Wheel Defect Detection With Machine Learning

Gabriel Krummenacher, Cheng Soon Ong, Stefan Koller, Seijin Kobayashi
Joachim M. Buhmann, *Senior Member, IEEE*

***Abstract*—Wheel defects on railway wagons have been identified as an important source of damage to the railway infrastructure and rolling stock. They also cause noise and vibration emissions that are costly to mitigate. We propose two machine learning methods to automatically detect these wheel defects, based on the wheel vertical force measured by a permanently installed sensor system on the railway network. Our methods automatically learn different types of wheel defects and predict during normal operation if a wheel has a defect or not. The first method is based on novel features for classifying time series data and it is used for classification with a Support Vector Machine. To evaluate the performance of our method we construct multiple data sets for the following defect types: flat spot, shelling and non-roundness. We outperform classical defect detection methods for flat spots and demonstrate prediction for the other two defect types for the first time.**

Motivated by the recent success of artificial neural networks for image classification we train custom artificial neural networks with convolutional layers on two-dimensional representations of the measurement time series. The neural network approach improves the performance on wheels with flat spots and non-roundness by explicitly modelling the multi sensor structure of the measurement system through multiple instance learning and shift invariant networks.

***Index Terms*—Machine learning, Statistical learning, Support vector machines, Pattern analysis, Railway safety, Railway accidents, Wavelet transforms, Supervised learning, Artificial neural networks**

I. Introduction

EARLY detection of serious wheel defects on freight trains are an essential part in preventing damage to the railway infrastructure and in providing the train operators with timely information on necessary repairs, that can prevent further deterioration of the wheels.

Wheel defects of railway vehicles directly cause an increase in attrition of and damage to the railway infrastructure, e.g., the track systems or the civil engineering works, thereby adding additional costs to maintenance and repair and leading to a reduced lifetime and availability of rolling stock. The life span of the railway infrastructure is significantly shortened by the negative effects of wheel defects. The life span of railway bridges for instance is calculated with an assumed maximal dynamical load of 21 tons. Due to wheel defects the actually occurring dynamical load can be up to 50 tons, or 270% higher than the theoretically assumed maximum, thus shortening the life span. Wheel defects also accelerate crack-growth on the rail tracks and lead to premature failure of the rail system.

Another important effect caused by wheel defects are ground vibration and noise emissions. In the European Union (EU) Project "Railway Induced Vibration Abatement Solutions" (RIVAS)[1] 27 partners from nine countries investigated the source and mitigation measures for noise and vibration emissions. They found that reducing wheel defects by wheel maintenance significantly reduces vibration and noise emissions directly [1]. Therefore, it is recommended to use timely and targeted maintenance of train wheels as an economic means to reduce emissions [2]. This measure is all the more important as the density and usage of modern railway networks is steadily increasing and failures quickly disrupt operation of the whole network or parts of it. Since 2008, all states in the EU are advised to employ noise emission ceilings. Switzerland started a noise abatement program based on emission ceilings that requires the infrastructure manager to curb emissions above the ceiling. This abatement programme leads to total costs of 1.5 billion CHF [3].

In this paper we propose a method of detecting defective wheels. This classification method promises to increase the reliability of the railway infrastructure, to reduce the cost of freight train operation and to save additional investments on noise protection measures. To reach this goal without the costly construction of further measurement sites or newly built sensors, we propose the use of statistical methods that allow us to automatically inspect the existing data and extract the information about defective wheels that is already present.

Our proposed methods do neither require a model of the measurement system, nor of train dynamics or wheel defects. The methods enable us to predict defects on wheels where there is no prior understanding of how these defects manifest themselves in the measurements. The methods detect and classify different types of defects based on measurements during normal operation where the trains pass the measurement sites in full operational speed. The features that we have developed for the use in supervised learning are general and can in principle be used for any time series data and are not restricted to specific defect types. In a second step we automatically learn features directly from the raw measurement signal.

A. Contribution

Our main contribution are two methods for automatic railway wheel defect detection and classification through vertical force measurements of trains running in full operational speed. For the first method we design novel wavelet features for time series data from multiple sensors and we learn a classifier

Gabriel Krummenacher, Seijin Kobayashi and Joachim Buhmann are with the Department of Computer Science, ETH Zürich, Switzerland (e-mail: gabriel.krummenacher@inf.ethz.ch; seijink@student.ethz.ch; jbuhmann@ethz.ch).

Stefan Koller is with the Department of Installations and Technology, SBB AG, Switzerland (e-mail: stefan.koller@sbb.ch).

Cheng Soon Ong is with the Machine Learning Research Group, Data61, CSIRO, Australia (e-mail: chengsoon.ong@anu.edu.au).

[1] http://www.rivas-project.eu

using a support vector machine. For the second method we design and train convolutional neural networks for different wheel defect types by deep learning.

We evaluate our novel and other classical methods for wheel defect detection on two labeled data sets with different types of wheel defects, that we have constructed from calibration runs and from maintenance reports.

B. Related Work

While there has been research on machine learning methods for railway track inspection [4–6] or condition based maintenance [7], to our knowledge machine learning methods for railway wheel defect detection have not been developed so far. There has been some research on sensor systems for wheel defect detection on freight trains. In Nenov, Dimitrov, Vasilev, *et al.* [8], the authors analyse the signal from acceleration sensors and demonstrate that they can visually see a difference between the measurements of wheels with flat spots and good wheels but they do not propose a method for detection. Another related work [9] advocates the use of Fibre Bragg Gating sensors for defect detection of rails to monitor track conditions. The authors investigate the wavelet decomposition of pressure signals but they do not propose a method or threshold for automatic defect detection. Jianhai, Zhengding, and Boshi [10] use continuous wavelet analysis of acceleration sensor data to visually inspect the measurements and conclude that there is a difference in the coefficients for wheel with flat spots and defect-free wheels.

Different kinds of track scales are in use in the field. They can in principle be used to detect flat spots. But to our knowledge they do not use machine learning to train a defect classifier. A general advantage of our proposed system is that the measurement system is relatively inexpensive, but we can show that it can still be used to detect wheel defects, thanks to our proposed machine learning methods.

II. Measurement System and Defect Types

A. Wheel Load Checkpoint

The infrastructure division of the Swiss railway operator SBB operates and maintains the one of the most heavily used railway network of the world. In 2010, 95.4 km of trains travelled one kilometer of track on average; this value documents the highest utilisation of network capacity in the world [11]. Automatically monitoring trains and network are thus important to minimise the risk of incidents that quickly affect the scheduling of trains on the network. SBB infrastructure operates an integrated wayside train monitoring system that controls safety relevant aspects of the railway traffic and infrastructure.

As part of this system, the wheel load checkpoints (WLC) measure vertical force through strain gauges installed on the rails. These devices are used for observing maximal axle load, maximal train load, load displacement and grave wheel defects. Our study investigates the use of machine learning methods to defect and classify wheel defects based on the data obtained through these wheel load checkpoints.

