Predicting NOx sensor failure in heavy duty trucks using histogram-based random forests

Ram B. Gurung, Tony Lindgren, and Henrik Boström

Department of Computer and Systems Sciences, Stockholm University, Kista, 164 07, Sweden
gurung@dsv.su.se
tony@dsv.su.se
henrik.bostrom@dsv.su.se

ABSTRACT

Being able to accurately predict the impending failures of truck components is often associated with significant amount of cost savings, customer satisfaction and flexibility in maintenance service plans. However, because of the diversity in the way trucks typically are configured and their usage under different conditions, the creation of accurate prediction models is not an easy task. This paper describes an effort in creating such a prediction model for the NOx sensor, i.e., a component measuring the emitted level of nitrogen oxide in the exhaust of the engine. This component was chosen because it is vital for the truck to function properly, while at the same time being very fragile and costly to repair. As input to the model, technical specifications of trucks and their operational data are used. The process of collecting the data and making it ready for training the model via a slightly modified Random Forest learning algorithm is described along with various challenges encountered during this process. The operational data consists of features represented as histograms, posing an additional challenge for the data analysis task. In the study, a modified version of the random forest algorithm is employed, which exploits the fact that the individual bins in the histograms are related, in contrast to the standard approach that would consider the bins as independent features. Experiments are conducted using the updated random forest algorithm, and they clearly show that the modified version is indeed beneficial when compared to the standard random forest algorithm. The performance of the resulting prediction model for the NOx sensor is promising and may be adopted for the benefit of operators of heavy trucks.

1. INTRODUCTION

In heavy duty trucks, it is important to ensure the availability of the truck and especially avoid any unexpected breakdown during operation. Such an unexpected breakdown not only can inflict heavy loss in terms of business income but also could result in life-threatening accidents. Therefore it is very important to accurately estimate the well-being of important components of trucks so that any impending faults could be discovered and dealt with early on. The information about the current health of the components of trucks may also be useful in organizing flexible maintenance plans rather than relying on fixed maintenance schedules, specifying when trucks should visit workshops independently of their condition (Lindgren, Warnquist, & Eineborg, 2013). This is also in accordance with a current trend in the truck industry which is shifting from selling products to selling transport service solutions for customers that demand up-time guarantees. In fleets of trucks, transportation tasks can be assigned to trucks according to their overall health condition, e.g., important transportation tasks are assigned to healthier trucks. The field of prognostics and health management (PHM) deals with such issues of predicting the impending failures, estimations of remaining useful life and assessment of the overall health of vehicles.

In PHM, for prognostics, there are mainly two frequently employed approaches; the model-based approach (Daigle & Goebel, 2011) (Bolander, Qiu, Eklund, Hindle, & Rosenfeld, 2009) and the data-driven approach (Si, Wang, Hu, & Zhou, 2011). The model-based approach concerns designing physical models to monitor degradation rates and then predict the remaining useful life of the components. However, this approach typically requires both extensive prior knowledge and effort, in particular since a separate model needs to be constructed for each specific component. The data-driven approach, on the other hand, is based on building models through statistical analysis and machine learning using data collected over time. It typically requires less involvement of domain experts and can therefore often be less expensive. Hybrid approaches, i.e., mixing both model-based and data-driven approaches, are also common (Liao & Köttig, 2016). This paper focuses on using a data-driven approach to

build a predictive model for the NOx sensor in heavy trucks. As heavy trucks are getting increasingly complex, building physical models are also getting increasingly challenging and therefore data-driven approach are gaining in popularity and attention. However, as will be seen, the increased complexity of the trucks is also a challenge for the data-driven approaches.

One particular challenge that will be considered in this paper is that currently large volumes of operational data are not transmitted immediately, but stored on-board and transmitted first during maintenance or workshop visit. Moreover, to save storage space, multiple measurements are aggregated into histograms, rather than keeping all individual measurements. It is therefore of utmost importance that the analysis methods can effectively handle such histogram data. Previous work on how to exploit such accumulated operational data from trucks in histogram format are quite rare. The authors are only aware of two such studies, (Frisk, Krysander, & Larsson, 2014) and (Prytz, Nowaczyk, Rgnvaldsson, & Byttner, 2015), which recently presented studies with very similar objectives of predicting failures of the components in the heavy duty trucks, but focusing on different components; Frisk et al investigated battery failure in trucks, while Prytz et al investigated compressor failure. Erik et al worked on data similar to the one used in this study, but they did not elaborate on how their data was collected. Prytz et al, on the other hand, have provided an elaborate explanation of how their dataset was prepared, but the number of considered vehicles in their study was very limited. Furthermore, multiple instances of the same vehicle were treated as independent observations, hence invalidating the standard assumption of data being drawn independently from an identical distribution (iid). However, some related work on predicting vehicle component breakdown could be found. For example Eyal et. al (Eyal et al., 2014) has published their work on survival analysis of automobile components. However, most of the paper is focused on explaining their proposed method and very few details on the data used. Earlier work by Lawless et. al (Lawless, Hu, & Cao, 1995) could also be mentioned where they have studied on failure distributions from automobile warranty data.

The component of interest for this particular paper is the NOx sensor, but the overall procedure is generic and reproducible for any other component of choice. The NOx sensor was selected because it is one of the most important components and very expensive to repair. NOx contents present in the exhust gas are atmospheric pollutants and so forced by legislation to cleanse below acceptable level before releasing into atmosphere. NOx sensor is an integral part of this cleansing system and must be in working condition at all time during truck's operation. Moreover, this is one of the most frequently failing components of heavy trucks. By using a predictive model for this component, one could save money by correctly predicting the failure early on, but one could of course also lose money when incorrectly replacing or inspecting the component at a workshop. The overall objective of using a predictive model is usually to optimize the total cost and/or up-time, for example by aligning workshop visits with the delivery schedule of haulage operators, parametrize the cost of down-time, etc. Whenever new operational data is extracted from a truck, such a predictive model can provide an estimate of how likely it is that the NOx sensor will fail in the near future and based on this information, one can decide whether there is a need to schedule a workshop visit or not. In this paper, we will also elaborate on various challenges encountered while preparing input data for training the NOx prediction model. The data that has been used for training has attributes represented as histograms and two dimensional matrices along with many numeric and categorical types laden with many missing values. We therefore have modified the standard random forest algorithm to exploit the fact that bins in a histogram are related rather than treating them as independent features.

In the next section, we provide more background about the NOx sensor. In Section 3, we describe and discuss the preparation of the input data. Details of the modified random forest algorithm are given in Section 4, while the results and analysis of the learned model are presented in Section 5 and 6. Finally, in Section 7 we summarize the main findings and point out directions for future research.

2. THE NOX SENSOR

Fuel combustion in internal combustion engines result in exhaust gas that contains particulate matter, oxides of Nitrogen (NOx) etc. which are atmospheric pollutants and can harm human health. Nitrogen oxides are responsible for photochemical smog that can harm respiratory functions and affect visibility. They form nitric acid in the atmosphere and eventually cause acid rain. NOx gases are also responsible for the global warming. Oxides of Nitrogen come in various forms such as nitric oxide (NO), nitrogen dioxide (NO_2) and nitrous oxide (N_2O)[1]. With the increasing number of heavy trucks produced every year, the legal limits of acceptable NOx emission by heavy trucks are getting increasingly stringent. In EU and EEA member states, European emission standards define acceptable limits of exhaust emissions. Emission standards have evolved over time from the first Euro I (1992) to the latest Euro VI (2013) and for heavy duty diesel trucks, standards are measured in engine energy output, g/kWh. For instance, the acceptable level of NOx emission in Euro IV (2005) emission standard was 3.5 g/kWh, which has now been made stricter to 0.40 g/kWh by the most recent Euro VI (2013) standard[2]. Cleansing the exhaust gas to keep the NOx content on an acceptable level has been one of the ma-

[1] http://www.eea.europa.eu/data-and-maps/indicators/eea-32-nitrogen-oxides-nox-emissions-1

[2] http://ec.europa.eu/environment/air/transport/road.htm

jor challenges in automotive industry. So it is very important for heavy truck manufacturers to install state-of-art emission control systems in their vehicles and keep track of the emission at all times when the vehicle is in operation.

