Data Min Knowl Disc (2018) 32:344–384
https://doi.org/10.1007/s10618-017-0538-6

Self-monitoring for maintenance of vehicle fleets

Thorsteinn Rögnvaldsson[1] · Sławomir Nowaczyk[1] · Stefan Byttner[1] · Rune Prytz[2] · Magnus Svensson[2]

Received: 27 November 2016 / Accepted: 7 August 2017 / Published online: 17 August 2017

Abstract An approach for intelligent monitoring of mobile cyberphysical systems is described, based on consensus among distributed self-organised agents. Its usefulness is experimentally demonstrated over a long-time case study in an example domain: a fleet of city buses. The proposed solution combines several techniques, allowing for life-long learning under computational and communication constraints. The presented work is a step towards autonomous knowledge discovery in a domain where data volumes are increasing, the complexity of systems is growing, and dedicating human experts to build fault detection and diagnostic models for all possible faults is not economically viable. The embedded, self-organised agents operate on-board the

Responsible editor: Fei Wang.

This work was supported in part by the Swedish Governmental Agency for Innovation Systems (Vinnova) and the Swedish Knowledge Foundation.

✉ Sławomir Nowaczyk
slanow@hh.se

Thorsteinn Rögnvaldsson
denni@hh.se

Stefan Byttner
stefan@hh.se

Rune Prytz
rune.prytz@volvo.com

Magnus Svensson
magnus.svensson@volvo.com

[1] Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Box 823, 301 18 Halmstad, Sweden

[2] Volvo Group Trucks Technology, 405 08 Göteborg, Sweden

cyberphysical systems, modelling their states and communicating them wirelessly to a back-office application. Those models are subsequently compared against each other to find systems which deviate from the consensus. In this way the group (e.g., a fleet of vehicles) is used to provide a standard, or to describe normal behaviour, together with its expected variability under particular operating conditions. The intention is to detect faults without the need for human experts to anticipate them beforehand. This can be used to build up a knowledge base that accumulates over the life-time of the systems. The approach is demonstrated using data collected during regular operation of a city bus fleet over the period of almost 4 years.

Keywords Data Mining · Knowledge Discovery · Empirical Studies · Vehicle Fleet Maintenance

1 Introduction

Current approaches for equipment monitoring, i.e., fault detection, fault isolation and diagnostics, are based on creating some form of a model, see e.g., Isermann (2006), Jardine et al. (2006), Hines and Seibert (2006), Hines et al. (2008a, b), Peng et al. (2010), Ma and Jiang (2011) for reviews. In the simplest case this model is a range that a signal should be within, but it can be a physics based reference model constructed prior to production and later compared to the actual operation of the system; or a pattern recognition model that is trained from collected data and later compared either to the real performance, or used directly to label the operation as normal or abnormal. Building such models requires a significant amount of "manual" expert work, e.g., trying out different model structures, different feature sets, collecting data, formulating suitable residuals to monitor, etc. These approaches have been very successful with safety critical products like airplanes and nuclear reactors. However, requiring that amount of effort from experts is not a scalable solution, in particular for the future as more systems need to be monitored. A different approach is going to be especially important in the future for complex, mass-produced mechatronic systems where the profit margins are slim. When considerable costs are needed prior to market release, designers must carefully limit which subsystems to have fault detection and diagnostics for, and do without such functions for the rest.

In order to build systems that can handle wide monitoring of today's and tomorrow's mechatronic equipment, it is necessary to take a different approach to health monitoring; an approach that can construct knowledge more autonomously and relies on human experts to a lesser degree. This should be an approach that does not require thinking of all possible faults beforehand, many of which will never happen in practice; that can do the best possible with signals that are already available, and does not need dedicated new sensors; that can scale up to "one more system and component"; that can address modifications and variants; and finally, that can do life-long learning, i.e., will remain relevant throughout the lifetime of the system.

From the industry's perspective real-time and life-long learning is important for maintaining quality in the products and to promptly react to unanticipated problems. Those techniques provide engineers and managers with better understanding of the

equipment they create, how it is being used by the customers, and the problems that they may face. A good illustration, from our results, is the case of engine liquid leaking onto an electronic control unit (ECU) and into the electrical connectors, leading to a short circuit making the coolant fan run continuously at full speed. Such a fault is hard to anticipate for the original equipment manufacturer (OEM), and thus no specific detectors for it are designed. This situation is a result of how components are placed within the engine compartment and where the oil leaks out from. Still, we could observe it repeatedly in the bus fleet, discovering it before either the workshop personnel or fleet operator became aware of it. Several other similar cases have been identified, one of them leading to a patent application, and others sparking further investigations.

The refinement of information from signals to knowledge can be represented by the Data–Information–Knowledge–Wisdom (DIKW) structure introduced in Ackoff (1989), often referred to as the "knowledge pyramid". The DIKW hierarchy describes that data is refined into information, which is refined into knowledge, which is further refined into wisdom (Rowley, 2007). The data level are the original measured signals, i.e., answers to the question of what physical qualities should be captured and how they should be represented. The information level is about answering questions like "who, what and how many", which are typical pattern recognition tasks. The knowledge level is about creating rules and requires combining different information sources, e.g., matching answers or external events to the patterns discovered previously. The wisdom level deals with the "why" question, i.e., reasoning and projecting into the future. Currently, in most cases human experts are performing critical parts of each of these steps, even in artificial intelligence (AI) and machine learning (ML) research: human experts define what signals to measure; human experts supply the training data to supervised ML and pattern recognition tasks; human experts define what information sources to combine when creating rules, etc. Autonomous knowledge creation is about reducing, or even removing, the need for human experts in these steps.

This paper showcases how some steps of the autonomous knowledge construction process can be approached in an industrial setting, and applied to the case of analysing maintenance needs for a city bus fleet. The steps are the autonomous selection of signals for monitoring (i.e., feature selection), without knowing what fault to look at, and detecting deviations under varying ambient conditions by repeated normalisation against a fleet of similar systems, and matching deviations to human curated maintenance records to create hypotheses of causes for the observed deviations. The results from a long-term field study, like the ones presented in this paper, have been analysed by experts at the OEM, leading to increased understanding of the equipment, its usage, and the problems that can occur—in one case, interesting enough to warrant a patent application. Currently, however, there is no clear formalisation of the discovered knowledge; one promising direction concerns identifying the type and amount of data that is worth collecting, processing and storing.

The paper is organised into eight sections, including this introduction, as follows. An overview of related work is provided in the next section, followed by a description of the general framework: the Consensus Self-organising Models (COSMO) approach. Section 4 presents implementation details for three example model families: histograms, linear relations and autoencoders. Additionally, Sect. 4.5 provides an overview of a

method that COSMO results are later compared against. The two subsequent sections describe the vehicle fleet data, together with actual up- and down-time statistics for this fleet. This is followed by presentation of the results from applying the COSMO approach in Sect. 7, and finally the conclusions.

This work demonstrates, in practice, self-exploration and self-monitoring using streams of data on-board vehicles, in combination with off-board data. One contribution is the demonstration of how simple models calculated on-board can be used to autonomously flag deviations related to maintenance needs and, by normalising against a fleet, avoid flagging changes that are due to external circumstances, e.g., season. Another contribution is to illustrate how "interestingness" can be measured for deciding which models to select, without prior information about what problems to monitor. A third contribution is the idea of how one can build a knowledge base from co-occurrences of faults repaired (as described in genuine maintenance records) and disappearances of fault signals.

2 Related work

Self-exploration and self-monitoring using streams of data are central within (for example) the concepts Autonomic Computing, Autonomous Learning Systems (ALS), and Cognitive Fault Diagnostic Systems (CFDS). We cannot provide here a full overview of work in these areas, and restrict ourselves to discussing work related to maintenance prediction, fault detection and diagnostics, and the use of model space to do this.

Particularly relevant within ALS is the work of Filev and Tseng (2006) and Filev et al. (2010), who presented a framework for using novelty detection to build an autonomous system for equipment monitoring and diagnostics, using dynamically updated Gaussian mixture model fuzzy clusters. They assume access to an external "expert" that defines relevant features and comment that this is a critical factor. Their clusters capture different operating modes (e.g., startup, normal, or idle). Those clusters are updated to account for drift and new clusters are created if the equipment is found to operate in a new state. The need for creating a new cluster signals that something could be wrong. The approach requires sufficient computing power on-board to run the Gaussian mixture model fuzzy clusters but does not require any off-board analysis. We have implemented this method (more technical description is provided in Sect. 4.5) and discuss how, in several aspects, COSMO outperforms it on the bus fleet scenario in Sect. 7.

The idea of doing fault detection in model space is not new. This is, for example, what motivates using autoencoders and principal component representations. Linear models were used by Byttner et al. (2007) and also by D'Silva (2008), who used correlations between signals to detect deviations. Linear correlations have been used in the Vedas and MineFleet® systems developed by Kargupta et al. (2004), Kargupta et al. (2010), Kargupta et al. (2007). They monitored correlations between on-board signals for vehicles and used a supervised paradigm to detect faulty behaviours, with focus on privacy-preserving methods for distributed data mining. Alippi et al. (2012), Alippi et al. (2014) used linear relationship models (including time lagged signals) in

their CFDS concept and provided a theoretical analysis that motivates this. Vachkov (2006) showed how self-organised neural gas models could capture nonlinear relationships between signals for diagnostic purposes. Chen et al. (2014) and Quevedo et al. (2014) used nonlinear reservoir computing models to represent the operation of a simulated water distribution network and detect faults through differences between model parameters. All these works, except Byttner et al. (2007), start from the assumption that the relevant features are provided by experts. Furthermore, neither of the approaches consider the system variability, i.e., how effective those solutions are when applied to a group of similar (but not identical) systems.

Lapira et al. (2011) and Lapira (2012) have used groups of systems with similar usage profiles to define "normal" behaviour. They created "peer-clusters" of wind turbines, i.e., ones with similar external conditions, and identified poorly performing ones. Zhang et al. (2009) also used fleets of vehicles for detecting and isolating unexpected faults in the production stage. Recently Theissler (2017) has provided the categorisation of anomalies in automotive data, and stressed the importance of designing detection methods that can handle both known and unknown fault types, together with validation on real data.

The ideas presented here, to use a consensus of self-organised models to detect deviations (faults or maintenance needs) on fleets of vehicles, were originally suggested by Byttner et al. (2007) and Hansson et al. (2008). The initial feasibility study was only done using simulation, showcasing the potential of the method but without validating that it will be able to handle the complexity of real data. First experimental results were presented in Byttner et al. (2009), based on a single heavy duty truck driving the same test route several times with different faults injected. This study was done in a supervised setting, focusing on accuracy of detection, and not on knowledge discovery. Based on several data sets (a city bus on a test track with injected faults, a simulated vehicle cooling system and a computer hard-disk measurements) Byttner et al. (2011) considered the "interestingness" of models, and looked into how to find and select good features on each individual vehicle. The first impressions of the bus fleet study, where the real "consensus" between models could be captured, has been presented in a short term study by Byttner et al. (2013). It was, however, only an overview focusing on linear models as the representation, without technical details and with no comprehensive evaluation of the results. The current work is the first time that we can, in considerable detail, demonstrate the usefulness of the COSMO approach on "off-the-shelf" vehicles (city buses) that are driven under normal traffic conditions, by different drivers, over a long period of time (4 years).

With time, the COSMO method has evolved, as more extensive experiments on real data have lead to identification of deficiencies. For example, the deviation level was initially calculated based on the assumption that the distribution of parameters across the fleet is either Gaussian or a Gaussian mixture model. The "most central pattern" concept, which allows direct use of empirical distribution, was introduced and evaluated using synthetic data in Rögnvaldsson et al. (2014). The current paper is the first time we present an analysis of different faults in different vehicle subsystems. For a specific component, the *air compressor*, Fan et al. (2015a) have recently evaluated the COSMO algorithm and shown how expert knowledge can be incorporated (Fan et al., 2015b); a comparison of performance difference between simple and complex data

models (histograms and Echo State Networks, respectively) was presented in Fan et al. (2016). That direction, however, did not include an analysis of signal "interestingness" nor the search for good data models—the automatic identification of which signals and which representations are suitable for different faults is crucial for a system that claims to be capable of knowledge discovery. In this paper, the "interestingness" concept that has been used previously has been formalised and the paper also includes autoencoders and a much more extensive description on the use of histograms as a representation. Finally, this paper is the first comparison of the results of COSMO algorithm against state of the art solutions, in particular the Evolving Novelty Detection Framework (ENDF) method suggested by Filev et al. (2010).

To summarise, the novelty of our contribution lies in describing the technical details of the COSMO method and experimentally showing that it allows for successful monitoring of city bus fleet in a a real, highly complex scenario, based on a long term field study.

3 The COSMO approach

This paper builds on the Consensus Self-organising Models (COSMO) approach, based on measuring the consensus (or the lack of it) among self-organised models (Byttner et al., 2007). The idea is that models are used to represent the streams of data on-board the systems; and fleets of similar equipments are used to agree on "normality".

A model is a parameterised representation of a stream of data consisting of one or more signals. This can be means, averages, correlations, distributions, or linear/nonlinear functional relationships between signals, signals with different time shifts, and so on. There are endless possible model families, and hierarchies of models of increasing complexity. It is interesting to study methods to automatically select models that are useful for detecting deviations and communicating system status to human experts. In this paper we showcase three quite different examples (histograms, autoencoders, and linear relations between pairs of signals), but this is by no means an exhaustive list.

The COSMO approach is especially applicable in settings where one has access to a fleet of equipments that do similar things, but it is challenging to precisely define the normal operation (for example due to influence of external conditions or differences in usage), where there are on-board data streams on those systems, but where it is expensive (difficult or impractical) to collect and store huge amounts of raw data at an off-board server, and where there is information available about historical maintenance and repairs done to the systems. Examples of suitable scenarios are fleets of buses and heavy duty trucks, or power wind mill parks at sea or in remote areas.

The approach consists of three parts: finding models, detecting deviations and determining causes. The first step is done, either fully or partially, on-board the systems and the two latter are done off-board.

