Machine learning methods for vehicle predictive maintenance using off-board and on-board data

Rune Prytz

LICENTIATE THESIS | Halmstad University Dissertations no. 9

Machine learning methods for vehicle predictive maintenance
using off-board and on-board data

Halmstad University Dissertations no. 9
ISBN: 978-91-87045-18-9 (printed)
ISBN 978-91-87045-17-2 (pdf)
Publisher: Halmstad University Press, 2014 | www.hh.se/hup
Printer: Media-Tryck, Lund

Abstract

Vehicle uptime is getting increasingly important as the transport solutions become more complex and the transport industry seeks new ways of being competitive. Traditional Fleet Management Systems are gradually extended with new features to improve reliability, such as better maintenance planning. Typical diagnostic and predictive maintenance methods require extensive experimentation and modelling during development. This is unfeasible if the complete vehicle is addressed as it would require too much engineering resources.

This thesis investigates unsupervised and supervised methods for predicting vehicle maintenance. The methods are data driven and use extensive amounts of data, either streamed, on-board data or historic and aggregated data from off-board databases. The methods rely on a telematics gateway that enables vehicles to communicate with a back-office system. Data representations, either aggregations or models, are sent wirelessly to an off-board system which analyses the data for deviations. These are later associated to the repair history and form a knowledge base that can be used to predict upcoming failures on other vehicles that show the same deviations.

The thesis further investigates different ways of doing data representations and deviation detection. The first one presented, COSMO, is an unsupervised and self-organised approach demonstrated on a fleet of city buses. It automatically comes up with the most interesting on-board data representations and uses a consensus based approach to isolate the deviating vehicle. The second approach outlined is a supervised classification based on earlier collected and aggregated vehicle statistics in which the repair history is used to label the usage statistics. A classifier is trained to learn patterns in the usage data that precede specific repairs and thus can be used to predict vehicle maintenance. This method is demonstrated for failures of the vehicle air compressor and based on AB Volvo's database of vehicle usage statistics.

Rune Prytz, Uptime & Aftermarket Solutions,
Advanced Technology & Research,
Volvo Group Trucks Technology, Box 9508, SE-200 39 Malmö, Sweden.
mail: rune.prytz@volvo.com

List of attached publications

Paper I	R. Prytz, S. Nowaczyk, S. Byttner, "Towards Relation Discovery for Diagnostics" in *KDD4Service '11: Proceedings of the First International Workshop on Data Mining for Service and Maintenance*,San Diego, California, 2011.
Paper II	T. Rögnvaldsson, S. Byttner, R. Prytz, S Nowaczyk, "Wisdom of Crowds for Self-organized Intelligent Monitoring of Vehicle Fleets", submitted to *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2014.
Paper III	R. Prytz, S. Nowaczyk, T. Rögnvaldsson, S. Byttner, "Analysis of Truck Compressor Failures Based on Logged Vehicle Data", in *In Proceedings of the 9th International Conference on Data Mining (DMIN'13),* Las Vegas, NV, USA. July 2013.
Paper IV	R. Prytz, S Nowaczyk, T Rögnvaldsson, S Byttner, "Predicting the Need for Vehicle Compressor Repairs Using Maintenance Records and Logged Vehicle Data" submitted to *Engineering Applications of Artificial Intelligence*, 2014.

Contents

Chapter 1
Introduction

The European Commission forecasts a 50% increase in transportation over the next 20 years. This will lead to a capacity crunch as the infrastructure development will not match the increase in traffic. It will require high efficient transportation solutions to maintain the transport performance of today. Together with the demand for sustainable transport solutions, more complex transport systems will evolve. Such transportation systems could be modal change systems for cargo and bus rapid transit systems for public transportation. In these the vehicle is an integrated part of the complete transportation chain. High reliability and availability become increasingly important as the transportation systems get more complex and depend on more actors.

High transport efficiency is also important in today's traffic as haulage is a low margin business with a high turnover. Profit can easily turn into loss by unexpected changes in external conditions such as fuel prices, economic downturns or vehicle failures. By continuously monitoring transport efficiency haulage companies can increase competitiveness and stay profitable. This is enabled with advanced Intelligent Transport System (ITS) solutions, such as Fleet Management Softwares (FMS), which provide efficient haulage management.

