/2} \tag{12}$$

$$\mathcal{L} = -\frac{1}{N} \sum_{n=1}^{N} \log p\,(t_n | \boldsymbol{x}_n, \boldsymbol{w}_n) \tag{13}$$

Here, the individual model weights have in all cases been inferred from the training set from year 1, with weights being shared for data points from the same turbine. The primary error measures for the year 1 test set are denoted by $\mathrm{RMSE}_{Y_1 \to Y_1}$ and $\mathcal{L}_{Y_1 \to Y_1}$ signifying that model parameters are inferred from year 1 and evaluated on year 1 data. Similarly, long-term measures are denoted with subscripts $Y_1 \to Y_2$ and $Y_1 \to Y_3$.

4.2. Temperature prediction results

For an overview of the various hyper parameters selected for each model architecture, we refer to the upper part of Table I. The window length was adjusted to $L = 24$ for all models investigated in this study. To gain a qualitative understanding of the predictive performance of the models, predictions for a single turbine have been visualized in Figure 1, covering the first 3 years of production. The residuals have also been visualized, split between models with and without AR terms. A notable feature of the bearing temperature data is the seasonal dependency clearly observed in the 3 year plots, which all models capture accurately, so the differences between the predictions from the individual models are most clearly observed in the detailed view and in the residual plots. As expected, the models with AR-inputs generally provide more

Table I. Overview of hyper parameters and performance metrics for each model architecture. The best predictive scores for each input class have been highlighted in bold.

	Exogenous				Exogenous + AR			
	Linear(μ)	Linear(μ, σ)	ANN(μ)	ANN(μ, σ)	Linear(μ)	Linear(μ, σ)	ANN(μ)	ANN(μ, σ)
No. hidden units	—	—	256	256	—	—	256	256
Weight decay, λ_{lin}	1.0	10^{-6}	10^{-4}	10^{-3}	1.0	10^{-4}	10^{-6}	10^{-4}
Weight decay, λ_{hid}	—	—	10^{-4}	10^{-1}	—	—	10^{-5}	10^{-2}
Learning rate, η	—	10^{-4}	10^{-3}	10^{-4}	—	$2 \cdot 10^{-5}$	10^{-3}	10^{-4}
Initialization, α	—	10^{-3}	10^{-2}	10^{-3}	—	10^{-5}	10^{-3}	10^{-3}
Momentum, γ	—	0.9	0.9	0.9	—	0.9	0.9	0.9
RMSE_{Y_1}	1.36	1.67	**1.13**	1.15	**0.31**	0.33	0.32	0.36
$\mathrm{RMSE}_{Y_1 \to Y_2}$	1.42	1.43	1.32	**1.22**	**0.31**	0.33	0.32	0.34
$\mathrm{RMSE}_{Y_1 \to Y_3}$	1.89	1.90	1.86	**1.78**	**0.32**	0.33	0.35	0.39
$\mathcal{L}_{Y_1}$	1.59	1.56	1.38	**1.35**	**0.22**	0.39	0.25	0.30
$\mathcal{L}_{Y_1 \to Y_2}$	1.79	2.02	**1.71**	2.28	**0.25**	0.38	0.29	0.35
$\mathcal{L}_{Y_1 \to Y_3}$	**2.43**	3.49	3.02	5.32	**0.29**	0.44	0.37	0.61

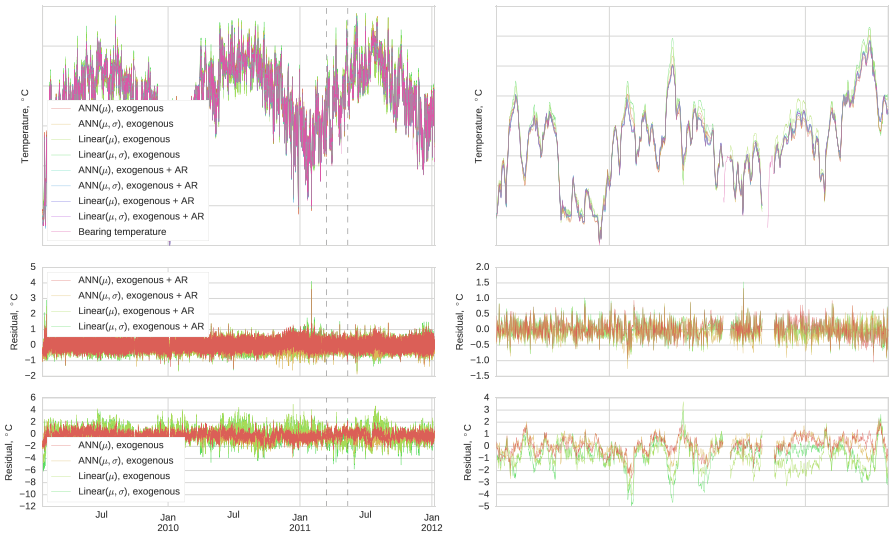


Figure 1. Example of predictions covering first three years of operation for a single turbine. Residuals have been plotted for the two input classes separately. A zoom-in of the time segment bounded by dotted lines can be observed in the right-hand side plots to provide a more detailed view of the model predictions. Gaps in plots are due to missing data.

accurate predictions. When comparing linear with non-linear models, the accuracy is comparable for AR models, but in the purely exogenous case, non-linear models seem to outperform the linear models. In the detailed view, the linear models are observed to overshoot near-temperature peaks, a trait that is not observable to the same degree for the non-linear models. Overall, the models provide accurate predictions also in the long term, and also in the purely exogenous case. To investigate if these observations for a single turbine applies generally, the performance metrics measured across the entire data set have been summarized in the lower part of Table I. Focusing on the exogenous case, we see that for the primary metrics defined for year 1 in the non-linear models provides the best fit with an RMSE as low as 1.13°C for the model ANN(μ). The RMSE for ANN(μ, σ) is slightly higher at 1.15°C, but the ability of this model to adapt to the noise distribution provides the best score measured on the likelihood measure $\mathcal{L}_{Y_1} = 1.35$. The results shift as we move to the long-term measures. In general, the (μ, σ) models do not provide robust likelihood models in the long term, or in other words, the more accurately the noise distribution can be learned from the given training period, the more any shifts in the data distributions due to system or external changes will penalize the prediction score. This is highlighted by the fact that ANN(μ, σ), which provided the best fit in year 1, provides the worst likelihood score in year 3. An interesting observation, however, is that ANN(μ, σ) still provides the best long-term fit measured in RMSE, both for years 2 and 3.

Moving on to the AR-models, we see that the predictive error is lower by a good margin, with the simple linear model Linear(μ) providing the best fits measured both in terms of RMSE and likelihood for year 1. This also holds for the long-term measures, and it can be concluded that the auto-regressive predictive problem is near linear in nature.

To complete the analysis of the predictive models, examples of the input-dependent noise terms are visualized in Figure 2. The noise distribution clearly has seasonal variations for both linear and non-linear models in this case, with a more pronounced variance shift for the exogenous models. In such a case, the added flexibility of the adaptive noise term allow the models to allocate more modeling capacity for the low-noise regions and penalize deviations in high-noise regions to a lesser degree during training. By extrapolating from the single turbine example in Figure 2, one could expect that an input-dependent noise term is more advantageous in the exogenous case, which the overall results in Table I confirms.

5. FAULT DETECTION

Next, we evaluate the models in terms of their fault detection capability. As in the predictive task, we will first discuss the test design and relevant test metrics in a detection setting, followed by the results obtained when tested on collected fault data.

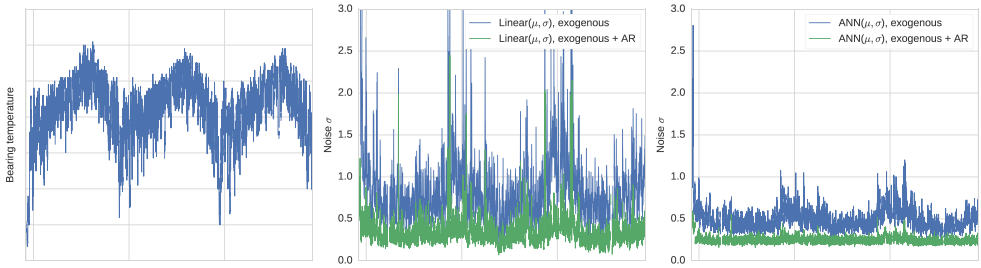


Figure 2. Illustration of the noise-adaptive capabilities of linear and non-linear models in a 3-year period for a single turbine. When comparing with the bearing temperature trend, the models are observed to be less certain in their predictions during colder periods.

Table II. AUC scores for the models evaluated at increasing values of the lead time Δt. Best scores for each input class have been highlighted in bold.

	Exogenous				Exogenous + AR			
Δt	Linear(μ)	Linear(μ, σ)	ANN(μ)	ANN(μ, σ)	Linear(μ)	Linear(μ, σ)	ANN(μ)	ANN(μ, σ)
0 days	0.995	0.995	0.996	**0.996**	**0.996**	0.996	0.995	0.967
1 days	0.992	0.992	0.993	**0.994**	0.992	**0.993**	0.991	0.963
2 days	0.987	0.987	0.989	**0.990**	0.986	**0.989**	0.986	0.948
3 days	0.984	0.985	0.986	**0.987**	**0.981**	0.975	0.981	0.946
5 days	0.973	0.975	0.979	**0.981**	**0.971**	0.964	0.958	0.953
10 days	0.956	**0.960**	0.960	0.957	**0.947**	0.930	0.938	0.915
15 days	0.947	0.947	0.950	**0.951**	0.936	0.895	0.926	**0.957**
20 days	0.932	0.932	0.938	**0.943**	**0.932**	0.874	0.920	0.882
30 days	0.859	0.861	**0.877**	0.869	0.864	0.804	0.861	**0.884**
60 days	0.781	0.784	**0.791**	0.780	**0.768**	0.720	0.758	0.760
90 days	**0.701**	0.693	0.666	0.668	0.661	0.613	0.638	**0.687**

5.1. Detection scoring

The detection scheme implemented for this work is structured as follows:

No-fault state modeling: The no-fault state assumption for the first 14 months of operation from Section 4, minus the first month of operation, is also replicated here.* The entire data set from this run-in period is then used to train the models on each turbine individually.

Detection approach: Once the no-fault state models have been established, the fault detection is formulated as an outlier detection problem. An obvious residual, or distance measure, to be used for this purpose would be a likelihood measure as full conditional probabilities are available for all models (explicitly or implicitly). However, the likelihood did not prove to be a robust long-term measure for the no-fault turbines, and the initial tests on the with-fault turbines also revealed that this measure did not provide any advantages in a fault detection setting. In the following, we will therefore strictly use $r_n = t_n - \mu(\boldsymbol{x}_n, \boldsymbol{w}_n)$ as a distance measure. Two filtering operation are also applied before computing the final residual. First, any data points where the active power output of the turbine is below 1000 kW are excluded from the data set. This is motivated by the fact that the temperature residual observed from a fault is caused by frictional losses in the bearing, which only occur during turbine operation. Periods with little or no production will therefore not provide much signal and is therefore excluded to improve the signal-to-noise ratio. Second, we average the filtered residuals using a 5 day sliding window approach†.

*One could argue that an adaptation period of such length would not be admissible for a critical fault detection system, but from the extensive experience of the authors from large-scale turbine monitoring, this run-in period is more than acceptable for a rotor bearing application. In any case, engineering solutions exist for mitigating these issues, and the approach taken here should still provide a relevant and valid baseline for further work in this area.

†A large window size will improve the estimate of the residuals and will also set a lower bound on the reaction time of the detection system. Conversely, smaller windows will provide more noisy estimates but with improved reaction times.

Detection timing: As the fault progresses, the fault signal-to-noise ratio is assumed to increase. All detection performance measures will therefore be a function of time. To align the fault data from many turbines and provide a principled approach for evaluating this time dependency, one must decide on a reference time. This reference time is for all turbines defined as the moment when the temperature reaches a critical level as defined in the built-in hard-threshold warning system. All measures will then be evaluated as a function of time relative to this reference point, with the relative distance in time denoted Δt, equaling the time to a turbine stop.

Data set: For each relative time distance Δt, a data set is compiled in the following way: (i) For each with-fault turbine, collect distance measure at given time stamp and add this data point as a 'positive'. (ii) Collect distance measures from four non-fault turbines from the same wind farm at the same absolute time stamp and add these data point to the data set as 'negatives'. The last step ensures that with-fault samples are balanced with a number of no-fault samples from turbines operating under similar conditions during this specific point in time. This step is taken to obtain smoother estimates of the time dependent metrics by reducing the biases induced from the limited fault data set size.