Looking for clues This corresponds to the data level in the DIKW hierarchy. Self-organising systems need to be able to collect information about their own state of operation; clues that can be communicated with other vehicles and any supervisory

system. This can be done by embedded software agents that search for interesting relationships among the signals available inside internal Electronic Control Units (ECUs). Such relationships can be encoded in many different ways, e.g., with histograms or probability density models describing a single signal, with linear correlations expressing relations between two or more signals, or with principal components, autoencoders, self-organising feature maps and other clustering methods. The choice of model family can be influenced by domain knowledge, but a self-organising system should be able to take any sufficiently general one and make good use of it.

Useful (interesting) relationships are those that look far from random, since they contain clues about the current state of the equipment. If, e.g., a histogram is far from being uniform, or a linear correlation is close to one, or the cluster distortion measure is low, then this is a sign that this model can be useful for describing the state of the system and then, possibly, also for detecting faulty operation. At the same time, the variation in the models is also considered an aspect of their interestingness. Relationships that have a large variation, i.e., shift a lot from one time to another, or from one vehicle to another, will be difficult to use for detecting faults. On the other hand, changes occurring in models that are otherwise usually stable are likely to indicate meaningful events.

This shows that there are many criteria one should consider when looking for suitable methods to distinguish interesting relations from irrelevant ones. In the city bus scenario presented here, it is particularly beneficial if this initial screening of models can be done on-board individual vehicles, without the need to know about the rest of the fleet. In this paper we explore two examples of such methods. The first is the *stability* of the models for a single vehicle, i.e., how much does 1 day differ from the next. The second is the *randomness* of the models, which we capture using entropy for histograms and relation strength for linear functions. As an example of a measure that also considers others aspects, looking beyond a single bus, we showcase the *consistency* of the models across the whole fleet.

Clearly, the fact that a model is considered "interesting" based on the above measures does not guarantee that it will be useful for fault detection and diagnostics. However, if it is not "interesting", then it is unlikely to contain information that can be used for that purpose. The goal of this stage is simply to weed out bad candidates, to make the subsequent steps more efficient.

Consensus in parameter space This corresponds to the information level in the DIKW hierarchy. In this step, all equipments compute the parameters for the most interesting model configurations and send them to a back-office server. The server then checks whether the parameters from different systems are in consensus. If one or more equipments disagree with the rest, then they are flagged as deviating and potentially faulty. There are many ways such a consensus test could be performed. It is an outlier or anomaly detection problem and there is a huge body of literature on this subject. Some recent reviews on this topic are: Chandola et al. (2009), Gogoi et al. (2011), Gupta et al. (2013), Patcha and Park (2007), Pimentel et al. (2014), Sodemann et al. (2012), Xie et al. (2011), Zhang (2013), Zimek et al. (2012). Furthermore, Laxhammar (2014) recently introduced the conformal anomaly predictor based on the work by Vovk et al. (2005), which is not covered in previous reviews. Our approach is similar to this conformal anomaly detection, as discussed by Rögnvaldsson et al. (2014). Most of

the available anomaly detection approaches can operate on data of almost any kind, only requiring a suitable distance metric to be defined between models. In the setting presented here it is desirable that the test produces a p value, i.e., the probability, given that the null hypothesis is true, for drawing a sample that is less likely than the observed sample. This allows for proper statistical handling when not only one, but several samples are drawn from each system, for example over a longer time period.

Fault isolation and diagnosis This corresponds to the knowledge level in the DIKW hierarchy. When a deviation is observed in the parameter space, then this particular system is flagged as potentially faulty. The next step is to diagnose the reason for the deviation. One way is to compare against previous observations and associated repairs, using a supervised case-based reasoning approach. It requires a somewhat large corpus of labelled fault observations; however, for most modern cyberphysical systems such data are available in the form of maintenance databases or repair histories.

Currently, in many domains, there are unfortunately practical problems that originate from the fact that maintenance databases have been designed with different purposes in mind and that the data is input manually. The specifics very much depend on individual systems, components and faults, but in general there are a number of quality issues with maintenance databases, including missing and superfluous repairs, lack of information about fault modes, and more, some of which are discussed later in this paper. Nevertheless, we show here that interesting results can be obtained already now, possibly with some manual curating of the data, and we argue that once the usefulness of the data is shown, its quality will continuously improve.

4 Data models

This section presents the implementation details for three examples of the COSMO approach that have been successfully applied in the field test. The first two are based on analysis of individual signals, modelled using histograms and autoencoders, and the third uses binary relationships in the form of linear functions.

4.1 Histograms

A single signal can be described using a one-dimensional histogram. It is a very robust representation, although it removes the time series characteristics of the signal. It is, for modern vehicles, possible to automatically compute key features like bin widths and signal ranges, since all signals that are communicated on the vehicle CAN (Controller Area Network) are described in databases that specify their digital representation (i.e., number of bits, min and max values, etc.). Essentially all on-board signals have special values that denote "out of range" or "error". These can be handled in different ways. In the experiments reported here they have been removed, in order to show the information that can be extracted from seemingly "normal" data.

We measure the "interestingness" of histograms in terms of their entropy (i.e., how random the signal is) and their stability (i.e., how much the histograms vary between two consecutive times). The entropy of histogram $\mathbf{P} = (P_1, \ldots, P_N)$ is defined as

$$E = -\sum_{i=1}^{N} P_i \log(P_i). \tag{1}$$

where P_i is the normalised frequency of data in bin i.

The entropy is dependent on how the bin sizes are chosen; it is proportional to the logarithm of the number of bins in the histogram. Thus, to enable comparison of two histograms with different number of bins (two different signals), a normalised entropy difference

$$\mathrm{NE} = \frac{\log(N) - E}{\log(N)} = 1 + \frac{1}{\log(N)} \sum_{i=1}^{N} P_i \log(P_i) \tag{2}$$

is used as a measure of a histogram's "interestingness". Furthermore, instead of N being the number of bins in the histogram, N is set to the number of occupied bins, to remove the effect of many empty, unused, bins.

The normalised entropy difference fulfils $0 \leq \mathrm{NE} \leq 1$. A low value of NE indicates a histogram where the data are spread evenly over all bins, which is an "uninteresting" case. A high value of NE, on the other hand, indicates that most of the data are concentrated in few bins, which is a more "interesting" situation. Thus, "interestingness" increases with NE. One particular exception is when NE equals one, since this corresponds to a constant or near constant signal, with only one bin occupied all the time. The normalised entropy difference, for a reasonable data distribution and histogram binning, will typically be below 0.5.

Another aspect is the variation in the histograms, i.e., how much they shift from one time to another, for the same vehicle. Many measures have been suggested for quantifying the difference between two histograms (discrete probability densities) and there is no "best" method for doing this. A review by Cha (2007) covers many measures, although not the earth mover's distance introduced in Rubner et al. (2000), and discusses the relationships between them. Pele (2011) presents a good overview of desired characteristics of histogram distance measures, emphasising the need for cross-bin distances in image retrieval tasks.

We used the Hellinger distance in this work since it has several attractive features. It is a proper metric, it is quick to compute and it has no problem with empty bins. The square of it is an f-divergence measure (Csiszár and Shields, 2004). It is not a cross-bin distance but this should not present a problem since the histogram domains are well aligned (Pele, 2011). The Hellinger distance between two histograms $\mathbf{P}$ and $\mathbf{Q}$ with N bins is defined as

$$H(\mathbf{P}, \mathbf{Q}) = \sqrt{\frac{1}{2} \sum_{i=1}^{N} \left(\sqrt{P_i} - \sqrt{Q_i} \right)^2} \tag{3}$$

where P_i and Q_i are the bin values for $\mathbf{P}$ and $\mathbf{Q}$, respectively. If $\mathbf{P}$ and $\mathbf{Q}$ are probability histograms then it fulfils $0 \leq H(\mathbf{P}, \mathbf{Q}) \leq 1$.

Once a set of interesting signals to monitor is determined, and appropriate histograms are calculated across the fleet, the central server is used to detect deviating

vehicles. To this end a set of histograms is sampled from the fleet. This could be, e.g., daily histograms for one signal over a week. If all vehicles operate for reasonable amount of time each day and we have M vehicles, then this gives $L = 7 \times M$ histograms. The pairwise Hellinger distances between the L histograms are computed, yielding a symmetric distance matrix that can be used for detecting deviations (as described in Sect. 4.4).

4.2 Autoencoders

Histograms are very robust representations but they do not capture the dynamics of a signal. This dynamics can be (and often is) a very important aspect and one should therefore explore models that can represent the time series characteristic of the signal. In this section we propose a model capable of capturing those aspects. We split the original signal time series into windows of fixed length and train an autoencoder to reconstruct them.

The autoencoder, introduced by McClelland and Rumelhart (1988), is an artificial neural network trained to reproduce the input as its output. They are commonly used for dimensionality reduction, due to their capability of learning compressed representations of the data. The input data $\mathbf{x} \in [0, 1]^d$ is first mapped into a hidden (or latent) representation, $\mathbf{y} \in [0, 1]^{d'}$ using a deterministic mapping:

$$\mathbf{y} = s\left(\mathbf{W} \cdot \mathbf{x} + \mathbf{b}\right), \tag{4}$$

where s is a non-linear function, $\mathbf{W}$ is a weight matrix, and $d' < d$. This mapping is used to reconstruct an output $\hat{\mathbf{x}}$ that should be as close as possible to the input data:

$$\hat{\mathbf{x}} = s\left(\mathbf{W}' \cdot \mathbf{y} + \mathbf{b}'\right), \tag{5}$$

The hat denotes that $\hat{\mathbf{x}}$ is a reconstruction (a model) of $\mathbf{x}$ and not the signal itself.

During the training phase, the weight matrices $\mathbf{W}$ and $\mathbf{W}'$ of the autoencoder are adjusted so that the reconstruction error is as low as possible. There are many ways to measure this error, and we used cross-entropy:

$$L_H(\mathbf{x}, \hat{\mathbf{x}}) = -\sum_{k=1}^{d} \left[\mathbf{x}_k \cdot \log \hat{\mathbf{x}}_k + (1 - \mathbf{x}_k) \cdot \log\left(1 - \hat{\mathbf{x}}_k\right)\right]. \tag{6}$$

The assumption is that $\mathbf{y}$ will capture the main factors of variability in the original data. In a sense it can be thought of as a non-linear version of principal components. Since the compression provided by $\mathbf{y}$ is lossy, the reconstructed output will closely match the input only if the data comes from the training distribution, and the autoencoder will demonstrate high error on samples chosen from other parts of the input space. The training procedure can be extended, in so-called denoising autoencoders, with an additional step of introducing noise into the input data during training, which improves the robustness of the discovered features and leads to less overfitting.

We used a denoising autoencoder with 150 hidden neurons, trained over 1 day of vehicle operation, split into windows of 250 samples. We used 5000 training epochs for each model, and a corruption level of 10%.

There are many different ways to compare trained autoencoders, i.e., measure the distance between two models. We chose the simple method of comparing the reconstructed signals over a predefined reference data set. In our experiments we have used a data selected randomly, averaging one sample window per month per bus. However, if an actual reference data corresponding to typical usage and wear patterns is available, it can be used instead.

Given the reference data $\mathbf{x}_r$ and two autoencoders a and b, which have been originally trained on data $\mathbf{x}_a$ and $\mathbf{x}_b$, respectively, we calculate $[\mathbf{z}_{ar}, \mathbf{z}_{br}]$, where $\mathbf{z}_{ij}$ is the reconstruction of data $\mathbf{x}_j$ by the autoencoder i. We then use the Normalised Mean Square Error (NMSE) to compare the outputs corresponding to the same $\mathbf{x}_i$:

$$D_{ab} = \mathrm{NMSE}_{ab} = \frac{1}{N\sigma^2} \sum_{n=1}^{N} [\mathbf{z}_{ar} - \mathbf{z}_{br}]^2 \tag{7}$$

This is based on the assumption that if both autoencoders were trained on inputs similar to the reference data, they will both reconstruct it quite well, and their outputs will be similar. On the other hand, if $\mathbf{x}_a$ and $\mathbf{x}_b$ are different and correspond to different underlying system behaviour, each autoencoder will make a large reconstruction error on the reference data, and thus their outputs are likely to differ significantly.

Interestingness of autoencoder models can be, in principle, measured in an analogous fashion to that of histograms. Randomness of a signal is captured by, instead of entropy, the reconstruction error (or a relation between reconstruction error and autoencoder complexity, i.e., the number of hidden neurons d'). The stability is still the similarity between the models calculated at two consecutive times. However, in the experiments reported in Sect. 7.2 we did not use interestingness measures for autoencoders at all, since we have decided to focus on a particular subsystem that was known by the fleet operator to be important and especially problematic.

As with histograms, the central server is used to detect deviating vehicles. A set of L models (e.g., daily autoencoders for one signal over a week) is collected from the fleet and the pairwise distances between them are calculated, yielding a symmetric distance matrix that can be used for detecting deviations (as described in Sect. 4.4).

4.3 Linear functions

In this section we propose the procedure for using linear functions as models. This approach allows us to capture not only the characteristics of an individual signal, but also the relations between pairs of signals. Such relations are often important artifacts of design decisions of physical properties, and their disturbance can be a valuable indicator of various faults.

The first step is to generate all pairwise linear model combinations between two signals x_i and x_j, where $i \neq j$, in the form of

$$\hat{x}_i = a_{ij} x_j + b_{ij} \tag{8}$$

If the total number of signals is K, then there are $K \times (K-1)$ different models for each vehicle: each pair of i and j, where order is important. The hat denotes that $\hat{x}_i$ is a model of x_i and not the measured value.

For each model ij we then calculate interestingness metric that estimates its potential usefulness for monitoring. As in previous two sections, the first aspect of this interestingness is how far from random the data is. We denote this value as α_{ij}. Here, instead of using entropy, we use the accuracy of each model, i.e., the strength of each pairwise signal relation. It is based on computing the NMSE for each model ij:

$$\mathrm{NMSE}_{ij}^{v} = \frac{1}{N\sigma^2}\sum_{n=1}^{N}[x(n)-\hat{x}(n)]^2 \tag{9}$$

where σ is the standard deviation of the signal x (the index on x is dropped for notation simplicity). For each model, the α_{ij} value is the average NMSE that was found onboard the vehicles in the fleet;

$$\alpha_{ij} = \frac{1}{V}\sum_{v=1}^{V}\mathrm{NMSE}_{ij}^{v} \tag{10}$$

where V is the total number of vehicles in the fleet.

