Improving rail network velocity: A machine learning approach to predictive maintenance

Hongfei Li a,⇑, Dhaivat Parikh b, Qing He c, Buyue Qian a, Zhiguo Li a, Dongping Fang a, Arun Hampapur a

a IBM T.J. Watson Research Center, Yorktown Heights, NY, United States
b IBM Global Business Services, Dallas, TX, United States
c The State University of New York at Buffalo, Buffalo, NY, United States

Article info
Article history:
Received 8 May 2013
Received in revised form 31 March 2014
Accepted 21 April 2014
Available online 16 May 2014

Keywords:
Big data
Condition based maintenance
Multiple wayside detectors
Information fusion
Predictive modeling
Rail network velocity

Abstract
Rail network velocity is defined as system-wide average speed of line-haul movement between terminals. To accommodate increased service demand and load on rail networks, increase in network velocity, without compromising safety, is required. Among many determinants of overall network velocity, a key driver is service interruption, including lowered operating speed due to track/train condition and delays caused by derailments. Railroads have put significant infrastructure and inspection programs in place to avoid service interruptions. One of the key measures is an extensive network of wayside mechanical condition detectors (temperature, strain, vision, infrared, weight, impact, etc.) that monitor the rolling-stock as it passes by. The detectors are designed to alert for conditions that either violate regulations set by governmental rail safety agencies or deteriorating rolling-stock conditions as determined by the railroad.

Using huge volumes of historical detector data, in combination with failure data, maintenance action data, inspection schedule data, train type data and weather data, we are exploring several analytical approaches including, correlation analysis, causal analysis, time series analysis and machine learning techniques to automatically learn rules and build failure prediction models. These models will be applied against both historical and real-time data to predict conditions leading to failure in the future, thus avoiding service interruptions and increasing network velocity. Additionally, the analytics and models can also be used for detecting root cause of several failure modes and wear rate of components, which, while do not directly address network velocity, can be proactively used by maintenance organizations to optimize trade-offs related to maintenance schedule, costs and shop capacity. As part of our effort, we explore several avenues to machine learning techniques including distributed learning and hierarchical analytical approaches.

1. Introduction

Rail network operators across the world are seeing an increase in demand for services driven by increased global trade and increasing cost of fuel. Accommodating this increased load on relatively fixed rail networks requires increase in network velocity without compromising safety. Network velocity is defined as system-wide average speed of line-haul movement...
between terminals and is calculated by dividing total train-miles by total hours operated. Higher network velocity represents efficient utilization of capital-intensive assets, and it is one of the most important metrics to measure performance of a railroad. Among many determinants of overall network velocity, service interruption is a key driver, which includes lowered operating speed due to track/train condition and delays caused by derailments. Railroads have put significant infrastructure and inspection programs in place to avoid service interruptions. One of the key measures is an extensive network of rolling stock or train monitoring detectors. Each of these detector systems consists of multiple sensors (temperature, strain, vision, infrared, weight, impact, etc.) and processing software and hardware. Detectors are installed along rail tracks and inspect the rail cars and locomotives passing over them to monitor and detect equipment health conditions (Ouyang et al., 2009). The detectors are primarily designed to alert for conditions that violate regulations set by government rail safety agencies.

Reducing the number of derailments attributed to mechanical faults in car and locomotive as primary cause and reducing intermediate maintenance calls due to false alarms can significantly improve rail network velocity. The extensive sensor network provides information sources to enable the solutions. One approach is to use machine learning techniques to automatically “learn rules” from historical sensor measurements and then better predict which rail cars are likely to have problems and thus maximize the hit rate of rail operator’s setouts. Machine learning techniques allow systematic classification of patterns or relationships contained in data and identification of the attributes, containing information about condition of physical assets that contribute associated failure mode, or class (Hastie et al., 2001).

Considering the complexity of sensor network, there are several challenges in developing machine-learning techniques for predictive maintenance in railway operations.

1. The first challenge is caused by spatio-temporal incompatible information collected through multiple detectors, which are not always co-located. The detector system consists of multiple detectors measuring temperature, strain, vision, weight/impact, etc. Existing systems issue alerts primarily using one or two types of detectors at a time and only partial information is used. For example, when the bearing temperature is above 170 °F, the system issues an alert to request an immediate train-stop. The rule is simple but does not consider measurement errors or impact of environmental variables on detectors, which may lead to high false alarms and lower hit rate of rail operator’s setouts. To better understand the conditions of a railcar, it is essential to integrate the information collected from various detectors. Since these detectors are not co-located, the measurements coming out of them are spatially and temporally incompatible, posing challenges when combining the information. We use bad truck/bad wheel prediction model as an example to show how we address this issue in Section 3.2.