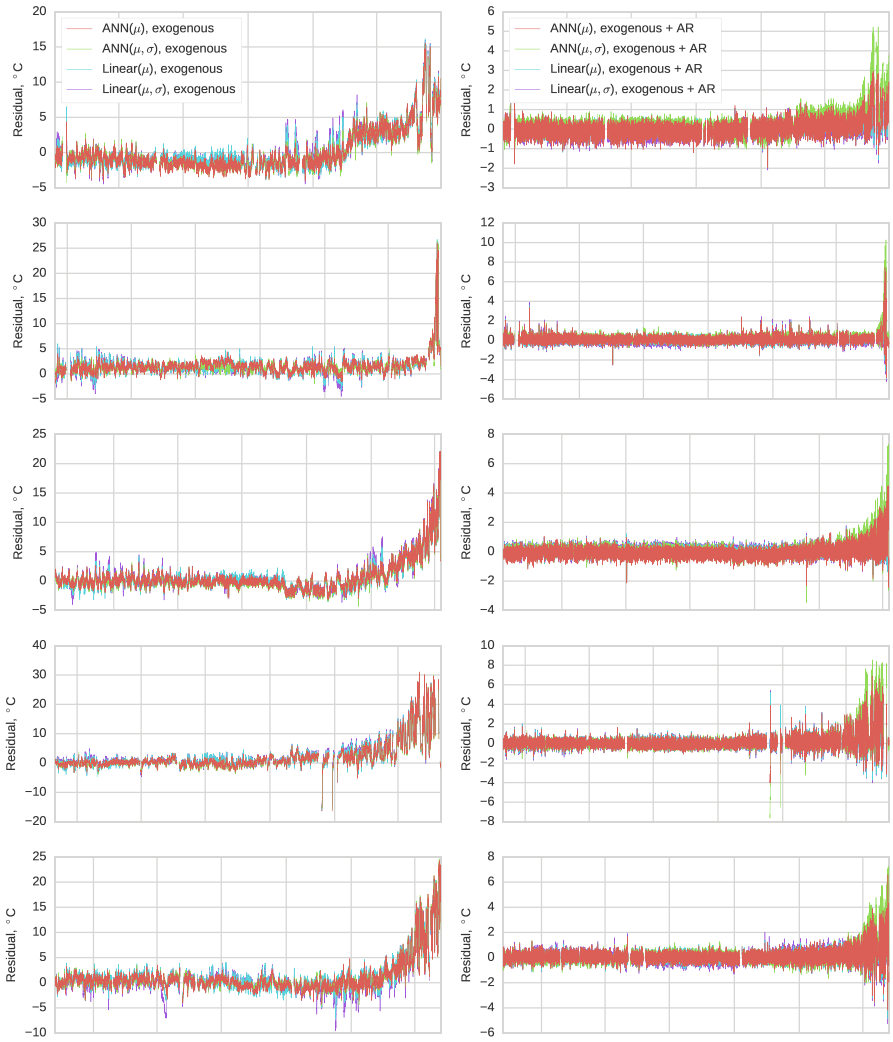


Figure 3. Temperature residuals for five different bearing faults covering the last 18 months before bearing failure with a split between exogenous (left plots) and exogenous + AR (right plots) input classes.

In the aforementioned description, a number of additional hyper parameters were introduced. An exhaustive study on how choices for these parameters influence the results have not been performed, but it was found that the overall conclusions drawn from this study were relatively insensitive to specific choices of these parameters.

The detection performance of the models are visualized using receiver operating characteristics (ROC)[15] defined by operating points by the classifier in (fpr, tpr)-space, where $fpr = \frac{FP}{N}$ is the false positive rate and $tpr = \frac{TP}{P}$ is the true positive rate defined from the number of actual positives P, the number of classifier true positives TP, the number of actual negatives N and the number of classifier false positives FP. Such plots are a vital tool in a large-scale monitoring setup, as they convey the trade-off between detection rate (tpr) and the number of false positives (fpr), and provides a structured approach for comparing different classifiers.

The ROC curves are also condensed into a single performance metric, the area-under-curve (AUC), defined as the area under a given ROC curve. A classifier that separates positives from negatives perfectly scores AUC $= 1$. A completely random classifier will score AUC $= 0.5$. The AUC can be interpreted as the probability that a classifier will rank a positive sample higher than a negative sample. To summarize, these metrics quantify how well the models rank fault-data and no-fault data based on the distance measures r_n.

5.2. Fault detection results

First, the temperature residuals have been plotted for four different turbines with bearing faults in Figure 3. Again, the plots have been split between the two different input classes. The results are promising in the sense that increasing residuals can be observed months before the bearings reach critical temperatures, with the exception of the fault observed in the second turbine that develops at a faster rate. The first plots in Figure 3 highlights some differences between the two different input classes. Here, the main differences between the exogenous models are observed during the no-fault state, while the responses during the fault state are very similar. The AR models on the other hand show similar responses during the no-fault period, but the responses during the fault state diverges as we are in an extrapolation regime. Also, the response of the AR models is delayed compared with the exogenous models. It must also be noted that the residuals during a fault state differ significantly for the two input classes. Exogenous models, if accurate enough, will provide an estimate of the actual residual caused by the failure mechanism, but the same principle will not apply to the AR models, which generally output lower residuals compared with the exogenous models.

The overall detection performance of the models is not only affected by the response during a fault-state but just as much by the response during no-fault states. The ROCs compiled for the entire data set have been visualized in Figure 4 for each individual model architecture. Each plot contains curves evaluated at different values of Δt as described in Section 5.1. As the fault signal increases for decreasing values of Δt, the curves are generally pushed towards the upper left corner

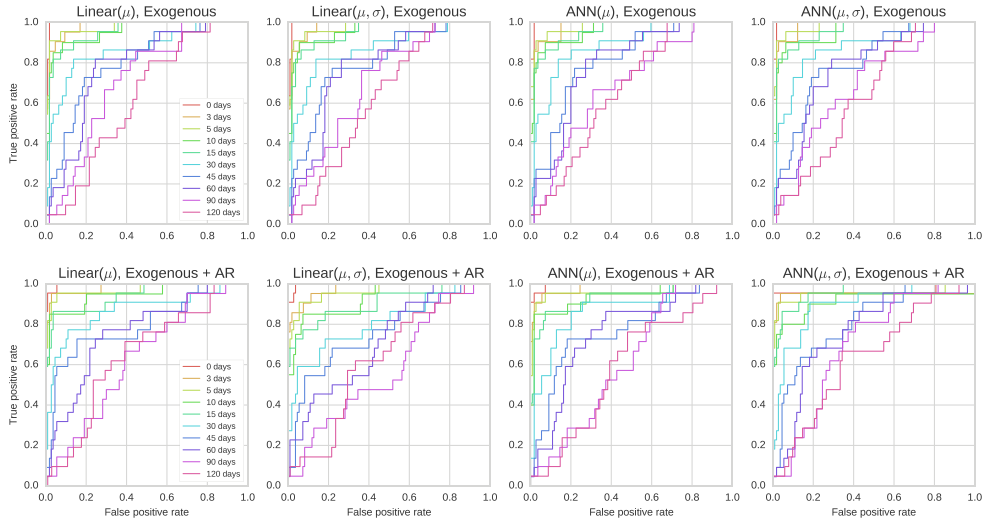


Figure 4. ROC curves for the various model architectures.

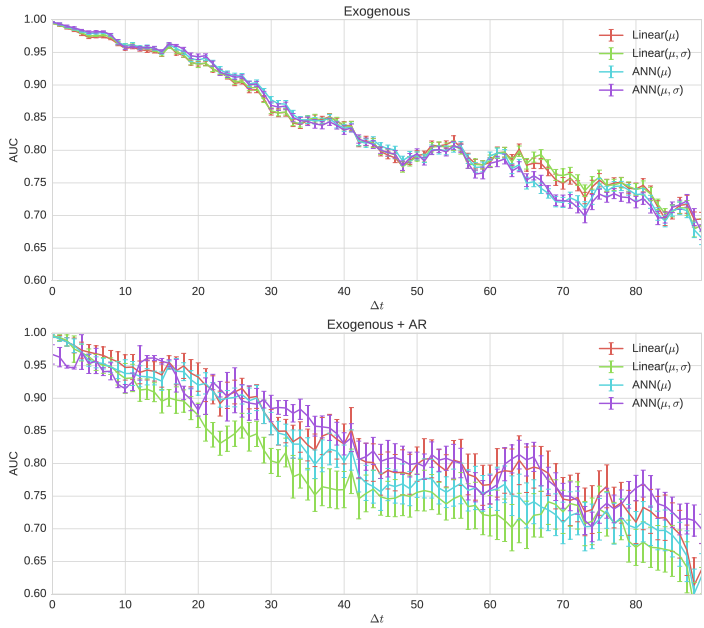


Figure 5. AUC score as a function of Δt for both input classes with error bars denoting 95% credibility intervals.

of the graph as expected. These curves have been condensed into AUC values evaluated in 1 day intervals in Figure 5 to summarize the overall classifier performance as a function of Δt. A sensitivity analysis has been performed by assuming normal distributed residuals within each 5 day evaluation window from which 5000 samples were drawn from the posterior of the individual interval means using a non-informative prior. The curves in Figure 5 are sample-based means, and the error bars indicate the sample-based 95% credibility interval. To quantify the overall performance in more detail, the AUC values are compared across model architectures in Table II. In the exogenous case, a significant performance gain can be observed for the non-linear models in the high-AUC region ($AUC > 0.8$), with the ANN(μ, σ) model leading the scores throughout most of this interval. In the AR case, the results overlap between the models, as also confirmed by the spurious rankings in Table II. The exogenous models provide overall the most stable results, and essentially approximates the upper bound on the accuracy obtained from the AR models.

When observing the ROC curves, isolated for each input class, the overall structure is very similar across models. From an operational point of view, the main differences between the highest and lowest AUC scoring models can be observed in the low-*fpr* and high-*tpr* regions. In these regions, high-AUC models can provide a substantially lower false positive rate for a given true positive rate at a given lead time Δt.

6. CONCLUSION

The main conclusions from this study are as follows:

- Based on very simple thermal considerations, data-driven flexible models were demonstrated to provide highly accurate predictions of the rotor-bearing temperature in wind turbines operating in a wide range of different conditions. The learned mappings proved to be robust years ahead once they had been fitted to a no-fault data period. This property is important if such models are employed for fault detection.
- The properties of model architectures with and without auto-regressive terms were investigated. Non-linearities present in the exogenous problem setting resulted in superior predictive performance by neural network-based models over linear models, with an overall root mean square predictive error as low as 1.13°C. As AR terms are included, the predictive task is simplified, and linear models provide the best overall predictive performance with an error of 0.31°C.
- Non-linear models with the added capability of an adaptive noise distribution did not provide robust long-term likelihood estimates. But, an adaptive noise distribution proved to be an effective regularizer as it allowed the models to

put less emphasis on prediction errors in high-noise regions during training, which resulted in more robust predictions in the distribution mean when evaluating years ahead. These models provided the best overall predictive results in the exogenous case.
- The potential for such models to be used in a fault detection system was quantified in detail using ROC and AUC metrics. The best performing models provided accurate detection ($AUC = 0.8$) approx. 45 days before reaching a critical bearing temperature.
- The benefits of strictly exogenous models over AR models was clearly demonstrated as (i) the objectives of the predictive task and the and the detection task are closely linked in the exogenous case, and the best predictive models were shown to also provide the best overall detection, which is an important point if only no-fault data are available; (ii) they provided a more robust detection system; and (iii) the residual from such models is interpretable as the residual from the fault mechanism itself.

Vibration-based systems can provide much earlier warnings of rotor bearing faults, as recently demonstrated in the literature[16] using a detection system using fully supervised learning. From a cost perspective, data enrichment strategies of the type presented in this study are of interest as they can provide somewhat early warnings at a fraction of the cost of advanced monitoring systems such as vibration-based systems. However, we believe that the question of how the outputs from various diagnostic system are used together is of greater interest. Even though a fault has already been confirmed using, e.g., vibration monitoring, additional modeling on temperatures give valuable insight into the failure progression and enables more accurate remaining useful life predictions. In all cases, this study provides industry practitioners with the performance data required to compare temperature-based modeling with other approaches to rotor-bearing monitoring.

7. FURTHER WORK

The work could be expanded in many different directions. With regard to the overall modeling choices, the ambition was to demonstrate how one could achieve high predictive accuracy using very basic engineering principles combined with data and flexible modeling techniques, but we did not perform an exhaustive analysis of how more complex input/output spaces could improve the results. Such an analysis could be based on more complex physical models and/or automatised feature selection. The inability of the models to provide robust long-term output distributions also warrants further investigation.

With regard to the model architectures, obvious alternative candidates would be support vector machines or Gaussian processes, but we do not expect the methods to improve the results in any significant way. Within the neural network framework, we did some initial testing with deeper networks, but these efforts did improve the overall results. As the data are sequential in nature, another obvious extension would be to use sequential models. Testing of long short-term recurrent neural network architectures[17] was also performed, but again, the results did not improve, which perhaps is not a surprise since sliding window-based feed-forward network already perform so well in the given problem setting.[18]

As for the detection itself, we only investigated a simple ranking based on the residuals, but much more complex methods could be tested, and the residual could also be used as additional input to a fully supervised fault detection scheme. The work has focused on the detection of critical faults, but the outputs of the models may also provide additional insight into the non-critical states of various turbine components. This also leads to the question of how the overall adaptation of the models to the turbines is optimized. Here, we used a fixed training window, but more elaborate online training methods could also be devised.

Finally, the fact that each turbine is part of a larger population is ignored in the current models, and applying hierarchical modeling techniques could also be an interesting extension to the presented analysis.