The second component if interestingness, denoted β_{ij}, captures how much the model parameters vary across the fleet. For each vehicle, the maximum Euclidean distance to model parameters of all other vehicles is computed:

$$d_{ij}^{v} = \max_{w=1..V}\left(\sqrt{(a_{ij}^{v}-a_{ij}^{w})^2+(b_{ij}^{v}-b_{ij}^{w})^2}\right). \tag{11}$$

The β_{ij} value is then defined as

$$\beta_{ij} = \sqrt{\frac{1}{V}\sum_{v=1}^{V}(d_{ij}^{v}-\bar{d})^2} \tag{12}$$

where $\bar{d}$ is the average d, i.e.,

$$\bar{d} = \frac{1}{VKK}\sum_{v=1}^{V}\sum_{i=1}^{K}\sum_{j=1}^{K}d_{ij}^{v} \tag{13}$$

The general procedure for finding an interesting model is thus to compute all pairwise linear combinations of signals on board each vehicle. For each model, an α_{ij} value is computed to determine what are the strong deterministic signal relations, as measured by the NMSE. A model where there is a strong linear relationship should have a small α_{ij} value. The β_{ij} value quantifies the variation in model parameters among the vehicles in the fleet. A large β_{ij} means that the models from each vehicle in

the fleet show a large variation, indicating that there is something potentially different about a vehicle. An interesting model for monitoring purposes is thus characterised by a small α_{ij} value and a large β_{ij} value.

It is possible that a useful model shows a small α_{ij} and a small β_{ij} since a fault has not occurred yet, which is why the search should be repeated from time to time, in order to discover those models as interesting. Once an interesting relation has been found, appropriate models are calculated across the fleet and the central server is used to detect deviating vehicles. To this end each vehicle v calculates a_{ij}^v and b_{ij}^v (as in Eq. 8) based on data from a fixed period of time (e.g., 1 day). The distance between two vehicles is then calculated as the Euclidean distance between model parameters in 2-dimensional space, i.e.:

$$D_{vw} = \sqrt{\left(a_{ij}^v - a_{ij}^w)^2 + (b_{ij}^v - b_{ij}^w\right)^2} \tag{14}$$

Again, the central server is used to detect deviating vehicles. A set of L models (e.g., daily linear relations for one pair of signals over a week period) is collected from the fleet and the pairwise distances between them are calculated, yielding a symmetric distance matrix that can be used for detecting deviations (as described in Sect. 4.4).

4.4 Detecting deviations

The deviation detection step requires having a metric for measuring the distance between models. That is, for N models, there exists a symmetric matrix

$$\mathbf{D} = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ d_{21} & d_{22} & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{NN} \end{pmatrix} \tag{15}$$

with pairwise distances $d_{ij} = d_{ji}$ between models. The statistics for finding outliers are computed in a leave-one-out fashion. It is therefore important to have sufficient number of models so that it makes sense to define a "normal" set even when one model is left out. This can be achieved either by having many vehicles with one model per vehicle, or by collecting several models for each vehicle. In our case, with a fleet of 19 vehicles, we collect daily models for each vehicle over 1 week, so that $N \leq 7 \times 19 = 133$ (sometimes vehicles drop out or are in repair). The process is then as described by Rögnvaldsson et al. (2014). All distances related to one bus, the "test bus", are removed from the matrix. In the remaining data, the row with the minimum row sum is selected and denoted "the most central model" c. The set of distances from all the other models to c are then used as the empirical distribution of fleet dissimilarities. The tail-frequency (z-score) for a test model m from the "test bus" is then estimated as the ratio of models in the empirical distribution that lie further away from the most central model c than this test model m, i.e.:

$$\text{z-score} = \frac{|\{i = 1, ..., N_s : d_{ic} > d_{mc}|}{N_s} \tag{16}$$

where d_{ic} is the distance from model i to the most central model and N_s is the size of the distance sub-matrix when the "test bus" data have been removed. The most central model is not included in the empirical distribution.

This "most central pattern method" is a simple and intuitive algorithm, which builds on the observation that if one uses the Euclidean distance metric then the sample that is most central, i.e., closest to the mean of all the samples, will be the sample that has the minimum row sum in the distance matrix **D**. The method is very similar to a conformal anomaly detection (Laxhammar, 2014) method with the distance to the most central pattern as conformity measure. Different methods for estimating the tail-frequency are explored and tested by Rögnvaldsson et al. (2014), with the conclusion that the method using the most central pattern works well in practice (it obviously does not handle multimodal distributions correctly).

Based on the z-score we want to estimate whether the test bus is operating in the same manner as the fleet. The null hypothesis is that all the distances d_{ij} are drawn from the same distribution, i.e., that the models are all drawn from the same distribution. Under this hypothesis the repeated samplings of z values are uniformly distributed between zero and one. We use the average of the "test bus" z values as a statistic and the one-sided (since we are only interested in vehicles that are far away from the fleet centre) p value for this average is calculated. If the average of the z values, over some suitable time horizon, for a bus has a p value below a pre-specified threshold, then that bus is marked as an outlier.

It is tempting to interpret this threshold statistically and e.g., guarantee a maximum number of false alarms. There are, however, many things that occur in reality so that this statistical interpretation does not hold. One practical issue is how to deal with longer repairs. It happens that the vehicles are in repairs for 10–15 days, which is a significant part of the time horizon. The ambient and operation conditions in the workshop hall are much different from the normal conditions experienced by the vehicles in the fleet. This can mean that the average z-score starts to deviate during the repair period, and will continue to deviate until the workshop time period has moved beyond the time horizon, even though the vehicle is working fine. This would be easy to deal with if there was an electronic indicator on the bus flagging that the bus is in repair, or if the dates and mileages in the service record database were always correct. Then data during repair times would be simple to just remove. However, this is not the case (yet) and there will therefore be more deviations in the data than there are faults.

4.5 Evolving novelty detection framework

To compare the results of COSMO against existing state of the art solutions, we have implemented the Evolving Novelty Detection Framework (ENDF) method based on Filev and Tseng (2006), Filev et al. (2010). In subsequent sections we present the results of analysing the same data using both techniques, and discuss some key differences. The authors describe the ENDF method as

> [...]a practical framework for autonomous monitoring of industrial equipment based on novelty detection. It overcomes limitations of current equipment monitoring technology by developing a "generic" structure that is relatively independent of the type of physical equipment under consideration. The kernel of the proposed approach is an "evolving" model based on unsupervised learning methods (reducing the need for human intervention). The framework employs procedures designed to temporally evolve the critical model parameters with experience for enhanced monitoring accuracy[...]

The ENDF method is based on fuzzy clustering, where data describing equipment operation are assumed to belong to several "significantly different, but repetitive machine signatures". Those so called "Operating Modes" (OMs) can correspond to different usage patterns, different external conditions, etc. They are first learned during an initialisation phase, and can later evolve, following the gradual changes in machine characteristics. Additionally, new clusters can be created when a big change in the feature space occurs.

Deviations are detected by continuously tracking "health status" for each cluster, based primarily on the number of data points assigned to it and its age. The idea behind this approach is that different regions in feature space are associated with different operating modes of the machine, either normal ones or those corresponding to a fault. However, faults are expected to include fewer data points, and to have limited life. Therefore, the deviation level of a machine can be measured based on the health status of the most recently visited clusters. In our data, the consecutive data points can vary a lot, due to variations in usage and in external conditions, which means that some form of aggregation (e.g., a mean) over time is necessary. We have decided to use the same time horizon as we use for z-scores in COSMO.

The ENDF method consists of three phases. First is the *Setup Phase*, including feature extraction and selection. Since our goal here is to compare the ENDF with COSMO, we have used exactly the same features and signals for both of them, according to the selection process described in the previous sections.

The second is the *Initialisation Phase*, where data are collected until the formative definitions of operating modes can be created. We followed the authors' recommendation to use Principal Component Analysis for dimensionality reduction. In order to estimate the parameters of the initial clusters we used the infinite Gaussian Mixture Model (GMM) with the Dirichlet Process (Blei and Jordan, 2006), since it is capable of estimating the number of clusters from the data. In the experiments we have used the first 200 days as the initialisation period.

The final phase is the *Monitoring Phase*, which consists of two main steps: updating of the OMs based on new data, and tracking of OM dynamics. For each new data point, the first step is to identify if it belongs to one of the existing clusters. If so, the parameters of this cluster are updated, including the mean, the covariance matrix, the age and the number of feature vectors (we have used a learning rate of 0.95). This is the expected behaviour, where new data matches one of the existing, common operating modes of the machine. The health status of the commonly visited clusters continuously grows, and they evolve over time to capture any gradual changes in machine characteristics. However, if neither of the existing clusters is a good enough

match for the new data point (as measured using a χ^2 test), a new cluster is created and added to the GMM. This new cluster could correspond to a previously unseen machine operating mode, or could be an indication of a fault.

Prediction of equipment faults is done based on the health status of the most recently visited clusters, as well as on the tracking of OM dynamics in order to predict the future trajectory of the machine operation. In our data, however, the predictions of cluster trajectories was very unreliable. The idea is that creation of many new clusters, with very few data points belonging to each of them, can be an indication of incoming fault. Essentially, if current observations fall into a region in feature space that was previously unoccupied, or into a region of space that is rarely observed, there is a high chance that the machine is not working correctly.

One issue with the ENDF approach is that finding the correct threshold for what is considered a "significant" deviation is not easy. The cluster health status for the ENDF method is between 0 and 1, where 0 means "good" and 1 means "not good". The value of 0 virtually never occurs, which is not very intuitive; a correctly operating equipment should be very close to "good" most of the time.

In order to compare the results between ENDF and COSMO, the trigger threshold was set so that 10% of the data were above this threshold. This means that both methods provide the same fraction of warnings. For visualisation (in figures) we linearly scaled the health status values for ENDF so that the lowest value equals 0, and both approaches have the same threshold.

5 Description of the data

The on-board data used in this study were collected between August 2011 and August 2015 on a bus fleet with 19 buses in traffic around a city on the west coast of Sweden. Four buses in the fleet were from 2009, one from 2008, and the remaining 14 from 2007. Each bus was driven approximately 100,000 km per year and the data were collected during normal operation.

More than one hundred on-board signals were sampled, at one Hertz, from the J1587 diagnostic bus and two of the CAN buses (the vehicle and the powertrain CANs). The vehicle positions were also sampled from a GPS receiver. The sampling equipment used was an in-house developed system called the Volvo Analysis and Communication Tool (VACT). This system is connected to a telematics gateway and can receive new sampling configurations wirelessly. It can both sample data for later analysis and return snapshots of relationships between sampled signals. The VACT system is non-invasive in the sense that it only listens to the data traffic on the network and does not affect the communication itself.

Data were, for the purpose of this research, also stored on USB sticks and collected periodically to allow more detailed analysis off-board. However, the idea of the VACT system and the algorithms described in this paper is not to collect raw data but only communicate models (compressed representations of the data) over a wireless link. The vehicles were not modified in any way for this project, except that a VACT system was installed on each bus to listen to the data streams.

The off-board data consists of the Vehicle Service Record (VSR) database. It collects information about all services that have been done on the vehicles. Each entry contains information about date, mileage, parts, operations, and free text comments by the workshop personnel. The VSR data builds on information that is entered manually by maintenance personnel and there are significant quality issues with it. Furthermore, the VSR is primarily designed for keeping track of invoicing, which means that while the parts and operations records are quite accurate, the date and mileage information is less than perfect. This was partly curated by comparing with GPS data for the vehicles and information from the bus operator's notebooks, where service dates were sometimes (but far from always) marked.

The bus operator has a maintenance solution that may be typical for a medium sized European city. All buses are on service contracts offered by the original equipment manufacturer (OEM) and should be taken to OEM garages for repairs. The OEM contract also includes on-road service that is available around the clock, at an additional cost. The OEM workshops, however, are about an hour's drive away from the bus operator and considerable time can be lost in the transport. A sub-contractor repair shop down the road from the bus fleet garage was therefore sometimes used for maintenance and repairs, which saved a lot of transportation time. Unfortunately, this decreased the VSR information quality significantly. Sub-contractors' operations are seldom entered into the database immediately; the typical case is that they are entered into the VSR database with dates that lie months after the actual repair. In this case the sub-contractor's mileage values were also more erroneous than the OEM workshop's.

The bus data were complemented with interviews with some of the bus drivers and the bus operator regarding quality issues with the buses.

6 Uptime and downtime for the bus fleet

For a bus operator the important metric is the "effective downtime", i.e., the amount of time that a bus is needed but not available for planned use (transportation). The effective downtime depends, in part, on how many "spare" buses the operator has (the more "spare" buses the less risk for effective downtime). In this case, the bus fleet operator's goal was to have one "spare" bus per twenty buses, i.e., that the effective downtime should be at most 5% and the bus operator took very good care of the vehicles in order to meet this goal.

Even with the interviews, there is no reliable way to measure the effective downtime for the bus operator. However, we could compute the times the buses spent in a workshop, in transportation to or from a workshop, or broken down on the road. This was done by analysing the VSR entries and the GPS signals for the fleet during the 4 years we observed it. The buses spent on average 11% of the time in or at workshops and in transportation to and from workshops. This number varied a lot between the vehicles; the lowest was 8%, the highest 19%, and the median 11% (i.e., the same as the average). These numbers do not include temporary stops on the road, which did happen. However, the numbers include weekends when the bus was in the workshop on the preceding Friday and following Monday.

During the data collection period, the buses had, on average, about eleven visits to workshops per year, which is more than twice the number of planned maintenance visits. They spent an average of 4 days in or at the workshop, or in transportation to or from the workshop, each visit. The variation in visit length was significant: the shortest were about half a day (visits shorter than half a day were not considered as downtime) and the longest was 35 days. In addition, a number of on-road assistances were needed.

It is notable that these uptime and repair statistics are very similar to those reported for US heavy duty trucks Riemer (2013a, b) so they are probably not specific for this city bus operation.

There are many reasons for the fairly high number of days at workshops. A big part is the waiting time; the bus is in the workshop but no work is being done on it. This means that the workshops optimise their performance (vehicles are available when work can be done on them) but the bus operator is unable to run an optimal operation. Some of the waiting time is due to insufficient planning. Unplanned repairs often lead to long waiting times while resources in the workshop are being allocated for the repair.