In heavy duty diesel trucks, one way of minimizing the NOx content in exhaust is by recirculating the engine exhaust back to the combustion chamber, which results in a lower temperature. Since nitrogen and oxygen need a higher temperature to form NOx, less NOx is hence generated. After-treatment purification methods could also be used. A NOx purification system uses a NOx Storage Reduction (NSR) and/or Selective Catalytic Reduction (SCR) system (Devarakonda, Parker, & Johnson, 2012) (Sawada & Imamura, 2012). Sensors to measure NOx concentration are positioned before (upstream) and after (downstream) NOx purification in the exhaust path. The NSR catalyst can absorb NOx in the exhaust gas when the air-fuel ratio of the exhaust gas is lower than a predetermined threshold and release the stored NOx as nitrogen when the air-fuel ratio of the exhaust gas is higher than the predefined threshold. Once the NSR catalyst cannot store anymore NOx, i.e., when it has reached a saturation state, then reducing agents are supplied to release the stored NOx. This process of releasing the stored NOx can begin when the NOx sensor downstream detects leakage of NOx. SCR systems are more popular than other exhaust NOx treatment process in heavy duty trucks as it is very effective at cleansing exhaust NOx. In a SCR system, Diesel Exhaustive Fuel (DEF, often urea solution) is used as a reducing agent in which NOx is subjected to and result in nitrogen, water and small amounts of carbon dioxide. The SCR system needs to replenish DEF on a periodic basis. Most of all, a correct measure of NOx concentration of the final exhaust gas has to be done to assess the performance of the purification system. These NOx readings are also used as feedback control. Hence, the NOx sensor is an important component in the exhaust purification system. Truck manufacturers are these days forced by legislation to design their vehicles such that the truck suffers from power limitations if NOx emissions levels are not met and within a certain period of time the trucks are forced to a standstill, to prevent it from being used with a defective emission control system. So it is extremely important to have a healthy and properly working emission control system for a vehicle to operate properly.

NOx sensors are an integral part of the emission control system. Usually one sensor is positioned in the tailpipe where they are exposed to harsh conditions, with very high temperature of exhaust emissions, varying from 500 to 1000 degrees Celsius. The NOx sensors are usually made of materials that can withstand such harsh working environments, such as ceramic type metal oxide, yttria-stabilized zirconia (YSZ) being the most common one. The benefit of using YSZ is that it can conduct oxygen ions in high temperature. Besides this functional advantage, YSZ is physically strong and stable at high temperatures (Schubert, Wollenhaupt, Kita, Hagen, & Moos, 2016). YSZ along with electrodes of noble metals such as platinum or gold is used to build a NOx sensor and the concentration of NOx is communicated via an electrical signal. Good NOx sensors usually have a high sensitivity, especially given a very low ppm (100 to 2000) of NOx to be measured at fluctuating high temperature. Moreover, the response time should be very short, since its readings are used for feedback control. This makes the NOx sensor very difficult to build, which also makes it one of the expensive components in the vehicle but at the same time also very prone to breakdowns. High exhaust temperature can de-laminate the electrodes over time and soot particles can degrade the material. Sometimes a tiny drop of water (e.g., dew) on very hot ceramic can crack the sensor rendering it useless. Because of these serious issues, it is one of the prioritized components in heavy trucks that need to be studied for its failure patterns to allow for accurate prediction of any impending failures.

3. DATA PREPARATION

The data that will be considered for analysis is collected from trucks manufactured by Scania. Because of their modular design, there is great differentiation in their configurations, i.e., two trucks only rarely have the same configuration. This in turn implies that data collected from different trucks may vary substantially, not only in terms of feature values, but with respect to what feature values are available. Moreover, more sensors are added to the trucks over time and the software in Electronic Control Units (ECU) gets upgraded, leading to completely new features or that the generation of values for old features changes. As a consequence, missing data are abundant.

Information about the operation of individual trucks and their operating environment is stored in the ECUs on-board the trucks, which usually are readings from various sensors. This information is normally extracted when trucks visit authorized workshops. Each extraction is called a snapshot and each truck will typically have multiple snapshots taken over time. It should be noted that the intervals between any two consecutive snapshots are not regular and the number of snapshots per truck will vary as well. Some trucks might not even have any snapshot at all. Various features are used in a snapshot to describe operation of the truck. All the snapshots are uploaded and stored in a central database at Scania. Two other databases that can be useful when constructing prediction models are the warranty claims database and the workshop orders database.

The warranty claims database stores all the information about the claims made by owners of trucks for its broken parts. Usually warranty claims cover any breakdown during the first year after the delivery of a truck, while warranty claims for some components could be extended well beyond a year. Sim-

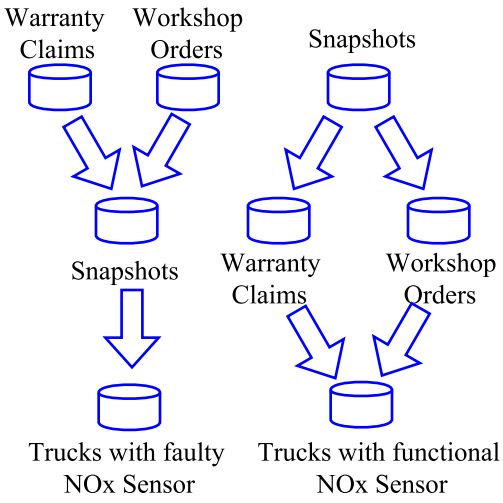

Figure 1. Selecting trucks for analysis

ilarly, the workshop orders database contains information of the components ordered by workshops when repairing the trucks. One can typically assume that a component has been ordered to replace a component that has failed. These two databases are important for identifying trucks that have had a faulty component of interest and at which date the fault occurred. In many fault prediction scenarios, the components of interest could be continuously monitored over time, but this is not the case here as the trucks normally visit workshops only a couple of times during a year. From these databases, the operational data (snapshots) for all the trucks that have had a faulty NOx sensor during 2008 and 2013 have been extracted. Similarly, we also select all the snapshots for trucks without any reported problems with the NOx sensor for the same period. The process of selecting data for the trucks is shown in Figure 1.

Most variables in the snapshots are cumulative in nature, e.g., if a variable in a snapshot is ambient temperature, it is represented as histogram of 10 bins where each bin would have a count for how long the truck operated under that particular temperature range defined by the bin boundaries. So, the count in the bin of ambient temperature variable is always increasing for snapshots taken afterwards. This means that different snapshots for the same truck often is highly correlated. However, for our purposes, we will only select one snapshot per truck. Rather than choosing a random snapshot, we want the snapshot to be the most informative, which in case of NOx sensor failure means the last snapshot taken before the breakdown occurred. For some trucks, the NOx sensor were broken multiple times, but for these we will only consider the first breakdown. The data does not state exactly when a breakdown occurred, but instead we consider the repair date registered in the warranty claims database or the truck arrival date in workshop order history information database as the approximate breakdown date. After an approximate breakdown date has been determined, we thus need to find the latest snapshot prior to this. In order to avoid that a snapshot is taken from after the breakdown, which e.g., may happen since snapshots are frequently extracted by mechanics while performing some tests in workshop. Moreover, it is also possible that the records of faults are not filed on the same day or when the truck arrives to the workshop some days after breakdown. So, we try to keep a safe margin of seven days such that the snapshot selected should be taken at least one week before the estimated breakdown day. Furthermore, the snapshot to be selected should be taken at least after truck has operated more than thirty days after delivery, in order to exclude trucks with very short operating history. For the trucks with a non-failed NOx sensor, we select the snapshot second from the last and again the selected snapshot should have been extracted at least a month after the truck was delivered. The overall procedure of selecting snapshots for trucks (with a faulty or non-faulty NOx sensor) is shown in Figure 2.