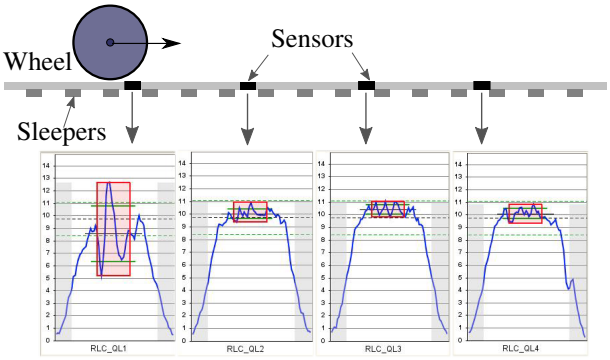


Figure 1: Multiple vertical wheel force measurements of a train wheel by the four sensors of one measurement bar. The wheel is affected by a discrete defect that manifests itself in the measurement of the first sensor. The remaining sensors do not directly observe the defect.

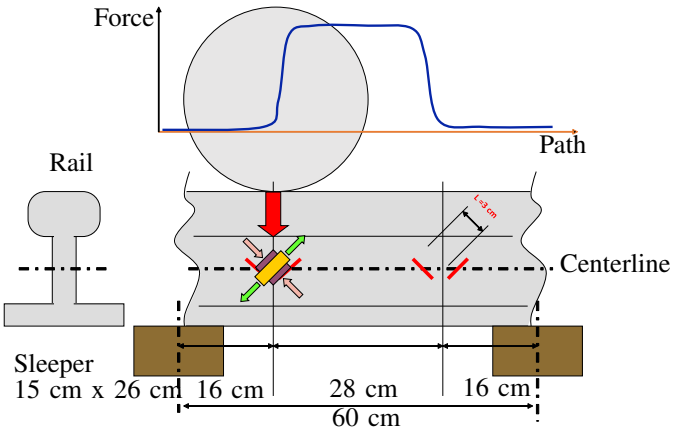
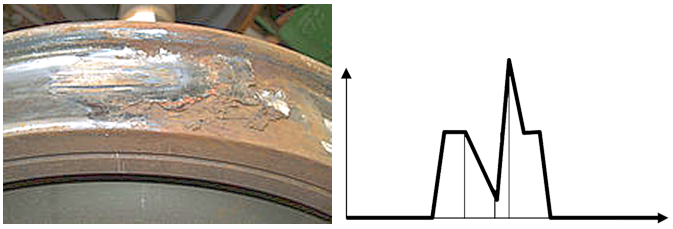


Figure 2: Diagram of one sensor on a measurement bar of the WLC. The strain gauges are attached to the side of the wheel between two sleepers and cover 28cm of vertical wheel force of the wheel rolling on the track.

Each WLC consists of four 1m long measurement bars with four strain gauges (referred to as sensors in the following) per measurement bar. Since on each side two measurement bars with 4 sensors are installed, each wheel that runs over the WLC is measured eight times at different parts of the wheel. Fig. 1 shows schematically the measurement of one wheel by one measurement bar. In this example a defect is directly observed by the measurement of the first sensor.

See Fig. 2 for a diagram of one sensor. The strain gauges are installed perpendicular on the centerline of the railroad track and they are combined into one vertical wheel force measurement. One sensor covers approximately 30cm of the wheel circumference.

The wheel load checkpoints are installed on multiple strategic sites on the railway network: ten on the border to Switzerland at the entrance to the railway network maintained by SBB and a dozen within the network.



(a) (b)

Figure 3: Picture of a serious flat spot on a train wheel of SBB (a) and the resulting idealized wheel load measurement (b). (Picture taken from Wikipedia/Bobo11)

B. Railway Wheel Defects

A relatively well understood wheel defect type is the flat spot or wheel flat. This defect occurs when the wheel stops rotating (for instance during an emergency brake) and is dragged along the track. Fig. 3 shows an image of a flat spot on a railway wheel of SBB and the corresponding idealized measurement obtained by the WLC if the flat spot directly hits a sensor of the measurement system. Grave wheel flats can be detected by looking at simple statistics (c.f. Section VI-B) of the measurement if the defect hits the sensor perfectly. To be able to detect flat spots that are less grave or that do not hit a sensor directly, more advanced machine learning methods are required. We demonstrate such cases on our first data set in Section VII-B.

Apart from flat spot, other common wheel defects on railway vehicles are *non-roundness* and *shelling* [12, 13]. Wheels with non-roundness have a high influence on the vibration and noise emitted by a passing train and, therefore, they are an important type of defect to detect [1, 13]. Non-roundness, in contrast to shelling and flat spot, is a non-discrete type of defect. This characterization means that the defect affects a large part of the wheel and changes its shape in a non-local way. We create an additional data set that contains the defect types flat spot, non-roundness and shelling (Section VI-C) and then, we compare the performance of our two machine learning methods in predicting these three defect types.

III. Time Series Representation for Defect Detection

An important step in any machine learning method is finding a representation of the original measurements that supports discrimination between different classes. For instance: the mean of the measurement signal of a wheel with or without a flat spot coincide if the weight of the axle is the same and the defect perfectly hits a sensor. The standard deviation on the other hand differs significantly, since the force exerted on the track is much higher for a wheel with the flat spot than for non-defective wheels. For other types of defects like shelling this observation does not hold, as the variance of the measured force does not significantly differ from a non-defective wheel, but there is a clear difference in higher frequency bands of the measurement, c.f. Fig. 4. These observations suggest to decompose the signal by a multiscale wavelet analysis in order to extract indicative frequency features for time series data.

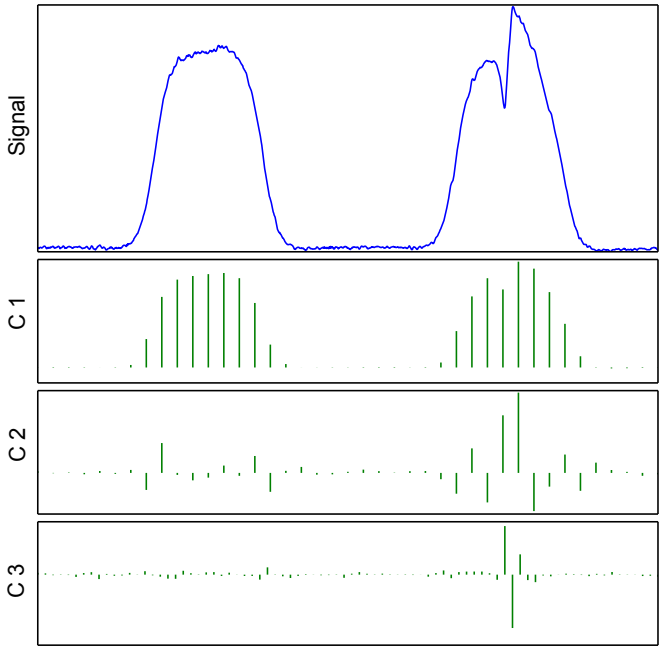


Figure 4: Signals and wavelet coefficients at different levels (C1 to C3) of a defective (right) and non-defective (left) wheel. The power in the high frequency coefficients C2-C3 reveal the defect.

A. Wavelet Transform

The Discrete Wavelet Transform (DWT) decomposes a signal over an orthonormal basis of dilated and transformed wavelets [14]:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t - k2^j}{2^j}\right), \qquad (1)$$

where ψ denotes mother wavelet, j and k the scale and shift parameters.

The orthogonal wavelets given by definition (1) at different scales 2^j resolve the original signal at different resolutions. The DWT can thus be employed to construct a multiresolution signal approximation [14]. An equivalent way of calculating the DWT is by passing the original signal through a series of appropriate high-pass and low-pass filters and sub-sampling operations, where at each level the output of the high-pass filter is stored as the detail coefficients for that level and the output of the low-pass filter is decomposed further at the next level until level $T = \log(n)$ is reached, where n is the length of the original signal. If the high-pass and low-pass filters in this filter bank are derived from the child wavelets in Equation 1, the detail coefficients (C1, ..., CT) correspond exactly to the wavelet coefficients.