2. The second challenge is big data. The ubiquity of connectivity and the growth of sensors have opened up a large storehouse of information. The bearing temperature detectors, for Class I railroad under this study, generate 3 terabytes of data in a year. Other industries offer possibilities of even larger amount of data generated under normal operating conditions, e.g., a Boeing jet generates 10 terabytes of information per engine every 30 min of flight-time (Higginbotham, 2010). The amount of data is only going to continue to rise. There has been a lot of recent progress in big data warehousing to manage, store and retrieve this information including Netezza and Teradata, but the true value is realized only if we are successful in mining and using the signals contained in the information effectively. In this paper, we will show a customized support vector machine (SVM) technique that effectively utilizes large-scale data and provides valuable tools for operational sustainability as described in the scenario of alarm prediction in Section 3.1.

3. The third challenge comes from the need to learn and create alarm rules in the context of industry operations, so that the rules generated can be interpreted by operators easily leading to efficient operational decision support. On one hand, subject matter experts (SME) can derive rules based on their knowledge and expertise in concert with industry standards. Those rules are easy to interpret, but do not accommodate the complexity required for accurate prediction based on large, spatially and temporally incompatible and dirty, heterogeneous data coming from multiple detectors. On the other hand, machine learning techniques provide good approaches to learn efficient rules to predict failures. But those rules are usually complicated and thus not easily understood by human operators. To facilitate predictive maintenance operations in railway, it is important to set proper trade-off in learning system to create efficient but human-interpretable rules. We address this issue by using two different approaches. One is to extract legalized rules from the complex machine-learning based algorithm outputs, and the other is to use certain machine learning techniques, such as decision tree, to derive rules that are easily to understand and implement. The approaches are described with more details in the two scenarios in Section 3.1 and 3.2, respectively.

In the area of condition monitoring and predictive maintenance, some work has been done to provide failure predictions using statistical and machine learning approaches. Lin and Tseng et al. (2005) presents reliability modeling to estimate machine failures. Saxena and Saad (2007) uses neural network classifier for condition monitoring of rotating mechanism systems. In railway applications, Yella et al. (2009) adopts a pattern recognition approach to classify the condition of the sleeper into classes (good or bad). Yang and Létourneau (2005) proposes an approach to predict train wheel failures but only using one type of detectors, Wheel Impact Load Detector (WILD), without considering the impacts of multiple detectors. Recently, Hajibabai et al. (2012) develops a logistic regression model to classify wheel failures based on WILD and Wheel Profile Detector (WPD). They claim that the classification accuracy is 90% with 10% false alarm rate. However, only two detectors are taken into account in that study. The problems that those papers have worked on are not as complicated as what we face and none of them has addressed all the challenges we describe above.
In this paper, we develop machine-learning approaches to predict impending failures and alarms of critical rail car components. The prediction will drive proactive inspections and repairs, reducing operational equipment failure. This work delivers significant gains in rail network velocity and safety. The rest of the paper is organized as follows. Section 2 gives the overview of sensor network operations in a US Class I railroad and various data sources provided. Section 3 describes the composite detector analysis, using two specific use cases as examples, including problem formulation, methodologies and results. Section 4 summarizes the work.

2. Overview of sensor network in a US Class I railroad

Railroad operators are keen to move from a reactive (respond to alerts from detectors and act) to a proactive stance around safety and network velocity. In this context, IBM works with a US Class I railroad to develop predictive failure analytics based on large-scale data from multiple-detector systems. The railroad manages about 20 K miles of tracks and has about 1000 detectors installed along their railway network. A huge amount of data has been collected over the past several years. For example, they manage about 800 hot box detectors (HBD) which comprises around 900 million temperature records collected from 4.5 million bearings just within 3 months in early 2011. They also have 90 hot wheel detectors (HWD), which contain 500 million records collected from 4.5 million wheels within 6 months in 2011. The above are examples of the scale of the data that needs to be analyzed in order to develop models. We expect the data volume to grow by a 100 fold as new detectors come online.

