Remaining Useful Life Prediction for Railway Switch Engines Using Classification Techniques

Thomas Böhm[1]

[1] *formerly German Aerospace Center (DLR) Institute of Transportation Systems Rutherfordstr. 2, Berlin, 12489, Germany*
dervincent@gmx.de, thomas.boehm@konux.de

ABSTRACT

A highly available infrastructure is a premise for capable railway operation of high quality. Therefore, maintenance is necessary to keep railway infrastructure elements available. Railway switches, especially, are critical because they connect different tracks and allow a train to change its moving direction without stopping. Their inspection, maintenance and repair have long been identified as a cost driver. Switch failures, particularly, are responsible for a comparable high number of failures and delay minutes. The reduction of failures would not only save maintenance costs, but also let more trains arrive on time and hence increase the attractiveness of the railway transport. Therefore, upcoming failures need to be revealed early enough to allow an effective planning and execution of failure preventing maintenance activities. Research is exploring ways to predict the remaining useful life of switches.

This paper presents an approach to predict the remaining useful life (RUL) of railway switch engine failures. The development is based on measurement data of the electrical power consumption of switch engines. The two year time series of 29 switches of Deutsche Bahn was recorded by a commercial switch diagnostic system leading to roughly 250 000 measurement tuples. Since earlier researched showed that the electrical data alone is not sufficient enough, additional data is integrated. It takes into account the dependency of the switch condition data from climatic conditions and certain properties of the switch construction type.

Predicting a RUL is quite challenging in many PHM applications. To avoid common problems with uncertainty in measurement data, a long prediction horizon (month) of small time units (hours) and to stabilise end user acceptance the approach transforms the RUL prediction problem into a classification problem of multiple classes. It, then, uses two different supervised classification techniques, Artificial Neural Networks (aNN) and Support Vector Machines (SVM), to predict the RUL in the form of classes. However, as known from the no free lunch-theorem of classification, there is no ultimately best performing technique. The success depends on the problem and data structure as well as on the parametrisation of the technique or the selected algorithm respectively. Especially aNN and SVM have a high number of possible parametrisations. They can fail the task or result in a very good performance under the heavy influence of their parametrisation. Hence, it is an important aspect of this paper to share how the different parameters effect the RUL prediction and which parameters result in maximum performance. In order to compare the performance, two metrics are chosen, the Matthews Correlation Coefficient (MCC) as single value metric and a visualisation of the confusion matrix as more comprehensible metric. Finally, deriving those parameters maximising the RUL prediction results enables one of the two classification techniques to reveal upcoming failures of the switch engine early enough to prevent them.

1. INTRODUCTION

A highly available infrastructure is a premise for capable railway operation of high quality. Therefore, maintenance is necessary to keep railway infrastructure elements available. Especially switches are critical because they connect different tracks and allow a train to change its moving direction without stopping. Switches consist of several components making them a complex infrastructure element (see Figure 1). Across Europe's railway companies, switch failures often account for the most infrastructure related failures or delay minutes (Cocciaglia, 2012; Rama & Andrews, 2013; Pool & Vlek, 2016). In Case of Deutsche Bahn 19 % of delay minutes are caused by switch failures (Rausch, 2010). If applied to the infrastructure delay minutes of 2015 (DB Netz AG, April 2016), switch failures sum up to around five years of delays. This does not only result in high maintenance cost but also leaves rail travellers highly dissatisfied. In order to reduce those negative effects, upcoming failures need to be revealed

Figure 1. Example of a simple switch

Figure 2. Pattern of active power during repositioning of tongues within SIDIS W, following (Stoll & Bollrath, 2002)
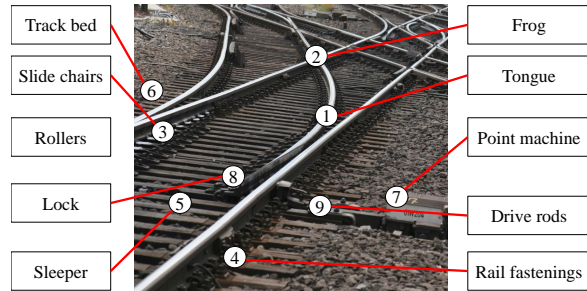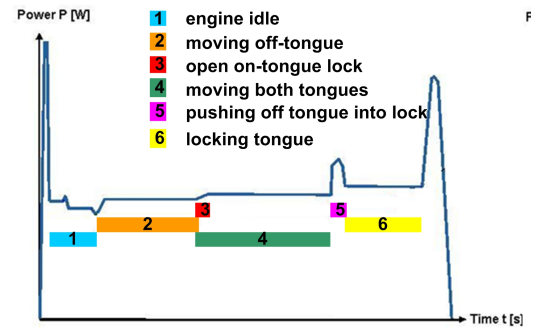

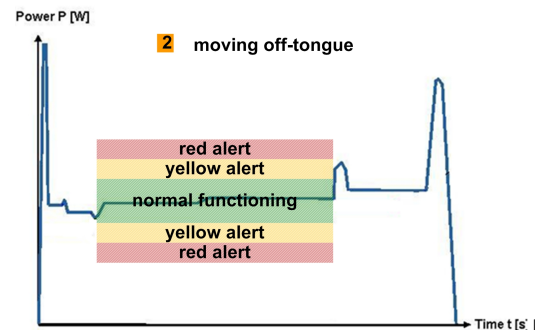Figure 3. Principle of alarms in SIDIS W, following (Stoll & Bollrath, 2002)

early enough to allow an effective planning and execution of failure preventing maintenance activities.

The switch engine (also referred to as point machine) moves the switch tongues (also referred to as blades), which make contact with one or another rail and thus enabling a passing train to take one or the other direction. The majority of switch failures is caused by faults of the movable components especially the locking device, drive rods and engine (Rama & Andrews, 2013). Therefore, monitoring these parts is an established way of failure detection.

So called switch diagnosis systems have been invented to monitor the function of the switch by monitoring the electrical power consumption of the engine during tongue repositioning. Since the level of power is proportional to the force necessary to move the tongues, related faults can be detected. Based on this well known principle, a number of different approaches has been developed to diagnose failures, e.g. (García Márquez, Weston, & Roberts, 2007; Chamroukhi, Samé, Aknin, & Antoni, 2008; Silmon, 2009; Atamuradov, Camci, Baskan, & Sevkli, 2009; Bai, 2010; Eker, Camci, & Kumar, 2010; Asada, Roberts, & Koseki, 2013). Neither those nor the stand-alone diagnostic system predict the remaining useful life as shown in (Böhm, 2012). But the RUL is essential in order to enable maintenance to act before the switch fails. Therefore, the following sections describe the RUL prediction based on supervised learning and on measurement data of the electrical power consumption of switch engine combined with additional data.

2. USED DATA SOURCES

2.1. Electric Power Consumption of Engine

There are only a few switch condition monitoring systems available on the market in Europe, e.g. Roadmaster 2000 from VAE, POSS from Strukton, and SIDIS W (compact) from Siemens. This research is based on the latter. SIDIS W only uses the switch motor as sensor. Relevant data, like voltage, amperage, and active power are directly measured at the engine and then processed in a remote diagnostic component. During the repositioning of tongues, the power time line graph shows a typical development (see Figure 2). Based

on that idealistic, general pattern, each individual switch is recorded with its characteristic level of power at the installation. Also threshold values are defined. A yellow and a red alert, each for an upper and a lower boundary, are set (see Figure 3). In case the measured values fall below or exceed their thresholds respectively, first a yellow alert is given. If the measured values decrease or increase further respectively, a red alert is given. SIDIS W presents eight indicators, each displaying the condition state during a different phase or of a different parameter. Does the measured power level during operation, respectively the corresponding indicator flag, reach the yellow or even the red alert, a failure might occur or might have occurred. At this time, at the latest, the responsible operation staff should maintain or at least inspect the corresponding switch. Theoretically, SIDIS W is able to detect tongue deformation, exceptional slackness of the locking, engine failures, and hardly moving tongues caused by a bad switch condition (Stoll & Bollrath, 2002). In reality, the SIDIS W produces a high rate of false alerts as proven in former analysis (Böhm, 2012). The main reason is that the condition is influenced by other parameters too, such as climate, switch characteristics, maintenance, age and usage (University of Birmingham, 2008; Zwanenburg, 2009).