Fleet Management Software, such as DECISIV-ASIST [Reimer, 2013a,b] and Cirrus-TMS [TMS, 2013] , monitors the utilisation and availability of the fleet of vehicles closely. These software streamline the day to day operation by offering services like route and driver planning, maintenance planning and invoice handling. This reduces waiting time at cargo bays, workshops and border crossings as the paperwork is automated. Thus the utilisation increases with smarter routes, increasing back-haulage and higher average speed due to less waiting time.

Vehicle reliability and thus availability, or uptime, is increasingly important to haulers as FMS systems become more widespread. Reliability is the next area of improvement and the demand for less unplanned stops is driven by the fierce competition in haulage as most of the other parts of their business already is optimised. Reliability is partly controllable through vehicle quality and partly by preventive maintenance actions and driver training. Preventive maintenance reduces the risk of unplanned stops, while it may increase the spending on maintenance. Other ways of handling

the risk of unplanned stops are by insurances and spare transport capacity, e.g. having redundant vehicles.

A vehicle lease program with a corresponding service contract is another way of handling the risk of unplanned stops. Relationship based business models, such as a lease program or service contract, give haulers more stability as their vehicle expense is predictable. Vehicle uptime is likely to improve as the maintenance responsibility is either shared with, or completely moved to, the vehicle manufactures. They benefit from vehicle expert knowledge and the experience of previous failures and maintenance strategies from other customers. This information levers manufactures above even the largest hauler when it comes to experience and expertise.

Nonetheless, relationship based business models are a huge challenge to the manufactures. Traditionally the profit originates from sales of vehicles and spare parts. To put it simple, the more vehicles and parts sold the larger the profit. A relationship based business model turns this upside down. The fewer spare parts used, while maintaining the uptime throughout the contract time, the larger the profit.

1.1 Background

1.1.1 Maintenance planning

Maintenance can be planned and carried out in different ways. The three common planning paradigms are corrective, preventive and predictive maintenance.

Corrective maintenance is done after a failure has occurred and it often causes downtime. This maintenance policy, or actually lack of policy, is common for infrequent failures or where the repair is very expensive. Corrective maintenance is also common practice in case the system has redundancy, e.g. for hard drive arrays in servers where a failing, redundant hard drive causes no downtime or loss of data.

Preventive maintenance is the common practise in the automotive industry, where vehicle components are replaced or overhauled periodically. It is a crude policy which enforces maintenance actions at a given vehicle age regardless of vehicle status. Some vehicles will get repairs in time while others fail prior to the scheduled repair date.

The maintenance periodicity is linked to the projected vehicle age and it is decided a priori. Common ways of estimating vehicle age is by calendar time, mileage or total amount of consumed fuel. The latter is used to individually adjust the maintenance interval of passenger cars based on usage and driving style.

Predictive maintenance employs monitoring and prediction modelling to determine the condition of the machine and to predict what is likely to fail and when it is going to happen. In contrast to the individually adjusted maintenance interval, where the same maintenance is conducted but shifted backwards or forward in time or mileage, this approach also determines what shall be repaired or maintained.

Predictive maintenance is related to on-board diagnostics featuring fault detection and root cause isolation. Predictive maintenance takes this one step further by predicting future failures instead of diagnosing already existing.

A vehicle, or machine, is often subject to several maintenance strategies. Different subsystems can be maintained according to different plans. That is, a vehicle is usually maintained both by preventive (fluids, brake pads, tires) and corrective (light bulbs, turbo charger, air compressor) strategies.

1.1.2 Fault Detection and Isolation (FDI)

A vehicle is a complex mechatronic systems composed of subsystems such as brakes, engine and gearbox. A typical subsystem consists of sensors, actuators and an electromechanical process which needs to be controlled. The sensors and actuators are connected to an Electronic Control Unit (ECU) which controls and monitors the process. It is also connected to an in-vehicle Controller Area Network (CAN) through which the different subsystems and the driver communicate with each other.

A *fault detection system* detects a change from the normal operation and provides a warning to the operator. The operator assesses the warning by analysing the extracted features provided by the detection system and takes appropriate action. No diagnostic statement is made by the system as opposed to a diagnostics system such as Model Based Diagnostics (MBD). Diagnostic statements are possible to derive analytically if the observed deviation is quantifiable and possible to associate with a known fault, or failure mode. A deviation observed prior to a repair while not seen after is likely to be indicative of the fault causing the repair.

The physical relationship between the inputs and outputs of the process can be modelled and the vehicle's health is assessed by monitoring the signals and comparing them to a model of a faultless process. This is the basis of Model Based Diagnostics (MBD), Condition Based Maintenance (CBM) and various fault detection methods, to detect and predict upcoming failures. These approaches are known as *knowledge based fault detection and diagnosis* [Isermann, 2006] methods and they require human expert knowledge to evaluate the observed variables and deduct a diagnosis.