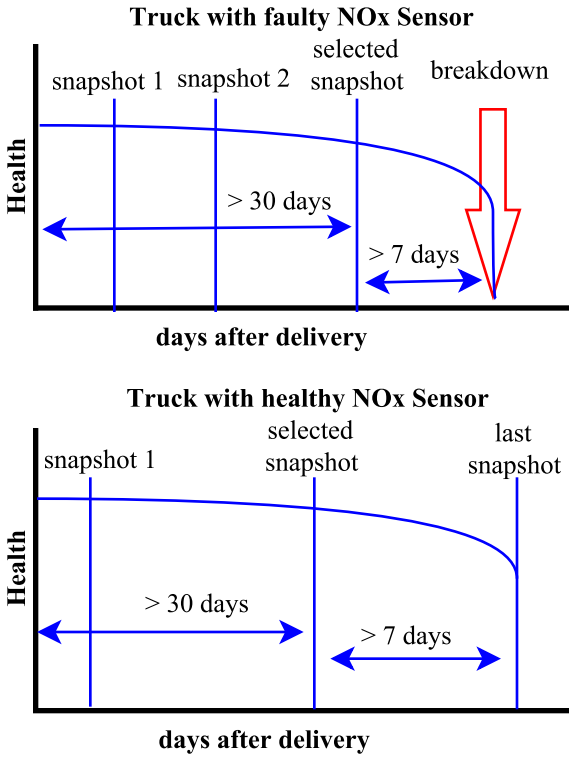

Figure 2. Selecting the best snapshot for a truck

To increase homogeneity of the trucks under consideration, only trucks built for a particular purpose of usage were selected, namely Scanias R series of trucks that are built for long distance haulage travels.

A total of 16 980 trucks were obtained from the above sources, out of which 951 had a faulty NOx sensor. Furthermore, the trucks were selected in such a way that none of the at-

tributes selected in the snapshot had missing values in them. Since the values were not missing at random, they needed to be treated specially and we intend to work on that in future. These trucks were considered ultimately for experimental dataset from around 72 000 trucks in the beginning. As mentioned before, there were many variants of same feature used in various trucks. For example, $Coolant\_Temperature$ histogram variable has at least two variants, some trucks use the first variant while others use the second variant. Other features also similarly have multiple variants. In order to keep the setup simple, we decided to select the variant that is used by most trucks. In doing so, the number of trucks at the end are largely limited to 16 980 only. So, we expect these final set of trucks to be of similar nature in their configuration and ECUs installed. Although random forest algorithm that shall be used for training the predictive model can handle missing values internally, we decided to refrain our analysis from how missing values were handled by the algorithm which can be something to be explored further in detail in future.

Attributes in snapshots were selected by consulting with experts from Scania. This was important as the number of technical specifications and operational variables were too many to consider all of them. Only very few technical specifications that would distinguish trucks were chosen while for operational variables only those that might have influence on NOx breakdown and exhaust system were selected.

Categorical attributes
Engine Type (16 unique values)
Engine Stroke Volume (3 unique values)
Power (9 unique values)
Generalized chassis number (4 unique values)

Numerical attributes
Age, Technical total Weight

Histogram Variables
Ambient Temperature: 10 bins
Atmospheric Pressure: 10 bins
Boost Air Pressure: 10 bins
Coolant Temperature: 10 bins
Fuel Consumption Speed Volume: 20 bins
Fuel Temperature: 10 bins
Inlet Air Temperature: 10 bins
Vehicle Speed: 10 bins

When histogram is normalized in such a way that the bins sum to one, it can be viewed as a probability distributions. The shape of this probability distribution depends on how the width of the bins are set. Nevertheless, we assume histograms to be normally distributed across bins and use formula for normal distribution to calculate mean and standard deviation to summarize its distribution. So, for each truck, for a given histogram variable, mean and standard deviations were calculated using frequency of the bins and midpoint of the bins (using bin breakpoints). In some of the histograms, the first and last bins have open boundaries ($< or >$), so we decided to assume the width of those bins to be equal to second and second last bin respectively. So, for the above listed eight histogram variables, 16 new additional numeric attributes were generated.

For a histogram variable H with m bins, mean (μ_i^H) and standard deviation (σ_i^H) for i^th observation are calculated as follows,

$$\mu_i^H = \sum_{j=1}^{m} x_j^H . f_j^H, s.t. \sum_{j=1}^{m} f_j^H = 1 \tag{1}$$

$$\sigma_i^H = \sqrt{\sum_{j=1}^{m} (x_j^H - \mu_i^H)^2 . f_j^H} \tag{2}$$

where f_j^H is the normalized frequency in j^{th} bin and x_j^H is the midpoint of j^{th} bin obtained from the bin breakpoints for histogram H. For histogram variables in the data set we have, we already know about the structure of histogram variable such as how many bins there are and what the bin boundaries are. For example, for the histogram variable ambient temperature, we can calculate bin midpoints using bin boundaries and its bin midpoints turns out to be

$(-35, -25, -15, -5, 5, 15, 25, 35, 45, 55)$.

In addition to this, an algorithmic approach to treating histogram variables is described in the next section.

Matrix Variable
Engine Load Matrix: 11 $\times$ 12 = 132 cells (Engine Load Percentage $\times$ Engine RPM)

For each matrix variable, marginal frequencies were calculated along the two axes. The matrix variable was simply split into two constituting histogram variables and mean and standard deviations for them were calculated. So, four new numeric variables were generated from each matrix variable. The algorithmic approach of handling matrix variables is described in the next section.

4. RANDOM FORESTS FOR HISTOGRAM DATA

The random forest algorithm (Breiman, 2001) is one of the most widely applied learning algorithms, often reaching state-of-the-art performance. In previous attempts of predicting component failure in vehicles by Frisk et. al (Frisk et al., 2014), they have used a variant of random forest to build their predictive model and similarly Prytz et. al (Prytz et al., 2015) had demonstrated that the random forest algorithm outperforms all the other considered learning algorithms. Specially

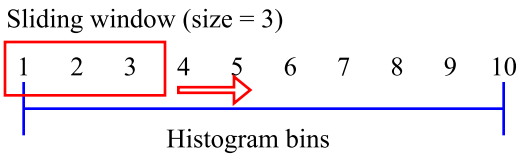

Figure 3. Sliding window for histogram with 10 bins

in survival analysis setup, tree based methods seem to be getting popular, for instance random survival forest by Ishwaran et. al (Ishwaran, Kogalur, Blackstone, & Lauer, 2008) and use of mutually exclusive forest by Eyal et. al (Eyal et al., 2014) to list the few. Interested readers can also look into work by Zhou et. al (Zhou & McArdle, 2015) for further details on rationale and applications of tree based survival methods. Because of this, we decided to use the random forest algorithm for developing the NOx sensor failure prediction model, but with a slight modification enabling it to learn from histogram and matrix variables. Below, we will briefly explain how the algorithm behaves when a histogram variable has been selected for evaluating splitting of a node while growing a tree. A more detailed description of learning (single) decision trees from histogram data can be found in our previous work (Gurung, Lindgren, & Boström, 2015), (Gurung, Lindgren, & Boström, 2016).