The detectors are installed along the wayside. When a rail car passes an equipped detector, the detector will report mechanical condition observations of the equipment or components including truck, wheel or bearing. Now we give a brief description of the detectors used in our analysis (Ameen, 2006). Hot Box Detector (HBD) is a pyrometer-based technology used to monitor the temperature of bearings and wheel (if equipped to do so). Wheel Impact Load Detector (WILD) is built into the track to detect defective wheels based on the impact they generate on the track structure. The system reports the features to describe dynamic impact load at wheel level, such as maximum force (KIP) and max lateral force (KIP). In addition, the detector collects observations at car level, such as speed and weight. Truck Performance Detectors (TPD) are usually built into S curves of the track to monitor performance of trucks (the assemblies that hold railcar wheels). The key feature collected through TPD is ratio of lateral force over vertical force. Both WILDs and TPDs use strain-gauge-based technologies to measure the performance of a freight car, its wheels and trucks in a dynamic mode, which tells us more about a car than a static inspection. Machine Vision (MV) technology uses computer algorithms to process digital image data of railcar underframes and side-frames into diagnostic information. The key wheel features captured by MV include flange height, flange thickness, rim thickness, diameter, etc. Optical Geometry Detector (OGD) uses angle of attack and tracking position to calculate the following features for each set of axles on the same truck – tracking error, wheel dimension and wheel impact load using multiple detectors, such as MV, OGD and WILD. It is a laser-based system with cameras mounted on tangent track. Acoustic Bearing Detectors (ABD) capture the noise signature emitted by bearing in motion and detector processes this information internally and issues alarms when anomalous acoustic signature is detected.

In addition to the detector data, the railroad also stores historical equipment failures, maintenance records and alarms issued in recent past. That information gets connected to the detector readings to give a whole picture of the cause and effect.

3. Composite-detector analysis

The goal of this work is to predict equipment failures in advance and improve network velocity by reducing derailments or at minimum reducing intermediate maintenance calls due to false alarms. We achieve this by developing models to create “learned rules” automatically from historical data to better predict which rail cars are more likely to have problems and to predict the most severe existing alarms in advance of the actual alarm event to reduce immediate train stops and service interruptions. Four major pieces of analytics work are listed:

1. Alarm prediction: predict the alarms alerting catastrophic failures caused by hot bearings 7 days in advance of actual alarm event using multiple detectors, such as HBD, ABD and WILD. The significance of alarm prediction is to reduce the immediate train stops and service interruption.
2. Bad truck prediction: detect truck performance issues earlier due to wear-out by identifying patterns in wheel movement error, wheel dimension and wheel impact load using multiple detectors, such as MV, OGD, TPD and WILD.
3. Bad wheel prediction: predict wheel defects earlier by identifying abnormal patterns in wheel dimensions, movement errors and wheel impact load using multiple detectors, such as MV, OGD and WILD.
4. Asymmetric wheel wear detection: detect one particular wheel defect, asymmetrically wearing wheels, which could lead to a shorter wheel lifespan and can also lead to truck performance issues, using MV and OGD.

In Sections 3.1 and 3.2, we describe in more details two of the above four applications – alarm prediction and bad truck/bad wheel failure prediction. These two use cases are good examples to show how to apply machine learning approaches in railroad predictive maintenance applications from different perspectives. Alarm prediction focuses more on how to handle large-scale data and extract human interpretable rules from complex machine learning algorithm outputs. We present bad
truck/bad wheel failure prediction to illustrate a way to combine information from multiple detectors and different data sources (even though this challenge also applies to the use case of alarm prediction) and demonstrate how to generate relatively simple rules for operation purpose by careful selection of machine learning techniques.

3.1. Alarm prediction

3.1.1. Background

The railroad is making efforts to transitioning from reactive maintenance to proactive maintenance. The existing alarm systems at the Class I railroad or in fact, at any reasonably efficient maintenance organization already catches the most catastrophic component failures before they happen, using their monitoring systems. But this alarm of catastrophic failure, called L1 alarm, is usually generated when the component failure is imminent, leaving little time for planning and thus putting the maintenance organization in reactive mode. One example of current operational alarms is that when HBD detects that bearing temperature has reached 170 °F degree, L1 alarm is issued and immediate train stop is required. The Class I railroad issues more than 1000 L1 alarms each year on average. L1 alarms cost significantly as they have big impact on network velocity. Unlike traditional condition-based monitoring in asset management (Jardine et al., 2006; Niu et al., 2010), we do not aim to change the current L1 alarm rules and improve their accuracy. Instead, we apply machine learning techniques and create new rules to predict the L1 alarms in advance so that operators have sufficient time to act on our L1 alarm prediction and avoid immediate service interruptions caused by L1 alarms. To our knowledge, no previous work has been done to address the problem of alarm prediction. In addition, our approach takes into account of inventory, shop capacity and planned used of the asset and results in higher network velocity and efficient utilization of scarce maintenance resources.