Thus, an ability to detect early symptoms of wear, before they become problems, and to deal with them during a planned maintenance visit has the potential to significantly decrease the downtime. The buses in the field study spent, on average, almost 1.5 months per year in workshops, including transportation time there and back again. It is not unrealistic to expect this time to shrink by half by using maintenance prediction methods and by decreasing the waiting time. We observed during 2015 that the maintenance procedures were being changed and waiting times shortened significantly.

7 Application of COSMO to the city bus fleet

The following examples have been selected in order to showcase the feasibility of the self-organising approach, but at the same time indicate several practical problems that emerge when applying it in a realistic setting. We believe that this research field is now mature enough that evaluations can (and should) be done not only on artificial data sets, but actually taking into account the full complexity of real-world scenarios. To this end, the first example is a relatively easy one, based on a signal that does not contain many deviations, which makes it possible to hand-pick, analyse and understand each of them. The second example is more complex, where we focus on a component that was a very problematic one from the fleet owner's perspective. In this case both the number of deviations and the number of repairs is too large to allow for as detailed analysis as in Sect. 7.1. Finally, our third example is chosen to demonstrate the ability of the system to not only detect deviations, but to also assign labels to them.

7.1 Histograms

The process described in Sect. 4.1 was used. Figure 1 shows the average normalised entropy (NE) plotted versus the stability (average Hellinger distance between two

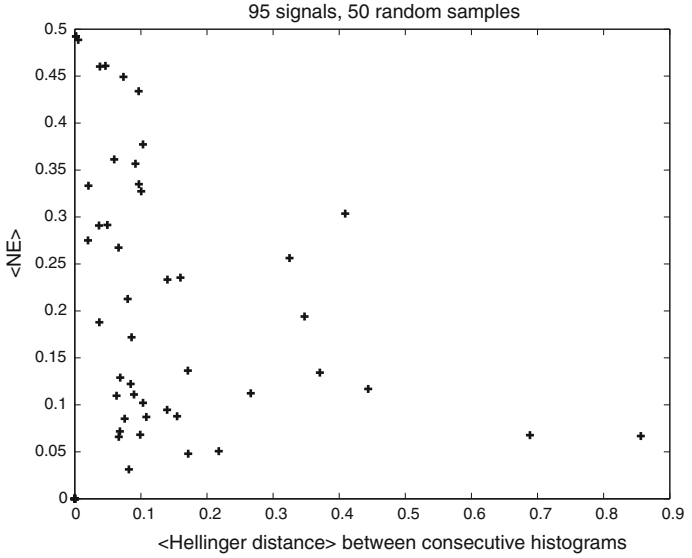

Fig. 1 "Interestingness" for the signal histograms. Angle brackets in axes labels denote averages. The *upper left corner* corresponds to histograms that are peaked around a few bins and that are fairly stable with time. The *lower left corner* corresponds to flat distribution histograms that are stable. The *lower right corner* corresponds to flat distributions that are non-stable. The *upper left corner* histograms are the most interesting and the lower right corner histograms are the least interesting

consecutive histograms) on the same vehicle, for all signals with nonzero entropies (about half the monitored signals had zero entropy). The values were estimated by collecting 100 daily histograms for 50 random dates and buses, i.e., two consecutive histograms each time. Consecutive histograms were chosen to decrease the effect from changes in ambient conditions. The upper left corner corresponds to the most interesting signals. The least interesting signals are those in the lower right corner.

Figure 2 shows example histograms from different parts in Fig. 1. In the far upper left corner, at (0.00, 0.49), is the *Cooling Fan* signal. This is a discrete control signal that has the same value most of the time (more than 99% of the time). This histogram has very low entropy and is very stable. In the lower right corner, at (0.86, 0.07), is the *Transm. Oil Temp*. This is the measured temperature for the transmission oil, which has high entropy and also high variation between consecutive histograms (not shown). The two other histograms represent points in between those extremes: *Boost Pressure* is located at (0.07, 0.13) and *Engine Coolant Temperature* is located at (0.33, 0.26).

Several of the signals close to the upper left corner are the relative speeds of the wheels, which are described in more detail in Sect. 7.3. One interesting signal close to the upper left corner is the *Coolant Gauge %*, located at (0.10, 0.43) in Fig. 1. This signal is the coolant gauge on the dashboard, which shows a filtered version of the coolant liquid temperature. It is equal to 50% (the center of the scale) most of the time during normal operation.

The upper plot in Fig. 3 shows the z statistic (cf. Eq. 16) for one of the buses when histograms of *Coolant Gauge %* are used as models (see Sect. 4.1 for details on

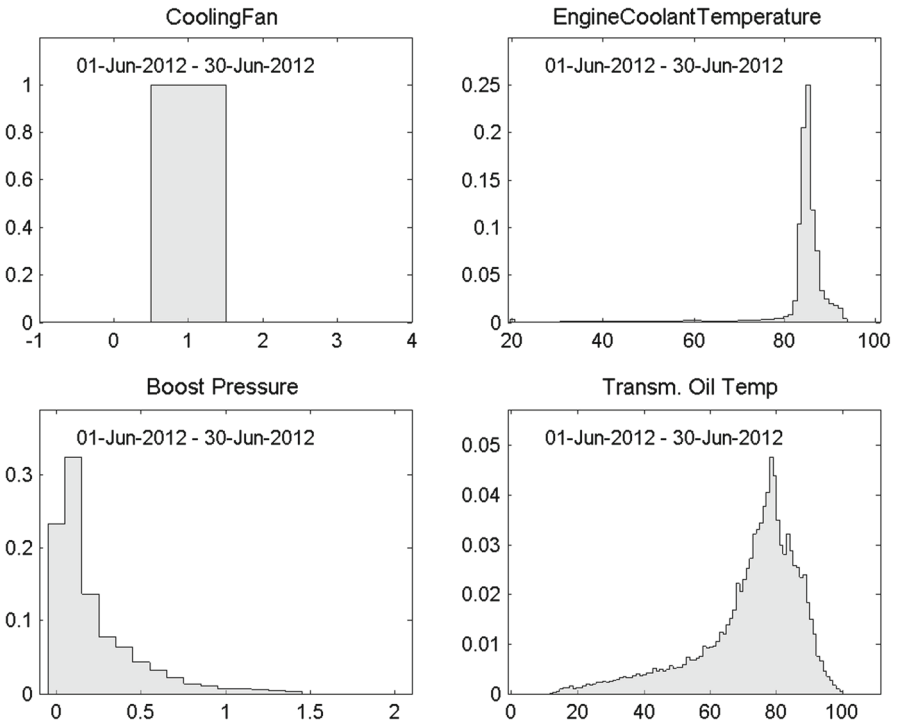

Fig. 2 Examples of histograms in different parts of the "interestingness" graph (Fig. 1). The shown histograms are average histograms for all vehicles in the bus fleet during June 2012. The *upper left plot* shows the *Cooling Fan* signal, which is a discrete control signal to the cooling fan. The *upper right* shows the *Engine Coolant Temperature* signal, which is the measured temperature of the engine coolant fluid. The *lower left* shows the *Boost Pressure* signal, which is the measured pressure after the turbocharger. The *lower right* shows the *Transm. Oil Temp* signal, which is the measured temperature of the transmission oil

the method). The z statistic is uniformly distributed between 0 and 1 under the null hypothesis that all histograms are drawn from the same distribution. The lower plot in Fig. 3 shows the p value for the arithmetic mean for the z statistic, for the same bus, when computed over a moving window of the previous 30 days. Occasionally some data are lost, so the moving average does not always include 30 values.

This particular bus (J) deviated from the fleet already during the first months of the data collection. At the beginning of the study we set the threshold for a significant deviation at p value of 10^{-5}. This value corresponded, if measurements were independent and our null hypothesis would hold, to less than one false alarm per 10 years per signal (we monitor 100 signals) if the vehicles were monitored daily (for the whole fleet of 19 buses). This initial deviation disappeared at the turn of October and November 2011, during a repair visit that lasted 28 days. One of the repairs that were done during this visit concerned a broken Electronic Control Unit (ECU); liquid has leaked into a contact and shorted the circuit, resulting in the cooling fan always running at full speed. The engine temperature distribution was, as a consequence, different from the fleet. However, this difference seems to not have been noticed by the customer. It

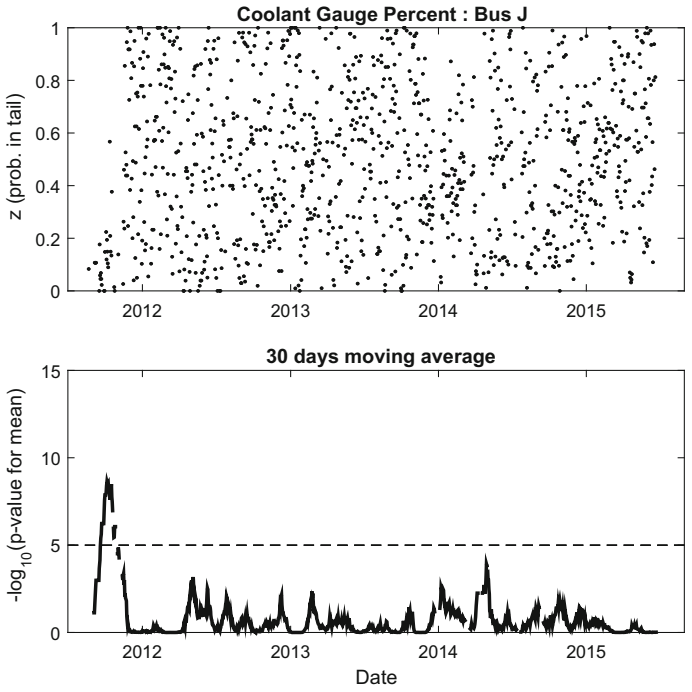

Fig. 3 The z statistic (*upper plot*) and the p value for the arithmetic mean over a 30 day moving average (*lower plot*) for bus J. Results are based on daily histograms. The *dashed line* in the lower plot marks p value $= 10^{-5}$

is questionable if it would have been noticed in the workshop if the repair visit, which focused on a gear box issue, had not been so long.

The *Coolant Gauge %* signal deviated also for other buses. Figure 4 shows the p values for all the buses during the period August 2011–August 2015. Here we set the threshold for a "significant" deviation at a p value of $10^{2.97}$, selected so that 10% of the data were above this threshold. This allows for better comparison with the ENDF method (see Sect. 4.5), i.e., we contrast the measurements in the 10th percentile (marked by a dashed line in Fig. 4) for both techniques.

Bus A started to deviate from the fleet in the second half of February 2012. During this period the *Coolant Gauge %* tended to be lower than for the rest of the fleet. This bus drove considerably shorter daily distances than the other buses in the fleet, less than half of the daily average for the other buses, and also significantly shorter daily distances than this bus normally drove. At the end of this period was a minor service and a week later a repair to the diesel fired burner for heating up the bus (the burner did not work). After this the bus ran normal daily distances again, and the deviation disappeared in March.

The ENDF method by Filev et al. flags three other deviations for bus A: in early 2013, in early 2014, and February and March 2015. None of these drop off as quickly as the COSMO deviations so it is harder to match them to specific workshop visits. However, in the spring of 2013 the bus visited the workshop for a longer time for

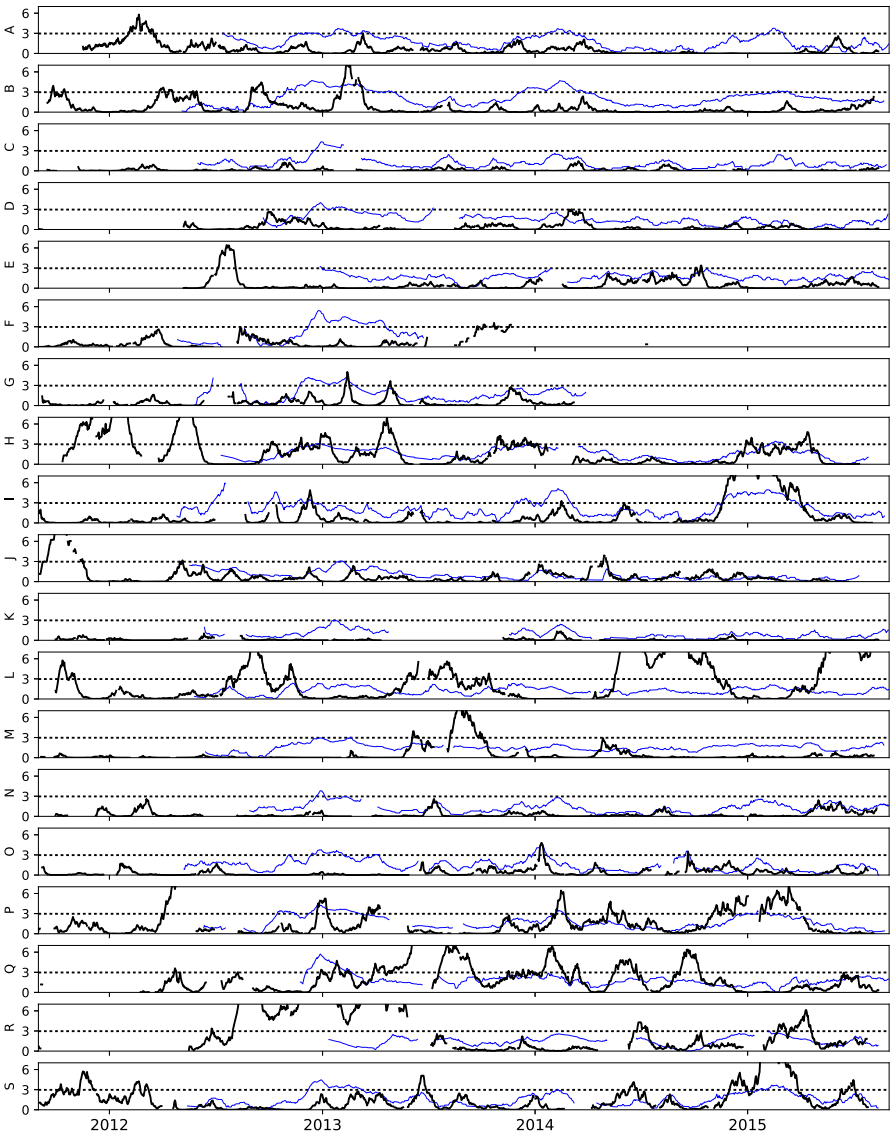

Fig. 4 In *black*, the p value (using logarithmic scale, cf. Fig. 3) for the arithmetic mean over a 30 day moving average for the signal *Coolant Gauge* % for 19 buses. Results are based on daily histograms. In *blue*, the (rescaled) deviation level based on a method by Filev and Tseng (2006), Filev et al. (2010) (Color figure online)

reprogramming the automatic gear box, in the spring of 2014 there was a large leak in the compressed air system and the bus visited the workshop for this, and there was a gear box repair in the spring of 2015. None of these workshop visits had any clear connection to the temperature in the engine.