For a histogram variable, the bins are sequentially arranged according to the bin boundaries. If for example the histogram variable is *ambient temperature*, the lower (higher) ordered bins would correspond to operation in cold (warm) weather. If the bins of a histogram are represented as separate numeric attributes, this can result in many correlated attributes. Apart from increasing dimensionality, which may have a negative impact on predictive performance, the standard random forest algorithm would underestimate the variable importance score for those variables. Furthermore, if there are any dependencies among the bins of a histogram, they might not be fully exploited when evaluating the bins individually. So instead, the modified version of the algorithm handle the bins of a histogram variable jointly, evaluating the regions of the histogram by considering groups of adjacent bins. For example, if the group of bins 1, 2 and 3 for ambient temperature gives good separation into trucks with faulty and healthy NOx sensors; then the operation in cold weather can be considered to be a useful factor for predicting failure. In order to select the size of region (how many adjacent bins to consider), we use a sliding window approach, where the size of the window is determined by a parameter that can be tuned. Figure 3 illustrates the use of the sliding window method for a histogram with 10 bins with a window size set to 3.

For example, consider the histogram variable *ambient temperature*, which has 10 bins whose midpoints are

$$(-35, -25, -15, -5, 5, 15, 25, 35, 45, 55)$$

corresponding to bins $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$. So, if we choose to let the sliding window size to vary from 2 to 4, we get the following groups of bins for evaluating split of a node, obtained by sliding the window of each given size along the ordered histogram bins:

$$\{(1,2),(2,3),(3,4),(4,5),(5,6),(6,7),(7,8),(8,9),$$
$$(9,10),(1,2,3),(2,3,4),(3,4,5),(4,5,6),(5,6,7),(6,7,8),$$
$$(7,8,9),(8,9,10),(1,2,3,4),(2,3,4,5),(3,4,5,6),$$
$$(4,5,6,7),(5,6,7,8),(6,7,8,9),(7,8,9,10)\}$$

When the ambient temperature histogram needs to be evaluated for splitting a tree node, the algorithm randomly selects $\lceil\sqrt{m\_combn}\rceil$ number of bin sets to investigate (similar to the original random forest algorithm which in a standard default setting considers the square root of the number of available variables for each node split). Here, in the example above $m\_combn = 24$. So, the algorithm would in this particular case randomly pick $\lceil\sqrt{24}\rceil = 5$ bin sets.

Let us assume that the set $\{3, 4\}$ is among the selected bin sets for evaluating the node split. In this particular case, all the observations (trucks) are represented as a point in a two-dimensional space of bins 3 and 4. Each point has a class label assigned to it as either faulty or healthy. Now the algorithm tries to find the linear hyperplane that can split the observations (trucks) into faulty and healthy trucks in best possible way as shown in the Figure 4. In order to find the best splitting hyperplane, a small number of special unique points are carefully selected first. Each splitting hyperplane in the given space is obtained by assuming it to pass through these points. The algorithm selects these special points such that they lie closest to the centroid of points from the opposite class. The number of special points to be used for creating splitting hyperplane is obtained using a tuning parameter sp as following:

$$number\_of\_split\_points = size(chosen\_bin\_set) + sp$$

Here sp is a natural number. As shown in the Figure 4, the special points (marked as asterisks) are chosen as the nearest points to centroids (two big dots) of points from the opposite class and later these points are used to generate a splitting hyperplane. Let us assume that sp is set to 10, so that the algorithm would select 12 special points to be used for generating the splitting hyperplane in the case where bin set $\{3, 4\}$ is selected. Out of the 12 selected special points, 2 (dimensions of the space) of them are chosen at a time to get the equation of the linear hyperplane that passes through these 2 points.

For this particular case, when the bin set $\{3, 4\}$ is selected, let the two points (x_1, y_1) and (x_2, y_2) be selected from the 12 special points. Let the equation of splitting hyperplane be

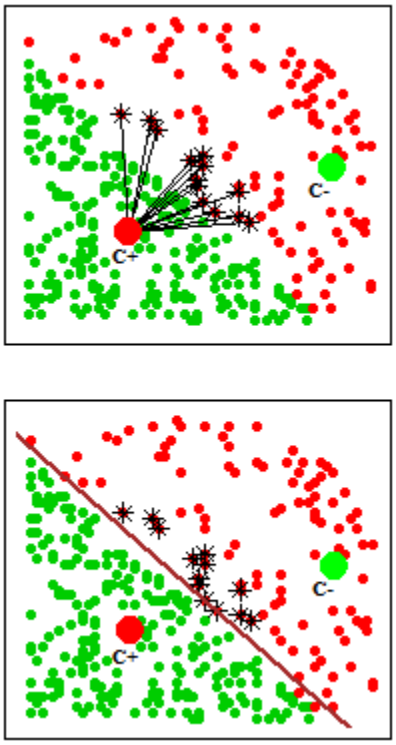

Figure 4. Selecting split points and best split plane

$$C_1.X + C_2.Y = 1 \tag{3}$$

Now, we use two selected points to solve for coefficients of this hyperplane such that

$$\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \times \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}^{-1} \times \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The hyperplane using these two points is possible only if the inverse of the matrix exists. Once the linear hyperplane has been generated, the algorithm tries to evaluate how well it separates the remaining points into two child nodes. The more homogeneous (or pure) the resulting child nodes are, the better the split is. The performance of the hyperplane is measured as information gain obtained after the split.

All the possible combinations of 2 points selected out of 12 special points, $\binom{12}{2}$, i.e., 66 splitting hyperplanes are evaluated and the best (most informative) one is selected. The best hyperplane from the bin set $(3, 4)$ is now compared with the best hyperplanes of 5 (i.e.$\sqrt{m\_combn}$) other randomly selected bin sets for ambient temperature histogram. Ultimately, the best splitting hyperplane and the bin set is determined and will represent the ambient temperature histogram to be compared with best splits from other histogram variables and numeric and categorical variables for the final split decision. The procedure is depicted in Figure 5 where the splitting process in intermediate node 2 is further elaborated for clarity.

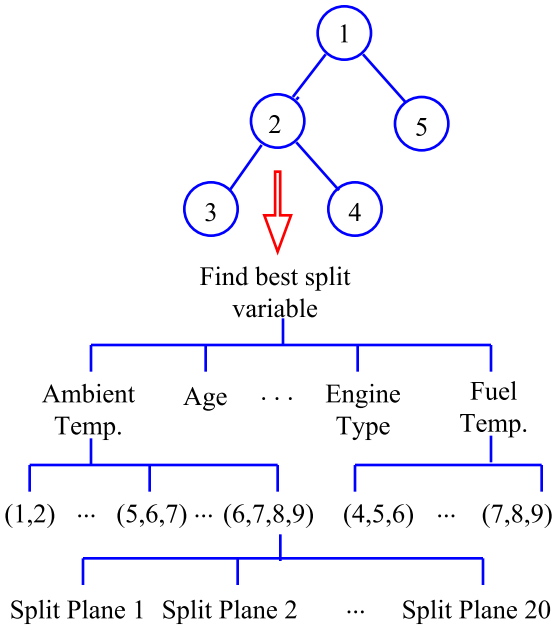

Figure 5. Node split evaluation process

The procedure for handling matrix variables is similar to the procedure for handling histogram variables, the only difference being the way in which groups of adjacent bins (cells) are assembled. Since a matrix variable has two dimensions, a sliding window should be able to move in both dimensions. For simplicity, a window of size 2×2 is selected, so that 4 adjacent cells of a matrix form a window which can sweep throughout the matrix cells as shown in the Figure 6. The matrix variable in the figure has 132 cells, X variable with 12 bins and Y variable with 11 bins. For the given matrix variable, 110 blocks of such 4 adjacent cells (size 2×2) can be generated. So if this matrix variable is to be evaluated for splitting a node, $\lceil\sqrt{110}\rceil = 11$ such blocks are randomly selected. For each of these blocks, the best splitting hyperplane is determined.