Traditional FDI systems are implemented on-board the monitored machine, as they require real-time data streams through which a subsystem can be monitored. Further, failure mode information is required to build failure mode isolation models. Typical sources are heuristic information of prior failures, fault injection experiments or a simulation of the system under various faulty conditions.

1.1.3 On-board data acquisition

Large scale data acquisition on vehicles (on-board) is difficult as the vehicles are constantly on the move. Data must be stored on-board for retrieval at a workshop or transmitted through a telematics gateway. As vehicles move across continents and borders, wireless downloads get expensive and hence in practice limited to small chunks of data. In-vehicle storage of data streams is not yet feasible as they require huge amount of storage which still is costly in embedded systems.

The development cost aspect of large scale on-board logging solutions is also a major reason to why it has not been done before. The logging equipment must be developed, rigorously tested and mass-produced. This does not fit well with the tough competition in the transport sector where vehicle manufacturers need to see a clear economic benefit for each function included in the vehicle.

The on-board data consists of thousands of signals from the sensors and ECUs, that are communicated through a CAN network. They are sent repeatedly with a specified frequency and form streams of continuous data which are used for vehicle control and status signalling between the different vehicle components.

So far, continuous vehicle on-board logging has been limited to fleets of test-vehicles and with retrofitted logging equipment. These systems are expensive and intended for product development purposes. It is probably safe to assume that any industrialized on-board monitoring or predictive maintenance algorithm must be limited to existing hardware with respect to sensors, signals and computational resources.

1.1.4 Off-board sources

Most large corporations, like the Volvo Group, have accumulated large amounts of data over the years in off-board databases. The data spans from drawings and test results to maintenance records and vehicle usage statistics. Structured right, the data can be transformed into knowledge and prove useful in various application areas. The maintenance records and usage statistics is of particular interest in this thesis because of their direct and indirect association with future vehicle maintenance.

Vehicle statistics

The usage statistics database, named the Logged Vehicle Data database (LVD), is limited to aggregated data. The data is aggregated on-board every Volvo vehicle and is either transmitted wirelessly through a telematics gateway or downloaded at a workshop. The update frequency is at best once per month but normally every three and six months. The frequency is unknown a priori even though vehicles regularly visit workshops for maintenance.

The LVD database includes aggregated statistics such as *mean vehicle speed* and *average fuel consumption*, which have been collected during normal operation. It provides valuable insights in how usage, market and customer segment affect key vehicle performance parameters. This is useful input into the development of future vehicles.

Maintenance records

The Vehicle Maintenance database (VSR) contains records of all maintenance conducted at a Volvo authorised workshops. The data is used for quality statistics during the warranty period as well as customer invoicing. The entries are structured with standardised repair codes and part numbers. The root cause is sometimes specified in

the repair record but, in most cases, must be deducted based on the reported repair actions.

Much of the data in the VSR database are manually entered and often suffer from human mistakes such as typos, empty records, or missing root cause. The VSR also contains systematic errors introduced by the process of entering new data to the database. The date of repair is not the actual date of the repair but rather the date of the entry to the database. Further, extensive repairs are reported as consecutive repairs a few days apart. This does not cause any problems in the day to day operation as the purpose of the system is to link invoices to repairs and give the mechanic an overview of the history of a vehicle.

Moreover, the records do not always reveal if the problem was solved by the performed repair. All these problems complicate the process of automatically determining the root cause of a repair and matching it to found deviations in e.g. the LVD data. Undocumented repairs, e.g. repairs done by unauthorised workshops, are also a problem as they cause the deviations to disappear without any records of the repair. The number of these repairs is unknown but it is believed that the frequency is increasing as the vehicle ages. They introduce uncertainty whether a found pattern can be linked to a given repair as the support might be low.

The aggregation of data smoothens any sudden change, which further reduces the possibility of finding statistically valid relations between deviations and upcoming repairs.

The database is extremely heterogeneous, as model year and vehicle specification affect the parameter set logged to the database. Only a small subset of parameters is common among all vehicles. In general terms, newer and more advanced vehicles provide more statistics. This makes it hard to find large datasets with a homogeneous set of parameters.