The wavelet transform has been extensively used in fields ranging from biomedical signal processing [15], geosciences [16] to image compression [17]. Since weight measurement signals and the defect effects on the signal are both localized in time and frequency the wavelet transform explicitly encodes this local perturbation and, therefore, has an advantage over the fourier transform in our application. The weight measurement signals also show a self-similar behavior which suggests the

wavelet transformation as an adapted set of basis functions with approximately the same amount of power per frequency band.

B. Wavelet Features for Defect Detection

To extract features from the measurement signals of the wheels, we first compute the wavelet decomposition of each signal. Each time series is now represented by the distributions of the wavelet coefficients at the different levels of the multiscale decomposition. To represent the distribution of the coefficients, n moments of the empirical distribution of the coefficients are computed. This representation captures higher order behaviour while still maintaining invariance to shift or scale of the defects as measured by the sensors. The procedure is summarized by Algorithm 1 and the function to compute the central moments is given below by Equation 2, where the average is used as the first moment.

$$\text{moment}_1(\mathbf{x}) = \bar{\mathbf{x}}, \quad \text{moment}_{m>1}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})^m \tag{2}$$

Algorithm 1 Wavelet feature computation

Input: W^t: coefficients at the t-th level of a T-level DWT.

```
k = 1
for t = 1 ... T + 1 do
    for m = 1 ... M do
        F^k = moment_m(W^t)
        k += 1
    end for
end for
```

Output: F^k: $k = 1, \ldots, M \cdot (T+1)$ wavelet features.

As explained in Section II-A, we observe eight signals for each wheel that we want to classify. To compute features for one wheel, we first concatenate the measurements of all the sensors and then compute the wavelet features on this single time series. When we are processing localized defects, like a flat spot, that are observable as a change in vertical force on one sensor, the specific information, which sensor has observed a defect, does not play a role due to the scale invariance of our feature construction method. For each sensor, the measurement signal can be divided into the regions of no load, raising slope, load measurement window and falling slope. Even though the load measurement window is relatively small we can still observe wheel defects that manifest themselves in one of the slopes or during the no load phase before and after the load measurement. To capture this information, a window of size three times the measurement window is used for feature construction. In all our experiments we use the Daubechies-5 wavelet family as basis functions [18].

C. Load Normalized Features

In addition to the wavelet features computed on the full concatenated signals of all the sensors we also compute wavelet features for each sensor separately. Whereas the feature construction based on the full signal pursued the strategy to capture as much information as possible, the goal here is to construct features that are normalized with respect to the load measurement.

To this end, we first subtract an idealized measurement curve from the signal of each sensor and then compute wavelet features with Algorithm 1 on the difference. Additionally we add the mean squared error of the signal to the measurement template as a feature per sensor.

D. Measurement Site

Each wheel load checkpoint exhibits different physical characteristics due to small differences in the ground below the tracks and the curvature of the track before the checkpoint. These characteristics change the wheel load measurement. Small unevenness in the tracks also manifest themselves as noise or small bumps in the signal. Therefore, we add the site of the wheel load checkpoint as additional feature to enable different predictions based on the origin of the measurement site. We encode this information as a unary code or a one-hot vector, where every dimension represents a site and is 1 only for measurements from that site. When in the future a new measurement site would be built on the railway network, training data for the new site would need to be collected.

E. Load

A train with different load, but the same waggons results in different wheel measurements for the same defect types, since the weight of the train plays a significant role how the defect exerts its pressure on the sensors. Another important reason to add information about the load to the feature set arises from the following observation: certain defect classes like non-roundness mostly change the average of a sensor reading, but only marginally affect higher order information. An oval wheel for instance will result in higher load measured by some of the sensors and lower load by others, but will not be detected as a defect wheel by individual load normalized measurements. The mean load of all the sensors, standard deviation over the mean load per sensor and the mean load for each sensor are added to the feature set.

IV. Automatic Representation Learning

An alternative to predefined feature representations are provided by deep neural networks that learn the features from data in a task specific way to maximize correct classification. In this section we introduce a learning method to automatically infer a representation of the measurements for the classification of wheel defects based on deep artificial neural network models (DNN). These models have gained considerable popularity in recent years, mostly due to their success in image classification and segmentation tasks [19, 20], in speech recognition [21] and quite recently in reinforcement learning for playing Go [22].

DNN for wheel defect detection alleviates the burden of the modeller to manually construct features and allows to learn representations from time series directly. Another benefit is

the flexibility that comes with designing decision functions as stacked activation layers. This flexibility allows us to design a network specifically for certain defect types.

A. 2-Dimensional Time Series Representation

Motivated by the success of convolutional neural networks on image classification tasks [23] we propose the use of 2D representations of the measurement signals for wheel defect detection. Recently Gramian Angular Fields (GAF) have been proposed [24] as a 2-dimensional encoding of time series data. This representation has been shown to capture cross-temporal dependencies and to enhance classification performance when used as input to a convolution network. A GAF is constructed by first transforming the time series to polar coordinates and then computing trigonometric sums between all points (See Wang and Oates [24] for details of the construction).

As a second 2D representation we also considered transforming the time series into the image of its 2D graph. This procedure is motivated by the fact that a human expert would also look at such a two-dimensional representation to classify wheel defects. The addition of the value of the signal as the second dimension allows the network to learn different filters for different values of the signal at the same point in time (the first dimension). The procedure is summarized in Algorithm 2.

Algorithm 2 Compute 2D time series representation

Input: $X = (X_t)_{1 \leq t \leq N}$: time series.
Input: $r > 0$: resolution.
Input: $[V_{min}, V_{max}]$: window.
1: $h = \lceil \frac{V_{max} - V_{min}}{r} \rceil$
2: $M = \mathbf{0}_{h \times N}$
3: $X = X - V_{min}$
4: **for** $t = 1 \dots N$ **do**
5: $\quad M_{\lceil \frac{X_t}{r} \rceil, t} = 1$
6: **end for**
7: **for** $m = 1 \dots N - 1$ **do**
8: $\quad$ Set all entries touching the segment $[M_{\lceil \frac{X_m}{r} \rceil, m}, M_{\lceil \frac{X_{m+1}}{r} \rceil, m+1}]$ to 1, drawing a line segment between the two points.
9: **end for**
Output: M: 2D graph of time series X.

B. DNN Network Architecture

We use a Convolutional Neural Network (CNN) based architecture to automatically extract the discriminating features. Here, we considered the 8 signals of the WLC as different channels. Our networks are composed of two modules: the mono channel feature extracting layers and cross channel feature extracting layers respectively from bottom (input layer) to top (classification layer). The mono channel feature extracting layers take each channel independently and compute high level features in parallel that can then be processed by the cross channel feature extracting layers. Furthermore, the weight of the mono channel feature extracting layers are shared across all channels, allowing it to learn from all channels at once. This approach is both computationally efficient, and also well suited for the data set. Since each channel represents a load measurement of the wheel from one sensor of the WLC the network learns features from the signals and also a relationship between the signals.