5. EXPERIMENT

Before training the adapted random forest algorithm on real data, it was first tested on synthetic data to verify that it works as it is expected. Two synthetic data sets were generated. All the experiments and the implementation of the modified random forest algorithm was done using the R language[3].

[3]https://www.r-project.org/

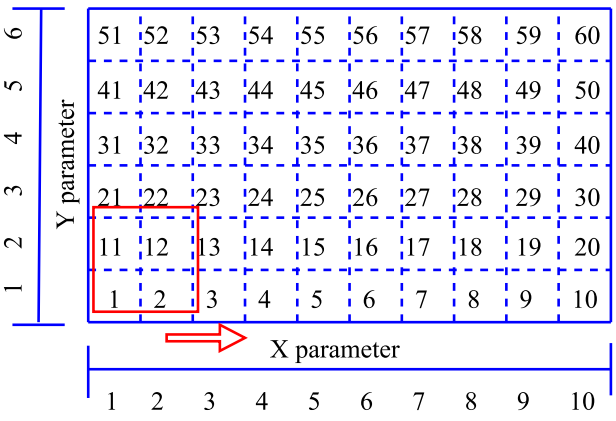

Figure 6. Sliding window for matrix variable

5.1. Dataset I: Linear pattern

The first synthetic dataset has two histogram variables with four and five bins, respectively, where the bins of each histogram sums to 1. Bin values are uniform randomly generated to lie between 0 and 100 first and then normalized to sum to 1. A class label for an observation is based on a linear pattern $H_1^1 + H_2^1 < 0.8$ in the first histogram and $H_1^2 + H_2^2 + H_3^2 < 0.8$ in the second histogram, where H_j^i represents the j^{th} bin of the i^{th} histogram variable. An indicator variable is used for each histogram to determine if an observation satisfies the condition in the linear pattern for both histograms. Indicator variables for an observation for each histogram variable are set to TRUE if the condition is satisfied. The class label of the observation is set to 1 if indicator variable for both histograms are TRUE otherwise 0. Noise s introduced around the boundary region (linear pattern) to blur the class boundary. Twenty five percent of the points for first histogram and twenty five percent of points for second histogram are selected and for randomly selected ten percent of those points, the indicator variables are flipped. After introducing the noise, the class label for each observation is re-evaluated i.e. class label for an observation is reassigned as 1 only if indicator variables for both histogram variables are still TRUE. However, indicator variables are there only for determining the class label and introducing noise, they are removed from the final experimental data. So the final dataset has only two histogram variables for 1912 observations among which 440 have class label 1.

5.2. Dataset II: NonLinear pattern

Second synthetic data-set has one histogram variable with four bins. Similar to first synthetic dataset, bin values are uniform randomly generated between 0 and 100 and later normalized to sum to 1. The class label for an observation is set to 1 if the histogram satisfies a condition $(H_1 - 0.3)^2 + (H_2 - 0.3)^2 < 0.3^2$ which is non linear in nature. Here, H_j refers to j^{th} bin of histogram H. Observations that satisfy the pattern, or observation for which point (H_1, H_2) in 2D space of first and second bin lies inside the circle with center $(0.3, 0.3)$ and radius 0.3 has class label 1 else 0. Noise is injected along the boundary region using similar technique as in first dataset. The final dataset has 1912 examples out of which 624 have class label as 1.

5.3. Experimental setup and results

A random forest model for the synthetic data was built. In the first setting, all the histogram variables in the dataset were treated as histograms (using sets of adjacent bins to make splitting decision). In the second setting, all the histogram bins were treated as standard numeric variables (using the standard random forest algorithm). The parameter that was varied was the number of special points to use (sp) for generating splitting hyperplanes. The values for sp were set to 1, 2 and 3 respectively. The larger the value for sp, larger would be the number of split planes to examine and better the chances of finding an informative split. However, the training time increases accordingly. For histogram approach of random forest model, sliding window size was set to vary from 2 to 3. The window size can be varied between any number from 2 to the size of histogram, but for simplicity only the smallest sizes from 2 to 3 were considered. Five-fold cross-validation was performed and the results were then averaged over these five folds. In total 300 decision trees were built in each forest. Again, a larger number of trees could have been chosen but the computational cost increases accordingly so only 300 trees were chosen. The node splitting process would stop when the number of examples in the node was less than 5 or the node had all the observations with same class label. In Table 1 and 2, the results for the experiments with synthetic data are presented. The area under ROC curve (AUC) and accuracy were used to evaluate the random forest models. The average number of leaf nodes in the trees of the forests built under different settings are also shown.

In addition to the above presented approach of finding the best splitting hyperplane for a histogram variable, commonly employed linear classifiers such as logistic regression, the perceptron algorithm and the Support Vector Machine (SVM) with a linear kernel were also examined to generate the best splitting hyperplane during tree construction. For logistic regression, the publicly available GLM R package was used, while for SVM, the e1071 R package was used. Logistic regression and linear SVM were trained with default settings as provided in their respective R packages. While searching for the best splitting hyperplane, several such linear models need to be trained for each histogram variable in every node

Table 1. Results of classification on synthetic data I.

Random Forest Models	AUC	Accuracy	Leaf Nodes
Hist. RF (sp=1)	0.9852	94.19	62
Hist. RF (sp=2)	0.9862	94.19	53
Hist. RF (sp=3)	0.9868	94.24	47.8
Hist. RF (Logistic reg.)	0.9222	92.62	27.6
Hist. RF (Perceptron)	0.9758	94.08	36.4
Hist. RF (Linear SVM)	0.8239	91.15	3
Standard RF	0.9633	93.25	83.4

Table 2. Results of classification on synthetic data II.

Random Forest Models	AUC	Accuracy	Leaf Nodes
Hist. RF (sp=1)	0.9580	87.02	115.2
Hist. RF (sp=2)	0.9590	86.92	100.4
Hist. RF (sp=3)	0.9597	87.12	92
Hist. RF (Logistic reg.)	0.7968	78.13	21.4
Hist. RF (Perceptron)	0.9283	84.25	23.2
Hist. RF (Linear SVM)	0.7159	76.09	3
Standard RF	0.9552	86.55	110.2

of many trees in the forest model. Tuning the parameters for every single model would be cumbersome, so the default setting was used here for convenience. The results obtained with the various implementations are also shown in Table 1 and 2. Implementations are compared in terms of AUC, Accuracy and average size of leaf nodes in each model. A model with small number of leaf nodes in average would indicate that the trees in the random forest are less bushy which further indicates that the splits in the tree nodes are few but compact. For instance, if two models are equivalent in performance (e.g., AUC) but differs in average tree size, we can assume that the model with smaller size carries compact information in each split.

The results of the experiments on synthetic data has shown that the histogram-based random forest approach performed better than the standard random forest algorithm. The result also shows that the histogram-based approach tend to perform better when the number of special points to be used for forming splitting hyperplane is increased, which also results in reduced average size (nodes) of the trees in the forest as indicated by decreasing average number of leaf nodes. The gain is more accentuated for the task regarding a relatively easy linear pattern compared to when a non-linear pattern has to be identified. When comparing the original histogram-based approach to alternatives that find splitting planes using logistic regression, the perceptron or a linear SVM, only the use of the perceptron algorithm gives comparable results, in particular in the first experiment that concerns linear patterns in the data set. However, the perceptron algorithm needs to update the weights of best plane sequentially and a very large number of repetitions is needed for convergence, or if there is no clear separation, the algorithm has to execute the maximum number of allowed repetitions, something which is very costly for large datasets. When using an SVM, the tree cannot typically be grown past two splits on average, hence leading to low variance. Even logistic regression did not do very well.