1.2 Problem formulation

The vehicle industry is extensively using manually engineered knowledge based methods for diagnostics and prognostics. These are development resource intense and limits the adoption to systems which are expensive, safety critical or under legal obligation of monitoring. The demand for more uptime requires less unplanned maintenance which in return drives maintenance predictions. This requires more universal deployment of diagnostics or maintenance predictions as the systems that today are under diagnostic monitoring only account for a small fraction of all repairs.

The heuristic information with regard to what failures to expect is hard to come by. A study, based on Failure Mode Analysis (FMEA) by [Atamer, 2004], compares anticipated failures on aircrafts to actual faults while in service. The overlap, illustrated in figure 1.1, is surprisingly low, only about 20%. A 100% overlap means that all faults are anticipated. This is not ideal as some faults are so severe that they should be completely avoided. Non-critical faults should be anticipated and it is likely that they account for more than the 20% overlap that was found in the survey.

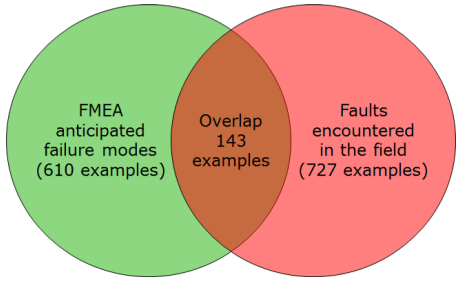


Figure 1.1: Overlap of known failure modes and actual failures of an aircraft.

This merits exploratory methods based on actual failures to deduct likely failure modes. This thesis presents two methods for data mining the vehicle maintenance records and vehicle usage data to learn usage or wear patterns indicative of failures. This requires detailed maintenance records where the failure root cause can be deducted with accurate date or mileages of the repair.

Further, more wide-spread adoption of predictive maintenance calls for automatic and less human-resource demanding methods, e.g. unsupervised algorithms with lifelong learning. Such methods are easier to scale up and they can thus be ubiquitously applied since much of the modelling is automated and requires little or no human interaction.

Supervised and unsupervised machine learning methods have proven successful to predict the need of maintenance. Kargupta et al. [2010], Rögnvaldsson et.al [Paper II] and Filev et al. [2010] all propose agent based methods while Frisk et al. [2014], Prytz et.al [Paper IV] and Zhang et al. [2009] propose centralised methods based on connected fleets of vehicles. The method developed by Kargupta is distinguished from the rest by being the only commercially available, third party, solution. All others are in cooperation with vehicle manufactures.

Maintenance predictions can be enhanced by combining the deviations in on-board data with off-board data sources such as maintenance records and failure statistics. This is exclusive product knowledge, only accessible to the vehicle manufacturers, which gives them an advantage in predicting maintenance. Still, data mining has yet to become a core competence of vehicle manufacturers, which makes the road to industrialisation long.

The aim of this thesis is to investigate how on-board data streams and off-board data can be used to predict the vehicle maintenance. More specifically, how on-board data streams can be represented and compressed into a transmittable size and still be relevant for maintenance predictions. Further, the most interesting deviations must be found for a given repair which requires new ways of combining semantic maintenance records with deviations based on compressed on-board data.

Chapter 2
Predicting maintenance needs in vehicles

2.1 Present business solutions

Solutions like remote diagnostics and monitoring are typically combined with predictive maintenance and other services like vehicle positioning and remote road-side assistance. These are often sold as an integrated part of a lager service offer, such as a FMS system.

Commercial vehicle manufacturers have not yet put any advanced predictive solutions on the market. Simpler predictive maintenance solutions exist, where wear and usage of brake pads, clutches and similar wear-out equipment is monitored and projected into the future. All of these are based on data streams being aggregated onboard and transmitted to back-office system. Mercedes [Daimler, 2014] and MAN [MAN, 2014], among others, offer direct customer solutions for preventive maintenance recommendations and remote monitoring. Volvo has chosen to incorporate predictive maintenance as dynamic maintenance plans offered in conjunction with service contracts.

Volvo [Volvo, 2014] has recently made advances in Remote Diagnostics by predicting the most likely immediate repair given a set of diagnostic trouble codes (DTC). Active DTCs are sent wirelessly over the telematics gateway to a back-office service team which, both manually and automatically, deducts the most probable cause of failure. Further, services like repair, towing and truck rental is offered to the customer. This is not predictive maintenance as such, as it does not comprehend any future failures. It is still a valuable service which reduces downtime and saves money.