1) Mono channel feature extracting layer: This module is a traditional CNN, composed of a sequence of convolutional layers, eventually followed by a fully connected layer:

a) Convolutional layer: A convolutional layer is a combination of a number of filtering layers, each followed by a non-linearity and a pooling layer. The settings chosen for each of these layers are specified below. The filtering layer outputs convolutional products of the input by learnable filters with a fixed receptive field. Every filter layer is followed by an activation function. We use a *Parametric Rectified Linear Unit* (*PReLU*), as it better back-propagates the gradient compared to the tangent hyperbolic or sigmoid functions, which can easily saturate. The PReLU non-linearity also prevents neurones from "dying out" as can be the case for the popular *ReLU* units, by introducing a learnable non-zero slope to the negative side of the input[25].

$$PReLU(x) = \max(0, x) + a \cdot \min(0, x)\,, \tag{3}$$

where a is an adaptable parameter.

The pooling layers reduce the resolution of the input time series and the learned features at each layer of the deep neural network. This max-pooling allows the classification to be robust to small variations of learned features at each layer. In all of our convolutional layers, we used a pooling layer with filters of size 2×2 applied with downsampling ratio of two, taking the maximum value among the four pixels in its receptive field.

b) Fully connected layer: Neurons in a fully connected layer have full connections to all units in the previous layer. The layer outputs biased linear combination of its input, followed by a non-linearity. As a non-linearity we used the hyperbolic tangent function ($\tanh$).

2) Cross channel feature extracting layer:

a) Cyclic Permutation Network: The cyclic permutation network (Fig. 5) is designed to learn cross-sensor features invariant to a cyclic permutation of the eight recordings. Depending on its phase, a given wheel can generate a set of possible recordings, which is approximately stable by cyclic permutation of the eight recordings. This network architecture serves the purpose to encode this characteristic of cyclic invariance. The network works in the following way:

1) The Cyclic Permutation Network sits on top of the Mono channel feature extracting layers. It takes as input the set of high level features of each channel computed independently by the weight shared CNN (represented as a dashed red box right of the signal in Fig. 5).
2) The network then distributes the set of 8 feature vectors v_i (the colored vertical bars in Fig. 5) across 8 permutation channels (the stack of colored horizontal bars in Fig. 5), one for each possible cyclic permutation of the feature vectors. Each permutation channel concatenates the feature vectors following the order of its specific cyclic

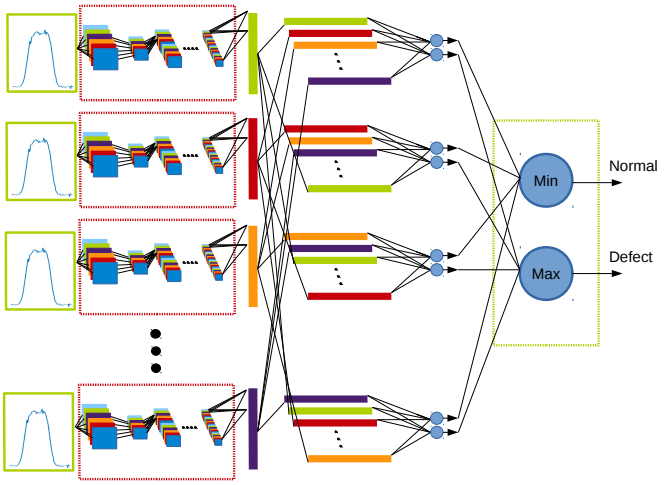


Figure 5: Structure of the cyclic permutation network that automatically learns cyclic shift invariant features. The red boxes on the left represent the weight shared CNN, the coloured bars designate features learned by the CNN, the stack of colored bars are permutations of the feature vectors, the blue dots the class log-likelihoods per permutation and the green box the final class probabilities.

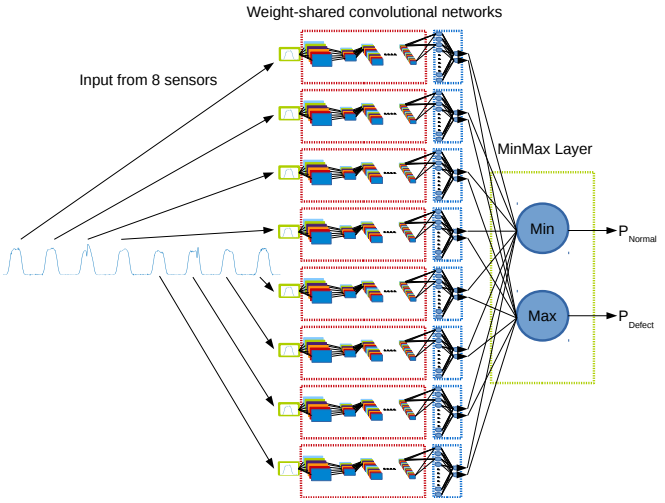


Figure 6: Structure of the MIL defect detection network for flat spots. The network consists of one CNN per measurement with weights shared across the networks. The defect likelihood of the whole wheel is given by the maximum defect likelihood across sensors.

permutation. Note the distinction between "channels" and "permutation channels", as the former refers to a specific sensor recording, while the latter refers to a permutation of the input channels, and contains the high level features of all initial channels.

3) Afterwards, the concatenated vector within each permutation channel is fed into a sequence of fully connected layers that extracts cross channel features and outputs the classification probability for the respective cyclic permutation (The blue circles in Fig. 5).
4) Finally, the multiple log-likelihoods (one for each permutation channel) are combined by returning the maximal log-likelihood for the defect class and the minimal log-likelihood for the non-defective class (The green dashed box in Fig. 5).

Formally, given a set of 8 feature vectors, $(v_i)_{1\leq i\leq 8}$, for a wheel the cyclic permutation network computes the probability of defect P^D as:

$$P^D = \max_{p\in\mathcal{P}} f(v_{p(1)} \| \dots \| v_{p(8)}), \tag{4}$$

where $\mathcal{P}$ is the set of all possible cyclic permutations of the numbers $[1, 8]$, $f(\cdot)$ is the function performed by the fully connected layers and $\|$ is the concatenation operator.

b) Defect Detection Network for Flat Spots: For tasks like flat spot detection, it is not necessary to learn complex cross channel features. Since a flat spot is a discrete defect and usually manifests itself only in one sensor reading, the Multiple Instance Learning (MIL) setting [26] is appropriate. In this setting a wheel is considered defective when at least one of the sensor readings is predicted defective. The Defect Detection Network encodes this idea by reducing the cross channel feature to the indicator function of whether a defect has been detected in one of the channels:

1) It takes as input the set of classification probabilities of each channel computed independently by the Mono channel feature extracting layer.
2) It combines the multiple log-likelihoods by returning the maximal log-likelihood for the defect class and the minimal log-likelihood for the non-defective class.