6. NOx SENSOR FAULT PREDICTION

6.1. Comparision of random forest models

The heavy truck dataset that was described in section 3 included two types of trucks; one with faulty NOx sensors and others with functional NOx sensors, and the considered task here was to classify trucks for which the status of the NOx sensor was not known into one of these groups. Table 3 presents the result of the random forest models built under four different set ups. *Histogram RF A* is a random forest model built using the dataset where histogram bins were expressed as percentages, such that bins sum up to 100. In the *Histogram RF B* model, the bins instead were represented by real values, which were normalized individually such that the values fall into the range between 0 and 1. An implementation of the standard random forest classification algorithm by Ishwaran et al (Ishwaran et al., 2008) was used to build *Standard RF A* and *Standard RF B* using data set that was used for building *Histogram RF B*. In addition, 20 attributes representing the mean and standard deviation of the histogram variables were included in *Standard RF B*. This was done to see if there would be any improvement in model performance by adding these additional derived variables. For all models, the number of trees was set to 500. Nodes with homogeneous set of observations or those with less than 5 observation were converted to a leaf node. For the histogram approach, the window size was varied between 2 to 4. However, for the matrix variable the size was fixed to 4 cells in a block. The parameter sp that concerns the number of special points used to form splitting hyperplanes was set to a minimum value 1, to keep the setup simple and computationally efficient. In Table 3, the results for all four models are presented. Since the data was highly skewed in terms of class distribution, accuracy would not give a clear picture of models performance, so it was dropped from the result table. For example, if 94% of observations in a dataset are of negative class, a useless model that always predicts a test observation as negative would still have accuracy of 94 percent which gives an impression of a good model although it is clearly not. On the other hand, in AUC measure, observations are ranked according to some measure assigned by the model (probability in this case). If all the positive observations are ranked higher than negative observations, the model has AUC score of 1. Any random guessing model would have AUC score of 0.5. Since, skewness of class distribution has no influence in AUC score, it is a preferred measure for model evaluation when dataset has highly skewed class distribution. The result from the classification experiment reveals that the histogram-based approach of building random forests delivered better results in term of the AUC measure. However, the average number of leaf

Table 3. Results of classification on NOx sensor data.

Random Forest Models	AUC	Leaf Nodes
Histogram RF A	0.8360	503.6
Histogram RF B	0.8479	522.8
Standard RF A	0.8108	422.3
Standard RF B	0.7955	411.9

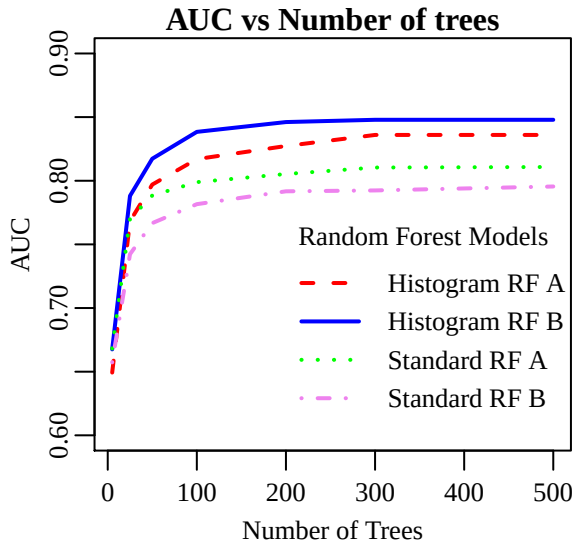

Figure 7. AUC vs Trees in random forest models

nodes for the histogram approach is higher. However, as was evident from the synthetic data experiment, the size can be reduced by increasing the parameter sp . Surprisingly, *Standard RF B* was the worst model among the four, probably because of including the 20 new features (means and standard deviations) introduced more noise than guidance. Although 500 trees were used in the random forest models, from the plot of AUC versus number of trees as shown in Figure 7, the AUC performance starts to stabilize after 200 trees. Again from the plot, it can be clearly seen that the histogram-based random forests outperform the two variants of the standard approach.

6.2. Variable importance

A variable importance rank for the best performing histogram-based random forest model *Histogram RF B* is shown in Figure 8, where the importance score has been normalized to sum to 100. The variable *Engine Loadmatrix* is ranked as the most important variable. It can be seen that all of the histogram variables are ranked relatively high in the list, which can be explained as various combinations of their bins were found to give the most informative split.

As previously explained, groups of adjacent bins of histogram (or cells of matrix) variables were used for evaluating splits for the histogram-based approaches. The variable importance rank in Figure 8 simply list whole histogram (including its

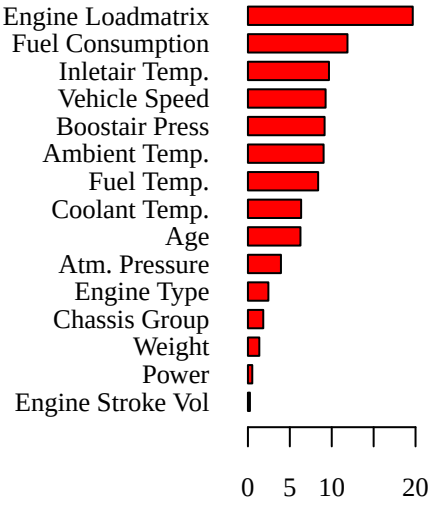

Figure 8. Variable importance rank

bins). However, we can further look into each histogram or matrix variable and see which set of bins were more useful than others during the node splitting phase while growing trees in the random forest model. Set of bins that were considered simultaneously while splitting a node are said to be more important if the split gives a better separation or it results in more homogeneous group of observations in child nodes. Description of bin boundaries also give a notion of a region in a histogram. For instance, bins 1,2 and 3 of ambient temperature histogram, $ambient\_temp$ would represent case when a truck was driving in cold and similarly bins 9 and 10 would represent operation in very hot region in histogram. Similarly, the matrix variable *Engine Loadmatrix* has two dimensions, so the region in the matrix towards the bottom left corner represents the case when truck was driving with light weight and with low RPM. Similarly, regions towards the top right corner represent cases when truck was driving with heavy weight and at high RPM.

Since *Engine Loadmatrix* turned out to be the most important variable as listed in the variable importance rank, we further looked into importance of its bins (cells) and plotted the importance score using a heat map. In our histogram approach, for a matrix variable, a square block of 4 adjacent cells (2×2) was used which is equivalent to sliding window size of 4 for one dimensional histogram variables. These 4 cells of a block were simultaneously used while evaluating the node split. In total 110 of such blocks could be generated by sliding the block of size 2×2 around the given matrix variable of size 11×12. However, not all 110 of such blocks were used for evaluating a split, rather only $\sqrt{110}$ of them were randomly used. If a block is used for splitting a node, it gets an importance score as an information gain obtained because of the split. Importance score for the matrix variable as a whole is

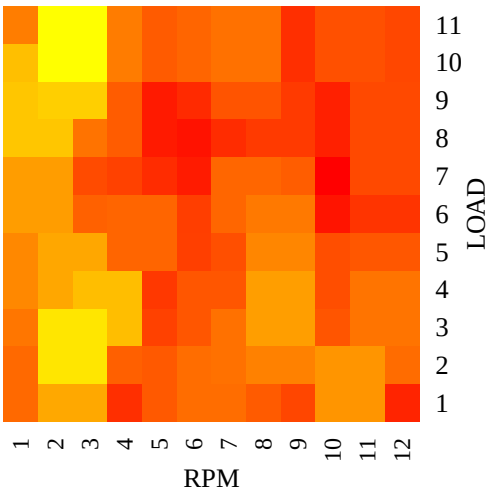

Figure 9. Important regions in Engine load matrix variable

an aggregation of these importance score by these blocks. In order to plot a heat map of importance score for matrix variable, we assumed that each cell of the 2×2 block gets the same importance score as does the block. If a cell appears in more than one block, the importance score for this cell will be the one that has the highest score. A heat map based on the importance score for each cell was then plotted and shown in Figure 9. The distinct yellow region towards the left of the heat map reveals that driving at low RPM might have some relation with NOx sensor breakdown as this region seems to be considered more important by the random forest model while training from operational data.