REFERENCES

1. Lu B, Li Y, Wu X, Yang Z. A review of recent advances in wind turbine condition monitoring and fault diagnosis. *Proceedings of Power Electronics and Machines in Wind Applications 2009*, IEEE, Lincoln, Nebraska, 2009.
2. Hameed Z, Hong YS, Cho YM, Ahn SH, Song CK. Condition monitoring and fault detection of wind turbines and related algorithms: a review. *Renewable and Sustainable Energy Reviews* 2009; **13**: 1–39.
3. Zaher A, McArthur S, Infield D, Patel Y. Online wind turbine fault detection through automated SCADA data analysis. *Wind Energy* 2009; **12**: 574–593.
4. Schlechtingen M, Ferreira Santos I. Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. *Mechanical Systems and Signal Processing* 2011; **25**: 1849–1875.
5. Guo P, Infield D, Yang X. Wind turbine generator condition-monitoring using temperature trend analysis. *Sustainable Energy, IEEE Transactions on* 2012; **3**: 124–133.
6. Zhang ZY, Wang KS. Wind turbine fault detection based on SCADA data analysis using ANN. *Advances in Manufacturing* 2014; **2**: 70–78.

7. Cross P, Ma X. Model-based condition monitoring for wind turbines. *Automation and Computing (ICAC), 2013 19th International Conference on*, IEEE, London, 2013; 1–7.
8. Talebi N, Sadrnia MA, Darabi A. Robust fault detection of wind energy conversion systems based on dynamic neural networks. *Computational Intelligence and Neuroscience* 2014; **2014**.
9. Kusiak A, Verma A. Analyzing bearing faults in wind turbines: a data-mining approach. *Renewable Energy* 2012; **48**: 110–116.
10. Butler S, O'Connor F, Farren D, Ringwood JV. A feasibility study into prognostics for the main bearing of a wind turbine. *Proceedings of IEEE International Conference on Control Applications 2012*, IEEE, Dubrovnik, 2012; 1092 –1097.
11. Wilkinson M, Darnell B, van Delft T, Harman K. Comparison of methods for wind turbine condition monitoring with SCADA data. *Renewable Power Generation, IET* 2014; **8**: 390–397.
12. Bishop CM. *Pattern Recognition and Machine Learning*. Springer, 2006.
13. Chen S, Billings SA, Grant PM. Non-linear system identification using neural networks. *International Journal of Control* 1990; **51**: 1191–1214.
14. Polyak BT. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics* 1964; **4**: 1–17.
15. Fawcett T. ROC graphs: notes and practical considerations for researchers. *Machine Learning* 2004; **31**: 1–38.
16. Bach-Andersen M, Winther O, Rømer-Odgaard B. Scalable systems for early fault detection in wind turbines: a data driven approach. *Annual Conference of the european Wind Energy Association*, EWEA, Paris, 2015.
17. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation* 1997; **9**: 1735–1780.
18. Gers FA, Eck D, Schmidhuber J. Applying LSTM to time series predictable through time-window approaches. In *Artificial Neural Networks, ICANN 2001*. Springer: Vienna, 2001; 669–676.

Appendix B

Article B

Martin Bach-Andersen, Bo Rømer-Odgaard, and Ole Winther. Scalable systems for early fault detection in wind turbines: A data driven approach. *Annual Conference of the European Wind Energy Association*, Paris, 2015.

Scalable systems for early fault detection in wind turbines: A data driven approach

Martin Bach-Andersen[1,2], Bo Rømer-Odgaard[1], and Ole Winther[2]

[1]Siemens Diagnostic Center, Denmark
[2]Cognitive Systems, Technical University of Denmark

Abstract

In order to leverage the complex data streams from modern wind turbines fully in large scale monitoring platforms, further automatization is required. This paper addresses this issue first by establishing a learning framework on which algorithms can learn from the diagnostic process performed by an experienced human expert, and second, establishing a structured methodology for evaluating the diagnostic performance of such algorithms. Building on this work and vibration data from 80 actual main bearing failures, deep networks are demonstrated to outperform shallow counterparts and rival human expert performance for this complex diagnostic task. The robustness of these methods make them suitable candidates for implementation in large scale monitoring platforms, and the study will provide a solid stepping stone for further research into data driven turbine diagnostics.

Keywords: Condition monitoring, data mining, machine learning

1 Introduction

The world has changed dramatically for wind farm operators and service pro-viders in the last decade. Organizations whose turbine portfolios was counted in 10-100s ten years ago are now managing large scale operation and service programs for fleet sizes well above one thousand turbines. A big challenge such organizations now face is the question of how the massive amount of operational data that are generated by large fleets are effectively managed and how value is gained from the data. A particular hard challenge is the handling of data streams collected from advanced condition monitoring systems. These data are highly complex and typically require expert knowledge to interpret correctly resulting in poor scalability when moving to large Operation and Maintenance (O&M) platforms.

In this paper we present a purely data driven fault detection method that has the potential to vastly improve the scalability of fleet wide condition monitoring systems as much of the complex diagnostic process can be put in algorithmic form. A feasibility study has been performed on data from 80 actual main bearing failures on both on- an offshore turbines.

2 Data driven fault detection

Recent reviews of condition monitoring applications in wind turbines can be found in [1] and [2]. The diagnostic task we focus on in this work is the detection of mechanical faults in the main bearing of geared turbines, typically spalling of raceways and/or rollers. From a diagnostic point of view, faults on main bearings are of particular interest because such faults can have a catastrophic impact on the operation of the turbine and significant resources and equipment are required to exchange a faulty bearing. For these reasons, significant cost savings can be gained by provi-

ding as early warnings of an impending bearing failure as possible. Vibration based diagnostics [3] is a mature technology for monitoring turbine drive trains including main bearings, and here we will restrict ourselves to vibration-based fault detection.

Two fundamentally different approaches exist for developing fault detection systems. One is a model-based approach typically using first principles. The second, is a data driven approach where the configuration of the diagnostic system is inferred from data. Data driven approaches for turbine diagnostics are typically unsupervised in the sense that models are fitted to no-fault conditions whereafter faults are identified as deviations from this model of normality [4] [5] [6] [7] [8] [9] [10] [11]. When moving to large scale monitoring platforms, a sufficient number of actual failure data is now available to approach the diagnostic problem using fully supervised classification, where models are trained to distinguish between the fault/no-fault state in the dichotomous case. One of the benefits of such an approach is for example that a probability of failure can be directly inferred from the output of the diagnostic models. The latter approach is adopted here.

3 Learning framework

3.1 Methodology

In order to frame the learning process we start by recognizing that state-of-the-art large scale vibration monitoring systems are based on a suite of automated outlier detection algorithms to generate warnings of potential faults, but the end diagnosis still heavily relies on human interaction as experts sift through the available data to provide early fault detection. Inspired by the diagnostic process performed by these analysts we will therefore explore whether the task can be fully or partly automated using deep and shallow learning architectures.

By observing how an experienced vibration analysis expert solves the task of detecting faults on a main bearing, a number of key elements was identified in this process, namely 1) feature selection, 2) time frame selection and 3) comparison with past observed faults. We will address these elements as follows:

Features A subset of two measurements was selected for this study, namely low-frequency acceleration autospectra (10-62 Hz) and low to mid-frequency acceleration autospectra (10-1000 Hz) from a single sensor mounted on the bearing housing. It is important to stress that we only perform feature selection on this high level. No attempts will be made to reduce these data to low dimensional diagnostic features using hand-crafted failure signal models or unsupervised methods. The aim is to have the models infer features of diagnostic relevance automatically from the high level feature set.

Time frame Based on input from the expert, a time window spanning back six months for both measurements was selected as input.

Comparison Full vibration data records were collected from turbines with known main bearing faults. The model was then trained to detect faults using data from these turbines as background knowledge.

In order to separate the fault from no-fault conditions on the selected turbines, the expert was tasked with specifying the change point from fault to no-fault condition using whatever data source he or she had available. This will be named as the Expert Retrospective Changepoint (ERC), as it is inferred from the full time record until present time. In an online setting, the time of detection will generally be later compared to the ERC as more data will have to be observed before an accurate detection can be made. By combining the information regarding the replacement of the faulty main bearing from sevice records with the specified ERC from the expert, time windows of faulty states could be obtained for each turbine.

Data windows used for model training and evaluation was now extracted using the following scheme where EXC denotes the component exchange date and U is the uniform probability distribution:

For each turbine: Sample a random time shift ΔT from $U(-90 \text{ days}, \min(+90 \text{ days}, \text{EXC} - \text{ERC})$. Shift 180-day window centered at ERC

by ΔT and add data in window to fault data. Repeat 10 times.

For each turbine: For period from first operation until ERC minus 2 years, extract non-overlapping 180-day windows and add data in windows to no-fault data. Repeat for period following component exchange.

The reasoning behind this scheme is 1) that the ERC itself can be viewed as a stochastic variable, and the data on which the models are trained should include this variability and 2) we seek models which robustly can recognize faults in both the early stages as well as the later stages of failure development.

To provide the data for model training and evaluation, failure records were obtained from the large scale monitoring platform operated by Siemens Diagnostic Center. This data set contains not only operational data and service records, but also detailed vibration data from more than 11,000 turbines collected during the last 12 years. From here, historic data from 80 turbines was collected in total. All turbines was from the same product family of geared turbines and all turbines had confirmed main bearing faults. These were the only criteria for turbine selection. The turbines selected are placed both on- and offshore and are spread across geographical regions. For model training and evaluation purposes, the set of turbines is now split into three subsets:

1. A training set of 40 turbines was selected randomly for model training.
2. A validation set of 15 turbines was selected randomly to tune the performance of the models on unobserved data.
3. A test set of 25 turbines was selected randomly for a final evaluation of the best performing models selected from the validation stage.

3.2 Data processing

To reduce the dimensionality of the inputs, each channel was first downsampled by a factor of 2 in the frequency domain. The available vibration data was not sampled equidistantly in time, so the data was further processed using linear interpolation on an uniform time grid with 100 samples in the time dimension. Following these steps the inputs comprised of two spectral images of size 100×198 and 100×1342 for the low- and mid-frequency channels respectively, with a total input dimensionality of $D = 154000$.

The initial preprocessing is followed by local contrast normalisation for each input image [12]. For each pixel $x_{i,j}$ the adjusted value is given by

$$\hat{x}_{i,j} = \frac{v_{i,j}}{\sigma_{i,j}}, \tag{1}$$

$$v_{i,j} = x_{i,j} - \sum_{p,q} w_{p,q} x_{i+p,j+q}, \tag{2}$$

$$\sigma^2_{i,j} = \sum_{p,q} w_{p,q} v^2_{i+p,j+q}. \tag{3}$$

Here, the weights $w_{p,q}$ are defined by an isotropic Gaussian kernel parameterized by scale parameters γ_f and γ_t for the frequency and time axes respectively. The time scale γ_t is shared across the two input channels, while the frequency scale γ_f is adjusted individually for each channel. The optimal values of these parameters are determined using cross validation for each model.

The local normalisation forces the models to learn richer pattern representations indicative of fault or no-fault states instead of relying on simple signal amplitudes.

4 Models

4.1 Logistic regression

The first model to be considered is a logistic regression model [13], where the posterior fault probability is parameterized as

$$p\left(\text{Fault}|\boldsymbol{x}\right) = \sigma\left(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}\right), \tag{4}$$

where σ is the sigmoid function, $\boldsymbol{w}$ is the parameters of the model and $\boldsymbol{x}$ is the input data vector with an implicit bias term included. To prevent overfitting the weights are constrained using L2 regularisation. With N training samples of input and output pairs $(\boldsymbol{x}_n, y_n)$, the error function to be minimized is

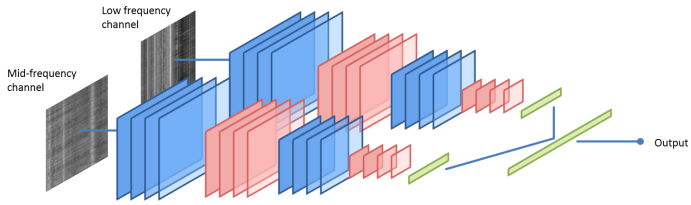


Figure 1: Schematic of the deep convolutional network investigated in this work. Convolutional layers are marked in blue, max-pooling layers are marked in red and fully connected layers are marked in green. Only four kernels are shown for each convolutional layer, details of the architecture are found in table 1.

then

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N} \ln\left(p\left(y_n|\boldsymbol{x}_n\right)\right) + \lambda \sum_{j=1}^{D+1} \boldsymbol{w}_j^2, \qquad (5)$$

where λ is a regularization parameter to be optimized via cross validation. The above error function is convex in the logistic case and can be minimized efficiently via iterative optimization procedures.

4.2 Deep network

As recent work on deep learning structures [14] has provided state-of-the-art results on complex image recognition tasks, a deep convolutional fault recognition model was implemented to investigate if a deeper architecture could provide increased diagnostic performance in the given learning setting. As in the case of the logistic regression we seek a parameterization of the posterior fault probability $p\left(\text{Fault}|\boldsymbol{x}\right) = f\left(\boldsymbol{w}, \boldsymbol{x}\right)$, but the function f is here constructed from a hierarchical structure of a number of nonlinear processing layers.
The model architecture employed here is depicted in figure 1. The overall architecture consists of two individual recognition networks covering each input channel. The input to these subnetworks is first processed by four convolutional and max-pooling operations followed by a 1024-unit fully connected layer. The individual networks are combined using a third fully connected layer, followed by a single sigmoidal output unit. For further details of the architecture we refer to table 1.