2.2. Temperature and Humidity

The humidity on the slide chairs and the locking device decreases friction a little bit. The temperature effects material contraction. Its change induces movement of the tongues alongside the rails. This can lead to cant and blocking in the locking device. Especially the temperature causes seasonal and daily fluctuation in the measurements. Therefore, temperature and humidity have a relatively high correlation with the electric power consumption of the switch engine, see (Böhm & Doegen, 2010) for details. Hence, such data should be combined with the SIDIS W measurements in the development. The data had been retrieved from a public weather station nearby the switch locations. Though, the actual temperature and humidity at a particular switch might differ from the general areal climate the results were convincing (E.g. for one Switch the correlation coefficient between maximum power of the engine gear and the temperature is -0.792. The coefficient of determination is 62.78 %). Besides, it was the only available data source.

2.3. Switch Construction Characteristics

As for the switch construction characteristics, the switch type, locking type, radius, sleeper type, rollers and engine type are used to group switches with similar behaviour in the electric measurements. For example, single switches are less complex than curve or even double switches and need less maintenance while a smaller radius induces higher lateral forces when a train is passing resulting in higher wear. Also, different locking types are responding differently to temperature changes due to their tolerance regarding tongue movement. Rollers cause less friction than simple slide chairs. Altogether, the characteristics of switches need to be considered when interpreting measurements. The level of high power consumption can be a normal behaviour within one group while meaning a fault in another group. This is reflected by developing a prediction according to groups. This also helps to transfer results to new switches. If the prediction follows groups instead of individual switches, then, a new switch can be interpreted according to its corresponding group right away instead of building an individual data pool over time before training and applying a prediction model.

As mentioned above, there are more parameters influencing the condition of switches, most notably the usage (e.g. number of trains passed, cumulated axle load passed) as well as the condition and maintenance of the track bed. Unfortunately, no sources were available to investigate the effects and incorporate them into the prediction.

2.4. Switch Failures as Condition Reference

The data driven approach chosen to develop a RUL prediction is based on supervised learning techniques (as described later). This requires some kind of reference for the RUL to derive and – equally important – to evaluate the prediction results. The actual condition of a switch is unknown at any time. Only when it fails it is revealed. Deutsche Bahn records failures in its enterprise resource system SAP. The database contains the start and end time of a failure along with predefined failure parts and causes. Also, each database entry includes freely editable text for the failed parts and their reason. This data is used as reference. Here, the before mentioned high rate of false alerts of SIDIS W produced an unintended benefit for the prediction development. Some time after the initial installation in 2002, the maintainer stopped paying attention to the alarms. But SIDIS W measurements were recorded continuously without any implication of the actual maintenance activities. Fortunately, this provides a seldom situation in which the condition monitoring is independent of the condition. Both sources can be used for a combined analysis with a retrospective evaluation of hypothetical and real condition. In contrast, if the maintainer acts according to the system, afterwards it will be unclear if a failure would have occurred actually. Hence, it would be impossible to validate the system performance.

2.5. Attributes and Amount of Used Data

After explaining the type of used data sources, the following section will describe the actual attributes and amount of data used for the prediction.

From the overall 56 parameters which SIDIS W records only 11 plus 4 meta attributes are relevant (see Table 1 for their description). Many of the discarded parameters only hold the information if a threshold of a measurement parameter was passed and what alarm was given. Others were simply static bearing no information useful for the prediction.

The measurements are combined with the temperature and humidity via their respective time attribute. Begin and end of a failure as well as the documented failing component and its cause are joined to the data base. Using the start time of the failure, each power measurement tuple is associated with an upcoming failure. It enables the calculation of the real RUL of each tuple. The static attributes of switch characteristics are used to define similar groups as described in section 2.3.

Deutsche Bahn provided electric power monitoring data from 29 switches in stations and on track. All are located within an area of 20 km at lines with mixed traffic of freight, regional and long distance trains. These switches are quite heterogeneous in their characteristics:

- Some are simple switches others are curved.
- Their radius cover 190, 500, 760 and 1200 m.
- Some are on concrete and some on wooden sleepers.
- The number of rollers ranges from 0 to 7.
- Some have a clamp lock others a pawl lock.
- But they all have a S700K electromechanical engine.

Table 1. Attributes chosen for prediction

Attribute	Description [Unit]
ID	Unique identifier of the measurement tuple
AKZ	number of the switch
ErfDatum	Date and time of measurement
R	Direction of tongue movement
Pleer	Avg. power during idle engine (phase 1) [P]
Plauf1	Avg. power during moving off-tongue (phase 2) [P]
Plauf2	Avg. power during moving both tongues (phase 4) [P]
Pverschl	Avg. power during locking (phase 6) [P]
Tumlauf	Time of repositioning (throw) [s]
Iunsym	Asymmetry of electrical power
Pzvorspg1	Power portion via mechanical tension of first tongue [P]
Pzvorspg2	Power portion via mechanical tension of second tongue [P]
Prutschpk	Maximum power at the end peak [P]
Umotor	Electric voltage of the engine [V]
PverschlPk	Maximum peak begin of locking (phase 5) [P]
Feucht	Humidity [%]
Temp	Temperature [C]
StorID	Unique identifier of the failure
Storbegin	Date and time when the failure occurred
Storende	Date and time when the failure was solved
Beschreib	Free text description of the failure
UrsachenT	Free text description of the failure cause
ProblCodeT	Predefined text of failure cause
ProblGrpT	Predefined text of failing component
WTyp	Switch type
VerschlTyp	Locking type
Zungenroll	Number of rollers
Radius	Radius of the switch [m]

$$A = \{A_m : 1 \leq m \leq 28\} \tag{1}$$

$$S = L \times A \text{ with } L := \{l \in \mathbb{N} | 1 \leq l \leq |S|\} \\ \text{and } |S| = 184278 \tag{2}$$

To give a perspective on that numbers, past research mainly used artificial (simulated) data or data from laboratory installations (García Márquez et al., 2007; Silmon, 2009; Atamuradov et al., 2009; Bai, 2010; Eker et al., 2010; Asada et al., 2013). If real data was used, only a few (less then five) switch engines had been investigated (Chamroukhi et al., 2008; Pedregal, García Márquez, & Roberts, 2009). Therefore, the diversity and number of switches as well as the amount and covered time of real live data provide a foundation for the prediction which distinguishes this work from other.

3. BASIC PRINCIPLE FOR THE PREDICTION

In theory, the condition is a function Z assigning a RUL of a real number value b in the interval between the failure 0 and the maximum RUL $maxRUL$ to each sample S_l.

$$Z: RUL \rightarrow \mathbb{B} \text{ with } \mathbb{B} = \{b: 0 \leq b \leq maxRUL\}_{\mathbb{R}} \tag{3}$$

In reality, predicting a RUL is quite challenging in many PHM applications. A common problem is the uncertainty in measurement data, which leads to uncertainty in the prediction, especially in cases in which a long prediction horizon (e.g. month) is given in small time units (e.g. hours). The predicted RUL might fluctuate and may appear unreliable in the view of the users. Besides, what is the benefit of a high resolution RUL which gives 3.582 month at one measurement and 3.574 month at the next while the time to act only requires a few days lead time? As in many applications, the lead time to timely act before a failure will define the best resolution. Sometimes, the measurements are only discrete and hence, do not allow for continuous prediction values. In case of switch failure prediction, all of that makes a discrete approach more suitable. Therefore, the RUL is binned in intervals covering the certain time periods as initially presented in (Böhm, 2015) and schematically shown in Figure 5.