6.3. Model for predicting failure

Classification of trucks into those with faulty vs. healthy NOx sensor is not very useful unless it can be used to make failure prediction for the future, for example to be able to say that the NOx sensor will probably fail in next three months from now. This can be done if the classification task is set up to meet this goal. In order to achieve this, the existing dataset for heavy duty trucks was slightly changed. A new variable *remaining useful life (RUL)* was generated which is the difference in the number of days between the selected snapshot extraction date and the NOx breakdown date. Now, trucks with a breakdown of the NOx sensor before 90 days in future, i.e., RUL less than or equal to 90, were considered as positive cases while trucks whose NOx sensor survived beyond that point were considered as negative (healthy) cases. Note that trucks whose NOx sensor broke after 90 days were hence considered healthy. Trucks in the existing dataset with no observed breakdown of the NOx sensor and whose RUL values were less than 90 days were simply removed because it cannot be determined for them if they have survived past the 90 days margin. In this way, a new dataset with 8633 trucks was obtained, out of which 540 were positive (non-healthy) cases.

Table 4. Results of NOx sensor failure prediction.

Random Forest Models	Trees Used	AUC	Leaf Nodes
Histogram RF	500	0.791	314.6
Standard RF A	1000	0.749	255.4
Standard RF B	1000	0.726	247.4

Three different random forest models were built with similar set up as explained in earlier experiments. The histogram approach with bins normalized to sum up to 100 was left out as it was outperformed by histogram approach where bins were real values. 1000 trees were used in the standard random forest models instead of only 500 as in previous case. This was done simply to see if performance would enhance further by increasing the number of trees. However, since the training time increases heavily with number of trees as large as 1000 trees in case of histogram approach, number of trees were limited to just 500.

Results are presented in Table 4 and as evident, again the histogram approach has outperformed the standard approaches. However, the AUC dropped well below what was observed in the previous experiment. This could probably be the result of how we set up the dataset. In this dataset, the trucks for which a broken NOx sensor was observed after the 90 days margin were labeled as healthy cases. There might be some common pattern among all the faulty trucks regardless of the 90 days margin that the model seems to capture and hence even for them it tend to assign higher probability of being faulty. This has been depicted in the Figure 10 where the probability assigned to all the faulty trucks of being faulty has been plotted against the RUL value (after how many days in the future the breakdown occurred). The vertical dashed line is the 90 days margin that separates positive from negative cases. The average probability of all the faulty trucks labeled as positive is indicated by the solid red line. Similarly, the average probability of being faulty for all the faulty trucks that were labeled as negative because they survived beyond the 90 days margin is shown with a dashed red line, which is very close to the average probability for the positive cases. The green dashed line shows the average probability of being faulty assigned to healthy trucks. The average assigned probability of NOx sensor being faulty in all these three cases reveal that the model assigns higher probability in general to the cases where actual fault has occurred regardless of 90 days margin. It should be noted that in this experiment, the choice of the future prediction horizon was arbitrarily set to 90 days. The predictive performance (AUC) is likely affected in a positive direction by considering more distant time horizons (similar to what is done in the first experiment), while moving the horizon closer in time will most likely lead to a further reduction in predictive performance. However, what is a suitable time point is

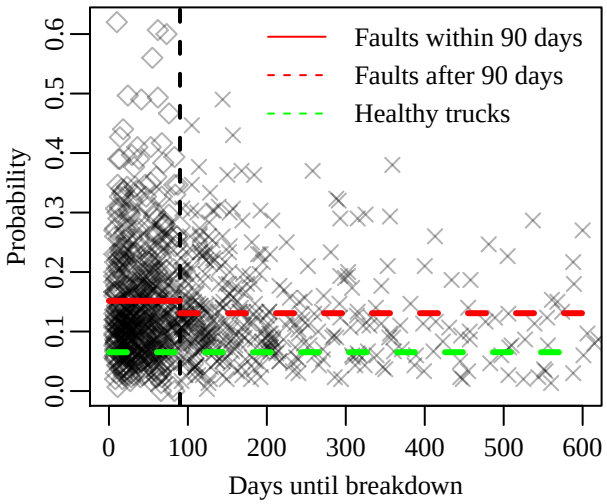

Figure 10. NOx sensor failure probability assigned to faulty trucks

not only determined by the predictive performance, but primarily by the business case, i.e., what time frames can be acted upon.

Once as a trained model has been obtained, it can provide an estimate of the probability that a truck is going to fail. But in order to make a decision, for example regarding whether or not to send a truck to a workshop, a cutoff point need to be chosen such that any truck with a probability higher than that should be selected. It is not trivial to choose such a cutoff point and it depends heavily on the business case. There are costs associated with each miss-classification made by the model. For instance, the expense associated with predicting a faulty truck as healthy can be very high compared to predicting a healthy truck as faulty. If we consider faulty trucks as positive cases, the cases of incorrectly predicting faulty trucks as healthy are the false negative (FN) cases. Similarly, incorrect predictions of healthy trucks as faulty are the false positive (FP) cases as shown in Figure 11. The cost associated with false negative (FN) and false positive (FP) cases could be very different depending on the business case. The optimal cutoff point should take these costs into consideration so that the total expected cost is minimized. For example, if the cost for false negatives is set very high compared to false positives, the cutoff should be set to avoid as many false negative as possible.

Using a simple business case where false negative cost was set to be five times higher than the false positive cost, we tried to find the optimal cutoff point for the model we trained earlier for predicting failure before the 90 days prediction horizon. Candidate cutoff points were searched in the whole region of 0 to 100 percent in 1 percent increments. The cutoff point with lowest average total cost was selected as shown

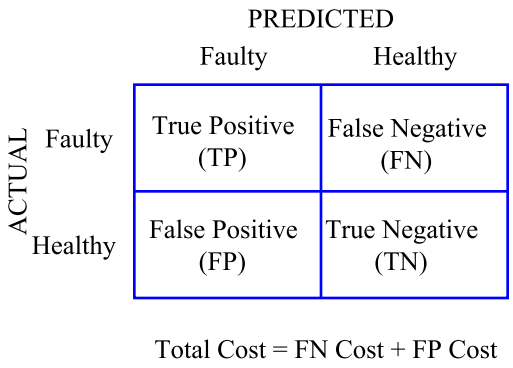

Figure 11. Confusion matrix for fault prediction

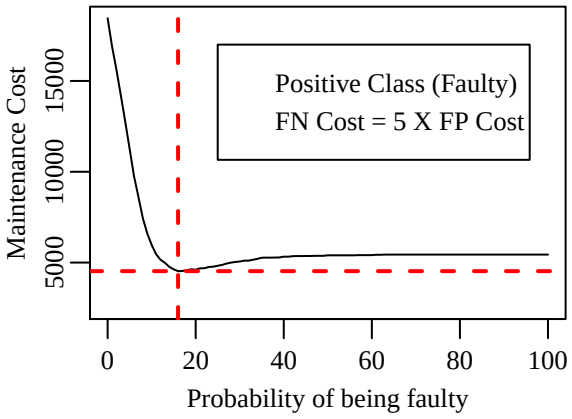

Figure 12. Selecting optimal cutoff point for fault prediction

in Figure 12, which shows that a cutoff threshold at around 16 percent minimizes the total cost. So, a truck with an estimated probability of being faulty equal to or higher than 16% should be called in for a checkup at a workshop.