There are two challenges we need to address when developing prediction model. One is the requirement of extremely low false alarm rate. Under limited budget, the railroad has constrained resources for maintenance operations. They require false alarm rate less than 0.014% (which translates into less than 21 out of 150,000 cars in daily operations). The other challenge is to generate human interpretable rules for operation and maintenance groups to facilitate their decision process. Our approach addresses these two issues primarily by combining two techniques. One is to develop customized SVM classification model by tuning several controlled parameters to achieve the low false alarm rate. The other is to use combined information from multiple detectors (e.g., HBD, ABD and WILD) to increase prediction accuracy.

HBD is used to monitor the temperature of bearings. There are more than 800 HBDs in the network and around 150,000 cars running each day on average. Data size for one-year HBD readings are about 3 TB. WILD system reports the features that describe dynamic wheel impact load, such as maximum peak force (KIPS), maximum lateral force (KIPS) and their ratios. ABD tables are composed of alarm event data such as the acoustic alarms for different bearing defect related noise signatures. We merge the three types of detector readings along with bearing alarms by bearing id and specified time window. Given a L1 hot bearing alarm, we identify the bearing id and date associated with the alarm. Then search the readings from HBD, WILD and ABD of the same bearing and corresponding historical time window. This procedure poses some challenges. There are only a dozen of WILD and ABD detectors in the entire railway network. As a result, we have much fewer observations from WILD and ABD than those from HBD. On average, there are 20 to 30 observations per wheel in a year from WILD. Since ABD only keeps alarm events, the records are even sparse. That means if we choose a fixed same historical reading window for those three detectors, such as past 14 days, it is very likely that no records can be extracted from WILD or ABD for the same bearing in the pre-determined time window. To overcome this challenge, we choose different historical reading windows for these three detectors, i.e., we choose 7 or 14 days historical readings for HBD and extend the reading time window to 1 month for WILD and 3 months for ABD.

Following the above discussion of how to choose historical reading window, now we describe the two sets of control parameters that can be customized in the model development. The first set of parameters is the time window for historical detector readings. It indicates how many days of past detector readings should be used to provide the forecast. Considering real-time data storage capability in detector database, we give two options here, 7 days and 14 days of historic HBD reading window. The second is the prediction time window that is how many days in advance the alarm prediction is generated. Based on the trade-offs between operational constraint and prediction accuracy, we offer predictions 3 and 7 days in advance that give sufficient buffer time for operators to prepare for inspections. By combining the two sets of parameters, we now have four settings, i.e., 7-7, 7-3, 14-7, and 14-3 for short. The first number shows the reading time window and the second one the prediction time window. For example, 7-3 means using the past 7 days of readings from current day, we can provide alarm prediction for day 3 in the future.

3.1.2. Model development

The model development consists of five major steps, i.e., feature extraction, dimension reduction, model training, prediction and confidence estimation and rule simplification. We now describe each step in details (Li et al., 2013).

3.1.2.1. Feature extraction. In this step, we combine and aggregate historical multi-detector readings including ABD, HBD and WILD and extract features using aggregated statistics, such as quantile, for each numeric value variable. For example, we calculate maximum, 95 percentile and mean of bearing temperatures in the historical reading time window. In addition, we take the variation and trending of bearing temperatures over time as features. The model shows that the features obtained from ABD do not have significant impact on prediction. This can be because that ABD data is noisy. In total, we have
55 features extracted from WILD and HBD measurements. In order to process the 1.5 TB amount of data over half a year in this step, we use two techniques, hashing and parallelization, to accelerate table merging and feature extraction (Li et al., 2013).

### 3.1.2.2. Dimension reduction.
For efficient model development, we reduce the 55 feature dimensions using principal component analysis (PCA) (Jolliffe, 1986). The basic idea is to maximize the variance over a set of instances. In our study, according to the results from cross validation, we keep 12 features (projected using the eigenvectors) for HBD and WILD.

### 3.1.2.3. Model training.
The goal is to predict whether a bearing will be alarmed at Level 1 in the next 3 or 7 days. We use a variant of SVM (support vector machine) (Cortes and Vapnik, 1995) to solve this classification problem. We start with a linear classifier. For each alarm or non-alarm record, $i$, $x_i$ is a feature description, such as bearing temperature and corresponding maximum wheel impact load; and $y_i$ is the class label, i.e., alarm and non-alarm. The formulation is as shown in Eq. (1).

$$\begin{align*}
\min_{w, \xi} & \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \\
\text{s.t.} & \quad y_i (w^T x_i + b) \geq \rho - \xi_i \\
& \quad \xi_i \geq 0 \\
& \quad \rho \geq 0
\end{align*}$$  
\tag{1}

where $w$ defines a hyperplane that maximizes the margin between the two classes. We introduce a non-negative “soft margin” slack variable parameter $\xi$ that allows the classification result on the training set to be slightly distant to (on the wrong side of) the given labels. $C$ is a parameter that controls the trade-off between the “margin” and the training error (in terms of misclassified labels). The first constraint in Eq. (1) enforces the penalty for misclassification, which pushes labeled instances to the correct side of the classifying hyperplane. Because of the requirement of achieving very low false positive rate, we add in another parameter $\rho$ to reduce the false positive samples. In general, a large $\rho$ produces less positive predictions, while a small $\rho$ leads to more positives.