Bus B had a deviation indicated by the COSMO method in September 2011. There was a repair to the diesel fired heater during this period (the burner did not start). There

was also a deviation between March and May 2012. This deviation period started with a repair of the diesel fired heater, followed by a longer workshop visit for renovating the gear box. The latter meant that the bus spent most time indoors when it was run, which in March in Sweden means that the ambient temperature is significantly higher than for the other buses that are outdoors. There was also a deviation during September 2012, which ended with the repair of an air leak. The deviation starts (grows very quickly) during a service operation in August 2012 and it may be the effects from this that lingers. The fact that the deviation is summed over the previous 30 days means that a few very deviating days can have an effect for almost a month onwards. Bus B has a larger deviation in *Coolant Gauge %* in late February 2013. This deviation disappeared after a major service and several repairs causing 16 days of downtime, in association with the mandatory Swedish annual vehicle inspection. The customer had complained about engine cooling fan overuse. This turned out to be the same fault as seen earlier on bus J: liquid had leaked into the ECU, creating a short circuit that caused the cooling fan to run at full speed all the time.

The ENDF method flagged three deviations for bus B: in early 2013, in early 2014, and in early 2015. Just as for bus A (and, in fact all other buses) these were much smoother than the COSMO deviations and thus harder to match to workshop visits. A comparison of Filev's method's deviations between bus A and B shows that they are very similar and that there may be a season effect. This can be expected since the method is based on calibrating it during a "normal" initial period.

On bus C the COSMO method did not identify any deviations, while the ENDF method flagged one, in January–February 2012. It ended with the diesel fuelled heater being repaired.

Bus D had a minor deviation in February–March 2014. This ended with a service during which the maintenance staff discovered that the radiator was clogged (and the radiator was subsequently cleaned). The ENDF method flagged one deviation December 2012–February 2013. Again, the character was very smooth and it looked more like seasonal variations than a fault. There was no workshop visit that matched the decrease in the indicated health status well.

Bus E deviated from the fleet in July 2012 as well as, to a smaller degree, during October 2014. The first (July 2012) disappeared after a repair that mentioned a faulty temperature sensor and a malfunctioning valve in the climate control system, which was causing internal heating to be continuously turned on. The climate control system uses heat originating from the engine, so this malfunction may cause deviations to the *Coolant Gauge %* histograms. The deviation in October started (grew quickly) during a longer (10 days) workshop visit, so the deviation may have been caused by the workshop visit and just lingered on for the following 30 days.

The ENDF method flagged a deviation in December 2012 for bus E. There was a repair to the diesel fuelled heater then (not working). However, the deviation from Filev's method was so smooth that it is hard to say if this repair really was the solution.

Bus F started deviating slightly towards the end of 2013, with an increased engine temperature. Unfortunately the data logging equipment on this bus stopped working, making it difficult to say what caused this. There was no way to match a decrease in the deviation to a workshop visit.

The ENDF method flagged a deviation for bus F during December 2012 – February 2013. It ended with the replacement of a temperature sensor for the fire extinguisher system (the fire alarm went off spontaneously).

Bus G deviated briefly in February 2013, when the *Coolant Gauge %* signal tended to be higher for this bus than the fleet. The bus was in repair, mostly indoors, for several days during this period. There was also a brief deviation in April 2013. We could, from the GPS signal, see that the bus had spent a day in the workshop on the day when this deviation peaked and started decreasing. However, there was no matching entry in the VSR data base. The ENDF method indicated a deviation December 2012–February 2013. This, however, again looked like a seasonal variation in the signal.

Bus H was enigmatic; it deviated from the fleet several times but none of the deviations could be linked to repairs on the cooling system. The first deviation lasted essentially until May 2012, the variations in deviation level were caused mostly by variations in the number of data points (data was lost during several days). This period ended with a repair of a modulator for the electronic brake system, but nothing related to engine temperature. It again had a deviation that stretched into April 2013, ending with a larger maintenance visit. This time the bus operator had complained about high consumption of coolant liquid. As a result was the cooling system checked for leaks, but no leaks found, and the radiator was cleaned properly. The bus had a third deviation between January and April 2015, ending with a renovation of the gear box, but again no repair that could be linked to the engine cooling. Bus H tended to have a colder engine than the fleet during the colder months. It was also in neutral gear more often than the fleet during the colder months (data not shown). When the bus drivers were interviewed about their experiences with the buses and asked whether there was any particular bus that they experienced as "problematic", bus H was the only one that the majority of drivers agreed on. The colder engine could possibly be a result of how this vehicle was operated.

The ENDF method by Filev et al. agreed to a large extent with the COSMO results on bus H but this may have been a coincidence; the deviation level showed the same seasonal behaviour as for buses A and B.

Bus I deviated in early December 2012, which dropped off after a repair of the diesel fuelled heater. It had a brief deviation in February 2014, which ended with a maintenance service. There were no operations during this maintenance related to engine temperature. Bus I also deviated quite a lot between December 2014 and March 2015. This deviation decreased after another repair on the heater in February 2015, which were done after the customer complained about poor heating in the bus.

The ENDF method showed a deviation during the same period in December 2014 and March 2015. It also marked a deviation in August 2012, which may have ended with a repair of an air leak, for bus I; this deviation peaked just before a period with loss of data why it is uncertain when it started decreasing. It also showed a deviation in February 2014, which ended with a minor maintenance service.

Bus J was discussed earlier. The biggest deviation disappeared in November 2011, after replacement of a broken ECU that controlled the engine cooling fan. The *Coolant Gauge %* signal also deviated in May 2012 and in late April 2014, although not as much as in the initial months. Both these peaks were caused by long visits for general

maintenance and annual vehicle inspection, when the bus spent most of the time indoors and did not operate like the rest of the fleet.

Neither the COSMO nor the ENDF method indicated any deviations on bus K.

Bus L deviated multiple times, initially for about 5 days in mid October 2011, then again from August to September 2012. The *Coolant Gauge %* signal was higher than the fleet during both these periods. The first deviation period ended with two repairs: an on-road action service due to a coolant leak to the diesel fuelled heater, and a repair of the diesel fuelled heater due to customer complaint on poor heating. The second deviation also probably ended with fixing a coolant leak (the mileage and date for the repair were uncertain). There was also a deviation around June–July 2013, which was associated with two longer stays at workshops; one for unknown reasons (no record in VSR) and one for the annual maintenance and checkup in association with the mandatory vehicle inspection. The latter had no repair related to engine temperature. It can be so that the deviations were caused by the workshops visits, rather than fixed by them. The *Coolant Gauge %* signal deviated again between May and October 2014, with a tendency towards higher temperatures than the fleet, but returned to normal after a repair when the cracked coolant expansion tank was replaced and a new coolant pump installed. This repair was surprising since the coolant pump had been replaced as late as July 2014, with very similar repair comments (leak in the weep hole). In May 2015, a deviation started that continued on after that. A new coolant pump was installed again in June 2015, with very similar repair comments as the two previous cases. The ENDF method had detected no deviations that passed the 10 percentile limit for bus L.

Bus M deviated in May 2013, and again from mid August until early October 2013. The May deviation ended with replacing the control unit for the heating system. The bus was sent for repair in early October 2013 because the customer had observed that the engine cooling fan was running at full speed all the time. This was due to a faulty ECU, the same problem as on buses J and B. The ENDF method discovered no significant deviations.

For bus N, the COSMO method detected no deviations, while the ENDF method indicated a short deviation in December 2012. It was not associated with any repair.

Bus O had the biggest deviation in January 2014. This appeared during a long period (27 days) in maintenance, mostly indoors, when the ambient conditions for this bus were completely different from the fleet. It also had a brief deviation between September and October 2014. The latter was flanked by a repair to the climate unit, and fixing a coolant leak by the diesel fuelled heater.

The ENDF method indicated a deviation from November 2011 until April 2013 for bus O. This ended with fixing a coolant leak. The ENDF method also flagged a deviation during the long maintenance period in January 2014.

Bus P had deviations six times: in April 2012, December 2012, March–May 2013, February 2014, April 2014, and from November 2014 until March 2015. The *Coolant Gauge %* signal for this bus was higher than the fleet during the first period. It ended with one of the engine cylinders jamming, resulting in a 4 week long unplanned stop for engine renovation. During the second period the signal tended to be lower than for the fleet. This deviation disappeared quickly in the first week of January 2013 without any explanation why; there is no mention in the VSR of any repair done at this time.

The third deviation started to appear in March and was gone in late May 2013. The data between April 9 and May 26 were lost and therefore it is unclear exactly when the deviation disappeared. However, there was one repair done in this period that could explain it: a broken air compressor in early May, caused in part by failed cooling of the compressor (congested pipes). The fourth deviation, in February 2014, disappeared after a replacement of the water pump for the diesel fuelled heater, which is likely to have an effect similar to what we reported earlier on bus E. The fifth deviation was not related to a repair relevant for engine temperature. Finally, the sixth deviation disappeared after the fleet operator complained about low temperature inside the bus. A malfunctioning connection for controlling a valve in the coolant pump was identified as the cause.

The ENDF method also indicated deviations for bus P in January 2013, January–February 2014, and January 2015. Several of these coincide with the deviations flagged by the COSMO method. However, the seasonal character of the indications is suspicious (they are all in January–February) and the deviation in March 2013 is missed.

Bus Q had several deviations. The first, and brief, deviation was in February 2012. There was a repair of the electronic control unit for the diesel fired heater around this time (mileage and date are uncertain). A second deviation began in January 2013 and disappeared in February, but there was no VSR entry or workshop visit that fitted this. A third deviation ended in September 2013 (the dip in mid 2013 was due to missing data) with the repair of a malfunctioning valve in the climate control system (heat was leaking out of the unit). There was a short deviation in early 2014, which also disappeared with a repair of the climate control system (malfunctioning fans). The fifth deviation, in May–June 2014, ended when a new coolant pump was installed. The final deviation, during September 2014, disappeared with the repair of a large coolant leak. This last repair was done since the customer complained about having to refill it with water daily.

The ENDF method also indicated a deviation in January–February 2013 for bus Q.

Bus R first deviated for a long time, between August 2012 and May 2013. The bus was in maintenance service on August 27, 2012, with a comment from the customer that there were warnings about high engine temperature. A fault search was done and an oil leak was repaired in late July 2012. In the weeks before this repair the *Coolant Gauge %* signal was a bit higher than the fleet average. However, after this repair the *Coolant Gauge %* signal tended to be lower, not higher, than the fleet average and the signal deviated quite a lot. The deviation disappeared after a repair of the cooling fan in May 2013, when the customer complained about cooling fan overuse. The fault was a broken magnetic valve, which caused the cooling fan to run at full speed all the time, similar to the faults on buses B, J and M. There was a short deviation June-July 2014, caused by a longer workshop stay for repairing the gear box, during which the ambient conditions were different from the fleet. There was also a minor deviation in March–April 2015. This latter deviation is mysterious; it peaks and starts decreasing in relation to a repair of the air dryer for the compressed air system. It was preceded by a repair to the diesel fuelled heater, after which the deviation started to grow.

The ENDF method did not flag any deviations for bus R.

Bus S had a deviation in November 2011 that ended with a visit to a workshop (as evident from the GPS signal). There was, however, no VSR entry that fit this date.

There was a shorter deviation in February 2012 that ended with a repair to a wheel bearing. The VSR record commented that the wheel got quite hot from friction due to the malfunctioning bearing. There was a deviation in mid 2013 that seemed to be caused by two longer workshop visits, one for gear box renovation and another for the service related to the annual vehicle inspection. There was a deviation in mid 2014, which was remedied with a longer service (in relation to the annual vehicle inspection). This repair included replacing the coolant pump and overhauling the cooling fan drive. There was also an approximately 3 month long deviation in the beginning of 2015, when the *Coolant Gauge %* signal tended to be below the fleet average. It was partly remedied with a new coolant pump in February, and completely remedied with a "no fault found" repair of the diesel fuelled heater in March 2015.

The ENDF method flagged deviations during some of the periods that COSMO method flagged for bus S, but only in the December–February periods.

Although these stories are anecdotal, the experiment demonstrates that a simple representation such as a histogram, selected based on it having low entropy, contains valuable information for a self-monitoring system; information that can be extracted using unsupervised outlier detection. The deviations tend to disappear when relevant maintenance and repair actions are done.

The COSMO method and the ENDF method are different in their design. COSMO builds on comparing models across the fleet of vehicles, and ranking them accordingly, e.g., by their distance to "the most central model"—but other measures can also be used. If a particular machine is consistently ranked low, i.e., far away from the group consensus, then that indicates a problem. This inherently requires that models and their ranks are aggregated over a period of time; we have used 30 days. This means that the COSMO approach needs a number of days to work well, and that abnormal operation will cause a deviation that can extend for some time after the fault has been removed. In our scenario we have seen several examples of deviations that appear during a workshop visit and remain for some time after that. The ENDF method builds on using clusters and can, in principle, give a health status based on a single day. In practice, however, the differences in usage and in external conditions that those buses encounter mean that daily deviation levels vary too much and are too unpredictable; any single one gives very little useful information. Therefore, decisions need to be based on consistent trends, and thus in the plots we are showing 30-days running average. This, however, makes the ENDF method vary less rapidly than the COSMO method, despite that both methods aggregate data over equally long time periods. The hypothesis testing framework of COSMO allows, in this case, a strong deviation to sway the p value faster than a simple averaging does. Therefore, it is easier to match COSMO deviations to workshop visit dates; they are more distinct.