7. CONCLUDING REMARKS

The primary objective of the paper was to investigate whether operational and environmental data from truck usage can be used for predicting component breakdown. This task is particularly challenging, since the trucks can be configured in many different ways and they operate under very different conditions. An additional complexity is that the task is also dependent on the way in which the driver uses the vehicle. There are hence many variables that can influence the risk for breakdown. Selecting the appropriate variables is hard, and, as a consequence, the task of coming up with an accurate prediction model is quite a challenge. Nevertheless, this paper provides some insights and findings from undertaking such a task.

Experimental data was collected from heavy duty trucks pro-

duced by Scania AB. The information about operation of these trucks were described by features that were expressed as histograms. Not many learning algorithms can train on histogram data. Random forest is one of the best machine learning algorithms for classification and regression purpose. Therefore, it was chosen and was slightly modified to allow it to handle histogram features. It was necessary to make the algorithm learn from histogram as it only seemed natural to treat histograms as they are. This modified algorithm was shown to outperform the standard approach. As evident from the experiment results, it seems that there are some common patterns among trucks with faulty NOx sensors vs. healthy trucks, since the observed AUC measure was observed to be around 0.85 in the best case. This means that faulty and healthy trucks can be quite accurately ranked with respect to risk of failure.

In the experiment with a prediction horizon of three months, even the cases where a failure of NOx sensor was observed after three months had around same average failure probability as the ones whose NOx sensor failed before that margin period. This clearly indicates that there could be some common pattern among the faulty trucks that the model was able to discover, even though the data was manipulated to treat such late failures as healthy ones. From a different perspective, trucks with apparently healthy NOx sensors were deemed to have a high risk of failing by the model. This apparently seems like a right thing to do since the trucks that eventually had faulty NOx sensor were assigned a higher failure probability on average. This also hints towards the fact that the prediction model should be working well.

The explicit decision of whether to call in a truck to a workshop for inspecting the NOx sensor should be done based on a rational selection of the cutoff point, which is used in conjunction with the probability scores output by the model. This selection depends heavily on the business case, such as the costs associated with incorrect predictions, and the choice of threshold should be as to minimize the total cost. For the considered scenario, where the cost of false negatives was set to be five times higher than the cost of false positives, the optimal cutoff point for the estimated probability of a breakdown was found to be 16 percent, hence leading to a decision that trucks with a relatively higher estimated probability for breakdown should be further investigated.

The overall result looks promising and seems to open up more opportunities to conduct research in various directions. This particular study has focused on NOx sensor breakdown, but the overall approach is generic and can be expected to work for any component of interest, as long as the functioning of the component can be determined by available operational and environmental data. One specific aspect that was not considered in this study is how to most effectively handling missing values. Further, only single snapshot for each truck was used for the analysis, in future, investigation could be carried out on how to effectively make use of multiple snapshots. Another direction for future research would be to consider other underlying models, including random survival forests (Ishwaran et al., 2008) to predict survival curves showing the probability of the considered component surviving a certain amount of time, given the current snapshot. This would allow for investigating various horizons even after the model has been built. Another direction for future research concerns the confidence in the predictions. The conformal prediction framework (Devetyarov & Nouretdinov, 2010) (Johansson, Bostrm, & Lfstrm, 2013) allows the user to determine a level of confidence in the predictions, which can be directly used e.g., for the three months prediction horizon experiment. Extending this to survival analysis, e.g., with confidence intervals around the survival curves, is another possible direction for future work.

ACKNOWLEDGMENT

This work has been funded by Scania CV AB and the Vinnova program for Strategic Vehicle Research and Innovation (FFI)-Transport Efficiency.

REFERENCES

Bolander, N., Qiu, H., Eklund, N., Hindle, E., & Rosenfeld, T. (2009). Physics-based remaining useful life prediction for aircraft engine bearing prognosis..

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Daigle, M. J., & Goebel, K. (2011). A model-based prognostics approach applied to pneumatic valves. *International Journal of Prognostics and Health Management*, *2*, 84.

Devarakonda, M., Parker, G., & Johnson, J. (2012, July 31). *Nox control systems and methods for controlling nox emissions.* Google Patents. Retrieved from `http://www.google.com/patents/US8230677` (US Patent 8,230,677)

Devetyarov, D., & Nouretdinov, I. (2010). Prediction with confidence based on a random forest classifier. In *Artificial intelligence applications and innovations.*

Eyal, A., Rokach, L., Kalech, M., Amir, O., Chougule, R., Vaidyanathan, R., & Pattada, K. (2014). Survival analysis of automobile components using mutually exclusive forests. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*, 246-253.

Frisk, E., Krysander, M., & Larsson, E. (2014). Data-driven lead-acid battery prognostics using random survival forests. In *Annual conference of the prognostics and health management society 2014* (p. 92-101).

Gurung, R., Lindgren, T., & Boström, H. (2015). Learning decision trees from histogram data. In *Proceed-*

ings of the 11th international conference on data mining (p. 139-145).

Gurung, R., Lindgren, T., & Boström, H. (2016). Learning decision trees from histogram data using multiple subsets of bins. In *Proceedings of the 29th international florida artificial intelligence research society conference (flairs)* (p. 430-435).

Ishwaran, H., Kogalur, U., Blackstone, E., & Lauer, M. (2008). Random survival forests. *Ann. Appl. Statist.*, *2*(3), 841–860.

Johansson, U., Bostrm, H., & Lfstrm, T. (2013). Conformal prediction using decision trees. In *2013 ieee 13th international conference on data mining* (p. 330-339).

Lawless, J., Hu, J., & Cao, J. (1995). Methods for the estimation of failure distributions and rates from automobile warranty data. *Lifetime Data Analysis*, *1*, 227-240.

Liao, L., & Köttig, F. (2016, July). A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction. *Appl. Soft Comput.*, *44*(C), 191–199.

Lindgren, T., Warnquist, H., & Eineborg, M. (2013). Improving the maintenance planning of heavy trucks using constraint programming. In *Proceedings of the 12th international workshop on constraint modelling and reformulation co-located with the 19th international conference on principles and practice of constraint programming (modref)* (p. 74-90).

Prytz, R., Nowaczyk, S., Rgnvaldsson, T., & Byttner, S. (2015). Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Engineering Applications of Artificial Intelligence*, *41*, 139 - 150.

Sawada, H., & Imamura, S. (2012, July 10). *Nox sensor malfunction diagnostic device and malfunction diagnostic method.* Google Patents. Retrieved from `http://www.google.com/patents/US8219278` (US Patent 8,219,278)

Schubert, F., Wollenhaupt, S., Kita, J., Hagen, G., & Moos, R. (2016). Platform to develop exhaust gas sensors manufactured by glass-solder-support joining of sintered yttria-stabilized zirconia. *Journal of Sensors and Sensor Systems*, *5*, 25-32.

Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). Remaining useful life estimation-a review on the statistical data driven approaches. *European Journal of Operational Research*, *213*, 1-14.

Zhou, Y., & McArdle, J. J. (2015). Rationale and applications of survival tree and survival ensemble methods. *Psychometrika*, *80*, 811-833.