After applying Lagrange method (Fletcher, 1987), we can rewrite the optimization problem in Eq. (1) as

$$\begin{align*}
\min_{w, \xi} \max_{\alpha} & \quad \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) - \rho + \xi_i) \\
\text{s.t.} & \quad \xi_i \geq 0 \\
& \quad \alpha_i \geq 0 \\
& \quad \rho \geq 0
\end{align*}$$  
\tag{2}

Taking the first derivative of function (2) with respect to $w$ and $b$ and setting them equal to 0, we obtain two equations. Plugging them back into Eq. (2), we can derive a dual formulation from the original primal optimization problem as

$$\begin{align*}
\min_{\alpha} & \quad \rho \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\
\text{s.t.} & \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \\
& \quad 0 \leq \alpha_i \leq C \\
& \quad \rho \geq 0
\end{align*}$$  
\tag{3}

In order to obtain desirable accuracy, we use a mapping function to project the original feature vectors to higher dimensional space where more accurate classification results can be achieved. The kernel function characterizes a notion of pairwise similarity between instances, without explicitly providing a mapping function. Here we apply the kernel function, RBF (Radial Basis Function), which is defined as

$$K_{ij} = e^{-\frac{(x_i-x_j)^2}{\sigma^2}}$$  
\tag{4}

where $\sigma$ is the length scale of the RBF kernel used to control the complexity of prediction model.

In summary, in model training, first we calculate the kernel matrix using feature vectors in Eq. (4). Then substitute the kernel to Eq. (3) and get the optimal $a$. Optimal $b$ is obtained by solving the equation $w^T x_i + b - 1 = 0$. These two parameters will be used for prediction in Eq. (5). Detailed discussion can be found in Li et al., 2013.

To evaluate the accuracy of the proposed model, we perform a fivefold cross validation using real data. In each trial, we randomly divide the data into five groups. One is used for training and the other four are used to test. We repeat this process by rotating the training group. Since the classification problem is imbalanced, the large proportion of negative case or normal railcars, dominates the learning accuracy/error. As a result, we measure both the true positive rate (TPR) and false positive rate (FPR) in Section 3.1.3, which naturally provide a trade-off that needs to be balanced in practice.
3.1.2.4. Prediction and confidence estimation. This step is to predict if a bearing will issue a L1 alarm in the next few days based on its “location” (in feature space) at the “support vectors” (the key samples that lie in the border area between positives and negatives). For any new observations, \( x' \), the predicted label can be expressed as

\[
y' = \text{sign} \left( \sum_{i=1}^{n} a_i y_i K(x, x_i) + b \right)
\]

In addition, we also estimate the confidence level of the alarm prediction based on the relative position to the support vectors.

3.1.2.5. Rule simplification. To make the classification model outputs human interpretable, we perform an exhaustive grid search in the entire feasible feature space and divide the feature space into non-overlapping small grids. In this way, we create a rule set by linearizing the separating hyperplane. Section 3.1.3 presents some examples.

3.1.3. Results

In this section we evaluate the performance of our proposed approach using decision tree as baseline technique. Decision tree is a rule-based classifier, which produces a decision model that is naturally interpretable. It is used to classify instances by sorting them based on feature values (Yuan and Shaw, 1995). However, it may not be valid in practice and as a result, it limits the applicability to real problems. We now compare the results of decision tree and our proposed SVM model using same data inputs. The comparison shows how our approach achieves significantly better performance than decision tree. The results explain why we choose customized SVM model in our context when the railroad requires significantly low false alarm rate.

To evaluate the performance of the algorithms, we use two measures, true positive rate (sensitivity) and false positive rate (one minus specificity). True positive rate (TPR) or sensitivity, measures the proportion of actual positives which are correctly identified as such (e.g., the percentage of the alarms predicted by the algorithm over total number of alarms). Specificity measures the proportion of negatives which are correctly identified (e.g., the percentage of non-alarms that are correctly identified as not having the condition). In other words, false positive rate (FPR) measures the percentage of falsely identified alarms out of total number of non-alarms.