Given a set of s log-likelihoods for binary classification from s sensors $x = (P_i^D, P_i^N)_{1\leq i\leq s}$, where P_i^D is the likelihood for defect and P_i^N for non-defect from sensor i. Since $P_i^N = 1 - P_i^D$ and $0 \leq P_i^D \leq 1$:

$$MIL(x) = (\min_i(1 - P_i^D), \max_i(P_i^D)). \tag{5}$$

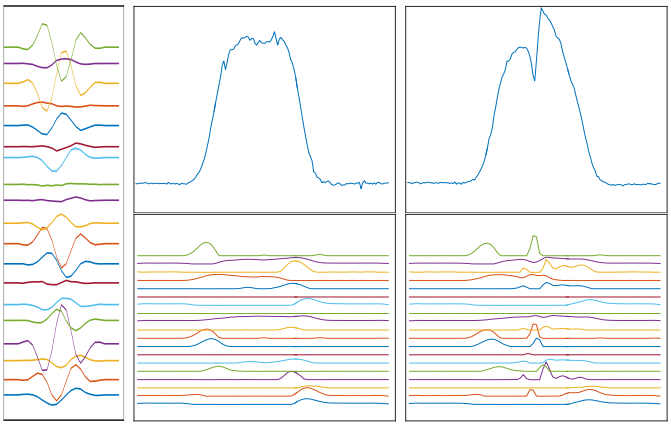
In Fig. 6 we depict the structure of the DNN that we use to train a model for the detection of flat spots.

We call the last layer MIL-Layer. It makes sure that if one measurement of the wheel captures the defect, the probability of the wheel having a defect is high. If defects are not "seen" by any sensor this probability will be low. Moreover, when training with defective wheels, only the error of the channel with the highest defect probability is backpropagated, thus preventing the Mono channel feature extracting layer to try to learn features for defective signals on signals that show no defect.

The MIL setting was already used for the SVM based MIL flat spot classifier in Krummenacher, Ong, and Buhmann [27].

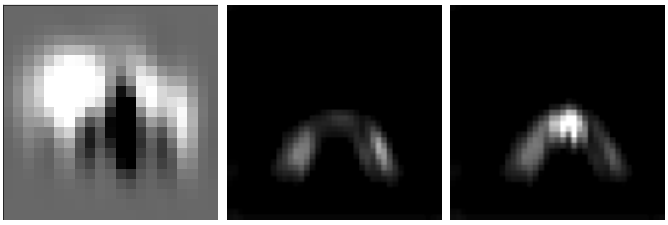
C. Top Layer Features learned by the DNN

In this section we look at the features learned by the DNN and compare the filters learned by the network on the 1-dimensional or 2-dimensional time series representation. The results in this section were obtained by training on data set 2 (SectionVI-C) and defect type flat spot.



(a) (b)

Figure 7: Top layer filters (a) and features (b) learned by the 1-dimensional defect detection network for flat spots for a measurement of a defective (right) and non-defective (left) wheel.



(a) (b) (c)

Figure 8: Example of a top layer filter (a) and corresponding features of the signal of a non-defective (b) and defective (flat spot) (c) wheel learned on 2D representations.

Examples of top-layer filters learned by the DNN directly on the 1-dimensional time series, as well as the features extracted by them are shown in Fig. 7. We can observe that the network has been trained to detect a short quick oscillation in the time series. The extracted features on the defective input clearly shows the successful training of the model in detecting defect regions.

Fig. 8 shows the top layer filters learned by the DNN on the 2-dimensional representation of the time series, and their respective extracted features on a defective and non-defective wheel. In general, the filters learned on the 2D representation encode high gradients in intensity, qualitatively presenting clear white/black delimitation. This suggests that the model focuses on 2D shape recognition rather than 1D pattern recognition as seen in filters learned on the time series directly.

V. Classification of Wheel Defects

Detection and classification of wheel defects amounts to infer from a vertical force measurement $\mathbf{x}$ of a wheel if a wheel is defective or not. Mathematically, a function $f(\cdot)$ either encode the binary information, that a defect is present or absent, or its defect class when we can differentiate the defect category. To achieve this goal we use sets of measurements of wheels to train decision functions for certain defect types and for non-defective wheels. We then use this *training set* of measurements and labels (the type of defect) to automatically find a function that is expected to predict the defects of wheels not seen during training accurately.

A. Support Vector Machine

One of the most popular models to find such a function are Support Vector Machines (SVM) [28]. A SVM finds a linear function parameterized by the vector $\mathbf{w}$ that maximally separates the two classes during training. It achieves this separation by maximizing the margin between the points of the two classes in feature space, or equivalently by minimizing the regularized empirical risk

$$\hat{R}(\mathbf{w}) = \left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))\right] + \lambda \left\|\mathbf{w}\right\|^2, \quad (6)$$

where we minimize the empirical risk over the parameters $(\mathbf{w}, b)$, that encode the hyperplane separating the two classes. $y_i \in (-1, +1)$ is the label (class membership) of the i^{th} example in the training set, $\mathbf{x}_i$ denotes the feature vector of the i^{th} measurements and $\max(0, 1 - z)$ is the hinge loss. Measurements of a new wheel $\mathbf{x}$ can now be classified with the following decision rule:

$$y := \operatorname{sgn}(\mathbf{w}^\top \mathbf{x} + b). \quad (7)$$

This decision rule (7) expresses its data dependence only by a scalar product between weights $\mathbf{w}$ and the feature vector $\mathbf{x}$. Therefore, we can model non-linear decision functions by replacing the scalar product with a kernel. A convenient choice is a Gaussian radial basis kernel function of the form $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2)$ on the feature vectors $\mathbf{x}_i, \mathbf{x}_j$. We can now express the minimization problem above (Equation 6) in the dual and employ the kernel trick to learn parameters α_i and get the new classification rule

$$y = \operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b\right). \quad (8)$$

To determine the optimal parameters for regularization λ and scale γ we maximize accuracy on cross-validation folds.

B. Classification with DNN

If we replace the hinge loss function in Equation (6) in the previous section with the logistic loss function $\log\left(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x})\right)$ we get regularized logistic regression. This optimization problem has the advantage that optimization algorithms estimate probabilities of the class likelihoods in addition to the binary labels. Using the softmax function instead of the logistic loss this benefit can be generalized to an arbitrary number of classes. We will use these probability estimates through a *SoftMax-layer* in our DNN to combine the output of multiple classifiers for different measurements of the same wheel.

For a given input and C classes, its log-likelihood for belonging to class i equals

$$p(\mathbf{v}|i) = \log\left(\frac{\exp(v_i)}{\sum_{j=1}^{C}\exp(v_j)}\right), \quad (9)$$

where $(v_i)_{1\leq i\leq C}$ are the top-layer features of the network.

The soft-max function above is not only used for DNNs but also in many multiclass classification methods, for instance for logistic regression or in dynamical system estimation with multiple model adaptive estimation (MMAE) [29, 30]

Unlike the previous section, where the classification function $f(\cdot)$ was modeled as a linear function in a Hilbert space, that takes a fixed representation of the measurements, DNNs model this function as a hierarchical structure (layers) of linear combinations and activation functions (non-linearities) directly on the time series of the measurement (Section IV).

VI. Data Sets and Models

Two data sets from different sources are assembled to evaluate the performance of different methods for wheel defect detection and classification and to train various classifiers. For both data sets the signals that we use to predict a wheel defect are measured by the wheel load checkpoint (Section II-A). The annotations or labels that provide the information about the defectiveness and defect class of a wheel are collected from different sources. These data sets contain information about different types of defects as described in the following. We also describe what models and features we will use for the respective data sets in this section.

A. Models and Features

On the first data set we compare the Wavelet-SVM with benchmark flat spot prediction methods. We show that it greatly outperforms prior art based on thresholding the dynamical coefficient (Eq. 10 below) and also on multiple instance learning with dynamic time warping.