One important aspect is the definition of intervals. They should consider the necessary lead time to prevent failures. The intervals were identified in a workshop with maintenance staff and planners of Deutsche Bahn. Two things are important:

1. An **emergency fault clearance** cloud be executed if **1 to 2 hours** remain before the break down. This time

The recorded period stretched from March 2007 to March 2009 but was not complete for all 29 switches due to recording gabs. Nevertheless, for at least 17 switches, one complete year has been recorded which is important to cover all seasonal effects. Figure 4 gives an overview of the recorded periods for each switch and the amount of daily measurements.

Since there are measurements for which a failure would eventually occur in the unknown future and hence unrecorded by the time the data was provided, some measurements are cut. So, the data is right censored when the desired event (failure) is missing. Here, it affects 61584 tuples. Additionally, 48934 measurements had to be discarded because the system itself marked them as invalid measurements without a comprehensible reason. Due to these circumstances, 184278 tuples remain out of 294796 original tuples.
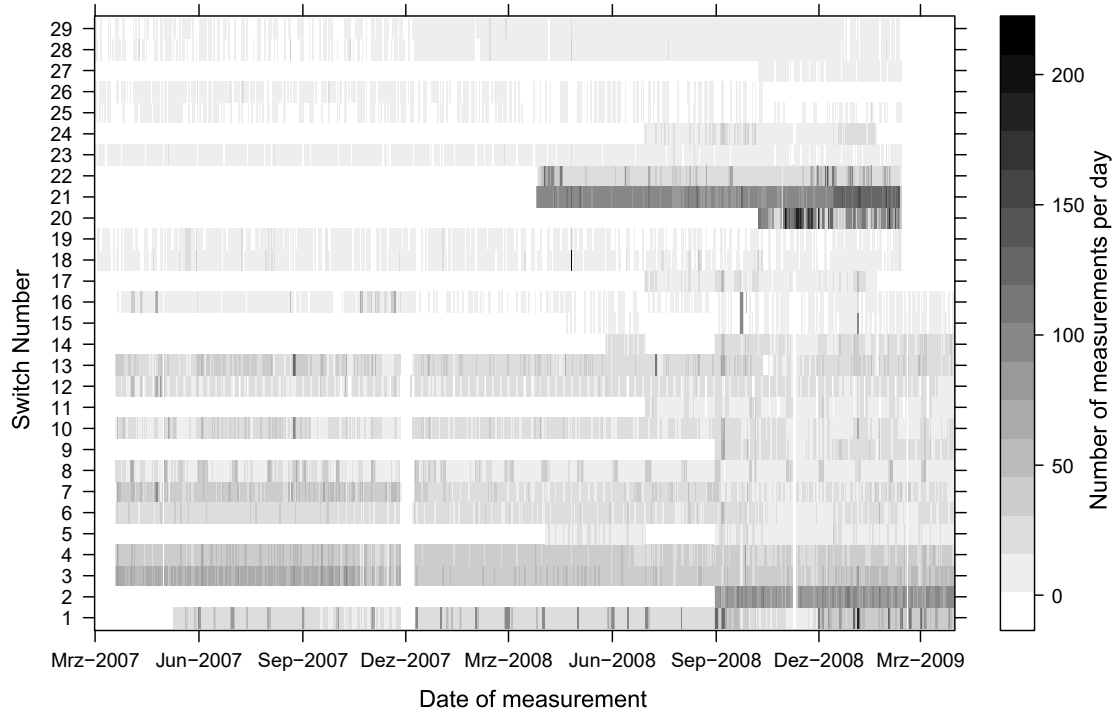
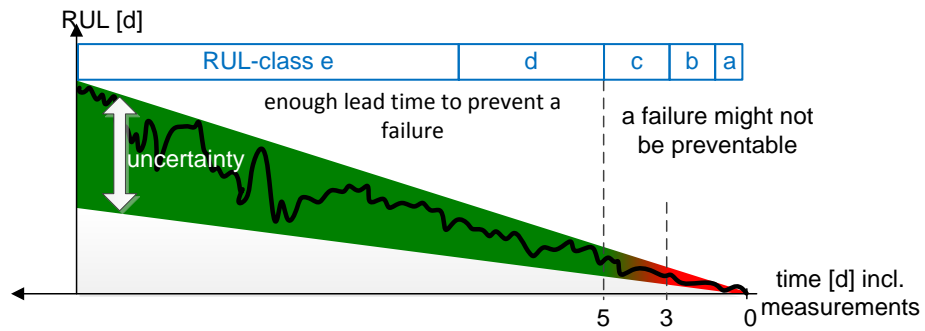Figure 4. Switches and their periods of recorded SIDIS W measurements

Figure 5. Principle of transforming continuous RUL into discrete classes

is mainly required by the staff to get to the concerning switch and execute maintenance even if they are on stand-by duty. In this way, an immanent operation disruption might be averted, assuming the track does not need to be blocked.

2. The emergency clearance does not guarantee a delay prevention and comes at high costs and depends on available resources. This is to be avoided. To **plan and timely execute maintenance** activities efficiently, potential failures need to be revealed **3 to 5 days** prior to the event. Activities and resources are invoked into the work schedule. This way, it is almost guaranteed that the problem will not affect the train operation, which makes the 3 to 5 days the pivotal mark.

Now, it is possible to understand the prediction as a classification problem with (at least) two classes. One contains all tuples with a RUL of less then 5 days and the other with a RUL of 5 days and more. But binary classes seem to be only slightly more helpful thus, additional distinction is required. Any further binning should consider the above mentioned uncertainty by covering larger periods for high RUL and smaller for the near future. This reduces fluctuation and stabilises end-user acceptance. In favour of an easy implementation, a geometric sequence (2^x) is chosen here to represent class intervals. The sequence is not only mathematically precisely defined but also fits well to time values if assigned to days. For example, $2^{-4} = 0.0625$ days is equal to 1.5 hours while $2^2 = 4$ days. Both represent the mean of the emergency and the scheduled maintenance marks.

$$Z \colon RUL \to I \text{ with } i \in I = \{1, 2, \ldots, 14\}$$
$$\text{and } z(RUL) = \begin{cases} 1 \colon 0 & \leq RUL \leq 2^{-4} \\ i \colon 2^{i-6} & < RUL \leq 2^{i-5} \\ 14 \colon 2^8 & < RUL \end{cases} \quad (4)$$

As described above, each sampled tuple has a continuous real RUL (the real condition TZ) derived from the next upcoming failure, which is documented in the SAP system. The attribute $A_{TimeToFailure}$ carries this information. The value is empty if measurements are not leading to a recorded failure.

$$TZ \colon S \to Z$$
$$s_l \mapsto z_i \quad \forall s_l \in S \colon A_{TimeToFailure}(s_l) \neq \emptyset \quad (5)$$

The prediction goal is to identify another function f that is learned from a training set of the samples. The function shall provide a hypothetical RUL (the hypothetical condition HZ) for a set of test samples.

$$f \colon S \to Z$$
$$s_l \mapsto z_i \quad \forall s_l \in S_{Training} \subset S \quad (6)$$

$$HZ \colon S \to Z$$
$$s_l \mapsto z_j \quad \forall s_l \in S_{Test} = S \setminus S_{Training} \quad (7)$$

4. CLASSIFICATION METHODS AND PERFORMANCE METRICS

4.1. Selecting Suitable Classification Techniques

The previous section transformed the continuous RUL into discrete classes with respect to lead time relevant for maintenance scheduling. The prediction became a classification problem with multiple classes. Each class is defined by a certain combination of feature (data) values of the observations which can be found in all or at least the majority of class members. A classification function identifies those describing feature values and then compares a new observation to it in order to asses and assign the class membership. If the actual class membership of observations is known, like in this case, a supervised learning technique can be applied based on the measured data (see section 2). A subsection of the observations is used to derive a classification function. Some techniques build a representing model (e.g. a class-defining boundary in feature space or series of transition functions leading to a class membership) or simply use a distance measure to the known class instances to asses and assign a new observation.