The passenger car industry is surprisingly ahead of the commercial vehicle manufacturers in predicting maintenance. Commercial vehicles, i.e. trucks, buses and construction machines, are designed for business purposes where *total cost of ownership* is generally the most important aspect. Passenger cars on the other hand, are designed and marketed to appeal the driver's feelings for freedom, speed and so on. There are several published attempts of offering predictive maintenance solutions. The level of

maturity varies from pilot studies to in-service implementations. Volkswagen [Volkswagen, 2014], BMW [Weiss, 2014] and GM [OnStar, 2014] all have methods to predict future maintenance needs based on telematics solutions and on-board data. VW and BMW offers predictive maintenance as a maintenance solution for an owner, while GM, through the *OnStar* portal, publishes recommended repairs.

2.2 State of the art

On-board solutions have unrestricted, real-time, access to the data streams. This enables fast detection as the detection algorithms are located close to the data source. On-board solutions typically have limited computational and storage capacity since the hardware needs to be of automotive grade, e.g. water, shock and EMC resistant, and inexpensive as well. Generally automotive electronics usually lingers two or three generations behind the consumer market.

On-board solutions impose challenges in the comparison of relevant reference models of normal and faulty conditions as the methods need to be computationally inexpensive and have a small memory footprint. The off-board methods have larger computational resources and access to historical data. The data can be clustered and labelled with failure root cause to form a knowledge base of operating modes linked to failures and normal conditions. This enables a data driven approach which is more precise in fault isolation although slower in fault detection compared to the on-board methods.

Moreover, on-board systems have direct access to the data which it can sample as often as necessary. It enables the detection of small deviations early in the progress of wear or failure. This leads to predictive requirements as the deviation acts as an early warning of failure. Off-board system require wireless technology to sample the off-board data. The sample frequency is low and it requires a predictive maintenance approach which relies on actual predictions rather than being reactive based on deviations caused by already existing failures. Reactive actions are likely to be too late when the sampling frequency decreases.

Three approaches can be used to predict vehicle maintenance; Remaining Useful Life (RUL) prediction, Deviation Detection and supervised classification. They can use both historical and real-time data as input. Table 2.1 illustrates the approaches presented in related work including the type of input data. Notably, no method is found which utilises both historical and real-time data, and this leaves room for further research in this area.

The deviation detection methods are distinguished from the other methods as they do not give specific information of what is wrong. The failing component is inherently known in the RUL approach as these models are developed for a specific component. The classification approach can both be used to detect deviations and to isolate the actual failing component.

Deviation detection is an easier problem to solve since the uncertainty in linking deviations to failures is left out. To make deviation detection methods complete (and comparable to RUL and Classification methods) they need to be combined with his-

torical examples of similar deviations linked to known failures. This introduces one more uncertainty into the complete prediction, which reduces the prediction accuracy.

2.2.1 Learning from streams of on-board data

The research in predictive maintenance and prognostics using on-board data streams in the automotive industry is small. Only a few different methods have so far been presented.

Filev et al. [2010] extends CBM with software agents and implements it on-board. An expert defined feature set is clustered on-board using Gaussian Mixuture Models and k Nearest Neighbours (kNN). The clusters are called operation modes (OM) and reflect different driving conditions like idling, driving and reversing. Every new data sample is assigned to an operating mode. New OMs are created as new and unique feature vectors are discovered. A health value is assigned to each OM. It is proportional to the number of observations assigned to it multiplied by the age of the OM. New OMs with frequent assignments of feature vectors are considered failures. The method works without any agent cooperation and thus without any communication requirements.

D'Silva [2008] presents an on-board method which assumes pair-wise correlations between signals of a complex system that is stationary. The cross correlation matrix is derived and the Mahalanobis distance is used as metric to find anomalies. The complete correlation matrix is used to define the vehicle status. The normal operation space is defined from experimental data. The system works in an on-board fashion with stored models of the normal operation and it has been demonstrated on simulated data.

Similarly, Vachkov [2006] and Kargupta et al. [2010] have exploited stationary signal relationships to find anomalies. Vachkov presents a method in which a neural gas models is used to represent data relationships. The system comprises of an on-board part which continuously monitors the vehicle and uploads the models to an off-board counterpart. The off-line system includes a database which stores data models of faulty and fault free systems. The state of health is determined by an off-board algorithm which compares the incoming data models with the stored examples in the database.

Kargupta et al. [2010] monitor the individual correlations between the on-board signals. A novel method of calculating the correlation matrix in a fast and efficient way enables deployment in an embedded system with scarce resources. Sudden changes in the correlation matrix are signs of wear or failures. These trigger a set of algorithms that collect data, calculate some statistics and transmit the result over a wireless link. The prognostic statement is made off-board.