Another difference is that the COSMO method recalibrates each time period by using the fleet, under assumption that the majority of the vehicles are OK. This requires the ability to communicate with the fleet. The ENDF method is calibrated only during the initial phase, which instead requires the vehicle to be operating normally during that phase. The most clearly visible consequence is that the ENDF method shows definite signs of season dependency, which is to be expected if the calibration to "normal" only happens in the initial phase, in particular if that phase is shorter than a year. The

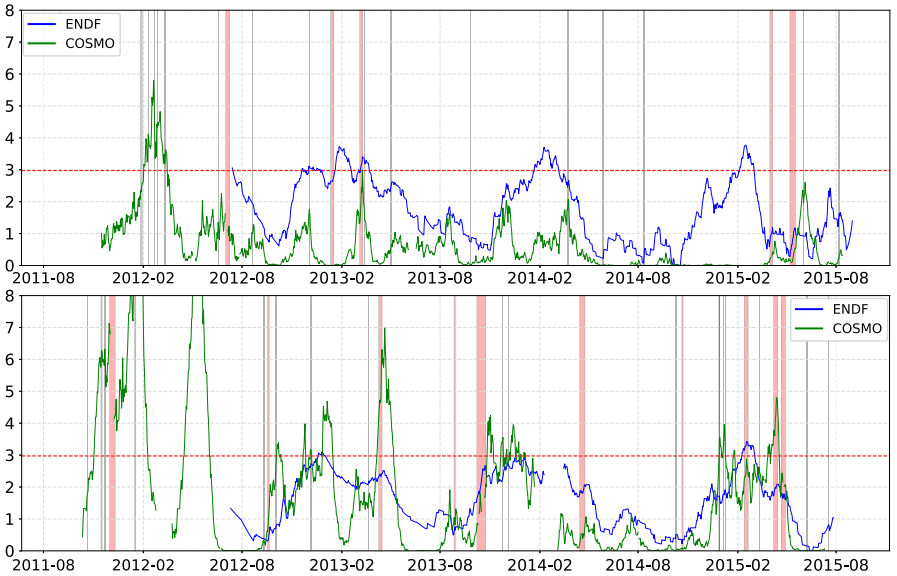

Fig. 5 Two examples of deviation levels: bus A (*top*) and bus H (*bottom*). *Vertical lines* correspond to workshop visits

COSMO method, on the other hand, does not suffer from seasonality since calibration is done all the time against vehicles operating under similar conditions.

As an example, Fig. 5 presents in more detail one case where the two methods differ (bus A), and one where they produce relatively complementary results (bus H). The first clearly visible thing is the unnatural similarity of the ENDF curves for both buses—clearly, the effects of seasonal changes overshadow the actual fault detection. For this presentation we have included, as vertical lines, workshop visits of the vehicles. In early 2012 bus A had a deviation, which disappeared after two repairs, related to Stroco heater and to coolant leak. This deviation is captured by COSMO, while Filev is still in the initialisation phase. There are no other significant deviations according to COSMO, while Filev discovers three more. In early 2013 there were five repairs, four of them related to Stroco heater, and one to gearbox. Deviation in early 2014 does not seem to have an explanation: the two repairs are related to broken air compressor hose, and to oil leak. Similarly in early 2015, repairs concern gearbox and fuel filters. In June 2015 there is a coolant leak, missed by Filev and discovered by COSMO (but not significant). In case of bus H, the period from September 2012 until June 2013 is generally problematic. COSMO has several peaks (not particularly well-aligned with repairs), while Filev has high deviation level all the time—although barely reaching the significance level. Subsequently, both methods warn during September 2013–April 2014, when three repairs related to Stroco heater and coolant-leaks are performed. Finally, in early 2015 there is period of deviations, with 8 different repairs, ending by gearbox renovation.

As is clear from the discussion in this section, the COSMO method deviations, for the signal *Coolant Gauge %*, are to a higher degree matched to VSR events related to

temperatures than what is the case with deviations from the ENDF method. However, it is difficult to put a concrete number on this, as there is quite a bit of subjective interpretation necessary to make use of VSR information.

It is impossible to say with certainty if the actual faults that were repaired were serious or not since the experiment was done on normally operating buses and with the normal record-keeping of maintenance operations. Neither is it possible to verify if the faults were present all the time that the deviations were there. There are a few comments in the maintenance records of the customer (the bus operator) complaining about having to refill the coolant system repeatedly, or about having cooling fans that run more frequently than expected, or that the bus is cold inside, which show that the faults have been present and noticeable at least for some time before they were repaired. Significant deviations, i.e., in the top 10 percentile, were observed in 16 out of 19 vehicles during the 4 year long observation time. Of these could at least five be explained with certainty from the maintenance records; four were caused by the same problem, a runaway cooling fan. Many of the other deviations were associated with repair events that very likely could have affected the engine temperature gauge.

The runaway cooling fan is a good example to motivate the COSMO monitoring approach. It is an example where no on-board fault detection (diagnostic trouble code) is implemented, it is a fault that is difficult to anticipate during design, and is neither a critical nor a frequent fault over the total vehicle lifetime. However, it is very uneconomical for the vehicle operator to have a runaway cooling fan once it happens. The energy consumption of the cooling fan increases with the cube of the speed; a cooling fan that runs at maximum speed consumes about 5–6% of the total engine power. The cooling fan is supposed to run at maximum speed less than 1% of the time under normal circumstances, as is evident from Fig. 2. A cooling fan that runs at full speed all the time is thus a significant waste of power, fuel and money.

This case was presented to illustrate what comes out from observing one signal that looks potentially interesting, without prior information on how this signal should be distributed nor prior information of whether it is interesting or not. There is no guarantee that signals that are in the far upper left corner in Fig. 1 will respond strongly to faults. That they are in the upper left corner means that their distribution is quite stable and has a low entropy. However, it might well be stable (i.e., not change) also when faults are present. Still, there are other signals, with lower values of the average NE in Fig. 1, that show stronger responses to different faults. One example is the signal *Temperature Upstream Catalyst*. Figure 6 shows the statistics for this signal for bus P, which experienced a jammed cylinder in May 2012 that resulted in a 1 month long unplanned stop. This signal indicated an abnormality already 6 months before the engine breakdown.

This kind of analysis can be performed manually, but the goal is for the system to autonomously explain detected deviations. It is possible to match deviations to repairs automatically, but this requires the number of similar failures to be sufficiently high. A fleet of 19 buses only experiences few cooling fan and engine cylinder problems. However, they may be very well correlated with certain deviations. If a search is done over the maintenance and repair events that lie close (within 1 week forward or backward in time) to drops in the *Coolant Gauge %* deviations, then the top operations that pop up as the most unique in this set, compared to the rest of the maintenance

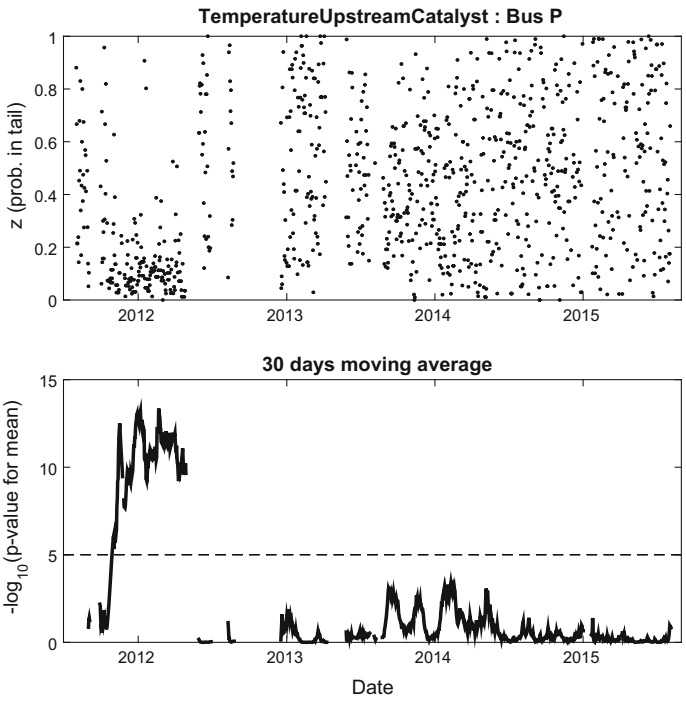

Fig. 6 The z statistic (*upper plot*) and the p value for the arithmetic mean over a 30 day moving average (*lower plot*), for the signal *Temperature Upstream Catalyst* on bus P. Results are based on daily histograms. The *dashed line* in the *lower plot* marks p value $= 10^{-5}$

events (collected since 2008) are: replacing the valve cover gasket, replacing oil filters and replacing a ECU. The first two are related to engine renovations and oil leaks, and the last is related to the runaway cooling fan. However, relating coolant leaks and engine heater problems to the deviations is more challenging since these are expressed in more free text.

The next two sections present examples of using different model families, autoencoders and linear relations instead of histograms, to detect more common problems that are easier to extract from the service records, namely *air compressor* and *wheel speed sensors and modulators* malfunctions.

7.2 Autoencoders

The example in previous section has been chosen based on the fact that there are not many deviations in that signal, and most of the ones that have been detected can be easily explained based on vehicle repairs. In this section we present a different scenario.

We focus on the air subsystem which was very problematic from the fleet owner's perspective. There were multiple issues related to the brakes and air suspension on every single bus. One particularly interesting component is *air compressor*, since it is

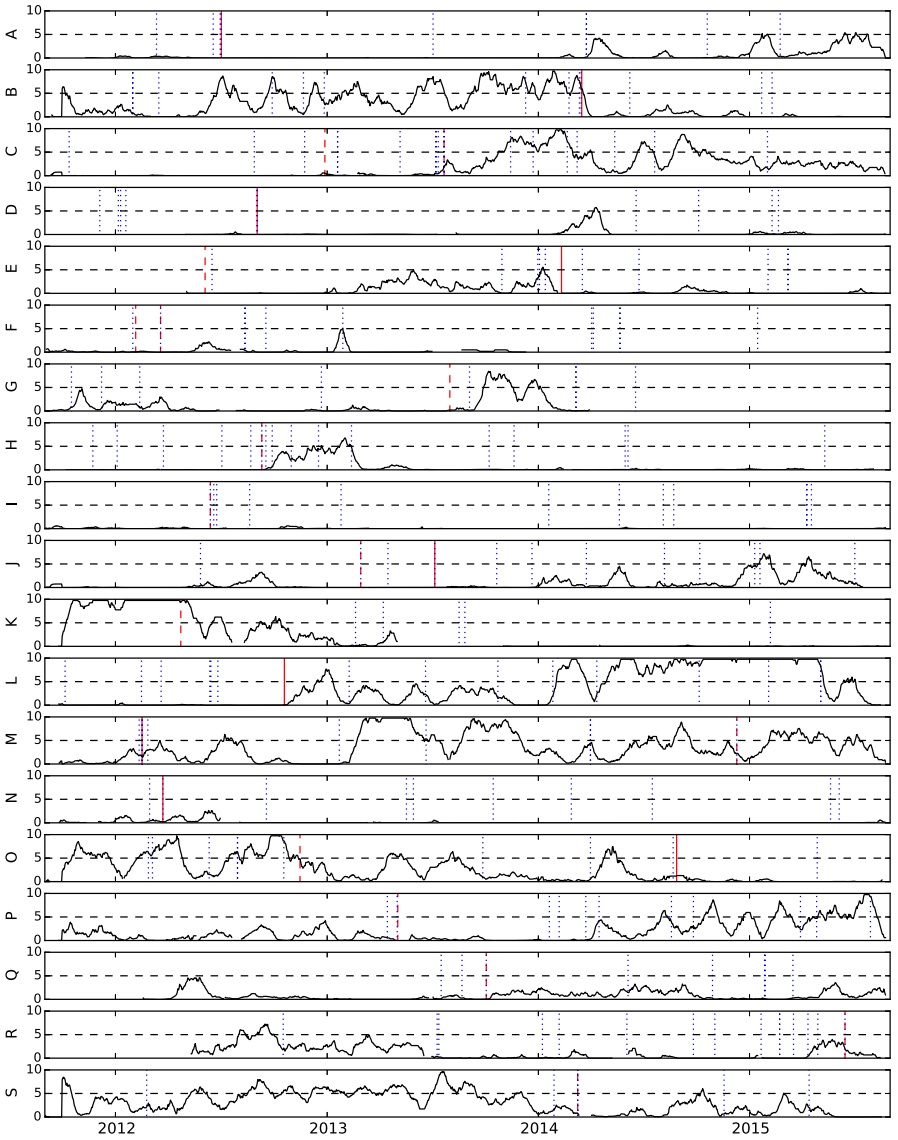

Fig. 7 The p value for the arithmetic mean over a 30 day moving average for the signal *Wet Tank Air Pressure* for 19 buses. Results are based on daily autoencoders, where fleet data are collected over a week. The *vertical lines* correspond to different kinds of repairs: *solid red lines* are compressor failures that happened on the road and required towing of the bus, *dashed red lines* are compressor replacements that took place in the workshop, and *dotted blue lines* are other repairs related to the air suspension and brakes (Color figure online)

quite expensive and there are no good predictive maintenance models for it. Therefore we have decided to specifically focus on the only relevant signal we have access to, *Wet Tank Air Pressure*. This section briefly presents our findings, using autoencoders as models, as explained in Sects. 4.2 and 4.4.

The results of the analysis can be seen in Fig. 7. There we present the deviation level computed using the COSMO method, as well as the relevant repairs. Solid vertical red lines are the dates for compressor failures that happened on the road and required towing of the bus. Those are the most serious failures and the primary reason, from the vehicle operator and workshop point of view, for doing this particular analysis. The dashed red lines correspond to compressor replacements that took place in the workshop, which are primarily preventive operations based on the expertise of mechanics. Finally, with dotted blue lines we have marked a number of other repairs that are related to the air suspension and brakes and which are likely to manifest in the *Wet Tank Air Pressure* signal.

Clearly, in this case both the number of deviations and the number of repairs is too large to allow us to present a comprehensive analysis of their relation as we did in Sect. 7.1. Therefore, we will limit ourselves to just highlighting some of the more interesting findings.

There are several cases of deviations that match compressor repairs quite well. Some examples include bus B in March 2014, bus K in April 2012 or bus O in November 2012. The overall accuracy of the method, however, is quite poor. What is particularly interesting are situations like bus C in July 2014 or bus L in October 2012, where deviations *start* when compressors are replaced. One possible explanation is that a new compressor is actually "stronger" than is typical for our (somewhat old) bus fleet, and therefore does indeed behave differently during the first couple of weeks. However, this effect is clearly not universal. In addition, one can also notice that deviations are more common in the later years, which may indicate overall worse condition of the air system in those vehicles.