In order to train the network we strictly used stochastic gradient descent (SGD) with mini-batches. The regularised cross entropy error function from (5) is used for training again, however with $\lambda = 0$ and averaging across errors in each mini-batch. To prevent overfitting of the network, dropout [15] is instead applied to all fully connected layers, with dropout probability $p = 0.5$. The update steps in the descent procedure are as follows:

$$\boldsymbol{w}_{\tau+1} = \boldsymbol{w}_\tau - \boldsymbol{\eta}(\tau)\nabla_{\boldsymbol{w}} E(\boldsymbol{w}_{\boldsymbol{\tau}}), \qquad (6)$$

where the learning rate η is both parameter dependent and training time τ dependent in the general case. Here, the learning rate is adapted using AdaGrad [16], and the batch size was set to 8 in all cases. Before training, all parameters were initialized according to the heuristics given in [17].

5 Results

From the training and validation stage the best performing models were tested on the test set of turbines. The results are listed in table 2. Both models provide high accuracies on the test set, but the deep network significantly outperforms the logistic model with an error rate of only 5.5%. To further understand the differences between the output of the two models, the posterior fault probabilities are plotted for a selection of four turbines from the test set in figure 2 as a function of time. These plots illustrate that both models provide a strong signal separation from the no-fault to the no-fault

Model layer	Low frequency channel	Mid-frequency channel
Convolutional layer	16 kernels, size 32×4	16 kernels, size 32×4
Max-pooling layer	4×2 pooling	3×2 pooling
Convolutional layer	32 kernels, size 16×4	32 kernels, size 16×4
Max-pooling layer	4×2 pooling	3×2 pooling
Fully connected layer	1024 units	1024 units
Fully connected layer	512 units	
Output	1 unit	

Table 1: Deep network details. All kernel and pooling operations are reported in frequency $\times$ time domain. Rectified linear units were employed for all layers exept the output unit which is sigmoidal. The network contains 67 million parameters in total.

state and back again when the bearing is replaced. Some differences between the outputs can however be observed:

1. The deep network outputs low fault probabilities throughout the no-fault periods. The output from the logistic model is more noisy, a most clearly observed in figure 2b where variation correlated with year cycles are present during the no-fault periods.

2. The deep network is more confident of its failure prediction at an earlier stage compared to the logistic model. An impressive result is observed in figure 2d where the network seemingly in an online setting predicts an impending failure more than a year in advance compared to the human expert.

From these more qualitative observations on a subset of turbines we now wish to evaluate the overall diagnostic performance on the entire test set in terms of Receiver Operating Characteristics (ROC) [18]. ROCs specify the classifier performance in terms of the True Positive Rate (TPR) and False Positive Rate (FPR), and a desirable property of any classifier is of course a high TPR coupled with a low FPR. We are however also interested in the performance of the classifiers as a function of time relative to the damage. Before the analysis can be completed across different failure cases, a common reference point in time is therefore required for each turbine. As in the case of the training procedure the ERC is chosen as this reference point, and all performance metrics will now be reported *relative* to the ERC.

Model	Test error rate
Logistic regression	9.9%
Deep network	5.5%

Table 2: Results on test set specified in terms of error rate

The resulting plots can be observed in figure 3 for both models. The plots have been generated by a moving average approach relative to the ERC where the output from the models has been averaged in 10 day windows. These outputs are then compared to similar averages from the models obtained during no-fault periods. Each line in the plots defines the classifier performance in terms of this relative distance to the ERC, and points on the individual lines indicate operating points measured in terms of TPR and FPR defined by an implicit model sensitivity, or p-value, selected for that classifier. A high p-value will result in robust predictions with a low FPR, but a lower TPR can also be expected, as also be observed from the plots.

The qualitative observations from the individual turbine plots are also reflected in the comparison across all test turbines. In general, the deep network will detect a higher rate of the actual failures with a lower rate of false positives compared to the logistic model, measured at a specific point in time relative to the ERC. Particularly in the region

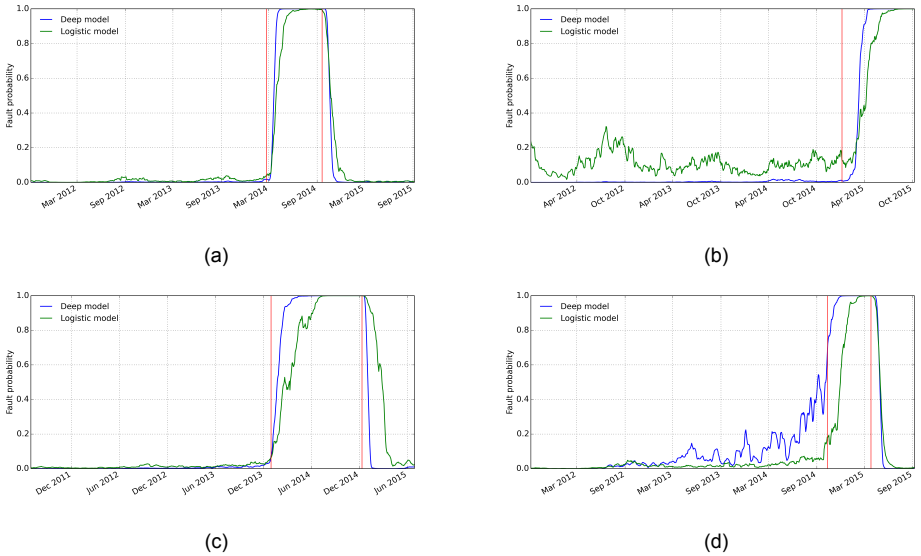
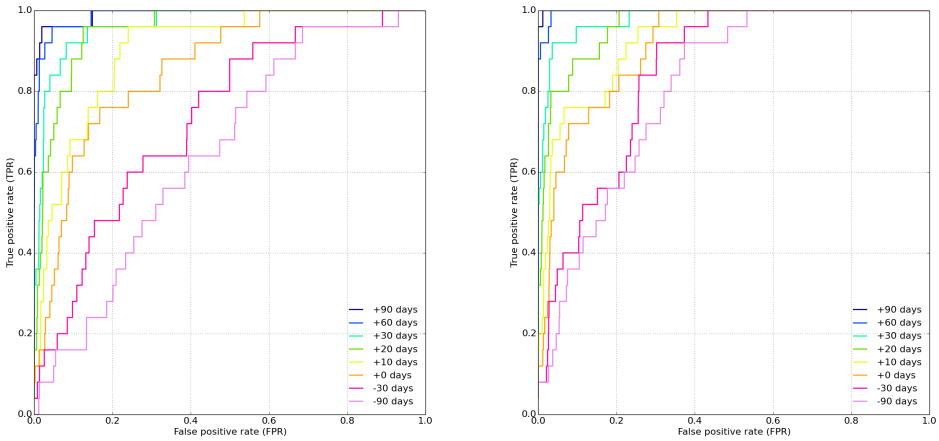


Figure 2: Examples of the model fault predictions near the fault onset time for four of the turbines in the test set. The ERC and the component exchange date has been marked with red vertical lines. A clear indication of a fault can be observed in all cases, as well as the return to a no-fault condition when the component is exchanged.

from +60 to +90 days the deep network is able to predict all failures at a very low false positive rate compared to the logistic model.

The overall classifier performance as a function of required detection lead time can be condensed to a single value, namely the Area-Under-Curve (AUC) which is the area below the given ROC-curve. The AUC-values are plotted in figure 4 and show that the deep model scores higher throughout the entire relative time range. From approx. +90 days the deep network is close to a perfect classifier (AUC=1) whereas similar performance is first obtained using the logistic model at approx. +150 days.



(a) (b)

Figure 3: Receiver Operating Characteristics for logistic regression (left) and deep network (right). The curves are generated by measuring the average output of the models over a 10 day window measured relative to the ERC for the turbines in the test set.

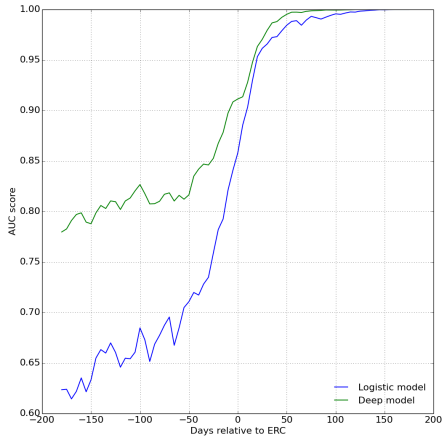


Figure 4: Comparison of overall classification performance of the two models using AUC-scores. Each score-value is calculated from a detection lead time specific ROC-curve as shown in figures 3a and 3b. The deep model is observed to outperform the shallow logistic model for the entire relative time range.

6 Conclusion

This study provides thorough insight into potential strategies for automating the complex diagnostic process involved in early fault detection in wind turbines. The key findings are as follows:

1. A learning framework has been proposed that enables algorithms to learn from the diagnostic process performed by experienced human experts.
2. The ability to train both shallow and deep architectures to detect faults on main bearings with a completely data driven approach was demonstrated using the above learning framework on 80 actual main bearing faults.
3. A structured methodology was etablished for evaluating diagnostic model performance across a turbine fleet using Receiver Operating Charateristics and AUC metrics.
4. With the ability of deep networks to extract high level abstract features from complex data streams, the performance of these models outperformed shallow architectures. In some cases the network was able to detect faults more than a year before a human expert although the network has strictly been tested in an online setting whereas the expert has the entire data history available for the diagnostic task.

The results provide a clear path towards a greater automatization of complex diagnostic tasks relevant for wind turbines as well as other type of machinery. The methods presented are very general in nature and can easily be adopted to other tasks.

References

[1] B. Lu, Y. Li, X. Wu, and Z. Yang, "A review of recent advances in wind turbine condition monitoring and fault diagnosis," in *Proceedings of Power Electronics and Machines in Wind Applications 2009*, IEEE, June 2009.

[2] F. P. García Márquez, A. M. Tobias, J. M. Pinar Pérez, and M. Papaelias, "Condition monitoring of wind turbines: Techniques and methods," *Renewable Energy*, vol. 46, pp. 169–178, 2012.

[3] R. B. Randall, *Vibration-based Condition Monitoring: Industrial, aerospace and automotive applications*. Wiley, 2011.

[4] A. Zaher, S. McArthur, D. Infield, and Y. Patel, "Online wind turbine fault detection through automated scada data analysis," *Wind Energy*, vol. 12, no. 6, pp. 574–593, 2009.

[5] M. Schlechtingen and I. Ferreira Santos, "Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection," *Mechanical Systems and Signal Processing*, vol. 25, no. 5, pp. 1849–1875, 2011.

[6] A. Kusiak and A. Verma, "Analyzing bearing faults in wind turbines: A data-mining approach," *Renewable Energy*, vol. 48, pp. 110–116, 2012.

[7] P. Guo, D. Infield, and X. Yang, "Wind turbine generator condition-monitoring using temperature trend analysis," *Sustainable Energy, IEEE Transactions on*, vol. 3, no. 1, pp. 124–133, 2012.

[8] Z.-Y. Zhang and K.-S. Wang, "Wind turbine fault detection based on scada data analysis using ann," *Advances in Manufacturing*, vol. 2, no. 1, pp. 70–78, 2014.

[9] P. Cross and X. Ma, "Model-based condition monitoring for wind turbines," in *Automation and Computing (ICAC), 2013 19th International Conference on*, pp. 1–7, IEEE, 2013.

[10] N. Talebi, M. A. Sadrnia, and A. Darabi, "Robust fault detection of wind energy conversion

systems based on dynamic neural networks," *Computational intelligence and neuroscience*, vol. 2014, 2014.

[11] S. Butler, F. O'Connor, D. Farren, and J. V. Ringwood, "A feasibility study into prognostics for the main bearing of a wind turbine," in *Proceedings of IEEE International Conference on Control Applications 2012*, pp. 1092 – 1097, IEEE, October 2012.

[12] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2146–2153, IEEE, 2009.

[13] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[16] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[17] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[18] T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," *Machine learning*, vol. 31, pp. 1–38, 2004.

Appendix C

Article C

Martin Bach-Andersen, Bo Rømer-Odgaard, and Ole Winther. Deep learning for automated drive train fault detection. Submitted to *Wind Energy*, 2017.

WIND ENERGY
Wind Energ. 0000; **00**:1–14
DOI: 10.1002/we

Research Article

Deep learning for automated drive train fault detection

Martin Bach-Andersen[1,2], Bo Rømer-Odgaard[1] and Ole Winther[2]

[1]Siemens Diagnostic Center, Denmark [2]Cognitive Systems, Technical University of Denmark

ABSTRACT

A novel data driven deep learning system is presented for large scale wind turbine drive train applications. The system is demonstrated to learn successfully from the actions of human diagnostic experts, and provide early and robust fault detection on both rotor bearing, planetary and helical stage gear box bearings from analysis of complex multi-sensor vibration patterns. Based on data from 251 actual wind turbine bearing failures, we are able to accurately quantify the fleet wide diagnostic performance of the diagnostic models, and deep architectures are shown to outperform the human analyst as well as shallow learning architectures. The results demonstrate that when applied in a large scale monitoring system, machine intelligence is now able to handle some of the most challenging diagnostic tasks related to wind turbines.