The second data set serves to demonstrate that the Wavelet-SVM can accurately classify all three defect types. We also compare the performance of the deep learning models on different time series representations by showing that the cyclic permutation network outperforms the simpler neural networks and also the Wavelet-SVM for non-roundness. For flat spots, the neural network with features learned on the 2D time series representation also outperforms the Wavelet-SVM.

We use different models and features for different defect classes, as this allows us to model network structure and feature construction adaptively to the effects the defects have on the measurements. Thus the problem differs from standard multi-class classification where one model predicts a vector of class probabilities over all classes. Instead we are looking at independent binary classification tasks per defect class, where the task is to distinguish between one defect type and non-defective. This enables clear comparison between the different models.

As there are no known methods to predict non-roundness or shelling we compare to baseline methods on a data set with flat spots (data set 1). To evaluate our Wavelet-SVM on non-roundness and shelling as well we use data set 2 to estimate classification performance on all three defect classes. We have proposed two different DNNs for defect detection in Sec. IV-B: the cyclic permutation network (cyclic DNN) and the MIL-DNN. We use the cyclic DNN to predict non-roundness as this is a non-discrete defect type with large-scale effects. We take the maximum probability of defectiveness over multiple inputs. As the region of the wheel that rolls over the first sensor is arbitrary we want to be able to be invariant to a specific way of shifting the sensors. Thanks to the symmetric way and the distances at which the sensors are installed we can look at cyclic shifts of the concatenated signal of all sensors to simulate different scenarios. The DNN trained to learn these cyclic shift invariant features is described in Section IV-B2a. The MIL-DNN is used to predict flat spot on data set 2 as the multiple instance learning setting lends itself nicely to this defect type as explained in Sec. IV-B2b.

B. Data Set 1: Calibration Run

To acquire a first training data set for flat spots, two wheels on different wagons were artificially damaged. The wagons were then added to a calibration train that was run over different measurement sites with different velocities and from both directions to calibrate the wheel load check points. This resulted in 1600 measurements, 50% of which are from a wheel with a flat spot.

We also consider another method to detect flat spots in this data set, that is not based on machine learning. It is a conservative threshold on the dynamic coefficient: a general measure of spread within one time series. For each sensor this coefficient is given by

$$d_{BW}(\mathbf{x}) = \frac{\max(\mathbf{x})}{\bar{\mathbf{x}}}, \tag{10}$$

where $\max$ and $\bar{\mathbf{x}}$ refer to the maximum and average value of a sequence of measurements $\mathbf{x}$, respectively.

C. Data Set 2: Reprofile Events

To generate data for training and testing a classifier that can predict additional types of wheel defects, we aggregated the time and date of reprofile events and linked them to railway wagons. We used two sources for these events: the protocols of repair workshops of freight trains and the regular maintenance measurements of passenger trains. These were annotated with a defect class by an expert before re-profiling the defective wheels. We then categorized measurements of the wheel load checkpoints of the same wagons around the date of re-profiling. Measurements up to a week before re-profiling were considered defective (according to the class label given by the expert), while measurements up to a week after re-profiling were considered defect free. Using this procedure we were able to obtain a large data set of annotated measurements from wheels of different defect classes over the span of multiple years. 1836 measurements are evaluated for flat spot detection, where 588 cases are classified as defective. For shelling, we received 6070 measurements, with 2678 being defective. For the non-roundness defect class, 688 cases out of 920 measurements are defective.

VII. Experimental Results

For performance evaluation of the methods we compute three metrics: accuracy, precision and recall. Whereas accuracy

Table I: Test set performance on data set 1.

Method	Accuracy (%)	Precision (%)	Recall (%)
Wavelet-SVM (ours)	92	94	93
eMIL	70	-	-
Dynamic coeff.	60	100	22

Table II: Test set performance of the Wavelet-SVM on data set 2.

Defect	Accuracy (%)	Precision (%)	Recall (%)
Flat spot	87 ± 3	89 ± 4	86 ± 6
Shelling	92 ± 2	92 ± 3	93 ± 3
Non-roundness	87 ± 6	87 ± 10	89 ± 4

gives the total fraction of correctly classified wheels, precision measures the fraction of correctly predicted defects out of all predicted defects and recall the fraction of correctly predicted defects out of all defects [31].

A. Model Selection and Evaluation

For all the experiments in this section the performance shown is computed on a test set that was not used for training or model/parameter selection. To make the evaluation robust against chance we repeat each experiment multiple times on new random train/test splits and report average and standard deviation over these repetitions. For data set 1 we only report the average as the standard deviation was not reported for the benchmark method. For data set 1 50% of the data is hold out for testing, for data set 2 20%. For the Wavelet-SVM the average performance is computed over 10 repetitions, for the DNNs over three repetitions. Using less experiments for the DNNs is due to computational reasons and justified by the low standard deviation over repetitions in all experiments $<= 2\%$. For the Wavelet-SVM three-fold cross-validation is performed on the training set to find the optimal hyper-parameters of the SVM and the Gaussian rbf kernel with grid-search on an exponentially spaced grid. For the DNN 10% of the training set were set aside as a validation set to benchmark performance online and decide on when to stop training.

As the class proportions for data set 2 are not balanced (c.f. Sec VII-C) training and evaluating the classifiers directly on this data would lead to bias and higher classification probability for the over-represented class. It would also make judging accuracy and comparing the methods and data sets hard, as the baseline for random chance would not be 50%. Therefore as a first step in all experiments we re-balance the class proportions of the data sets by randomly over-sampling the smaller class through sampling with replacement. While balanced data sets are useful for comparing methods and data sets, in a real-world setting the true proportion of the classes is important and mistakes for different types of error might have different cost. Therefore we recommend to give class probability estimates for each class when implementing such a system and then adapting a threshold for raising an alarm iteratively based on the test performance of the system.

Table III: Test set performance of the deep models on flat spots in data set 2.

Model	Accuracy (%)	Precision (%)	Recall (%)
Deep 1D	88 ± 1	96 ± 2	79 ± 3
Deep 2D	89 ± 1	93 ± 2	85 ± 2
Deep GAF	87 ± 2	91 ± 1	81 ± 5
Wavelet-SVM	87 ± 3	87 ± 2	86 ± 5

Table IV: Test set performance of the deep models on non-roundness in data set 2.

Model	Accuracy (%)	Precision (%)	Recall (%)
Deep MIL	81 ± 1	89 ± 3	71 ± 3
Deep Concat	81 ± 2	82 ± 2	78 ± 3
Deep Cyclic	88 ± 1	93 ± 1	82 ± 1
Wavelet-SVM	84 ± 9	80 ± 13	95 ± 3

B. Data Set 1

In a study prior to this publication [27], this data set was used to empirically demonstrate the effectiveness of a new algorithm for MIL [26]. Krummenacher, Ong, and Buhmann [27] beat state-of-the art MIL algorithms on this data set and get a classification accuracy of 70% with ellipsoidal multiple instance learning (eMIL). In this study features based on the Global Alignment (GA) kernel for time-series [32, 33] were used.

Using the features described in Section III with a SVM (Section V) we were able to improve accuracy to 92% (Table I).

With the current operational threshold of $\theta = 3$ on the maximal dynamic coefficient (Eq. 10) an accuracy of 60% is achieved. This is relatively low, as with random guessing already 50% accuracy could be achieved. It is thus important to note that the precision of this method is perfect with 100% of reported wheels being defective. So even though the method misses defective wheels it never raises a false alarm.