The world of data mining offers various techniques and a wide range of algorithms to solve such a classification problem. (Note: Here, technique is understood as a family of algorithms belonging to the same basic principle. For example, a Support Vector Machine is the technique and the particular kernel-function is the algorithm) Not every technique suits the underlying structure and the data. The case at hand is characterised the following:

- The number of classes is relatively high and can vary because switches have different number of failures during the recorded two year period. Hence, not every switch has measurement data in each class. That requires classification techniques with low sensitivity to the number of classes.

- The data contain a lot of noise against which a technique must behave robustly.

- Any technique must deal with the heavily unbalanced class membership distribution, see Figure 6.

- The number of used attributes (dimensions) is not small. So, a technique which is applicable for a high number of

Number of Samples

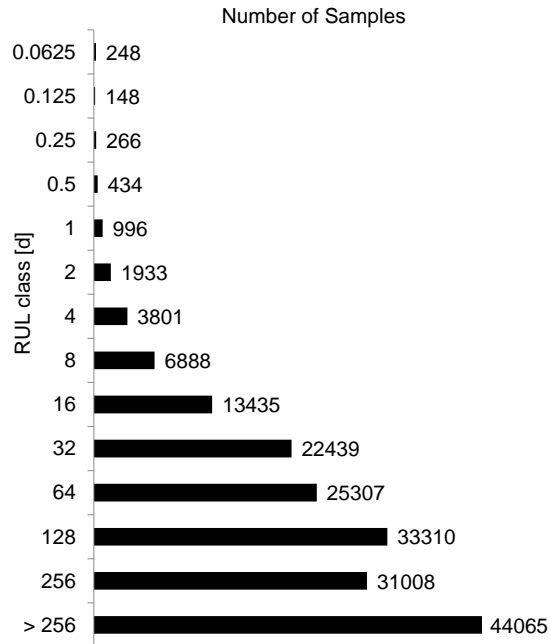| RUL class [d]	Number of Samples
0.0625	248
0.125	148
0.25	266
0.5	434
1	996
2	1933
4	3801
8	6888
16	13435
32	22439
64	25307
128	33310
256	31008
> 256	44065

Figure 6. Number of samples in each RUL-class is heavily unbalanced

dimensions is preferred.

- Switches with certain design characteristics and resulting similar behaviour are grouped. The measurements are significantly different between these groups. The classification technique should account for that and the fact that not every group is represented in training. Therefore, the technique should not tend to over-fitting.

Techniques like k-Nearest-Neighbour (kNN), artificial Neural Networks (aNN), Decision Trees (DTree) and Support Vector Machines (SVM) are suitable while the popular Bayesian Networks are not. The later is influenced by the class distribution and does not handle noisy data well (Lee, 2010; Sebastiani, Abad, & Ramoni, 2010). Although, all of the four mentioned techniques had been investigated, only the work with aNN and SVM will be described further because of the various parameter options they have. Even with generally suitable techniques, there is no ultimately best algorithm to solve a problem. This is known as the "No-Free-Lunch-Theorem" (introduced by Duda et. al in 1973) and still holds true (Duda, Hart, & Stork, 2012). The success of a technique to solve a problem sometimes heavily depends on the algorithm and the parameters. The same algorithm can produce excellent results or completely fail the task depending on the configuration. Therefore, identifying the right parameter systematically is an important scientific contribution, not only in this field but in various research areas like biology, medicine, and finance.

Since aNN and SVM are well known techniques, only a very brief description is given here. **Artificial Neural Networks**

take the human brain as role model. Like in its biological origin, aNN consist of neurons that process an input to an output. In a single neuron, the sum of all weighted input signals is passed through depending on a threshold value. The processed signal can be the result or the input for another neuron. The network is build of multiply neurons in one to many layers of neurons. During the supervised learning phase, initial (random) weights are adapted in each epochs of output calculation until the overall discrepancy to the desired output is acceptable or a certain number of epochs has been reached. Thought the theory of aNN was developed in the 1940s not until computers reached a level of computation power aNN could be applied to real problems. aNN are highly capable to map complex, non-linear functions between input and output vector as well as to parallelise computation. They also tolerate noisy data. Their biggest disadvantage is a tendency to converge towards a local optima instead of a global one. Furthermore, it is difficult for a user to asses weather a training resulted in an global optimum or not. Sometimes, over-fitting can be a problem with aNNs too. For more details about calculation, transition functions and network designs of aNNs see (Hand, Mannila, & Smyth, 2001) or (Zhang, 2010).

Support Vector Machines have been introduced by Vapnik in 1995 (Vapnik, 2008) and gained popularity in recent years. Simply speaking, SVM try to find a separation axis (hyperplane) in the feature space that maximises the distance (margin) between data points and best divides classes. The hyperplane is defined by points along the margin – the name giving support vectors, see figure 7. Many real life problems do not consist of linear separable classes. To solve these, SVM uses Kernel-Functions to transform the feature space into a higher dimension and to identify a hyperplane there. Many Kernels had been defined over time, some are linear, polynomials of n-th grade or sigmoid. SVM are very robust against noise, permutation in the classes, and the number of dimensions. Contrary to aNN, they almost always converge to the global optima because of the margin maximisation. This comes at the cost of high computation time which exponentially growth with the number of classes. The right choice of Kernel is essential to a SVMs success. Finding the right Kernel at high computation time is the biggest drawback of SVM. For more details about optimisation, Kernels and parameters see (Burges, 1998), (Shmilovici, 2010) or (Ben-Hur & Weston, 2010).

4.2. Selecting Suitable Metrics

Before selecting metrics, it is important to introduce the confusion matrix $\mathbf{C}$ as the basis of many metrics for classification problems. The two Eqs. (5) and (7) in section 3 already defined one vector containing the real class of all samples and another vector with all hypothetical classes as the result of a classifier (parametrised instance of a classification algorithm).
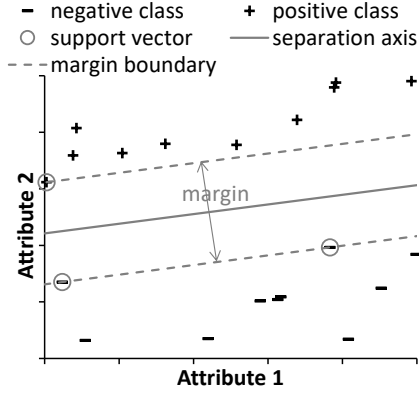
Figure 7. Principle of SVM for linear separable binary classes

$$\mathbf{C} \in \mathcal{M}(I \times I, \mathbb{N})$$
$$\text{mit } C_{i,j} = |\{s \in S \colon TZ(s_l) = z_i, HZ(s_l) = z_j\}| \tag{8}$$

The confusion matrix $\mathbf{C}$ is quadratic and each cell contains the number of samples belonging to the real class in the row and being classified as the corresponding class in the column.

Finding the right classification technique and the parameter is essential for success. In order to evaluate the success, selecting or defining a suitable performance metric is vital too. The metric, of course, must be independent of the classification technique when comparing their performance. The assessment must be as impartial as possible. Metrics have different properties determining how they react to certain characteristics of the data structure. For example, the very simple and commonly used accuracy. It is defined as the ratio from the number of correct classified samples of all samples. The accuracy is highly variant to the class distribution, as can be seen from two simple examples of a binary classification. Table 2 shows the confusion matrices of the same classifier but with different distributions between the classes *true* and *false* by factor 1000.

Table 2. Confusion matrices of one classifier but with different distributions between the binary classes

Classifier 1 with class distribution a

	$C_I, 1$ **positive**	$C_I, 2$ **negative**
C_1, J **true**	3000	2000
C_2, J **false**	3	12

Classifier 1 with class distribution b

	$C_I, 1$ **positive**	$C_I, 2$ **negative**
C_1, J **true**	3	2
C_2, J **false**	3000	12000

Though, it is the same classifier which performs exactly the

same, the accuracy suggests otherwise. Misleadingly, at 0.6 ($\frac{3000+12}{3000+2000+3+12} = 0.6$) the first seems to perform quite not as good as the second with 0.8 ($\frac{3+12000}{3+2+3000+12000} = 0.8$). Since, at the moment of application the actual class distribution will be unknown, it is important to identify the best classifier independent of the distribution at the moment of development.