KEYWORDS

Condition Monitoring, Fault-Detection, Data Analysis, Artificial Intelligence

Correspondence

E-mail: martin.bach-andersen@siemens.com

Received . . .

1. INTRODUCTION

Accurate and early detection of faults in wind turbine components play an important role in modern wind farm maintenance programs. This is particularly true for off-shore wind farms, where limited turbine accessibility and overall cost and complexity of turbine maintenance are some of the key cost drivers behind developing advanced diagnostic systems. In this work we focus on the detection of faults in the drive train in geared wind turbines as these faults are typically random in nature with a potentially severe negative impact on the continued operation of the turbine and with a high risk of further damage to other components.

The application of machine learning techniques in a wind turbine fault detection context is not new, but have typically been restricted to low-dimensional SCADA-based data streams and unsupervised learning paradigms where no explicit fault data is included [1, 2, 3, 4]. Similarly, such models have also been applied to vibration data, but is typically preceded by a low-dimensional feature selection [5, 6, 7] based on a classical signal processing approach, where relevant diagnostic features are extracted based on theoretical and semi-empirical fault signal models [8].

The work presented here departs from these approaches. First, our focus is on large scale deployments with thousands of turbines where enough fault data is available to make fully supervised learning feasible. Supervised means that the diagnostic system is trained directly to separate fault-states from no-fault states. Second, we base this learning on the actions of non-causal human diagnostic experts. Third, we lessen the requirements for detailed feature engineering and instead feed the learning algorithms with a full high-dimensional multi-sensor data stream. The question we wish to answer is therefore if machine learning architectures embedded in a large scale monitoring system can learn from its

Prepared using weauth.cls [Version: 2010/06/17 v1.00]

human operators and potentially provide a fully automatized scalable diagnostic solution for complex fault scenarios. The main contributions of the presented work are:

1. A novel deep learning architecture is trained in a fully supervised sense to provide robust fault detection in complex multiple-fault, multiple-sensor scenarios.
2. Using a large data set of actual fault data we are able to accurately quantify the diagnostic performance of various model architectures. To the knowledge of the authors, this study represents the largest ever conducted on the performance of machine intelligence when applied to complex fault detection in wind turbine drive trains.

The paper is structured as follows: the data set is presented in section 2, the formulation of the learning problem in section 3, the model architectures in section 4, fault detection results in section 5, visualization of the trained convolutional network in section 6, findings are concluded in section 7 and potential extensions to the work are discussed in section 8.

2. DATA SET

In this section we discuss the selection of drive train faults we want to detect, the sensor input and the expert output we want our model to predict.

2.1. Drive train faults

The drive train in a modern sized geared wind turbine is a complex mechanical system with a multitude of different failure modes. To facilitate a structured and detailed analysis of the performance of the various diagnostic models, we will restrict our attention to only a few, but important failure modes of the drive train. As a first restriction, we will only consider bearing faults. Bearing failures are a common cause of turbine downtime and a key component of any wind turbine condition monitoring system is the ability to detect bearing failures early and accurately. As a second restriction, we will only consider bearing faults at three specific locations, namely rotor stage bearings, planetary stage bearings and helical stage bearings. This choice has been made to demonstrate the fault detection capability of the models across different rotational speed regimes in the drive train.

From the service records from a large scale monitoring platform containing data many thousands of turbines, a number of turbines were selected to provide the data set on which the models were trained, evaluated and tested on. Only turbines from a specific turbine family were selected to ensure that the overall mechanical configuration was similar across the turbines. The set is from a mechanical point of view however not entirely homogeneous. Most importantly the set consists of two major subgroups each with a different gear box configuration. Mechanical variation also exists within each of these groups, as the subcomponents (bearings etc.) can vary from gear box to gear box. This also highlights the difficulties with existing monitoring solutions, as the mechanical configuration must be specified in detail in order to detect faults at a very early stage. The individual fault classes are also highly heterogeneous as the faults can be classified into further fault subcategories, for example bearing inner ring spalling, outer ring spalling and combinations hereof. To simplify the analysis, we did not pursue a further breakdown into subcategories. This introduces additional variation of the input data with respect to the fault detection task. The methods presented do however still provide a key functionality for a practical implementation, namely the overall location of the identified fault within the drive train.

In total, data from 251 confirmed bearing faults were included in this study divided on 85 rotor bearing faults, 63 planetary stage bearing faults and 103 helical stage bearing faults. The faults have been collected from both on- and off-shore turbines. From the fault data set, five random training, validation and test splits are generated on the turbine level. For each random split, 25% of the turbines are held out for testing, 10% for validation and the final 65% are used for model training. The models are then trained and evaluated across all splits.

2.2. Sensor data

The sensor data used for fault detection are based on vibration signals measured by an online vibration monitoring system installed in every turbine. Data from four accelerometers are utilized in this study; a sensor mounted on the rotor bearing housing (sensor 1), a sensor mounted on the gearbox housing near the planetary stage (sensor 2) and two sensors mounted near the helical stage (sensor 3 and 4). The data collected are point measurements evaluated on small time intervals of less than 2 minutes of duration. Two primary types of data are extracted:

Wind Energ. 0000; **00**:1–14 © 0000 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

Acceleration autospectra are extracted from sensor 1, 2 and 3. The upper frequency range of the spectra is scaled from 63 Hz at sensor 1, to 125 Hz at sensor 2 to 300 Hz at sensor 3, reflecting the increase in rotational speed, and frequency range of interest, from the rotor stage towards the high-speed output of the gear box.

Envelope acceleration autospectra are extracted for sensors 2, 3 and 4. Such measurements often provide an increased signal-to-noise ratio for bearing diagnostics on gear boxes [8]. The upper frequency of the envelope spectra is 500 Hz in all cases.

In total, six different measurements are collected across the four different sensors. To avoid frequency smearing, all recordings have been synchronized with the rotational speed of the turbine and normalized to the nominal rotational speed to provide a stable frequency pattern despite speed variation within and across the individual recording time windows.

For each measurement, a sliding window approach is used where data windows spanning a given time interval is extracted as explained in section 3. As a result, each data point $\boldsymbol{x}_n$ is a collection of six time $\times$ frequency feature maps with each map containing 50 samples in time. The combined sensor input is extremely complex with a data dimensionality of 210,000.

As found in [9], a final local normalization [10] of the data windows is required in order for the models to provide robust predictions. Here, we strictly employ normalization in the frequency-dimension. For each pixel $x_{i,j}$ the adjusted value $\hat{x}_{i,j}$ is given by

$$\hat{x}_{i,j} = \frac{v_{i,j}}{\sigma_{i,j}}, \tag{1}$$

$$v_{i,j} = x_{i,j} - \sum_k w_k x_{i,j+k}, \tag{2}$$

$$\sigma^2_{i,j} = \sum_k w_k v^2_{i,j+k}. \tag{3}$$

The weights w_k are defined by an Gaussian kernel parameterized by a scale parameter κ measured in feature map pixels adjusted individually for the groups of vibration and envelope spectra. The optimal values of these parameters was determined using validation data and was set to $\kappa = 100$ pixels for the vibration spectra and $\gamma = 40$ pixels for envelope spectra, with a kernel truncated at $\pm 4\kappa$. Without this normalization strategy the models were found to base their predictive strategy mostly on amplitude levels and not complex frequency pattern recognition, which in turn leads to inferior predictive performance for unobserved data points.

It must finally be emphasized that the input selection should not be considered to be a complete set for detailed drive train diagnostics, nor has the selection itself been optimized towards better diagnostic performance across the different architectures. Instead, a high-level feature selection has been made solely based on observations from the diagnostic process performed by human experts.

2.3. Expert output

To provide a learning target for the discriminative models, an experienced human expert has performed a detailed analysis of the data for each of the included faults. Here, the expert sets a point in time where the first signs of a fault is visible in the data streams. This point in time is denoted the Expert Retrospective Changepoint (ERC). From a fault detection perspective, this setup is quite favorable as the expert has prior knowledge of the fault history of the turbine, and has access to the entire data time history when designating the ERC, which makes the human detection non-causal. The time of repair or replacement of the component in each case has also been extracted from the turbine service records. This point in time is denoted EXC. How these data are used in the training procedure is described in detail in section 3.

3. TRAINING

Given the full historic sensor data tracks and expert outputs for each turbine, the data must now be structured in order to facilitate learning for the considered model architectures. This process will be described in detail here.

The learning framework is a multi-task extension of the framework described in [9]. First, we will consider input to the models $\boldsymbol{x}$. As already outlined in the preceding sections we will not restrict ourselves to instantaneous data points. Instead, a full data window spanning a predefined time interval is fed into the diagnostic models which allows the models to adapt to not only the current input pattern but also the development of these patterns over time. This choice of input architecture

is founded in how the human experts actually perform the analysis of the data. The data window has been set to a duration of $L = 6$ months, again based on the input from human experts. No further optimization of this hyperparameter was attempted.

As the next step, we will consider the target values y_n to which the models must learn to map the inputs $\boldsymbol{x}_n$ to. An example of the event history for a turbine is illustrated in figure 1 with a fault recorded for both the rotor and helical stage. For each component, the fault signal has been plotted against time (no fault $= 0$, fault $= 1$). In the fault period between the detection date ERC and the exchange date EXC, it is straightforward to set the target as $y = 1$. For the target values *before* the ERC, we simply discard the data for a period of 2 years back in time. The motivation for this strategy is as follows: The ground truth, being the actual state of the component, is generally unknown and the component may actually be in a fault state leading up to the ERC. To avoid a possibly wrong target value in this period of uncertainty we simply discard these data points such that model configurations that gives a high fault probability in this period is not punished during training. An important point here also is that by using the ERC as a fault-onset threshold, we are by design forcing the models to learn to detect faults earlier than the human experts as the ERC is determined non-causally with a prior knowledge of the fault history.

From the time tracks for each turbine, pairs of $\boldsymbol{x}_n, y_n$-values are now sampled randomly with a fixed sample count for each turbine, where $\boldsymbol{x}_n$ is a data window containing 50 equidistant data points from the last 6 months. The diagnostic models are then trained on the combined data sets across all turbines.

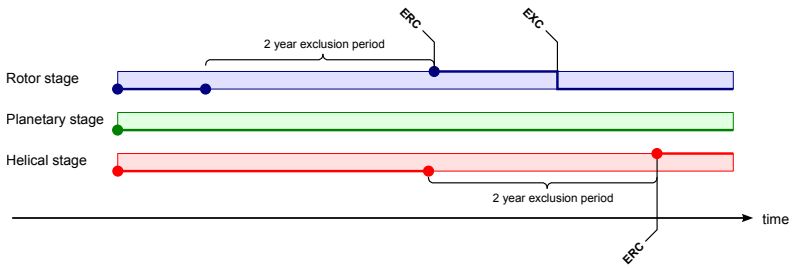


Figure 1. Fault signals for rotor stage, planetary stage and helical stage components for a single turbine. A fault on the helical stage has been recorded for this turbine as illustrated by the interval enclosed by the ERC, EXC-dates. For components with recorded faults, a 2-year interval leading up to the fault detection is excluded from the data set.

4. MODEL ARCHITECTURES

We consider three different model: logistic regression, fully connected and convolutional neural networks.

4.1. Logistic regression

To establish a baseline for the predictive performance we first consider a logistic regression model [11], where the posterior fault probability is parameterized as

$$p\left(\text{Fault}|\boldsymbol{x}\right) = \sigma\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\right) \tag{4}$$

where σ is the logistic function $\sigma(z) = 1/(1 + \exp z)$, $\boldsymbol{w}$ is the parameters of the model and $\boldsymbol{x}$ is the input data vector with an implicit bias term included. We encode the No fault/Fault label with $y \in \{0, 1\}$. With N training samples of input and output pairs $(\boldsymbol{x}_n, y_n)$, the regularized negative log-likelihood cost function takes the form

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N} \ln p\left(y_n|\boldsymbol{x}_n\right) + \sum_{j=1}^{D+1} \lambda_j w_j^2 \tag{5}$$

$$= -\sum_{n=1}^{N} \left(y_n \ln\left(\hat{y}_n\right) + \left(1 - y_n\right) \ln\left(1 - \hat{y}_n\right)\right) + \sum_{j=1}^{D+1} \lambda_j w_j^2 \,, \tag{6}$$

DOI: 10.1002/we
Prepared using weauth.cls

where $\hat{y}_n = p\left(\text{Fault}|\boldsymbol{x}_n\right)$ are the posterior fault probability estimated by the model and λ_j is a per-weight decay regularization parameter to be optimized using validation data. The optimal choice of the weights $\boldsymbol{w}$ cannot be solved analytically for any of the models considered here, so one must reside to numerical optimization techniques. We strictly employ first-order variants of stochastic gradient descent for this optimization, as these methods scale well with respect to both data size and model complexity. For the logistic regression model specifically, the weights are adjusted using stochastic gradient descent with a momentum term [12]:

$$\boldsymbol{v}_{\tau+1} = \boldsymbol{v}_\tau \gamma - \eta \nabla_{\boldsymbol{w}} \tilde{E}(\boldsymbol{w}_\tau) \tag{7}$$

$$\boldsymbol{w}_{\tau+1} = \boldsymbol{w}_\tau + \boldsymbol{v}_{\tau+1} \, , \tag{8}$$

where γ is the momentum parameter, η is the learning rate and $\tilde{E}(\boldsymbol{w}_\tau)$ is the estimate of the true gradient of the cost function evaluated on a subset (mini-batch) drawn randomly from the training set. The velocity is initialized to $\boldsymbol{v}_0 = 0$. Weights are initialized according to the scheme outlined in [13]. The training algorithm hyperparameters are tuned using validation data.