C. Data Set 2 - SVM

Equipped with our general method of constructing features from multiple wheel vertical force measurements (Section III) and learning a classifier from them (Section V) we are now ready to predict other types of wheel defects as well. We also evaluate the DNN based method (Section IV) in this section.

The SVM classifier (Section V) are trained on the labels obtained by this method for the defect types flat spot, shelling and non-roundness.

In Table II the performance on the reserved test set is reported for each defect type including standard deviation over the permutations. The performance on shelling is the best out of the three defect types. This observation can be explained by the fact that the training set for this defect type was by far the largest, so we were able to train a classifier with higher accuracy. This defect type also affects the wheel globally, so it is harder to miss for the sensors than a flat spot. To improve the performance on flat spot and non-roundness we trained custom deep neural networks and give the results in the next section.

For the defect type non-roundness, the load normalized features based on the load observed by individual sensors (c.f. Section III) substantially contributed to an increase in accuracy. This effect can be explained by the observation that wheel non-roundness errors do not cause a large variation on the within measurement time series since they are a non-discrete type of wheel defects. They do introduce variations between the different measurements per wheel on the other hand and so features based on averages per measurement sequence are important. We will improve the classification performance for flat spot and non-roundness in the next section by using a custom deep neural network (DNN) that is cyclic-shift invariant for classification of these defect types.

One complication of this data set arises from the lack of knowledge if the wagon passes the wheel load checkpoint with the same orientation as the wheels were annotated in the workshop. This lack of information leads to uncertain labels for the class of defective wheels, as not all wheels on a wagon necessarily share a defect. For the class of non-defective wheels this uncertainty does not pose a problem, since all wheels of a wagon are re-profiled and therefore are non-defective in our data set. We deal with this problem by adding both possible orientations of each wagon to the data set for the defective class of wheels. This augmentation of the data set introduces additional noise to the learning problem during training as non-defective wheels might be labeled defective. Nonetheless, we are able to train classifiers with high accuracy for all three types of defects (flat spot, non-roundness, shelling) based on data generated from this source. Since during testing the same uncertainty exists and actually non-defective wheels might have a defect class assigned the error rate of the classifier appears to be over-reported . Therefore the numbers reported in Table II and in the next section are a lower bound on the performance of the classifier.

D. Data Set 2 - Deep Learning

Using the same data set as in the previous section we evaluate the deep learning method (Section IV) on the two defect types flat spot and non-roundness. To simplify the experiments we do not include additional features like speed, measurement site or template fit, but only consider the wheel vertical force measurements from the WLC sensors. Therefore, the performance of the SVM is slightly worse compared to the previous section.

To compute the 2D image of the time series we proceeded as following: first, the recording from each of the 8 channels have been preprocessed via PAA [24], with bin number $N = 156$. The GAF encoding as well as the 2D graph were computed for each channels (we took the following parameters for the 2D graph: $V_{min} = -4, V_{max} = 6$ as the window captures more than 99.9% of all the values, and $r = \frac{V_{max} - Vmin}{N} = \frac{10}{N}$ to generate square pictures of size $N \times N$). Finally, the picture size was further reduced by averaging every 2×2 non-overlapping pixels for computational reasons, resulting in 8 channels of size 78×78 for both GAF and 2D graph encoding.

To prevent overfitting to the training set and to enable the model to explore a larger parameter space, we augmented the data by adding Gaussian noise and by randomly shifting and re-scaling the time series before applying image transformations.

We applied dropout regularization [34] on all the fully connected layers. To further improve generalization, we added an additional ℓ_2 weight regularization penalty term in the cost function ("weight decay") to encourage smooth solutions by favouring small weights. We have employed stochastic gradient descent with Nesterov Momentum [35] to accelerate convergence. The learning rate was set to decay inversely proportional to the number of epochs.

1) Flat Spots: In Table III we compare the performance of the different DNN models and the Wavelet-SVM. The only deep model that is able to out-perform the accuracy of the Wavelet-SVM is based on the 2D image of the time series. All of the deep models have smaller standard deviation and higher precision.

2) Non-roundness: In Table IV we compare the performance of the cyclic DNN with the DNN used for flat spot prediction (Deep MIL), a DNN that is trained on the concatenation of all the sensors (Deep Concat) and the Wavelet-SVM. Remember that the MIL-DNN used for flat spot prediction is trained by looking at the time series of each sensor individually and computing the loss on the sensor with highest probability of observing the defect. The performance of the different methods on the test set shows that MIL is an inadequate model for this type of defect since a wheel with a non-roundness defect can not be reliably identified on the basis of only one sensor measurement. This non-local behavior is in contrast to the challenge of predicting flat spot. Concatenating the sensors as is and not looking at the possible cyclic permutations resulted in training set accuracy similar to the cyclic shift network, but performance on the test set is significantly worse (Table IV). Intuitively ignoring the permutations leads to overfitting as the measurements in the test set might be shifted arbitrarily.

In comparison with the Wavelet-SVM the cyclic DNN shows higher accuracy and precision and reduced variance. Unlike the DNN for flat spot we only trained the cyclic DNN for non-roundness directly on the 1D time series, as the increase in parameters due to the concatenation of measurements of the sensors prohibited efficient training of the model on the 2D representation.

VIII. Conclusion

We have presented two machine learning methods for defect detection on railway train wheels. The methods analyse multiple time series of the vertical force of a wheel under operational speed and output if a wheel has a defect or not. Both methods are trained automatically on measurements gathered from defective and non-defective wheels. The first method is based on novel general wavelet features for time series. The second method employs deep convolutional neural networks to automatically learn features from the time series directly or from a 2-dimensional representation. We design cyclic shift invariant artificial neural networks for the detection of wheel flats and non-round wheels that model the relationship between

the measurements inherent to these defects. To evaluate our methods we collect two data sets from different sources and demonstrate improved performance for predicting flat spot, shelling and non-roundness.

The methods that were developed for this work are currently being implemented as part of the SBB wayside train monitoring system. To improve the quality of the training and test data RFID tags will be deployed to enable perfect association between defect labels and measurements. Further future work consists of integrating external features into the deep learning models, optimizing for precision and predicting severity scores for the defects. For the prediction of severity scores we obtained promising preliminary results on regressing the flat spot length using support vector regression [36] and the wavelet features.

REFERENCES

[1] R. Müller, D. Leibundgut, B. Stallaert, L. Pesqueux, and E. A., "Validation of wheel maintenance measures on the rolling stock for reduced excitation of ground vibration", SBB, D2S, Alstom, Trafikverket, Tech. Rep., 2013.

[2] P. Huber, B. Nélain, and R. Müller, "Rivas–mitigation measures on vehicles (wp5); experimental analysis of sbb ground vibration measurements and vehicle data", in *Noise and vibration mitigation for rail transportation systems*, Springer, 2015, pp. 531–538.

[3] E. Verheijen and F. Elbers, "Future european noise emission ceilings: Threat or solution? a review based on swiss and dutch ceilings", in *Noise and Vibration Mitigation for Rail Transportation Systems*, Springer, 2015, pp. 71–78.

[4] Y. Li and S. Pankanti, "Anomalous tie plate detection for railroad inspection", in *Pattern Recognition (ICPR), 2012 21st International Conference on*, IEEE, 2012, pp. 3017–3020.

[5] Y. Li, H. Trinh, N. Haas, C. Otto, and S. Pankanti, "Rail component detection, optimization, and assessment for automatic rail track inspection", *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 760–770, 2014.