This demonstrates how essential the metric can be to asses the solution and hence to impartially find the best way to solve the problem. The structure and the data of switch failure prediction require metrics with the following properties:

- The metric must be applicable from binary to multiple classes.

- It must be invariant against the class distribution. The class distribution is not only unknown for future use but also changes during training iteration when samples are drawn randomly.

- Any metric must tolerate noise in the class membership.

- The classes shortly before the failure contain little data, but any metric should treat these small classes the same as others.

- Also small changes in the performance should be distinguishable by the metric.

The literature provides many metrics, but only a few of them meet the criteria above. A systematic investigation of metrics properties can be found in (Leao, Yoneyama, Rocha, & Fitzgibbon, 2008; Sokolova & Lapalme, 2009; Ferri, Hernández-Orallo, & Modroiu, 2009). The selection is based on the descriptions and results therein. Since none of the known metrics fits ideally, Matthews Correlation Coefficient (MCC) and a simple visualisation of the confusion matrix will be used in the paper.

Matthews Correlation Coefficient (MCC) was introduced by (Matthews, 1975). It can be used for binary and multiple classes. The MCC handles noise and unbalanced class distributions very well while being good to display small changes in the confusion matrix. Though, it is not completely invariant to a changing class distribution it reacts far less sensitive than most other metrics (see (Jurman & Furlanello, 2010) for more details). The MCC is calculated from the confusion matrix (Eq. 8) with additional indices o, p, q as follows:

$$MCC =$$

$$\frac{\sum\limits_{i,j,o=1}^{I} C_{ii}C_{oj} - C_{ji}C_{io}}{\sqrt{\sum\limits_{i=1}^{I}\left(\sum\limits_{j=1}^{I} C_{ji}\right)\left(\sum\limits_{\substack{p,q=1 \\ p\neq i}}^{I} C_{qp}\right)}\sqrt{\sum\limits_{i=1}^{I}\left(\sum\limits_{j=1}^{I} C_{ij}\right)\left(\sum\limits_{\substack{p,q=1 \\ p\neq i}}^{I} C_{pq}\right)}} \tag{9}$$

As a single value metric, the MCC makes comparing several results easy. Besides, the MCC is foremost used to evaluate classification at different parameters because of its high resolution. A MCC value of 1.0 shows a perfect classification while -1.0 represents a completely wrong result, see (Jurman & Furlanello, 2010) for more details about the MCC.

5. CLASSIFICATION RESULTS FOR PREDICTION

Now, that classification techniques have been selected as well as the metrics to measure the success, this section will identify the most successful parametrisation and compare the prediction results.

During the development, the data set is separated into a 70% portion as training set and the disjunct remaining 30% as test set. The sets are drawn randomly while approximately keeping the class distribution intact (stratified random sampling). Since the classification result can depend on the samples within the training respectively test set, the steps of drawing, training, and testing are repeated 100 times. Afterwards, the evaluation is expressed with the arithmetic average of the metric from these iterations. The number of iterations could be different as long as it ensures that performance is not the result of a lucky or unlucky combination of samples in the training or test set.

5.1. Best Individual Parameters

In this case, the **aNN** is a Multi-Layer-Feed-Forward network implemented with the standard algorithm from (Riedmiller & Braun, 1993). The investigation of the best parameter settings is done systematically. Therefore, the number of hidden layers is increased stepwise by 1 while the number of neurons per layer is incremented from 8 to 38. (The training is stopped after 100 epochs, though, the result does not any longer change significantly after 60 to 80 epochs).

Figure 8 and Table 3 show a mediocre success of the aNN, though, it is stable throughout all configurations. The performance is clearly above 0.0, hence, it is better than a random classification. A closer look into the range from the minimum to the maximum MCC (visualised in Figure 9) reveals that the performance decreases if more than one hidden layer is selected. The best performance is reached with 28 neurons but it does not really make a difference between 24 to 34 neurons. The standard deviation from the 100 iterations ranges from 0.0127 to 0.0322. This indicates a very good overall precision.

The data is noisy and the samples in the feature space are close to each other and often overlapping. Paired with the aNN tendency to converging into local minima, this might explain the results. The aNN learns a particular model relatively fast, but this is not general enough.
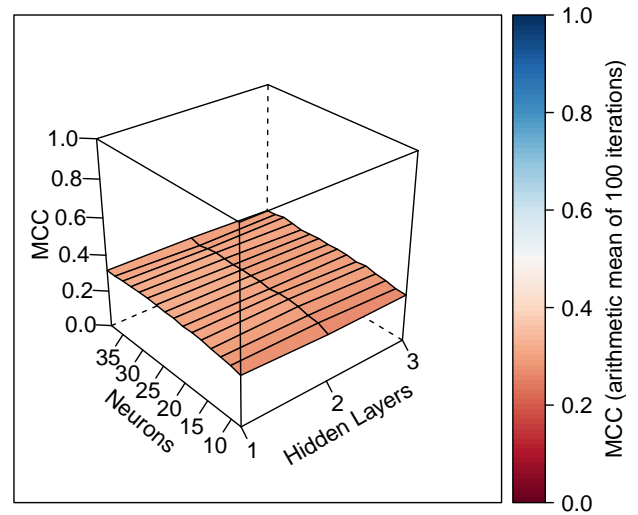

Figure 8. aNN prediction performance at various number of hidden layers and neurons
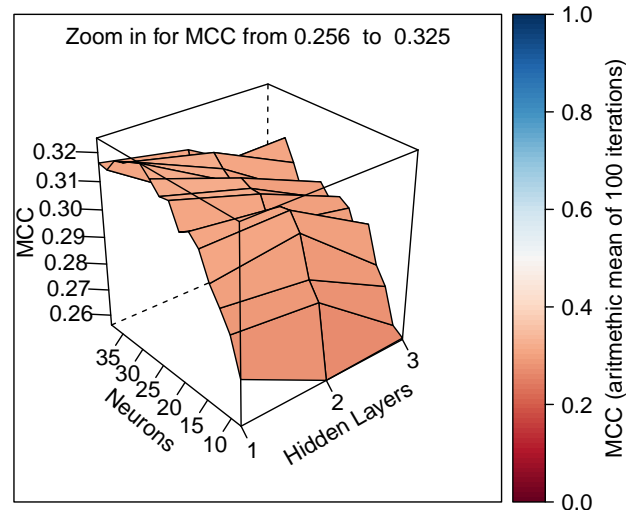

Figure 9. aNN prediction performance at various number of hidden layers and neurons (zoomed into the MCC range $[0.256, 0.325]$)

9

Table 3. MCC as metric of the aNN prediction performance at various number of neurons and hidden layers

Neurons	Hidden Layers		
	1	2	3
8	0.273	0.256	0.257
10	0.286	0.283	0.259
12	0.293	0.289	0.272
14	0.299	0.303	0.278
16	0.308	0.309	0.292
18	0.312	0.308	0.284
20	0.310	0.300	0.294
22	0.318	0.309	0.296
24	0.318	0.309	0.294
26	0.322	0.311	0.298
28	0.325	0.308	0.294
30	0.324	0.311	0.291
32	0.322	0.315	0.300
34	0.321	0.307	0.308
36	0.316	0.295	0.297
38	0.316	0.311	0.293

Nevertheless, aNN has been subject for research since the 1940s and many network types – sometimes very special ones – have been developed since. Therefore, it can not be excluded that some aNN exists which is capable to produce better results. But the extensive effort to further investigate this would only make sense if the SVM do not reach sufficient results.