It is also worthwhile to point out that there are multiple situations where the compressor was replaced even though there are no indications of any problems (for example on bus D in September 2012, bus E in June 2012, bus F in February *and* in March 2012, or bus J in February *and* July 2013). In only two cases those were caused by actual failures on the road—typically, such replacements are performed in the workshop, as a preventive measure. We believe that at least in some of the cases this was done too early. Currently the mechanics do not have reliable ways to measure condition of the air compressor, so a system like the one we propose here could lead to significant cost savings. This component is problematic enough to warrant a dedicated, supervised solution (Prytz et al., 2015).

Finally, we need to mention buses F and J, where compressors were replaced twice in a short period of time. For bus F we believe this to be an issue with the service records database, where a single operation is recorded twice, in February and then again in March 2012 (it was done by a sub-contractor). For bus J, the two repairs were done in February and in July 2013. The 6 months period is not enough to wear the compressor down under normal circumstances. It leads us to believe that either there was some other problem with the air system (for example, a severe clogging of the pipes could lead to much faster compressor wear), or that this particular unit had some kind of manufacturing fault. Neither of the repairs was accompanied by any deviation—in fact, bus J is the only bus in the fleet that never shown any *Wet Tank Air Pressure* anomalies.

7.3 Linear relations

In this section we model binary linear relations between signals to showcase how a complete self-monitoring system operates. It covers the whole process, from the identification of relevant data to monitor, all the way to automatically discovering repairs that correspond to fixing the problem.

The procedure described in Sect. 4.3 was followed and the search was applied to data from a 1 week long window starting on January 12, 2012. The window contained 100,000 samples from each signal. The algorithm built linear models for all pairwise combinations of the 48 signals that had non-zero entropy, resulting in a total of 2256 models. These models were generated on each of the 19 vehicles in the fleet and two different interestingness metrics, denoted α and β, were computed for each signal pair. The result of this is shown in Fig. 8. The most interesting models, from a monitoring point of view, are models that have small α values and large β values. Exactly how many models that will be returned as interesting then depends on where the user sets a threshold on how many that are possible (or of interest) to study in more detail. For the purpose of this demonstration, the threshold was set conservatively at $\alpha < 0.4$ and $\beta > 0.05$.

One of the most interesting relationships concerned *Relative Wheel Speed Rear Left* and *Relative Wheel Speed Rear Right*. Those two signals are related to the operation of Electronic Brake System and are critical for the safety of the vehicle, which makes them interesting to study further. We have used the procedure described in Sect. 4.4 to calculate deviation level and present it in Fig. 9. In addition, we mark with vertical red

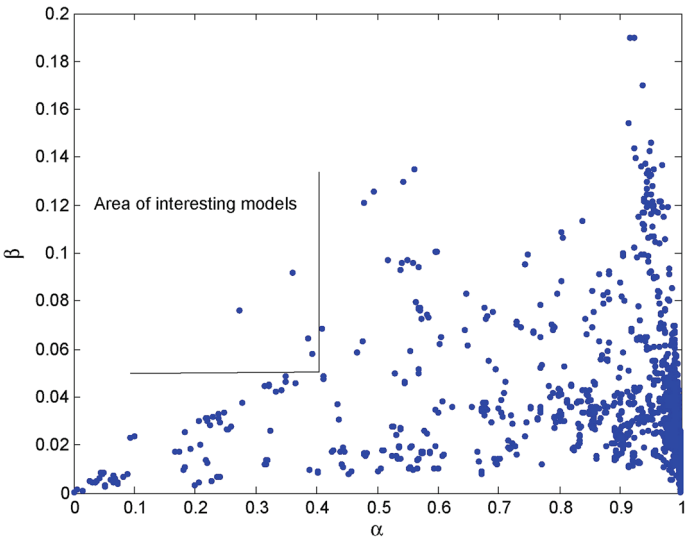

Fig. 8 Interestingness metrics α and β. The α measures the modelling accuracy of the models that were fitted on-board the vehicles. The β value measures how large the variation is between the models (of the same type) in the fleet (cf. Sect. 4.3). The most interesting models are in the *upper left corner* because they model strong relations that exhibit large variation

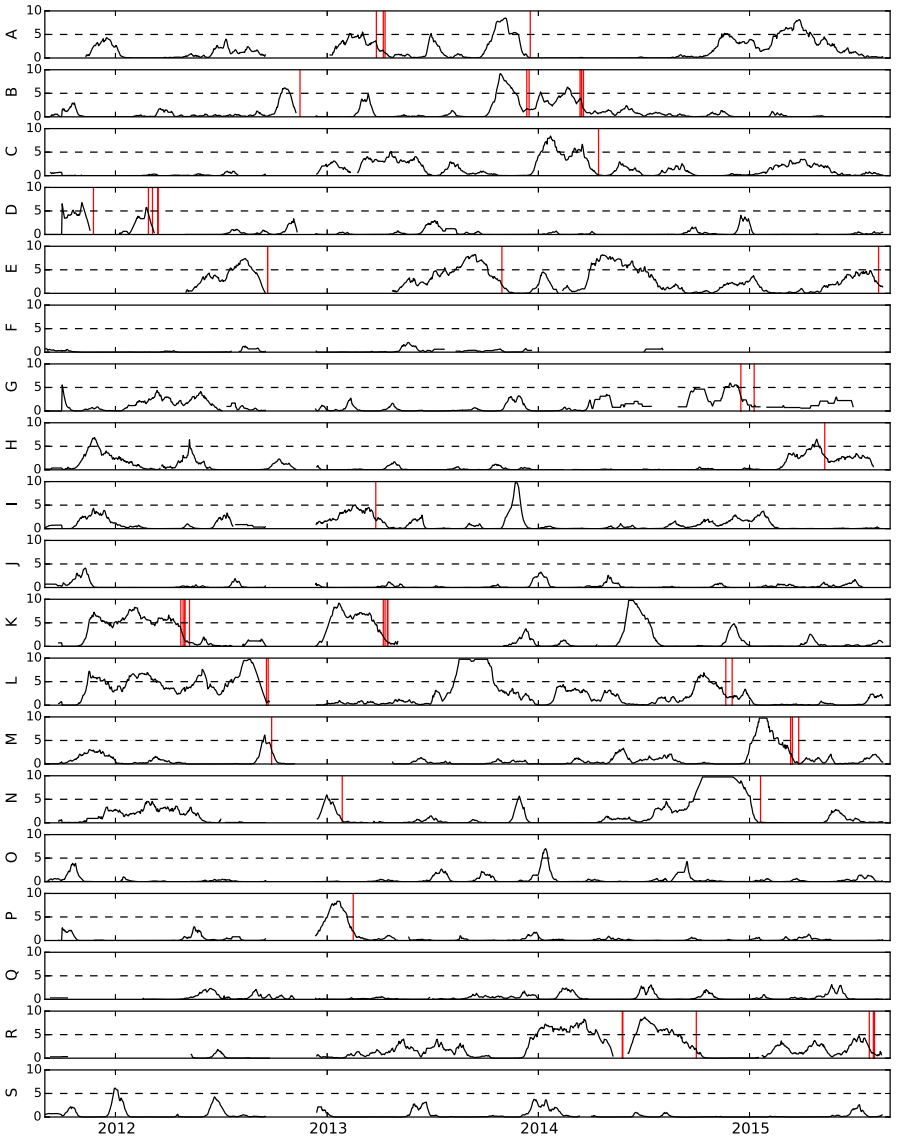

Fig. 9 The p value for the arithmetic mean over a 30 day moving average for the parameters of linear relation between *Relative Wheel Speed Rear Left* and *Relative Wheel Speed Rear Right* for 19 buses. Results are based on daily linear relations, where fleet data are collected over a week. The *vertical red lines* correspond to workshop visits that occurred close in time to strong deviations disappearing, i.e., repairs expected to contain operation(s) that have fixed the problem (Color figure online)

lines all the workshop visits that occur when a significant deviation *disappears*. Due to the fact that the dates in our repair history database are quite uncertain, we take into account the period of 4 weeks: 1 week before and 3 weeks after the deviation drops below the threshold.

In this particular case, we have identified a total of 33 deviations in the selected model, and 51 workshop visits within the aforementioned 4 week time periods. As described in Sect. 5, each workshop visit contains information about date, mileage, parts, operations, and free text comments by the workshop personnel. In this work we only focus on the structured part of the service records. Volvo uses unique codes for both the physical parts as well as for the typical operations that can be performed on the vehicles. A self-organising system needs to be able to automatically identify a small subset of those codes that are relevant for a particular deviation.

A simple database query shows that there are 150 codes that occur more than once within the 51 workshop visits of interest. However, it is important to notice that different codes occur in the service records at different frequencies (for example, "read fault codes" is a very common operation, while "replace turbocharger" is not). Therefore, it is not the absolute number of occurrences of various codes that indicates their relevance, but rather the difference between typical and observed frequencies.

Figure 10 presents the expected number of occurrences for each of those codes as the grey bars, with standard deviations shown in blue. This value is calculated by repeatedly selecting a bus and a random period of 4 weeks, then counting the number of occurrences of each repair code. By doing this for 33 independently chosen time intervals, we obtain occurrence counts that are comparable to those within our periods of interest. We repeated this process 50000 times, and the mean of those trials is an approximation of the expected number of times each repair code should occur, assuming they were not related to the wheel speed deviations. This procedure is not flawless, since the frequency of operations is definitely not constant over the 4 years of our field study, however, we believe it is good enough for our needs here.

Having calculated the expected number of occurrences for each code, as well as its standard deviation σ, we are now able to identify operations which are likely to be related to the deviations identified in Fig. 9. Intuitively, we are interested in the operations that are more common during our periods of interest than they are at random 4 weeks time intervals. In order to visualise this, we have shown the actual number of occurrences for each code in Fig. 10. Red stars correspond to codes which are deemed to have the expected number of occurrences, while green stars correspond to codes which are significantly more common (i.e., have at least 3σ more occurrences) and are therefore considered relevant.

Overall, there are 16 such codes. We are not able to provide the full list here (this is considered sensitive information). However, this list includes several codes related to components which, according to experts, are likely to cause deviations in relative wheel speed signals. This concerns both codes corresponding to physical parts (e.g., "inner wheel bearing" or "speed sensor modulator") as well as to the operations (e.g., "replace of rear modulator"). Some of the codes are very generic and it is unclear how interesting they really are (e.g., a part described simply as "bearing").

This process is not foolproof, however, and there are two codes which we believe are not related to wheel speed sensors: "lamp" and "air compressor test". It is likely that this is simply a matter of the amount of data we have, and with bigger fleet, more deviations and repairs, we would be able to exclude them. However, other explanations are also possible. For example, there actually could be a relation between "lamp" and wheel speed sensors: those two components are physically located close to each other,

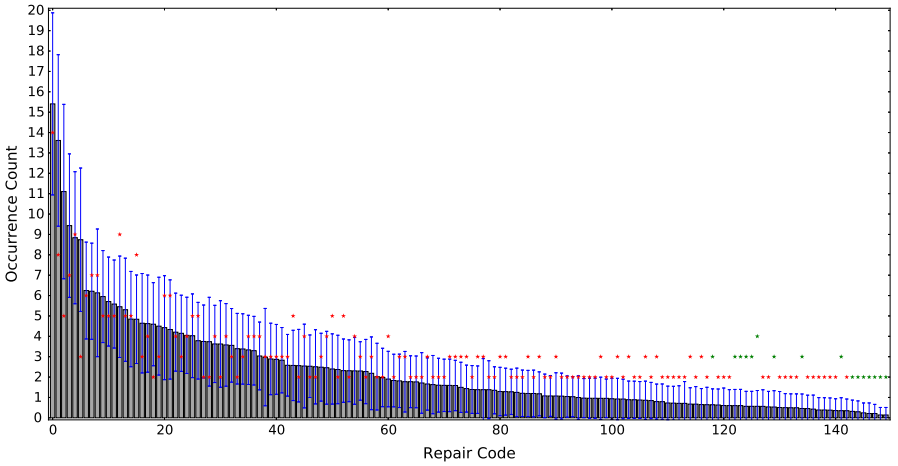

Fig. 10 The expected frequency of 150 repair codes (*grey bars*), with standard deviations (in *blue*). The *stars* show the actual number of repair codes during the repairs presented in Fig. 9. *Green stars* indicate repair codes where the actual frequency is more than 3σ above the mean (Color figure online)

so it is possible that the mechanic who looks at one of them is likely to notice a problem with the other.

The experiment shows how deviations can be detected using models that are automatically selected based on "interestingness". Those deviations correspond to real faults on the vehicles, and the system can use VSR database to automatically identify a set of candidate components for replacement. For the particular case of wheel speed sensors there are diagnostic functions implemented on-board since this is a critical fault. Nevertheless, one of the malfunctioning wheel speed sensors, which our method detected in good time, resulted in an on-road emergency repair.

7.4 Knowledge discovery

There were several cases of discovered knowledge during the time of the field test on the city bus fleet. We describe a few of them here.

A first example is the runaway engine cooling fan. One of the city buses showed a clear downwards deviation in the engine temperature already from the start of the experiment, in August 2011. Representatives of the bus operator and engineers at the OEM were consulted but could not provide good explanations why one of the vehicles would have a significantly cooler engine than the others. There were also no complaints from the bus drivers that could be related to this phenomenon. It was believed to be a data collection problem, i.e., a "bug", until the deviation suddenly disappeared after an ECU was replaced. Engine liquid had leaked onto and into the ECU connector glove, resulting in a short circuit that caused the cooling fan to always run at full speed. A cooling fan running at full speed all the time represents a considerable waste of energy (see Sect. 7.1). After the fact, this was an obvious reason for having a colder engine. Before the fact, however, the reason for the observed symptoms was not that obvious.

The runaway cooling fan has occurred more times, also on other buses, and it became clear that leaks onto the ECU was a problem to consider in future designs of the engine compartment.

The knowledge discovered was not just the connection between liquid leaks and runaway cooling fan; this could, in theory, have been discovered by text mining the service record database (although we believe it would have been difficult). The important knowledge discovered was that the runaway cooling fan can be easily detected by a comparison of engine compartment temperatures across the fleet under similar ambient conditions.