[6] X. Gibert, V. M. Patel, and R. Chellappa, "Deep multi-task learning for railway track inspection", *ArXiv preprint arXiv:1509.05267*, 2015.

[7] T. Idé, "Formalizing expert knowledge through machine learning", in *Global Perspectives on Service Science: Japan*, Springer, 2016, pp. 157–175.

[8] N. Nenov, E. Dimitrov, V. Vasilev, and P. Piskulev, "Sensor system of detecting defects in wheels of railway vehicles running at operational speed", in *Electronics Technology (ISSE), 2011 34th International Spring Seminar on*, IEEE, 2011, pp. 577–582.

[9] T. K. Ho, S. Liu, Y. Ho, K. Ho, K. Wong, K. Y. Lee, H. Tarn, and S. Ho, "Signature analysis on wheel-rail interaction for rail defect detection", pp. 1–6, 2008.

[10] Y. Jianhai, Q. Zhengding, and C. Boshi, "Application of wavelet transform to defect detection of wheelflats of railway wheels", in *Signal Processing, 2002 6th International Conference on*, IEEE, vol. 1, 2002, pp. 29–32.

[11] W. Badran and U. Nietlispach, "Wayside train monitoring systems: Networking for greater safety", *European Railway Review*, vol. 17, no. 4, pp. 14–21, 2011.

[12] J. C. Nielsen and A. Johansson, "Out-of-round railway wheels-a literature survey", *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 214, no. 2, pp. 79–91, 2000.

[13] J. Nielsen, "Out-of-round railway wheels", in *Wheel–Rail Interface Handbook*, R. Lewis and U. Olofsson, Eds., Woodhead Publishing, 2009, pp. 245–279.

[14] S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.

[15] M. Unser and A. Aldroubi, "A review of wavelets in biomedical applications", *Proceedings of the IEEE*, vol. 84, no. 4, pp. 626–638, 1996.

[16] P. Kumar and E. Foufoula-Georgiou, "Wavelet analysis for geophysical applications", *Reviews of geophysics*, vol. 35, no. 4, pp. 385–412, 1997.

[17] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The jpeg 2000 still image compression standard", *Signal Processing Magazine, IEEE*, vol. 18, no. 5, pp. 36–58, 2001.

[18] I. Daubechies *et al.*, *Ten lectures on wavelets*. SIAM, 1992, vol. 61.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[20] L. Deng and D. Yu, "Deep learning: Methods and applications", *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[21] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks", *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[22] 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.

[23] 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.

[24] Z. Wang and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks", in *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", in *Proceedings of the IEEE*

International Conference on Computer Vision, 2015, pp. 1026–1034.

[26] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles", *Artificial intelligence*, vol. 89, no. 1, pp. 31–71, 1997.

[27] G. Krummenacher, C. S. Ong, and J. Buhmann, "Ellipsoidal multiple instance learning", in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 73–81.

[28] C. Cortes and V. Vapnik, "Support-vector networks", *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[29] M. Imani and U. Braga-Neto, "Optimal gene regulatory network inference using the boolean kalman filter and multiple model adaptive estimation", in *2015 49th Asilomar Conference on Signals, Systems and Computers*, IEEE, 2015, pp. 423–427.

[30] P. S. Maybeck and P. D. Hanlon, "Performance enhancement of a multiple model adaptive estimator", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 4, pp. 1240–1254, 1995.

[31] C. J. V. Rijsbergen, *Information Retrieval*, 2nd. Newton, MA, USA: Butterworth-Heinemann, 1979.

[32] M. Cuturi, "Fast global alignment kernels", in *ICML 2011*, 2011.

[33] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui, "A kernel for time series based on global alignments", in *ICASSP*, vol. 2, 2007.

[34] 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.

[35] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning", in *Proceedings of the 30th international conference on machine learning (ICML-13)*, 2013, pp. 1139–1147.

[36] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression", *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

Gabriel Krummenacher is a Ph.D. student at the Institute for Machine Learning at the Department of Computer Science of ETH Zurich. He is working on scalable methods for large-scale and robust learning, on wheel defect detection in a collaboration with SBB and on sleep stage prediction with deep learning. He received a M.Sc. in computer science from ETH Zurich in 2011. In February and March 2013 he was an academic guest at the NICTA Bioinformatics group in Melbourne. From September 2008 to February 2009 he did a software engineering internship in the trading technology team of Axa Rosenberg in San Francisco. He is interested in solving complex real world problems arising from industry or the medical domain through machine learning.

Cheng Soon Ong is a principal researcher at the Machine Learning Research Group, Data61, CSIRO. He is also an adjunct associate professor at the Australian National University, and an honourary research fellow at the University of Melbourne. His Ph.D. in Computer Science was completed at the Australian National University in 2005. He was a postdoc at the Max Planck Institute of Biological Cybernetics and the Fredrich Miescher Laboratory in Tübingen, Germany. From 2008 to 2011, he was a lecturer in the Department of Computer Science at ETH Zurich, and he has been with NICTA/Data61 since 2012. interested in enabling scientific discovery by extending statistical machine learning methods. In recent years, he has developed new optimization methods for solving problems such as ranking, feature selection and experimental design, with the aim of solving scientific questions in collaboration with experts in other fields.

Stefan Koller is head of the Wayside Train Monitoring System department at Swiss Federal Railways (SBB). He graduated from the Swiss Federal Institute of Technology Zurich (ETH) in Physics with a Ph.D. He has been working a number of years as senior scientist in mirco electro mechanical sensor system (MEMS). After that he has been working a couple of years as a senior consultant for software testing. Stefan Koller has been with the Wayside Train Monitoring Systems department of SBB AG since 2008. As system engineer for Wayside Train Monitoring Systems he was responsible for the development and rollout of SBB's unique fire and chemicals detection system and the wheel load checkpoint system.

Seijin Kobayashi Seijin Kobayashi is a graduate student from ETH Zurich and Ecole Polytechnique in Paris. From September 2015 to March 2016 he worked on his Master Thesis at ETH Zurich on defect wheel detection and sleep stage staging using deep learning. He received a M.Sc. in computer science from ETH Zurich in 2016. He is currently ongoing an internship at Google Zurich. He is interested in applying deep learning for real world problems as well as improving artificial neural network architectures and learning algorithms.

Joachim M. Buhmann is professor for Information Science and Engineering at the Computer Science Department of the Swiss Federal Institute of Technology Zurich (ETH). He received his Ph.D. degree in theoretical physics from the Technical University of Munich, Germany, in 1988. He has held postdoctoral and research faculty positions at the University of Southern California, Los Angeles, and the Lawrence Livermore National Laboratory, Livermore, CA between 1988 and 1992. Until October 2003, he headed the Research Group on Pattern Recognition, Computer Vision and Bioinformatics in the Computer Science Department, Rheinische Friedrich-Wilhelms Universität Bonn, Germany. In October 2003 he joined ETH Zurich. His current research interests cover machine learning, statistical learning theory and its relations to information theory as well as applications of machine learning to challenging data analysis questions. The machine learning applications range from image understanding and medical image analysis, to signal processing, bioinformatics and computational biology. Special emphasis is devoted to model selection questions for the analysis of large scale heterogeneous data sets. Dr. Buhmann has served as an associate editor for IEEE-TNN, IEEETIP and IEEE-TPAMI.