For any **SVM**, the user must first select a Kernel function which transforms the data into a higher dimension to optimise the separation hyperplane. Besides, the penalty term ($\in \mathbb{R}^{>0}$) needs to be chosen. It defines how much a point that violates the margin (see Figure 7) is tolerated. Depending on the Kernel, additional parameter might be necessary, e.g. the grade, scalar, and rest term of a polynomial Kernel or the width Sigma ($\sigma \in \mathbb{R}^{>0}$) of the Gauss form of a Radial-Base-Function (RBF) Kernel. The research at hand used the SVM implementation of (Platt, 1999), respectively the improvement of (Keerthi, Shevade, Bhattacharyya, & Murthy, 2001). At first, a linear Kernel is used, because it only has the penalty as parameter and is therefore easy and fast to use. The results at various penalty values lay in the MCC range of 0.140 to 0.181 with a standard deviation below 0.01. The bad performance is not unexpected considering that the classes are hardly linearly separable. A RBF Kernel promises better results although the costly transformation makes the systematic parameter search a very time consuming task. To reduce this effect, a raster search for penalty and sigma is performed following the suggestions of (Hsu, Chang, & Lin, 2003) as well as (Min & Lee, 2005).

How much the performance depends on the parameter is clearly

displayed in Figure 10 and corresponding Table 4. Certain combinations return a random classification ($MCC = 0.0$) while others are almost perfect ($MCC = 1.0$). A zoom into a finer grid (see Figure 11) shows a maximum performance at a penalty value of 128 and sigma of 0.2. With a $MCC = 0.975$ and a standard deviation below 0.0019, the RBF clearly outperforms aNN. Also, the investigation of a polynomial or sigmoid Kernel are obsolete, especially because both have more parameters and often show worse results (shown in (Hsu et al., 2003, p. 4), (Min & Lee, 2005, p. 610), (Ben-Hur & Weston, 2010, p. 4)). Therefore, only the SVM with an RBF Kernel, a penalty of 128, and sigma of 0.2 will be given a closer look at the performance of predicting the RUL-classes.
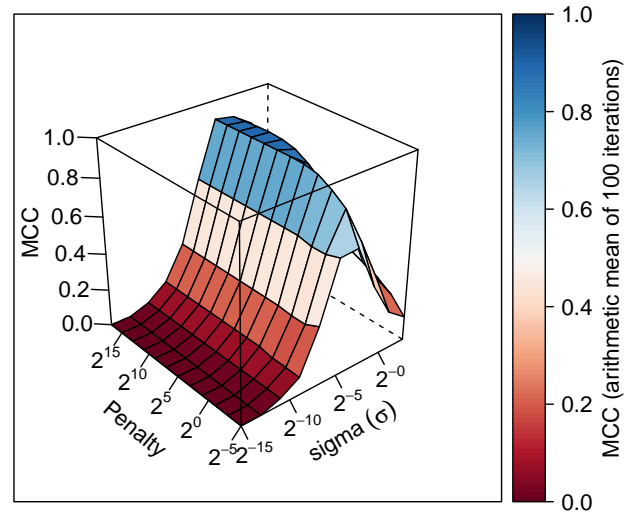

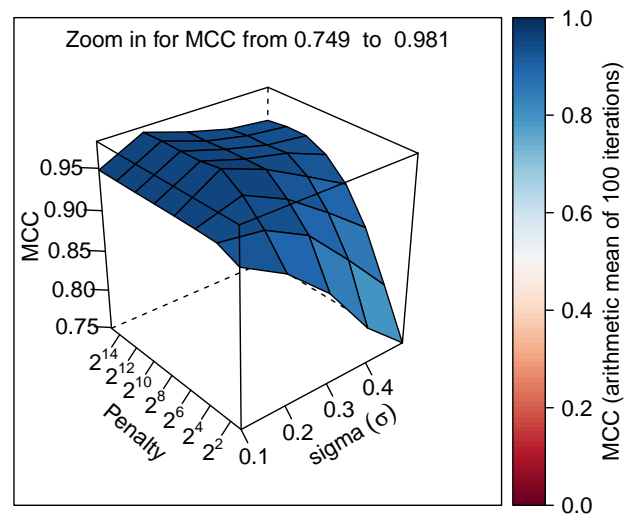Figure 10. Prediction performance of SVM with RBF Kernel at various parameters

Figure 11. Prediction performance of SVM at an alternate class structure

Table 4. MCC as metric of the SVM prediction performance at various parameters of a RBF Kernel

Penalty	sigma														
	2^{-15}	2^{-13}	2^{-11}	2^{-9}	2^{-7}	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}	0.1	0.2	0.3	0.4	0.5
2^{-5}	0.00	0.00	0.04	0.10	0.32	0.62	0.63	0.39	0.20	0.13	-	-	-	-	-
2^{-3}	0.00	0.00	0.04	0.10	0.28	0.60	0.79	0.58	0.34	0.21	-	-	-	-	-
2^{-1}	0.00	0.00	0.04	0.10	0.28	0.60	0.84	0.65	0.39	0.26	-	-	-	-	-
2^{1}	0.00	0.00	0.04	0.11	0.29	0.62	0.88	0.72	0.42	0.30	0.937	0.907	0.861	0.794	0.749
2^{3}	0.00	0.00	0.04	0.11	0.29	0.62	0.90	0.78	0.47	0.33	0.948	0.940	0.914	0.863	0.809
2^{5}	0.00	0.00	0.04	0.11	0.29	0.62	0.90	0.83	0.52	0.36	0.948	0.965	0.941	0.906	0.867
2^{7}	0.00	0.00	0.04	0.11	0.29	0.62	0.90	0.87	0.59	0.39	0.948	0.981	0.962	0.935	0.908
2^{9}	0.00	0.00	0.04	0.11	0.29	0.62	0.90	0.89	0.66	0.41	0.948	0.977	0.961	0.951	0.931
2^{11}	0.00	0.00	0.04	0.11	0.29	0.62	0.90	0.90	0.69	0.45	0.948	0.976	0.961	0.948	0.943
2^{13}	0.00	0.00	0.04	0.11	0.29	0.62	0.90	0.89	0.74	0.50	0.948	0.976	0.958	0.948	0.943
2^{15}	0.00	0.00	0.04	0.11	0.29	0.62	0.90	0.89	0.79	0.54	0.948	0.974	0.958	0.945	0.940
2^{17}	0.00	0.00	0.04	0.11	0.29	0.62	0.90	0.88	0.82	0.58	-	-	-	-	-

5.2. SVM as favourable predictor

The excellent results of the SVM with RBF Kernel in the previous section arise the question how good the classifications of the individual RUL-classes are. To answer this question, the confusion matrix is visualised in Figure 12 and in Table 5 accordingly. They show the ratio of the real class each cell has. Along the diagonal line, the true positive rate (TP_r) is displayed. It represented the ratio of correct predictions within the corresponding real class. All other cells contain a misclassification.
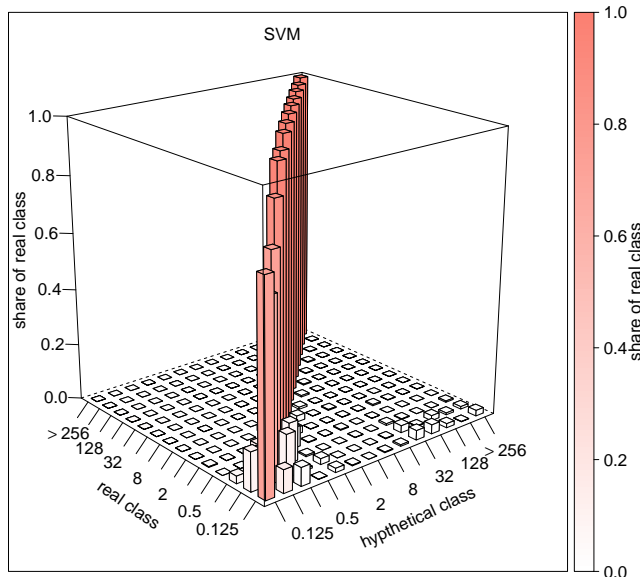

Figure 12. Confusion matrix of the prediction with a SVM shown as share of real RUL-class

A few things shall be pointed out from Figure 12:

- There are rarely misclassifications of classes with long time differences because only very small numbers exist offside the diagonal.