4.2. Fully connected neural network

To explore a more expressive non-linear model, the second model is a fully connected two-layer neural network parameterized by

$$p\left(\text{Fault}|\boldsymbol{x}\right) = \sigma\left(\boldsymbol{w}_2^T \boldsymbol{f}(\boldsymbol{W}_1^T \boldsymbol{x})\right) \, , \tag{9}$$

where $\boldsymbol{W}_1$ is the weight matrix for the first layer, f is the hidden layer non-linearity (activation function) applied element-wise, $\boldsymbol{w}_2$ is the weight vector for the second layer and σ is the sigmoid function. Like for the logistic regression model we add one constant dimension to both the input and hidden layer in order to include bias terms. The hidden units are non-linear feature extractors that allow the neural network to adapt to complex data distributions compared to the logistic regression model which only adapts a linear decision boundary. The number of hidden units for the fully connected networks has been fixed to $M = 60$ in all cases with a $\tanh$ activation function. This number has been chosen to approximately match the number of free parameters between the fully connected neural network (FC NNs) and the deep convolutional neural networks (Deep CNNs) explored in the next section. The neural network models' cost function is not, contrary to logistic regression, guaranteed to be convex. This makes good heuristics for initialization and regularization more important for successful training.

4.3. Deep convolutional neural networks

The architectures considered so far are all permutation invariant in the sense that the models perform equally well on arbitrary permutations of the input data as no assumptions are made regarding the spatial structure of the data. The vibration patterns fed into the models are however highly structured in both frequency and time dimensions, and we now seek a model architecture that is able to capture and utilize these spatial structures in the diagnostic process. So spatial in the following is to be understood as frequency$\times$time.

The three key architectural elements of our extended neural network models are listed below:

Convolutions: Convolutional layers are network layers with local spatial connectivity and weight sharing, or in other words, a bank of non-linear local spatial filters applied across the spatial dimensions of the input. The weight sharing greatly reduces the number of free parameters and makes convolutional layers less amenable to overfitting. The primary hyperparameters of convolutional layers are the kernel size (width$\times$height), the kernel strides and the number of kernels.

Pooling: Each convolutional layer is followed by a max-pooling operation that provides spatial downsampling of features and robustness towards small spatial deformations of the input. The primary hyperparameters of the pooling layers are the window size (width$\times$height) and strides.

Network depth: The major achievements in the last years within the domains of image recognition [14], text translation [15], speech recognition [16] and reinforcement learning [17] have all been driven by advancements in the field of deep learning. As the name implies, the major difference between classical shallow learning architectures and deep learning architectures is the network depth, or number of layers. The success of deep architectures compared to shallow counterparts is rooted in their ability to extract exceedingly higher level, abstract features of the data which in turn enables the models to generalize well and provide accurate inference on new observations [18].

These points, combined with the promising results in [9], motivated us to investigate if deep convolutional neural networks can provide an overall robust diagnostic model framework for fault detection in wind turbine drive trains based on complex input signals from multiple sensors. For a more detailed description of convolutional neural network architectures we also refer to [10, 14].

The architecture found to provide the best overall results is illustrated in figure 2. The model consists of four different processing pipelines, one for each of the spectrum input windows, and one combined pipeline for the envelope spectrum windows. A kernel size of $[4 \times 8]$ with strides of $(1, 1)$ is used throughout the network. Pooling window sizes have been restricted to $[1 \times 2]$ (first layer) and $[2 \times 2]$, with non-overlapping strides. Compared to the model used strictly for rotor bearing faults in [9], we use smaller kernels but more layers. Different weight sharing strategies were first tested, from full weight sharing between all kernels across all pipelines, to partial sharing across the first N layers, and finally no sharing at all. The last strategy provided the best results, meaning that kernels are adapted individually on each pipeline with the exception of the envelope spectra which were concatenated before the first convolution. These individual measurements are from different sensors, but are processed exactly the same way in the data acquisition unit. Due to this high similarity, depth-wise kernels in the first layer improved the feature extraction. The other input pipelines are much more heterogeneous with different frequency ranges and resolutions, which is the reason for the share-nothing setup here. Following the convolutional layers, outputs are fed to a 128-node fully connected layer per pipeline, and then merged by a 1024-node fully connected layer followed by a final logistic output. Rectified linear units [19] are used as activation functions throughout the entire network.

As for the fully connected neural network and logistic model, we train the deep networks using the cost function specified in equation 6 but with changes to the regularization and optimization procedures. The final architecture has more than 13 million trainable parameters, and proper regularization must be considered in order to avoid overfitting. Regularization was performed by employing dropout [20] in both input, convolutional and fully connected layers during training. Weight decay was disabled ($\lambda_j = 0$). We found that an adaptive learning rate scheme [21] provided the fastest training of all deep models.

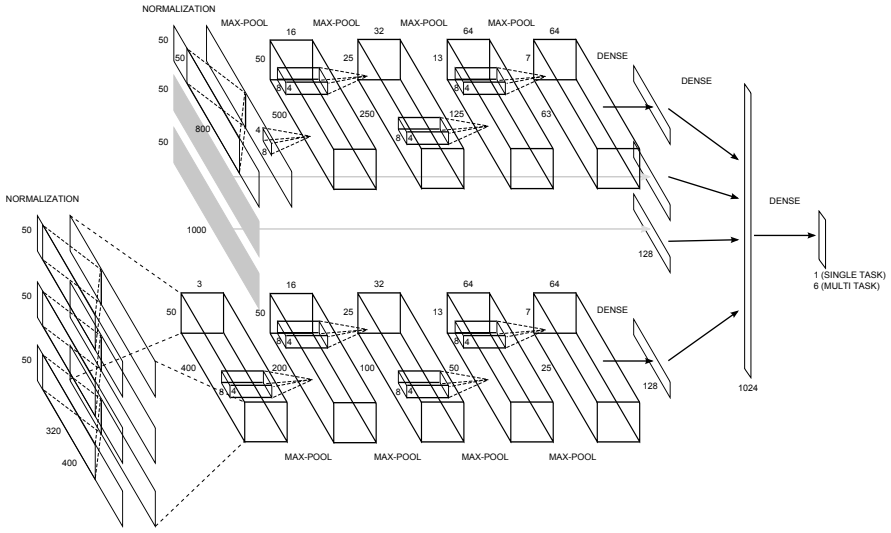


Figure 2. Deep network topology showing the processing flow from inputs (left) to outputs (right). The architecture is split in four processing pipelines, one for each of the spectral feature maps and one for the combined envelope spectral feature map. The second and third pipeline have only been outlined as the structure is the same across the first pipelines.

4.4. Transfer learning

By solving a number of distinct but related predictive tasks simultaneously, the performance of learning algorithms can in some cases be improved compared to single-task training [22]. This concept is also known as transfer learning. In the transfer learning set-up the model can possibly utilize internal representations helpful for one task to improve the

Wind Energ. 0000; **00**:1–14 © 0000 John Wiley & Sons, Ltd.
DOI: 10.1002/we
Prepared using weauth.cls

predictions for a different related task. We therefore test this hypothesis on all neural network architectures, as these contain internal (hidden) representations that may benefit from multi-task learning. The motivation in our specific problem domain is that relevant features may be shared both across both fault types (rotor, planetary and helical stage) and turbine classes (A and B). In total, we have six different predictive tasks.

To support multi-task training, the models and training procedures are changed as follows: Given K different tasks, the output layer of the models is extended to K logistic outputs. With a training set of tuples $(\boldsymbol{x}_n, y_n, k_n)$ where the index $k_n \in \{1, 2, \dots K\}$ denotes which task data point n belongs to, we use a masked cost function

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N}\sum_{j=1}^{K} \delta_{jk_n} \left(y_n \ln\left(\hat{y}_n\right) + (1 - y_n) \ln\left(1 - \hat{y}_n\right)\right) + E_{\text{reg}}(\boldsymbol{w}) \tag{10}$$

where $E_{\text{reg}}(\boldsymbol{w})$ denotes an arbitrary regularization cost. All parameters apart from those connected the last hidden layer to the specific logistic functions are shared across the different tasks. Using the masked cost function, the training of the multi-task models follow the same procedures outlined for the single-task models in the preceding sections.

5. FAULT DETECTION

We measure the fault detection performance in several ways. Below we give the results for the overall test performance and for several time-dependent measures.

5.1. Overall test metrics

In order to evaluate the diagnostic performance of the various model architectures, we will first consider the negative log-likelihood score for each of the models evaluated on the hold-out test sets. These metrics provide a basic measure of how well the model output distributions match the actual distributions of the target data. We will also evaluate the models in terms of the Area-Under-Curve score (AUC) which is a widely used metric for evaluating diagnostic classification algorithms [23]. The AUC is derived from the Receiver-Operating-Characteristic (ROC) of the classifier which is the curve in (fpr, tpr)-space, where fpr is the false positive rate and tpr is the true positive rate. The AUC-score can be interpreted as the probability that a given classifier will rank a positive sample higher than a negative sample, and is independent on the balance of the data set (number of positives vs. number of negatives). A perfect classifier has a score of AUC $= 1.0$, while a random classifier has a score of AUC $= 0.5$. AUC and ROCs are critical tools for evaluating diagnostic algorithms in large scale settings, as they enable direct comparison between different algorithms and highlight the trade-offs made between model sensitivity (the true positive rate) and the number of false positives. These metrics will be explored in more detail in the following sections. The test log-likelihood is straightforward to evaluate using equation 6. The score for each individual data point on which the AUC is calculated is simply the posterior fault probability $\hat{y}$. All tasks are evaluated individually, with separate log-likelihood scores and AUC-values.

The results are listed in table I given as mean values and standard deviations measured across the five random test set splits. Focusing on the likelihood scores, a significant margin is observed between the shallow and deep architectures, with lower errors for the deep architectures across all turbine classes and fault types. The largest margin is observed for helical faults on turbine class A, where the deep networks outperform the shallow models with a factor of ten. Within the family of deep architectures the performance between single- and multitask-models are comparable. The same applies for the shallow type family, with no significant difference across these models.

The superior log-likelihood scores of the deep networks is also observed to translate to better performance in AUC-scores, with deep networks achieving the best scores for all tasks considered. Again, no particular performance gap between single task and multi-task models can be observed. The AUC-scores are generally high, with average scores not below 0.93 for all models, and not below 0.97 for the deep architectures alone.

5.2. Time dependent detection metrics

The test metrics reported in section 5.1 are based on data with the same distribution as the training data, namely data points sampled uniformly in time for the individual turbine fault tracks. These results therefore provide an overall measure of how accurately the fault tracks spanning the entire life time of the turbines are predicted. To gain a more detailed understanding

		Negative log-likelihood			Area-Under-Curve		
		Rotor stage	Planetary stage	Helical stage	Rotor stage	Planetary stage	Helical stage
A	Logistic, single-task	0.16±0.04	0.21±0.10	0.22±0.11	0.960±0.017	0.947±0.047	0.926±0.058
	FC NN, single-task	0.19±0.03	0.21±0.10	0.19±0.05	0.958±0.016	0.938±0.066	0.931±0.055
	FC NN, multi-task	0.17±0.03	0.20±0.08	0.20±0.05	0.961±0.012	0.941±0.037	0.940±0.051
	Deep CNN, single-task	**0.11±0.04**	0.10±0.08	**0.02±0.01**	0.987±0.010	0.971±0.029	**0.998±0.002**
	Deep CNN, multi-task	0.12±0.08	**0.10±0.06**	**0.02±0.01**	**0.988±0.011**	**0.973±0.033**	0.994±0.005
B	Logistic, single-task	0.17±0.09	0.13±0.08	0.23±0.05	0.937±0.026	0.909±0.083	0.965±0.015
	FC NN, single-task	0.15±0.06	0.12±0.05	0.28±0.04	0.936±0.031	0.906±0.075	0.957±0.014
	FC NN, multi-task	0.15±0.05	0.11±0.05	0.25±0.05	0.935±0.028	0.924±0.072	0.960±0.014
	Deep CNN, single-task	**0.03±0.02**	**0.03±0.02**	0.11±0.06	**0.993±0.005**	**0.992±0.008**	0.995±0.003
	Deep CNN, multi-task	0.06±0.04	0.04±0.04	**0.08±0.04**	0.974±0.016	0.974±0.046	**0.996±0.003**

Table I. Overview of performance metrics for each model architecture. The best predictive scores for each input class have been highlighted in bold.

of how well the models perform in an online fault detection setting, the dependency of time is now included in the analysis. The methodology is based the work in [9], but we will reiterate the basic procedure here for the sake of completeness.