In order for comparison, we tune parameter sets to achieve different combinations of true positive rates (TPR) and false positive rates (FPR). The alarm prediction results of decision tree and SVM are shown in Tables 1 and 2, respectively. We use two settings, 7-7 and 14-3, and different scenarios for illustration.

Scenario I gives the case of the highest TPR that the two methods can achieve. It is apparent that for Scenario I in both tables (Tables 1 and 2), the SVM method outperforms decision tree by achieving higher TPR and lower FPR under both settings of 7-7 and 14-3. Comparatively, Scenario II presents the case when the lowest FPR can be achieved. It turns out that the FPR of SVM can go down to 0.000% for both settings (Table 2). That means nearly no false alarms, even though the TPR under this scenario is not high compared to Scenario I. However, from Table 1, we can see that the lowest FPR decision tree can achieve is around 1% which is much higher than the requirement from our client. Considering the huge amount of running cars each day and limited resources and budget available, the railroad requires the FPR to be as low as 0.014%. Obviously, decision tree cannot satisfy this strict constraint. This is the main reason we use customized SVM instead of decision tree for the alarm prediction application. In Table 2, we also give the result of TPR under Scenario III using SVM method when FPR is lower than 0.014%, our client’s requirement threshold. Under this situation, the TPR can still reach 38.5% for the setting of 7-7 and 45.4% for 14-3.

To make SVM model outputs human-interpretable, we perform an exhaustive grid search in the entire feasible feature space, which consists of all possible values that each sensor (such as HBD and WILD) can take. Specifically, as the first step, we divide the feature space into non-overlapping small grids that cover the entire feature space. In Fig. 1, we use 2-dimensional feature space for illustration. In the second step, we input the center point of each grid to the predicting model, and get a class assignment for that grid and its associated confidence level. After obtaining the class labels covering the entire grid, we are able to create a set of logical rules by combining the ranges of all dimensions of each grid-defining hyperplane.

Fig. 1 visualizes the results of alarm prediction. The horizontal axis is one derived feature, a representation of a linear combination of all measurement features coming from HBD. The vertical axis is the other derived feature, a linear combination from WILD features. To be specific, feature 1 on horizontal axis can be represented as \( \mathbf{w}^T \mathbf{x} \), where \( \mathbf{w} \) is the weight vector and \( \mathbf{x} \) is the feature vector from HBD. Since there are 33 features extracted from HBD, there is no much point to list all of

<table>
<thead>
<tr>
<th>Scenario</th>
<th>7-7</th>
<th>14-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>I: Highest true positive rate</td>
<td>91.546</td>
<td>92.568</td>
</tr>
<tr>
<td>II: Lowest false positive rate</td>
<td>61.256</td>
<td>68.463</td>
</tr>
</tbody>
</table>

Table 1

Decision tree results for the settings of 7-7 and 14-3 under two scenarios.
them. Instead, we select three important ones as examples. \(x_1\) is the accumulated value of bearing temperature when it is larger than 90°C; \(x_2\) is the number of \(K\) values larger than 2 where \(K\) value is a specific standardization of bearing temperature; \(x_3\) is the maximum \(K\) value during the historical reading time window. Correspondingly, \(a_1\) equals 0.35245, \(a_2\) is 0.0066 and \(a_3\) is 0.001402. Feature 2 on the vertical axis is represented as \(b_0y\), where \(b\) is the weight vector and \(y\) is the feature vector from WILD. For example, \(y_1\) is the 95 percentile of the equipment gross weight; \(y_2\) is 95 percentile of maximum peak wheel load and \(y_3\) is average dynamic maximum wheel load. Correspondingly, \(b_1\) equals 0.016389, \(b_2\) is 0.009667 and \(b_3\) is 0.004745.

**Fig. 1.** Performance comparison of granular rule logicalization.

### Table 2
Customized SVM results for the settings of 7-7 and 14-3 under three scenarios.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>7-7</th>
<th>14-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>I: Highest true positive rate</td>
<td>TPR (%) FPR (%)</td>
<td>TPR (%) FPR (%)</td>
</tr>
<tr>
<td>II: Lowest false positive rate</td>
<td>97.585 5.657</td>
<td>99.775 3.966</td>
</tr>
<tr>
<td>III: Highest true positive rate under constraint</td>
<td>7.459 0.000</td>
<td>8.987 0.000</td>
</tr>
<tr>
<td>of 0.014% false positive</td>
<td>38.542 0.014</td>
<td>45.368 0.012</td>
</tr>
</tbody>
</table>

Fig. 1(a) and (c) represent the decision maps. Blue points are the normal records or non-alarms while the red points are alarms. Correspondingly, the pink areas indicate alarm decision regions and blue areas are normal regions. One logical rule, for example, is when feature 1 is larger than 100 and feature 2 is larger than 6300, i.e., \(\alpha'x > 100\) and \(\beta'y > 6300\), then alarms.
The other example is when feature 1 is between -60 and 20 and feature 2 is between 6300 and 7200, i.e., $-60 < x' x < 20$ and $6300 < \beta y < 7200$, then an alarm is issued.