A second example is the engine emission control system. A fault that occurred several times on the city buses was failing NOx sensors used to verify the engine emission control; this was not a sudden failure but a slow degradation. The buses had been operating for long enough that the NOx sensors were approaching their expected end of life. The COSMO algorithm indicated many cases when sensors were about to fail, and highlighted the differences in characteristics that separated them from well-functioning ones. For example, a failing sensor had reduced bandwidth; the mean value was unchanged, but the value range was narrower. This was new knowledge for the vehicle manufacturer and was used to improve existing on-board diagnostic (OBD) of the sensor, as described in a patent application by Karlsson et al. (2013). Similar characteristics of the NOx sensor were also exploited in a recent patent by Rajagopalan et al. (2011).

What this illustrates is how a generic method, i.e., not aimed at an a priori defined expected problem, could mine the data streams on-board the equipment and highlight failure characteristics of a component in a subsystem that then became a starting point for a further study that resulted in a patent application. There was no specificiation prior to the field test that the NOx sensor was of particular interest, but repeated deviation detections in relation to repairs where failing sensors (in the field) gave indications that there was something important here. This exemplifies a vision for a system like this; when the on-board search with intelligent agents leads to discoveries that become new knowledge, exploited in, e.g., a patent.

A third example is a jammed cylinder. Just as with the runaway cooling fan, one of the vehicles was showing a strong deviation already from the start of the field test period; in this case, in the engine exhaust temperature and manifold pressure. After about 6 months of operation, the charge air cooler (CAC) was checked for leaks and repaired; this was a relevant operation, since the symptoms of a CAC leak are increased exhaust temperatures and lower manifold pressures. However, the signal deviations remained after this repair and 2 months later the engine jammed. This resulted in about 30 days unplanned downtime for the bus.

The discovered knowledge here was that the repair of the CAC did not have the desired effect. The deviations in the signals, compared to the fleet, were still there. An immediate consideration of alternative reasons for the atypical behaviour would potentially have saved a lot of downtime.

A fourth example concerns malfunctioning wheel speed sensors. There are many possible causes for incorrect signal readout, including failure of the sensor itself, broken wiring or faulty modulator. Wheel speed sensors are monitored by the OBD system, since the Electronic Braking System (EBS) will not operate unless it has good

signals from all those sensors. However, the driver may not observe or care about diagnostic warnings and they can exist for a long time before any repair action. This kind of situation was experienced in November 2011, when a front wheel speed sensor had been deviating for a month, the OBD system warned about an EBS problem and the driver complained about weak brakes. Ultimately, however, the vehicle had to be stopped on the road in order for technicians to diagnose the fault and replace the wheel speed sensor. This example shows that a self-monitoring system can be used to discover, and potentially avoid, issues before a human driver reacts to a problem.

It is important to understand that the field test targeted none of these components or subsystems specifically. It was a general search for deviations, their appearances and disappearances, and matching these to service record events that led to the insights about these systems. It was also important that the study extended over a long period after the buses were produced, since many of the faults started to appear after 3–5 years of operation.

8 Conclusions

We have presented results from a long term real-life study using the Consensus Self-organising Models (COSMO) approach, which builds on using embedded software agents, self-organising algorithms, fleet normalisation and deviation detection for health monitoring of mechatronic systems. The usefulness of COSMO was demonstrated on data collected during a 4 year long study of 19 seasoned city buses in normal operation. As such, this work is a contribution towards self-learning and self-monitoring systems, particularly useful in domains where the product complexity or subcomponent pricing strategy makes it difficult to justify dedicating human expert efforts to build fault detection and diagnostic models.

The COSMO approach is formulated in a general way, using embedded agents that can monitor many variables and build various kinds of models. It is also general in the sense that it does not require much knowledge about the monitored system. It has been demonstrated on hard-disk drives, heavy duty trucks, wheel speed sensors and engine cooling alike. It is important to note that the monitoring was not done with expert defined features, instead COSMO suggests features, ranks them and then observes what happens. This is a new step towards autonomous knowledge discovery.

In this paper three examples of models were described in detail: histograms, autoencoders and binary linear relations. All of them produced clear deviations that were relevant to practical vehicle maintenance issues, issues that in many cases resulted in on-road stops or extended unplanned workshop visits. An analysis of the maintenance statistics for the vehicle fleet indicates that predictive maintenance solutions like the one proposed here, combined with more efficient handling of the workshop waiting times, should allow to decrease the number of workshop days per vehicle significantly, by a factor of two or more. The COSMO results were also favourably compared against state-of-the-art method for deviation detection in equipment monitoring.

It is, after this extensive field study, clear that it is feasible to construct self-learning and self-monitoring systems for a fleet of vehicles in normal operation. COSMO can already today be used for decision support to flag vehicles that need attention. In order

to have a fully working system, however, it is absolutely necessary to improve the data quality in service record databases. Future condition monitoring systems must in large part be self-learning and rely much less, or not at all, on human expertise to define suitable features, model structures and provide labelled training data.

References

Ackoff RL (1989) From data to wisdom. J Appl Syst Anal 16:3–9

Alippi C, Roveri M, Trovò F (2012) A learning from models cognitive fault diagnosis system. In: Artificial neural networks and machine learning—ICANN 2012(Lecture notes in computer science), vol 7553. Springer, Berlin, pp 305–313

Alippi C, Roveri M, Trovò F (2014) A self-building and cluster-based cognitive fault diagnosis system for sensor networks. IEEE Trans Neural Netw Learn Syst 25(6):1021–1032

Blei DM, Jordan MI (2006) Variational inference for dirichlet process mixtures. Bayesian Anal 1(1):121–143

Byttner S, Nowaczyk S, Prytz R, Rögnvaldsson T (2013) A field test with self-organized modeling for knowledge discovery in a fleet of city buses. In: Proceedings of 2013 IEEE international conference on mechatronics and automation (ICMA), pp 896–901

Byttner S, Rögnvaldsson T, Svensson, M (2007) Modeling for vehicle fleet remote diagnostics. Technical paper 2007-01-4154, Society of Automotive Engineers (SAE)

Byttner S, Rögnvaldsson T, Svensson M (2011) Consensus self-organized models for fault detection (COSMO). Eng Appl Artif Intell 24:833–839

Byttner S, Rögnvaldsson T, Svensson M, Bitar G, Chominsky W (2009) Networked vehicles for automated fault detection. In: Proceedings of IEEE international symposium on circuits and systems

Cha S-H (2007) Comprehensive survey on distance/similarity measures between probability density functions. Int J Math Models Methods Appl Sci 1:300–307

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41:15:1–15:58

Chen H, Tiňo P, Rodan A, Yao X (2014) Learning in the model space for cognitive fault diagnosis. IEEE Trans Neural Netw Learn Syst 25(1):124–136

Csiszár I, Shields PC (2004) Information theory and statistics: a tutorial. Found Trends™ Commun Inf Theory 1:417–528

D'Silva SH (2008) Diagnostics based on the statistical correlation of sensors. Technical paper 2008-01-0129, Society of Automotive Engineers (SAE)

Fan Y, Nowaczyk S, Rögnvaldsson T (2015a) Evaluation of self-organized approach for predicting compressor faults in a city bus fleet. Procedia Comput Sci 53:447–456

Fan Y, Nowaczyk S, Rögnvaldsson T (2015b) Incorporating expert knowledge into a self-organized approach for predicting compressor faults in a city bus fleet. Front Artif Intell Appl 278:58–67

Fan Y, Nowaczyk S, Rögnvaldsson T, Antonelo EA (2016) Predicting air compressor failures with echo state networks. In: PHME 2016: proceedings of the Tthird European conference of the prognostics and health management society 2016. PHM Society, pp 568–578

Filev DP, Chinnam RB, Tseng F, Baruah P (2010) An industrial strength novelty detection framework for autonomous equipment monitoring and diagnostics. IEEE Trans Ind Inform 6:767–779

Filev DP, Tseng F (2006) Real time novelty detection modeling for machine health prognostics. In: Annual meeting of the North American fuzzy information processing society NAFIPS, IEEE Press

Gogoi P, Bhattacharyya D, Borah B, Kalita JK (2011) A survey of outlier detection methods in network anomaly identification. Comput J 54:570–588

Gupta M, Gao J, Aggarwal CC, Han J (2013) Outlier detection for temporal data: a survey. IEEE Trans Knowl Data Eng 25:1–20

Hansson J, Svensson M, Rögnvaldsson T, Byttner S (2008) Remote diagnosis modelling. U.S. Patent 8,543,282 filed May 12, 2008, and issued Sept 24, 2013

Hines J, Garvey D, Seibert R, Usynin A (2008a) Technical review of on-line monitoring techniques for performance assessment. In: Volume 2: theoretical issues. Technical review NUREG/CR-6895, vol 2. U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, DC 20555-0001

Hines J, Garvey J, Garvey DR, Seibert R (2008b) Technical review of on-line monitoring techniques for performance assessment. In: Volume 3: limiting case studies. Technical review NUREG/CR-6895, vol 3. U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, DC 20555-0001

Hines J, Seibert R (2006) Technical review of on-line monitoring techniques for performance assessment. In: Volume 1: state-of-the-art. Technical review NUREG/CR-6895. U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, DC 20555-0001

Isermann R (2006) Fault-diagnosis systems: an introduction from fault detection to fault tolerance. Springer, Heidelberg

Jardine AKS, Lin D, Banjevic D (2006) A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech Syst Signal Process 20:1483–1510

Kargupta H, Bhargava R, Liu K, Powers M, Blair P, Bushra S, Dull J (2004) VEDAS: a mobile and distributed data stream mining system for real-time vehicle monitoring. In: Fourth international conference on data mining

Kargupta H, Gilligan M, Puttagunta V, Sarkar K, Klein M, Lenzi N, Johnson D (2010) MineFleet®: the vehicle data stream mining system for ubiquitous environments (Lecture Notes in Computer Science), vol 6202. Springer, pp 235–254

Kargupta H, Puttagunta V, Klein M, Sarkar K (2007) On-board vehicle data stream monitoring using minefleet and fast resource constrained monitoring of correlation matrices. New Gener Comput 25:5–32

Karlsson N, Svensson M, Nowaczyk S, Byttner S, Prytz R, Rögnvaldsson T (2013) A method for monitoring the operation of a sensor. U.S. Patent 20,160,232,723 filed Oct 4, 2013

Lapira ER (2012) Fault detection in a network of similar machines using clustering approach. Ph.D. thesis, University of Cincinnati

Lapira ER, Al-Atat H, Lee J (2011) Turbine-to-turbine prognostics technique for wind farms. U.S. Patent 8,924,162 filed Nov 12, 2012, and issued Dec 30, 2014

Laxhammar R (2014) Conformal anomaly detection. Ph.D. thesis, University of Skövde

Ma J, Jiang J (2011) Applications of fault detection and diagnosis methods in nuclear power plants: a review. Prog Nucl Energy 53:255–266

McClelland JL, Rumelhart DE (eds) (1988) Explorations in parallel distributed processing: a handbook of models, programs, and exercises. MIT Press, Cambridge

Patcha A, Park J-M (2007) An overview of anomaly detection techniques: existing solutions and latest technological trends. Comput Netw 51:3448–3470

Pele O (2011) Distance functions: theory, algorithms and applications. Ph.D. thesis, The Hebrew University of Jerusalem

Peng Y, Dong M, Zuo MJ (2010) Current status of machine prognostics in condition-based maintenance: a review. Int J Adv Manuf Technol 50:297–313

Pimentel MA, Clifton DA, Clifton L, Tarassenko L (2014) A review of novelty detection. Signal Process 99:215–249

Prytz R, Nowaczyk S, Rgnvaldsson T, Byttner S (2015) Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. Eng Appl Artif Intell 41:139–150

Quevedo J, Chen H, Cugueró MÀ, Tiňo P, Puig V, Garciá D, Sarrate R, Yao X (2014) Combining learning in model space fault diagnosis with data validation/reconstruction: application to the Barcelona water network. Eng Appl Artif Intell 30:18–29

Rajagopalan S, Wang Y-Y, Feldmann S (2011) Offset and slow response diagnostic methods for NOx sensors in vehicle exhaust treatment applications. U.S. Patent 8,930,121 filed Apr 7, 2011, and issued Jan 6, 2015

Riemer M (2013a) Days out of service: the silent profit-killer—why fleet financial and executive management should care more about service & repair. White paper, DECISIV

Riemer M (2013b) Service relationship management—driving uptime in commercial vehicle maintenance and repair. White paper, DECISIV

Rögnvaldsson T, Norrman H, Byttner S, Järpe E (2014) Estimating p-values for deviation detection. In: 2014 IEEE eighth international conference on self-adaptive and self-organizing systems. IEEE Computer Society, pp 100–109

Rowley J (2007) The wisdom hierarchy: representations of the DIKW hierarchy. J Inf Sci 33:163–180

Rubner Y, Tomasi C, Guibas L (2000) The earth mover's distance as a metric for image retrieval. Int J Comput Vis 40:99–121

Sodemann AA, Ross MP, Borghetti BJ (2012) A review of anomaly detection in automated surveillance. IEEE Trans Syst Man Cybern 42:1257–1272

Theissler A (2017) Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. Knowl Based Syst 123:163–173

Vachkov G (2006) Intelligent data analysis for performance evaluation and fault diagnosis in complex systems. In: Proceedings of the IEEE international conference on fuzzy systems, 2006. IEEE Press, pp 6322–6329

Vovk V, Gammerman A, Shafer G (2005) Algorithmic learning in a random world. Springer, New York

Xie M, Han S, Tian B, Parvin S (2011) Anomaly detection in wireless sensor networks: a survey. J Netw Comput Appl 34:1302–1325

Zhang J (2013) Advancements of outlier detection: a survey. ICST Trans Scalable Inf Syst 13:e2:1–e2:26

Zhang Y, Gantt GW Jr, Rychlinski MJ, Edwards RM, Correia JJ, Wolf CE (2009) Connected vehicle diagnostics and prognostics, concept, and initial practice. IEEE Trans Reliab 58:286–294

Zimek A, Schubert E, Kriegel HP (2012) A survey on unsupervised outlier detection in high-dimensional numerical data. Stat Anal Data Min 5:363–378