- The Classes z_1 to z_4 are less precise than others. In this area, with RUL values below 0.5 days, the number of samples is small and the overlapping is high. The good thing is that the misclassification mainly stays within these small classes.

This leads to the conclusion that an improvement can be reached with an alternate class structure in which the smaller classes are joined. A few alternatives were defined to investigate the effect of different RUL-class structures. Some structures joined small classes, some are defined with equal days distance instead of geometric while others consist of only four classes. Due to limited space in this paper, only one especially interesting candidate is elaborated.

The alternate structure joins the smaller classes but no longer follows a mathematically simple sequence. Moreover, it is oriented to calendar periods as well as the lead time required by the maintenance staff. The interval boundaries are manually chosen at $\{0.125, 1, 3, 5, 14, 28, 61, 122, 244, \infty\}$ days. This provides 3 hours for an emergency fault clearance, one full day to initialise an activity, three to five days for timely planning, two and four full weeks and month after that. While the $MCC = 0.989$ is only a slight improvement compared to previous results, the confusion matrix in Table 6 and its visualisation in Figure 13 show a clearer picture. All but the smallest class have a very high TP_r, which makes the prediction really successful.

6. CONCLUSION

The paper presented an approach to predict the Remaining Useful Life of railway switches. The data driven prediction was based on electric measurements of the engines (point machine) of 29 switches of Deutsche Bahn and combined

Table 5. Confusion matrix of the prediction with a SVM shown as share of real RUL-class, class structure 2^i

Real class	Hypothetical class [d]													
	0.625	0.125	0.25	0.5	1	2	4	8	16	32	64	128	256	> 256
0.0625	0.724	0.081	0.065	0.001	0.015	0.003	0.000	0.003	0.005	0.034	0.032	0.015	0.001	0.020
0.125	0.136	0.623	0.150	0.007	0.020	0.000	0.000	0.000	0.002	0.025	0.002	0.018	0.007	0.009
0.25	0.026	0.084	0.730	0.140	0.005	0.000	0.000	0.001	0.000	0.006	0.002	0.000	0.000	0.005
0.5	0.000	0.001	0.062	0.858	0.072	0.002	0.001	0.003	0.000	0.001	0.000	0.000	0.000	0.000
1	0.003	0.001	0.001	0.015	0.946	0.024	0.001	0.000	0.001	0.001	0.005	0.002	0.000	0.001
2	0.000	0.000	0.000	0.000	0.021	0.950	0.022	0.001	0.003	0.000	0.001	0.000	0.001	0.001
4	0.001	0.000	0.000	0.000	0.000	0.010	0.979	0.007	0.001	0.000	0.001	0.000	0.000	0.001
8	0.000	0.000	0.000	0.000	0.000	0.000	0.006	0.986	0.005	0.001	0.001	0.000	0.000	0.000
16	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.991	0.004	0.001	0.000	0.000	0.000
32	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.994	0.002	0.000	0.000	0.000
64	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.994	0.003	0.000	0.000
128	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.995	0.002	0.000
256	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.995	0.002
> 256	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.999

Table 6. Confusion matrix of the prediction with a SVM shown as share of real RUL-class, class structure manual

| Real class	Hypothetical class [d]									
	0.125	1	3	5	14	28	61	122	244	∞
0.125	0.824	0.088	0.000	0.000	0.004	0.021	0.034	0.017	0.000	0.013
1	0.024	0.952	0.017	0.000	0.000	0.001	0.004	0.003	0.000	0.000
3	0.000	0.012	0.976	0.008	0.000	0.001	0.001	0.000	0.001	0.001
5	0.000	0.000	0.015	0.974	0.010	0.000	0.001	0.000	0.000	0.000
14	0.000	0.000	0.000	0.002	0.993	0.002	0.001	0.000	0.000	0.000
28	0.001	0.000	0.000	0.000	0.002	0.994	0.003	0.000	0.000	0.000
61	0.000	0.000	0.000	0.000	0.001	0.001	0.995	0.002	0.000	0.000
122	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.995	0.002	0.000
244	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.995	0.002
∞	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.999
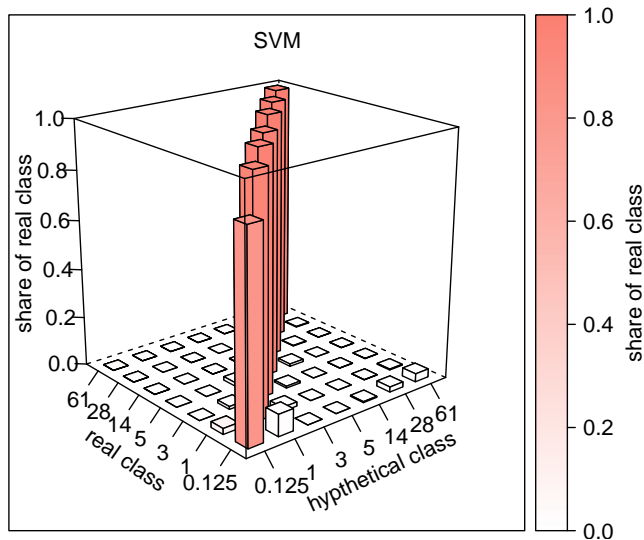
Figure 13. Confusion matrix of the prediction with a SVM shown as share of real RUL-class

with additional data about temperature, humidity, and switch design characteristics. Instead of a continuous prediction, the RUL was binned in intervals and therefore the prediction was transformed into a classification problem of multiple discrete classes. The structure of the classes (the binning intervals) followed the required lead time of maintenance to timely plan and execute activities. This transformation does not only make the prediction itself easier but also reduces uncertainty and increases user acceptance. The documentation of failures was used as reference for supervised learning with artificial Neural Networks and Support Vector Machines. Since finding the right parameter is crucial to the success, these were identified systematically. During this process, SVM with an RBF Kernel clearly outperformed the aNN (Multi-Layer-Feed-Forward network). Applied to the 98 documented failures and the roughly 245000 measurements, it was possible to at least predict 50 of them correctly. So to speak, 50% of failures could be prevented.

Though, the research revealed a few good guidelines about how to structure the classes and select intervals that consider maintenance requirements. Future work should further investigate this with direct feedback about experience with the prediction. Additionally, other techniques of classification or unsupervised learning should be applied to the prediction. Nevertheless, the biggest challenge is to test the results at a much bigger data base. Even with the data based used here, the instances of a certain switch and failure type had been so few that it remains unclear if the results generalise good enough for a real life application. But ultimately, only a real life application can reduce maintenance cost and prevent failures.

NOMENCLATURE

A	Set of attributes selected as data base for the prediction
I	Set of prediction classes (real condition)
i	Index of prediction class (real condition)
J	Set of prediction classes (hypothetical condition)
j	Index of prediction class (hypothetical condition)
L	Set of all tuples within S
l	Index of tuple within S
$m)$	Index of attributes A
S	Set of all sample tuples for the prediction
$\mathbb{B}$	Set of all real numbers in the interval between 0 and the maximum Remaining Useful Life $maxRUL$
b	Single value of the RUL
f	An arbitrary function
HZ	Hypothetical condition
MCC	Metric Matthews Correlation Coefficient
o	additional index of the confusion matrix $\mathbf{C}$ to calculate the MCC
p	additional index of the confusion matrix $\mathbf{C}$ to calculate the MCC
q	additional index of the confusion matrix $\mathbf{C}$ to calculate the MCC
TZ	Real condition
Z	Set of classes, which represent the RUL and hence the condition

REFERENCES

Asada, T., Roberts, C., & Koseki, T. (2013). An algorithm for improved performance of railway condition monitoring equipment: Alternating-current point machine case study. *Transportation Research Part C: Emerging Technologies*, *30*(0), 81–92.

Atamuradov, V., Camci, F., Baskan, S., & Sevkli, M. (2009). Failure diagnostics for railway point machines using expert systems. In *Ieee international symposium on diagnostics for electric machines, power electronics and drives, 2009* (pp. 1–5). Piscataway (NJ USA): IEEE.