First, a common point of reference must be selected in order to move from an analysis of fault state predictions for individual turbines, to an analysis of the model performance in a fleet-wide sense. The ground truth, being the actual fault state of each component, is generally unknown. The best estimate we can obtain of the fault onset is the ERC. This point in time is therefore selected as the common point of reference when comparing predictions across turbines. As a result, model performance will now be measured in terms of $\Delta t = t - \mathrm{ERC}$. For negative values of Δt we are at a point in time *before* the retrospective detection by an human expert, for positive values we are at a point in time *after* human detection. Using this structured approach, Δt-dependent metrics are now compiled in the following way:

1. For each test set turbine, model predictions are calculated for all three components for entire turbine life span. Model fault probabilities are smoothed using a 5-day rolling average.
2. For each turbine, for each model, for each turbine class, for each component, for each Δt, the model output at $t = ERC + \Delta t$ is added to data set as a with-fault model score if a fault has been registered for this component.
3. For each turbine, for each model, for each turbine class, for each component, model outputs uniformly sampled during no-fault states (without the 2-year exclusion period) are added to data set as a no-fault model scores.
4. From the compiled data set, ROC and AUC metrics can now be calculated for individual models and components, as a function of Δt.

The above procedure is then repeated for all five random data set splits to provide uncertainty estimates of the calculated metrics. The key points to grasp from this evaluation procedure are 1) The models are now partly evaluated on out-of-bounds data points, namely in the exclusion zone ($\Delta t < 0$), and 2) all models are tested in a fully on-line manner compared to the human estimates which are based on retrospective evaluation with full knowledge of fault history and access to future data points. If accurate online fault predictions can be obtained just near non-causal human detection $\Delta t = 0$, the results will then indeed be promising.

5.2.1. Area-Under-Curve

The overall detection performance, specified by the AUC metrics, have been visualized in figure 3. Here, the mean value of the AUC has been plotted as a function of Δt together with min, max values as non-parametric uncertainty estimates. Results have been shown for each individual turbine class and component, and also as an average across the two turbine classes in the rightmost figures. We will comment on these results for each component individually. For rotor stage faults, the separation in results between shallow and deep architectures from table I is also reflected here, particularly for turbine class B. Deep models are observed to maintain an very accurate detection (AUC $= 0.9$) at $\Delta t \approx 0$ days. The response from the shallow models at this point is delayed approx. 50 days, and only achieves near-perfect classification (AUC $= 1.0$) at a much later point in time.

DOI: 10.1002/we
Prepared using weauth.cls

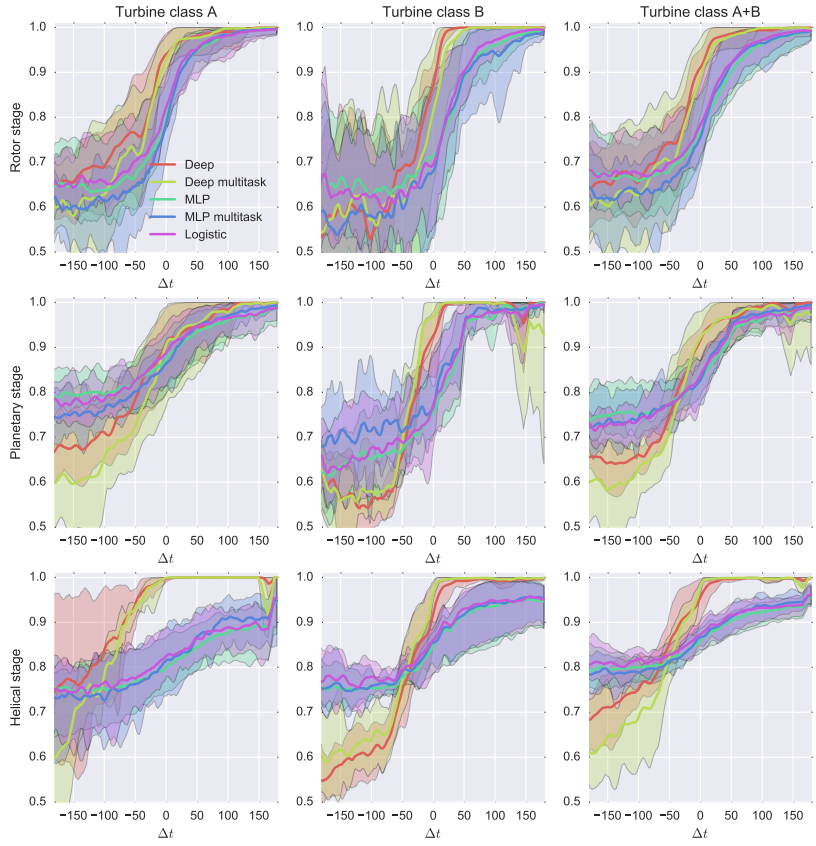


Figure 3. AUC-scores for various models evaluated by turbine class and component classes plotted as mean values across test splits. Non-parametric uncertainty estimates given by min, max values are shown as shaded regions for each curve. Aggregates scores for both turbine classes have been summarized in the right column.

For planetary stage faults, the results show even greater differences between turbine classes. For turbine class A, the deep models have a small lead in performance, but the results are not markedly different across the investigated models, with $\Delta t \approx 0$ days for AUC $= 0.9$. For turbine class B, the picture is different, where the performance gap between deep and shallow models is much more prominent. Here, the deep models show near-perfect detection accuracy at $\Delta t \approx 10$ days.

Finally, the superior detection performance of the deep architectures is highlighted for the helical stage faults, where the deep models show near-perfect detection from $\Delta t = 0$ days and forward in time, while the shallow models does not provide AUC-values higher than 0.95 even at the latest stages of fault development.

An interesting observation is that the shallow models provide higher AUC-values for large negative values of Δt. We contribute these effects to a form of underfitting of the shallow models with respect to the actual fault detection task where the models adapt to overall qualities of the turbine signals that carries some statistical weight in terms predicting a future fault on a turbine. This weight could for example be inferred from a learned fault risk clustering of the turbines or a learned representation of the turbine wear and tear which again is correlated to some degree with the fault onset for the specific

fault categories. Further testing using no-fault data strictly from the same turbines on which the fault data were collected, lowered the early phase AUC-values significantly in most cases for the shallow models, while the deep models were unaffected. These results largely support the clustering hypothesis, but some mixture of both effects cannot be ruled out. A deeper analysis is required to quantify and understand these results further, but this analysis is left as a future extension to the presented work.

5.2.2. Receiver Operating Characteristics

The AUC-metrics stand as the overall indicator of classifier performance, but further insight can be gained from analyzing the Receiver Operating Characteristics on which the AUC values have been calculated. From before, a significant gap was observed between the shallow model families and the deep model families. Visualizations of ROCs for each of these families can be observed in figure 4, restricted to the logistic model and the Deep CNN multi-task model. Each curve in these figures represent a given point in time relative to human detection (Δt), and for each curve an AUC-value can be calculated. The curves are based on the combined set of random test splits, providing an average representation of the ROCs.

In the very early phase, $\Delta t = -120$ days, the logistic model (upper row) provides better-than-random performance across all turbine classes and components. For the deep model, the performance is much closer to random performance, signifying that no signal can be found. For increasing values of Δt, the difference between and shallow model is clearly observed. Approaching $\Delta t = 0$ days most curves for the deep models approach the upper left corner, indicating almost perfect classification performance. At the same value of Δt the shallow models provide lower AUC-values, meaning much higher false positive rates for a fixed true positive rate, or much lower true positive rates for a fixed false positive rate when compared to a deep model.

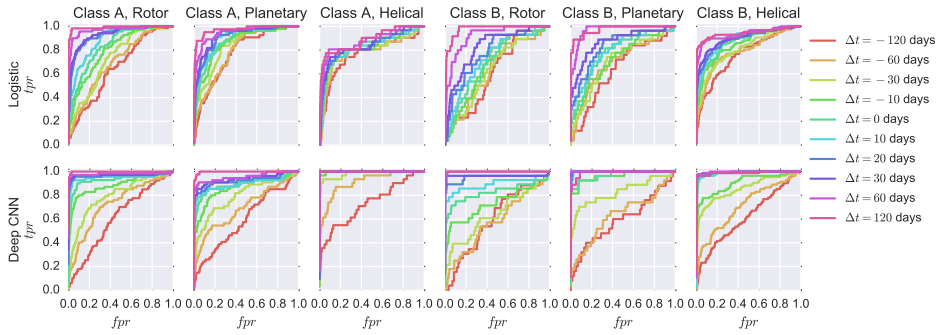


Figure 4. A comparison of Receiver Operating Characteristics for the logistic (top row) and deep multi-task model (bottom row). Each curve is calculated for a discrete point in time relative to the ERC, under which the area is mapped to an AUC-value in figure 3. The curves shown here are averages, calculated across all random test splits data points.

5.2.3. Time tracks

As a final analysis we will investigate the detailed model output time tracks for selected test turbine faults, one for each turbine and fault class. The outputs have been visualized in figure 5. These plots clearly illustrate the superior performance of the deep models reflected in ROC and AUC metrics. Two features are noticeable: 1) In the no-fault regions, the deep architectures consistently output low fault probabilities except for the upper left turbine where some fluctuations can be observed following the component exchange. The outputs of the shallow models are more noisy. As we approach the ERC, the outputs of the deep models display rapid increases with almost switching-like behavior. In some cases, the models picks out a clear fault signature months before the human analyst. The shallow models do not exhibit a similar behavior. The responses are delayed and less attenuated compared to the deep models. In some cases (turbine class B, rotor bearing) the shallow models struggle to pick up on the fault signature at all.

Figure 6 shows the output of one of the best performing models (Deep CNN multi-task) for a single turbine. This turbine has two different faults reported (rotor bearing and helical stage bearing). The model is observed to be able to clearly differentiate between the different fault classes without cross-talk.

DOI: 10.1002/we
Prepared using weauth.cls

A better understanding of how the test metrics are calculated can also be gained by iterating the visualized results in reverse order. Starting from the time tracks in figure 5, one can choose a fixed point in detection time Δt. The detection threshold, or p-value, can now be be moved up (for reduced sensitivity) or down (for increased sensitivity) and each threshold setting sets a point on the corresponding Δt ROC-curve in figure 4. The value of the detection threshold is a choice of the system designer and the ROC-curve is a visualization of this choice effects the diagnostic system in terms of true and false positive rates. Each ROC-curve is then integrated to provide an overall AUC value, as shown in figure 3.

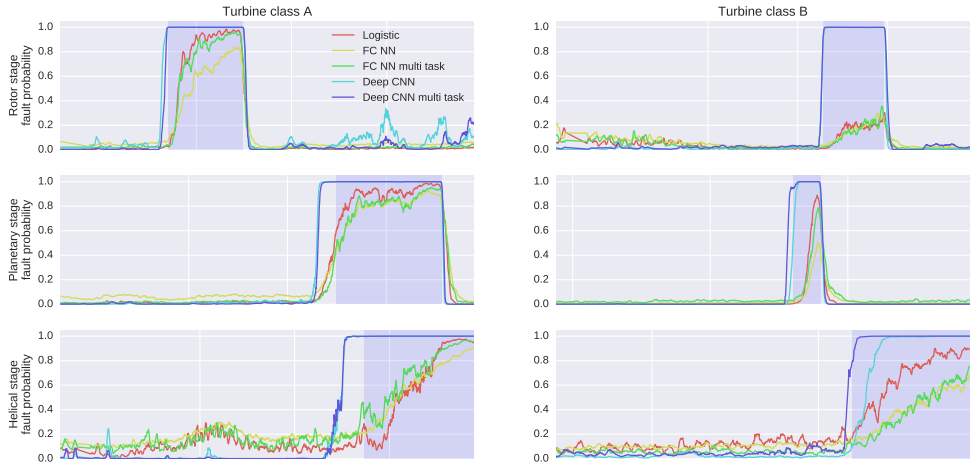


Figure 5. Time tracks displaying the actual predictions from the individual models on a number of faults. Examples are given for all turbine and fault classes. The fault interval specified by a human analyst has been highlighted by the blue regions in each plot. The vertical grid lines denotes years in time.

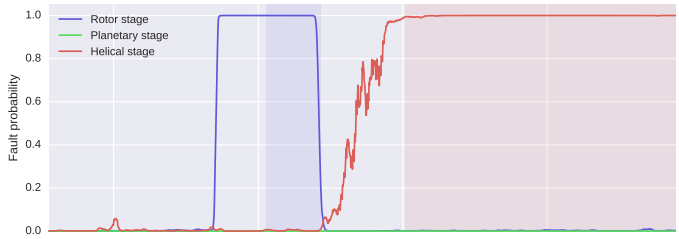


Figure 6. Multiple faults have been observed on some turbines as shown in this figure using the outputs from the deep multi-task model. The model is able to clearly separate the different fault regions.