Now we compare the impact of granular sizes on decision rule logicalization. The results based on coarse and fine granular searches are shown on the top two plots and bottom two plots, respectively. Fig. 1(a) and (c) shows the rules as described above. Fig. 1(b) and (d) provide the probability maps that indicate the confidence level of each point being classified as an alarm. The color represents the probability. Red means high possibility being an alarm while blue means low possibility. In other words, blue means high possibility of being classified as non-alarm. With finer grids in bottom two plots, the rules become more complex, but with more accuracy, since the rules derived from finer grids represent the original SVM-based rules more closely.

3.2. Failure prediction

Unplanned equipment failures cost significantly to railway operations. Prediction of railcar defects ahead of time will give operators time inspecting the potential failures before they really occur. Among all defect types, wheel failures cause about one half of all train derailments (Salient Systems Inc., 2012). Rail car failure prediction has recently attracted significant attention along with increasing volume of data collected through wayside detectors. Yang and Létourneau (2005) propose an approach to predict train wheel failures before they disrupt operations. They claim that their study is the first application of using data mining techniques for prognostics of train wheel failures. However, they only use WILD detector in the study. In practice, measurements collected from different detectors are correlated and also essential for failure prediction. For example, a “bad” truck with severe defects will accelerate the wheel wearing process. Thus, a “bad” wheel can not only be detected by dynamic impact track load from WILD, but also by other types of detector signals, such as wheel profile vision data from MV detector, and truck geometry parameters from OGD.

In order to improve the prediction accuracy, we combine and integrate the information collected from several data sources, i.e., condition assessment data, historic failures, multi-detector observations, and equipment configurations. These diverse data sources are used to model failure risk and to discover hidden patterns. In the project, we provide bad truck and bad wheel predictions as separate tasks. But since the approaches are similar, we present the methods by combining them as one bad truck/bad wheel prediction model, which consists of five major steps: feature selection, data aggregation, bad truck/bad wheel labeling, prediction, and rule extraction. The modeling processes of the two prediction models, alarm prediction and bad wheel prediction, are similar. But the detailed procedures of each step vary a lot due to different concerns from the data and the challenges in modeling. We will show how to address various concerns in bad wheel/bad truck prediction in the following sections.

**Feature Selection.** In this step, we apply ANOVA test (Devore, 1995) to select key reading attributes from four different detectors: MV, OGD, WILD and TPD, as listed in Table 3.

**Data Aggregation.** About 500 GB of raw multi-detector data including MV, OGD, WILD and TPD are collected over the rail network from January 2010 to March 2012. Since the wayside detectors are not co-located and they are sparsely installed across the network, trains get different measurements when traveling through different routes. In addition, detector readings are highly influenced by various factors, such as truck load from internal status and temperature and snow from

<table>
<thead>
<tr>
<th>Detector type</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV</td>
<td>Wheel flange height</td>
</tr>
<tr>
<td></td>
<td>Wheel flange thickness</td>
</tr>
<tr>
<td></td>
<td>Wheel rim thickness</td>
</tr>
<tr>
<td></td>
<td>Wheel diameter</td>
</tr>
<tr>
<td></td>
<td>Wheel tread hollow</td>
</tr>
<tr>
<td></td>
<td>Brake shoe upper thickness</td>
</tr>
<tr>
<td></td>
<td>Brake shoe lower thickness</td>
</tr>
<tr>
<td>OGD</td>
<td>Truck hunting peak-to-peak (PTP) measurement</td>
</tr>
<tr>
<td></td>
<td>Truck hunting amplitude</td>
</tr>
<tr>
<td></td>
<td>Truck inter-axle misalignment (IAM)</td>
</tr>
<tr>
<td></td>
<td>Truck rotation measurement</td>
</tr>
<tr>
<td></td>
<td>Truck tracking error</td>
</tr>
<tr>
<td></td>
<td>Truck shift measurement</td>
</tr>
<tr>
<td>WILD</td>
<td>Wheel average downward load reading</td>
</tr>
<tr>
<td></td>
<td>Wheel peak downward load reading</td>
</tr>
<tr>
<td></td>
<td>Wheel average lateral load reading</td>
</tr>
<tr>
<td></td>
<td>Wheel peak lateral load reading</td>
</tr>
<tr>
<td></td>
<td>Difference between peak and average downward load reading</td>
</tr>
<tr>
<td></td>
<td>Truck hunting index</td>
</tr>
<tr>
<td>TPD</td>
<td>Ratio of later and vertical load</td>
</tr>
</tbody>
</table>
external environment. For example, in winter, wheel flange thickness readings from MV detector could be overestimated due to snow accumulated around wheels. Considering the irregularity and heterogeneity of multi-detector data, we aggregate all the readings in each month for each unique truck/wheel using specific summary statistics, such as mean, median, various quantiles, trending and variation in a month. Those statistics are calculated for each feature defined in Table 3. Finally, the aggregated data tables from different detectors are merged into one large table through unique truck/wheel identifier and time window using Netezza data warehouse (Fransico, 2011).