Bai, H. (2010). *A generic fault detection and diagnosis approach for pneumatic and electric driven railway assets* (Unpublished doctoral dissertation). University of Birmingham, Birmingham.

Ben-Hur, A., & Weston, J. (2010). A user's guide to support vector machines. In O. Carugo & F. Eisenhaber (Eds.), *Data mining techniques for the life sciences* (Vol. 609, pp. 223–239). New York (NY USA): Humana Press. Retrieved from http://dx.doi.org/10.1007/978-1-60327-241-4_13 doi: 10.1007/978-1-60327-241-4_13

Böhm, T. (2012). Accuracy improvement of condition diagnosis of railway switches via external data integration. In C. Boller (Ed.), *Structural health monitoring 2012* (pp. 1550–1558). Germany.

Böhm, T. (2015). Präzise vorhersage von weichenstörungen. *EI - Eisenbahningenieur*, *66.*(10), 50–53.

Böhm, T., & Doegen, C. (2010). Diagnosis without sensors - integration of external data for condition monitoring of railway switches. In S. Okumura, T. Kawai, P. Chen, & R. B. Rao (Eds.), *Comadem 2010 - advances in maintenance and condition diagnosis technologies towards sustainable society* (pp. 619–622). Hikone (Japan): Sunrise Publishing.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*, 121–167.

Chamroukhi, F., Samé, A., Aknin, P., & Antoni, M. (2008). Switch mechanism diagnosis using a pattern recognition approach. In *The 4th iet international conference on railway condition monitoring* (pp. 1–4). Derby (UK).

Cocciaglia, D. (2012). Case study: Switch&crossing diagnostics. In London Business Conferences (Ed.), *Cost optimisation track maintenance and renewal congress 2012.*

DB Netz AG. (April 2016). *Infrastrukturzustands- und -entwicklungsbericht 2015: Internetversion.* Frankfurt am Main. Retrieved 15.10.2016, from http://www.eba.bund.de/SharedDocs/Publikationen/DE/Finanzierung/IZB/IZB_2015.html;jsessionid=35B3BE62A8AA227E32055FF48043F670.live11292?nn=491736

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification* (2. Aufl. ed.). s.l.: Wiley-Interscience.

Eker, O. F., Camci, F., & Kumar, U. (2010). Failure diagnostics on railway turnout systems using support vector machines. In U. Kumar (Ed.), *emaintenance 2010* (pp. 248–251). Lulea (Sweden): Univ.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*(1), 27–38. Retrieved from http://www.sciencedirect.com/science/article/pii/S0167865508002687

García Márquez, F. P., Weston, P., & Roberts, C. (2007). Failure analysis and diagnostics for railway trackside equipment. *Engineering Failure Analysis*, *14*(8), 1411–1426. Retrieved from http://www.sciencedirect.com/science/article/pii/S1350630707000556

Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge (Mass. USA): MIT Press. Retrieved from http://www.gbv.de/dms/bowker/toc/9780262082907.pdf

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). *A practical guide to support vector classification.* Retrieved from http://www.csie.ntu.edu.tw/cjlin/papers.html

Jurman, G., & Furlanello, C. (2010). A unifying view for performance measures in multi-class prediction. *ArXiv e-prints*(Aug), 1–5.

Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (2001). Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, *13*(3), 637–649. doi: 10.1162/089976601300014493

Leao, B. P., Yoneyama, T., Rocha, G. C., & Fitzgibbon, K. T. (2008). Prognostics performance metrics and their relation to requirements, design, verification and cost-benefit. In *International conference on prognostics and health management, 2008* (pp. 1–8). Piscataway (NJ USA): IEEE.

Lee, J. (2010). Design of selfmaintenance and engineering immune systems for smarter machines and manufacturing systems. In S. Okumura, T. Kawai, P. Chen, & R. B. Rao (Eds.), *Comadem 2010 - advances in maintenance and condition diagnosis technologies towards sustainable society* (pp. 1–13). Hikone (Japan): Sunrise Publishing.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et biophysica acta*, *405*(2), 442–451. Retrieved from http://view.ncbi.nlm.nih.gov/pubmed/1180967

Min, J. H., & Lee, Y.-C. (2005). Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Systems with Applications*, *28*(4), 603–614. doi: 10.1016/j.eswa.2004.12.008

Pedregal, D. J., García Márquez, F. P., & Roberts, C. (2009). An algorithmic approach for maintenance management based on advanced state space systems and harmonic regressions. *Annals of Operations Research*, *166*(1), 109–124.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization: Advances in kernel methods. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods* (pp. 185–208). Cambridge (Mass. USA): MIT Press. Retrieved from http://dl.acm.org/citation.cfm?id=299094.299105

Pool, C., & Vlek, J. (2016). Making the switch: Predictive maintenance on railway switches. In databricks (Ed.), *Spark summit europe 2016.* Retrieved 23.04.2017, from https://youtu.be/HEDMET3K6nA

Rama, D., & Andrews, J. D. (2013). A reliability analysis of railway switches. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, *227*(4), 344–363. doi: 10.1177/0954409713481725

Rausch, W. (2010). Diagnosesysteme für weichen als grundlage für eine optimierte instandhaltungsplanung. In VDEI (Ed.), *Symposium zum*

thema: „moderne instandhaltungsverfahren für weichen – qualitätsansprüche – wirtschaftlichkeit“. Brandenburg-Kirchmöser.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: therprop algorithm. In *Proceedings of the ieee international conference on neural networks* (pp. 586–591). Piscataway (NJ USA).

Sebastiani, P., Abad, M. M., & Ramoni, M. F. (2010). Bayesian networks. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 175–208). Boston (MA USA): Springer Science+Business Media LLC.

Shmilovici, A. (2010). Support vector machines. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 231–247). Boston (MA USA): Springer Science+Business Media LLC.

Silmon, J. A. (2009). *Operational industrial fault detection and diagnosis: Railway actuator case studies* (Unpublished doctoral dissertation). University of Birmingham, Birmingham.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437.

Stoll, H., & Bollrath, B. (2002). Weichendiagnosesystem sidis w. *Signal + Draht*, *94*(4), 26–29.

University of Birmingham. (2008). *Innotrack (fp6 eu project): Deliverable d3.3.1 – list of key parameters for switch and crossing monitoring.* Brüssel.

Vapnik, V. N. (2008). *The nature of statistical learning theory* (2. ed., 6. print ed.). New York: Springer. Retrieved from http://www.loc.gov/catdir/enhancements/fy0816/99039803-d.html/http://www.loc.gov/catdir/enhancements/fy0816/99039803-t.html/http://www.gbv.de/dms/hbz/toc/ht012773848.pdf

Zhang, P. G. (2010). Neural networks for data mining. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 419–444). Boston (MA USA): Springer Science+Business Media LLC.

Zwanenburg, W.-J. (2009). *Degradation processes of railway switches & crossings: To improve maintenance & renewal planning on the swiss railway network*. Saarbrücken: Südwestdeutscher Verlag für Hochschulschriften.

BIOGRAPHIES

Thomas Böhm (Boehm) studied at University of Magdeburg in Germany and the EPFL Lausanne in Switzerland. He holds a Master in Computer Science in Engineering and has joined the DLR (German Aerospace Center) Institute of Transportations Systems in 2007. Since then, his main field of research is condition based maintenance and health management using statistics and data mining. In 2010, he became leader of the Life Cycle Management team with whom he further advanced the topic of efficient, predictive maintenance of railway infrastructure. He has several publications in this field and is session chairman of the International Congress of Condition Monitoring and Diagnostic Engineering Management. He worked in and led several international and national projects as well as development projects for diagnostic and prognostic algorithms for Deutsche Bahn and its asset monitoring platform DIANA. In 2015, he became Chief Officer and Head of the Department Data Management and Knowledge Discovery. In this role, he was responsible for sensor system development, data management, monitoring, and situation interpretation for road traffic and autonomous driving as well as railway system. Since 2017, he is the Chief Data Scientist at KONUX a company focusing on analytics and predictive maintenance.