6. DEEP NETWORK VISUALIZATIONS

From the various test metrics it was shown that the deep architectures are able to generalize extremely well in the given learning framework. Neural network models are often seen as black-box models, but the architecture employed here allows us to peak inside the network to gain a further understanding of *what* the deep models actually learn from the data. One way to visualize the high-level abstract features that the models extract from the input data is shown in figure 7. Here, we map the activations of the units in the top-most convolutional layer in the first pipeline in the deep architecture. These activations represent a low-dimensional, high-order spatial feature map from which the network base its final output via the fully connected layers. In the same figures we also provide the original normalized high-dimensional inputs and the

fault probability output from the network. Two different states have been visualized as a rotor stage fault progresses. The first figure shows the state of the activations before the fault occurs, while the second figure shows the activations well into the fault state.

A number of interesting findings can be extracted from the activation maps. First, many of the kernels show high activity in the right edge regions, representing the region with the most recent inputs. This is hardly surprising; to detect faults at an early stage, and to accurately reflect when the fault disappears again, the model must allocate much of its capacity on the most recent data points. Overall, the activation maps display some highly interesting features, resembling harmonic families in the frequency dimension, with varying attenuation across time. Again, it is hardly surprising that such frequency harmonics are important features for diagnosing faults in bearings (or any rotating machinery), but it is an impressive feat that the model successfully can learn to extract these features from a complex input signal in a purely data-driven fashion. A noteworthy observation is also that the changes in the input signals are almost invisible by visual inspection. The ability of the model to extract relevant diagnostic information from such weak signals shows why the deep architectures are able to adapt to the given diagnostic task so well.

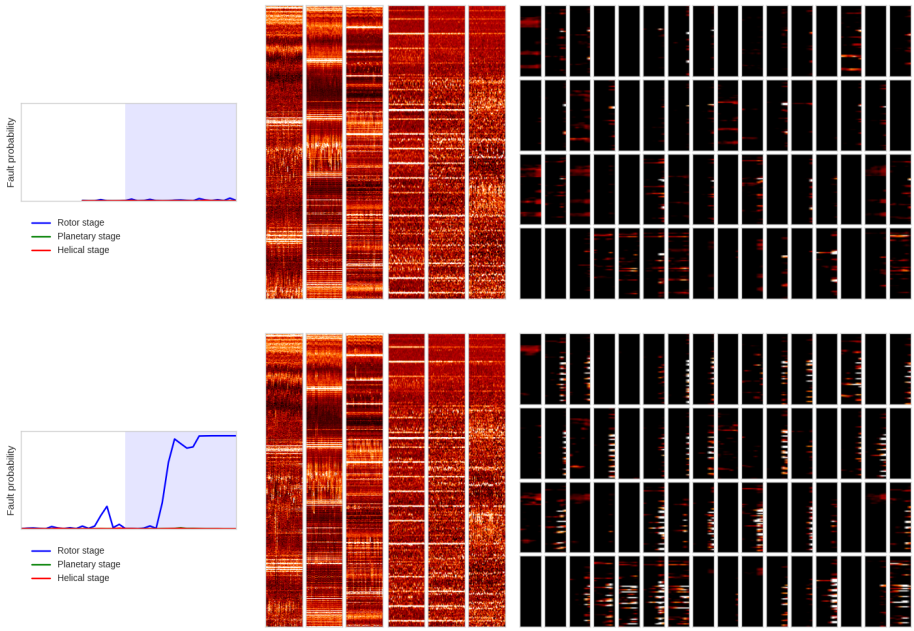


Figure 7. Deep network activations in the last convolutional layer in the first processing pipeline for a rotor stage bearing fault with the top most figure showing the state *before* fault initiation, and the bottom figures showing the state *after* fault initiation. The left plots show the fault probability output of the model as a function of time. The time length of the input data window is highlighted in blue. The middle plots show the normalized input vectors for each channel/pipeline as (time, frequency)-plots. Similarly the 64 activation maps are shown in the right-most grid.

7. CONCLUSION

In this work we investigated how machine intelligence could be applied to the problem domain of large scale wind turbine monitoring with a focus on complex diagnostic tasks based on vibration pattern analysis. The main conclusions from the work are as follows:

1. We were able to successfully train a range of discriminative models to mimic the outputs of human non-causal diagnostic experts in a fully supervised sense using only a high-level feature selection.

DOI: 10.1002/we
Prepared using weauth.cls

2. A novel deep learning architecture was demonstrated to provide extremely robust fault detection capabilities for three different bearing fault classes. Measured on both the static and time-dependent test metrics, the deep architectures consistently outperformed more shallow architectures with a wide margin.
3. The best performing models demonstrated accurate detection (AUC > 0.9) for all fault classes at the time of human non-causal detection. In many cases, the models could detect a fault signal weeks and even months before the human non-causal expert.
4. Given the current setup, the models did not benefit from multi-task learning. The performance did however not suffer from multi-task learning, meaning that a single model has the the capacity to detect multiple faults.

The broader, and more practical, perspectives of these results are interesting. The detection by a non-causal expert with prior fault knowledge should be viewed as an absolute best case performance if a system based on final human analysis is tested in an online scenario. The performance will in general be worse, and depends in a complex way of the architecture of the underlying monitoring platform. With the results demonstrated here, we have shown that the scalability issues with such platforms can be sidestepped using advanced machine intelligence. Hereby, a continuously up-to-date accurate evaluation of component health is feasible, a knowledge which in turn can be turned into direct operational value. An added benefit is that the models also provide a probabilistic output, which easily lends itself to further processing in decision support systems. It must finally be noted that the methods presented in this work are very general in nature, and it should be straightforward to adapt these to other fault detection scenarios.

8. FURTHER WORK

Ample opportunities exist for extending the work presented in this paper. For the fault detection scope, one could extend the analysis with a further breakdown of failure modes and different stages of failure progression. As for the choice of failure modes, other critical drive train faults such as gear wheel damage could also be included.

From a methodological perspective, the supervised training is strongly linked to the human expert evaluations and the inherent limitations in these evaluations. To fully utilize all available data, one could pursue a semi-supervised learning paradigm, where unsupervised learning is leveraged by the expert evaluations. One could then dispense with the exclusion period (or period of uncertainty) and instead let the model adapt to the data in these regions also. Although feature selection was performed here only at a relatively high level, the models could also adapt to the raw data itself being the high-sample rate vibration signals, with the aim of achieving a full end-to-end training of the diagnostic system.

Finally, the higher-than-random AUC-values observed for large negative values of Δt for the shallow models in also deserves further attention as already noted in section 5.2.1.

REFERENCES

1. Zaher A, McArthur S, Infield D, Patel Y. Online wind turbine fault detection through automated scada data analysis. *Wind Energy* 2009; **12**(6):574–593.
2. Schlechtingen M, Ferreira Santos I. Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. *Mechanical Systems and Signal Processing* 2011; **25**(5):1849–1875.
3. Kusiak A, Verma A. Analyzing bearing faults in wind turbines: A data-mining approach. *Renewable Energy* 2012; **48**:110–116.
4. Bach-Andersen M, Rømer-Odgaard B, Winther O. Flexible non-linear predictive models for large-scale wind turbine diagnostics. *Wind Energy* 2016; .
5. Zhao W, Siegel D, Lee J, Su L. An integrated framework of drivetrain degradation assessment and fault localization for offshore wind turbines. *IJPHM Special Issue on Wind Turbine PHM* 2013; :46.
6. Straczkiewicz M, Barszcz T. Application of artificial neural network for damage detection in planetary gearbox of wind turbine. *Shock and Vibration* 2015; **2016**.
7. Verma A, Zhang Z, Kusiak A. Modeling and prediction of gearbox faults with data-mining algorithms. *Journal of Solar Energy Engineering* 2013; **135**(3):031 007.
8. Randall RB. *Vibration-based Condition Monitoring: Industrial, aerospace and automotive applications*. Wiley, 2011.
9. Bach-Andersen M, Winther O, Rømer-Odgaard B. Scalable systems for early fault detection in wind turbines: A data driven approach. *Annual Conference of the European Wind Energy Association*, EWEA, 2015.

10. Jarrett K, Kavukcuoglu K, Ranzato M, LeCun Y. What is the best multi-stage architecture for object recognition? *Computer Vision, 2009 IEEE 12th International Conference on*, IEEE, 2009; 2146–2153.
11. Bishop CM. *Pattern recognition and machine learning*. Springer, 2006.
12. Polyak BT. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics* 1964; **4**(5):1–17.
13. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. *International conference on artificial intelligence and statistics*, 2010; 249–256.
14. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012; 1097–1105.
15. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 2014; 3104–3112.
16. Xiong W, Droppo J, Huang X, Seide F, Seltzer M, Stolcke A, Yu D, Zweig G. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256* 2016; .
17. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, *et al.*. Mastering the game of go with deep neural networks and tree search. *Nature* 2016; **529**(7587):484–489.
18. Bengio Y. Learning deep architectures for ai. *Foundations and trends in Machine Learning* 2009; **2**(1):1–127.
19. Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010; 807–814.
20. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 2014; **15**(1):1929–1958.
21. Kingma D, Ba J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* 2014; .
22. Caruana R. A dozen tricks with multitask learning. *Neural networks: tricks of the trade*. Springer, 1998; 165–191.
23. Fawcett T. Roc graphs: Notes and practical considerations for researchers. *Machine learning* 2004; **31**:1–38.

DOI: 10.1002/we
Prepared using weauth.cls

Bibliography

[1] Energinet.dk. Dansk vindstrøm slår igen rekord – 42 procent. [Online]. Available: http://energinet.dk/DA/El/Nyheder/Sider/Dansk-vindstroem-slaar-igen-rekord-42-procent.aspx

[2] "Global Wind Energy Outlook 2016," Global Wind Energy Council, Tech. Rep., 2016. [Online]. Available: http://www.gwec.net/publications/global-wind-energy-outlook/global-wind-energy-outlook-2016/

[3] R. Wiser, K. Jenni, J. Seel, E. Baker, M. Hand, E. Lantz, and A. Smith, "Expert elicitation survey on future wind energy costs," *Nature Energy*, vol. 1, p. 16135, 2016.

[4] Vindmølleindustrien. Kystnære havvindmøller fortsat billigst. [Online]. Available: http://www.windpower.org/da/aktuelt/aktuelt_i_vindmoelleindustrien/news_q4_2016/kystnaere_havvindmoeller_fortsat_billigst.html

[5] "Cost Reduction Monitoring Framework 2016, Summary Report," ORE Catapult, Tech. Rep., 2016. [Online]. Available: https://ore.catapult.org.uk/our-knowledge-areas/knowledge-standards/knowledge-standards-projects/cost-reduction-monitoring-framework/

[6] "ISO 13372:2012 Condition monitoring and diagnostics of machines – Vocabulary," International Organization for Standardization, Geneva, CH, Standard, Sep. 2012.

[7] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*,

"Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[8] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[9] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[11] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[14] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[15] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2146–2153.

[16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] M. Sharif and R. Grosvenor, "Process plant condition monitoring and fault diagnosis," *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 212, no. 1, pp. 13–30, 1998.

[19] A. Zaher, S. McArthur, D. Infield, and Y. Patel, "Online wind turbine fault detection through automated SCADA data analysis," *Wind Energy*, vol. 12, no. 6, pp. 574–593, 2009.

[20] M. Schlechtingen and I. Ferreira Santos, "Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection," *Mechanical Systems and Signal Processing*, vol. 25, no. 5, pp. 1849–1875, 2011.

[21] S. Butler, F. O'Connor, D. Farren, and J. V. Ringwood, "A feasibility study into prognostics for the main bearing of a wind turbine," in *Proceedings of IEEE International Conference on Control Applications 2012*. IEEE, October 2012, pp. 1092 – 1097.

[22] M. Wilkinson, B. Darnell, T. van Delft, and K. Harman, "Comparison of methods for wind turbine condition monitoring with scada data," *Renewable Power Generation, IET*, vol. 8, no. 4, pp. 390–397, 2014.

[23] G. E. Morales-Espejel and A. Gabelli, "The progression of surface rolling contact fatigue damage of rolling bearings with artificial dents," *Tribology Transactions*, vol. 58, no. 3, pp. 418–431, 2015.

[24] W. Yang, P. J. Tavner, C. J. Crabtree, Y. Feng, and Y. Qiu, "Wind turbine condition monitoring: technical and commercial challenges," *Wind Energy*, vol. 17, no. 5, pp. 673–693, 2014.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches," in *Artificial Neural Networks ICANN 2001*. Springer, 2001, pp. 669–676.

[27] J. Herp, M. H. Ramezani, M. Bach-Andersen, N. L. Pedersen, and E. S. Nadimi, "Bayesian state prediction of wind turbine bearing failure," *Renewable Energy*, 2017.

[28] K. Fischer, F. Besnard, and L. Bertling, "Reliability-centered maintenance for wind turbines based on statistical analysis and practical experience," *IEEE Transactions on Energy Conversion*, vol. 27, no. 1, pp. 184–195, 2012.

[29] C. M. Harris and A. G. Piersol, *Harris' shock and vibration handbook*. McGraw-Hill New York, 2002, vol. 5.

[30] R. B. Randall, *Vibration-based Condition Monitoring: Industrial, aerospace and automotive applications*. Wiley, 2011.