Bad Truck/Bad Wheel Labeling. Historical failure records consist of two data sets: bad order data and repair data. Wayside detectors help generate the alarms that result in bad order data. Once it is generated, the rail car is sent to a workshop for examination. When a real defect is observed, and actual repair work is completed, a repair record corresponding to this bad order is stored. Combining bad order data and repair data, we are able to track and identify real equipment failures and thus label the wheel/truck as bad. Next, the detector readings of bad truck/bad wheel within a certain time window before the bad order date are extracted. This reading window selection should be determined long enough to allow the railroad to conduct necessary corrective maintenance action but not too long to avoid false alarms (Yang 2005). We use three-month window based on the discussions with the domain experts in order to meet the practical needs.

Prediction. Now we describe how to predict bad truck/bad wheel in the next 3 month based on the past three month aggregated readings from multiple detectors. As the Maintenance Department emphasizes the importance of human interpretability of failure prediction rules, we choose decision tree, a popular technique that provides logic-based rules (Breiman et al., 1984). Users can understand the classification rules by tracing down the nodes and branches in the tree. We develop decision tree model using the method proposed in Hothorn et al. (2006). It turns out that decision tree model gives good results. It satisfies the accuracy requirement and generates simple logic rules that can be interpreted straightforward by operators.

We divide the entire data into training and testing sets (80% & 20%). Fig. 2 gives an example of a subset of branches that shows the conditions triggering bad truck alarms based on detector (OGD, MV and WILD) readings. The true positive is equal to 97% for both training and testing data, and false positive is 0.20% for training set and 0.23% for test set.

Rule Extraction. Human interpretable rules are extracted given the decision tree classification results. Every leaf of the decision tree corresponds to a primitive production rule (Han et al., 2011).

4. Summary

We have described the work on large scale multi-detector predictive modeling for railway maintenance operation management. Using huge volumes of historical detector data, in combination with failure data, maintenance action data, inspection schedule data, train type data and weather data, we explore several analytical approaches including, correlation analysis, causal analysis, time series analysis and machine learning to automatically learn rules and failure prediction models. Information integration of multiple detectors plays an important role in the analysis. Combining data collected from disparate sources shows the better results than when these sources are used individually. In addition, in the implementation of predictive modeling rule sets, we emphasize the capability of human interpretability which facilitates the decision making process of maintenance personnel. We show two different approaches to achieve this goal using two applications of alarm prediction and failure prediction.
The solution we provide has shown significant business values. Given the constraints of extremely low false alarm rate due to the cost of attending to false alarms, only the scenarios of FPR less than 0.014% is interesting, which allows no more than 20 setouts per day. Since the cost of service interruption depends on several factors including the interruption location and traffic, it can range from several thousand to several million dollars. We create sensitivity analysis around different maintenance cost assumptions. These saving calculations assume that each alarm and related repairs cost several thousand dollars and a preventable interruption can save several thousand to hundreds of thousand dollars depending on alarm location and traffic. Given the total railroad traffic, detector network density and total L1 alarms generated system-wide, it turns out that savings generated with our alarm prediction module range between 200 K to 5MM USD per year, depending on the TPR and FPR trade-off chosen in the implementation.

Although our implementation specifically discussed application in rolling-stock maintenance for railroad operations, the algorithms described above is more generally applicable to many other industries that use sensor network for equipment health monitoring, such as aircraft maintenance, heavy earth-moving equipment maintenance, power-plant and chemical plant maintenance, etc. In all these cases our machine-learning approach of large-scale multi-detector predictive modeling will bring significant business values to maintenance operations.

References