Mosallam et al. [2014] also propose an unsupervised method that takes training data labelled with RUL as input to an initial step. One RUL model per machine is derived during training and stored in a database as references of degradation patterns. The database is deployed on-board and an algorithm tries to match the degradation of the monitored system to one of the references systems stored in the database. A

Bayesian approach is used to calculate the most likely RUL by looking at the similarities to the reference models.

Quevedo et al. [2014] propose a method for fault diagnosis where the deviation detection is made in the model space. Quevedo et.al focuses on echo state networks (ESN) and proposes a model distance measure which is not based on the model parameters, but rather on characteristics of the model. It makes the distance invariant to the parametrisation resulting in a more stable result. The approach is much like Rögnvaldsson et.al's approach [Paper II], although he either use the model parameter space or model space as the space for learning and does not use ESNs. Quevedo et.al further proposes an ensemble of one-class classifiers for learning the fault modes.

Angelov et al. [2008] present an approach to design a self-developing and self-tuning inferential soft-sensor. It is a similar research topic which shares the problem of finding a model automatically, without deeper knowledge of the underlying process structure. His method automatically learns new operating modes and trains a separate linear model. The operating modes are learned using a fuzzy rule set and the linear models are identified using weighted recursive least square (wRLS). A new fuzzy set is added each time the estimation residual suddenly increases. Fuzzy sets that have not been used fome some time are removed to keep the number of sets low. This method share the same underlying principle of Filev et al. [2010]' Operating Modes but differs with respect to model structure and application.

2.2.2 Learning from already collected records of aggregated data

There is not much research on the topic of using existing data sources for predicting vehicle maintenance and only three methods are known to specifically predict vehicle maintenance using historical data. The area has a lot in common with the general data mining such as supervised classification, which is not covered here.

Zhang et al. [2009] propose a method in which a small amount of aggregated data is collected from thousands of passenger cars. The data are collected using telematics and the material is analysed off-board using rules that have been defined by an expert. The method has been demonstrated on real-world data to predict no-start events caused by drained starter batteries.

The starter battery is yet again considered by Frisk et al. [2014]. This time Remaining Useful Life (RUL) is modelled from aggregated data. The main difference from Zhang et al. is that they model the full RUL curve instead of assessing if the vehicle is likely to start or not. Their dataset has been collected from heavy duty trucks and contains quantitative and qualitative data, in total about 300 variables. Furthermore, the data is highly unbalanced or right censored. The important variables are discovered using the AUC criterion and the Random Survival Forest technique is used to model the RUL.

Buddhakulsomsiri and Zakarian [2009] rely solely on warranty or maintenance records. By analysing the records for patterns of sequential repairs they derive simple IF-THEN rules that capture the relationship between common repairs. This can be used to predict the most likely next repair based on what has already failed on a

Method	Historical data	Real-time data
RUL prediction	Frisk et al. [2014]	Mosallam et al. [2014]
Deviation detection		Rögnvaldsson et.al , [Paper II] Filev et al. [2010], Quevedo et al. [2014], Angelov et al. [2008]
Classification the need of maintenance	Zhang et al. [2009], Prytz et al. [Paper IV], Buddhakulsomsiri and Zakarian [2009]	D'Silva [2008], Vachkov [2006], Kargupta et al. [2010]

Table 2.1: Table of methods and references for predictive maintenance

vehicle without any communication with the vehicle. The vehicle is only indirectly captured through prior failures linked to specific usage.

2.2.3 Contributions

Two approaches to predict vehicle maintenance is put forward in this thesis. The first one, presented in paper II and partly in paper I, presents an unsupervised self-learning method for predictive maintenance based on streaming on-board data. It specifically aims at tackling the bottleneck of manually crafted predictive algorithms by using life-long learning of upcoming failures.

The second approach relies on off-board data sources for maintenance predictions and uses supervised classification to predict the future maintenance of vehicles. It is first presented in paper III and developed further in paper IV.

Paper I is an attempt of using linear relations, which are automatically discovered in on-board data, for fault diagnosis. The method finds the strong relationships within the parameters of a dataset. It was evaluated in a controlled fault injection experiment where a truck drove the same route with a known set of faults. The method presented is unsupervised with the exception of some thresholds set by the user and manual clustering of the system signals as an ambient or system originated signal.

Paper II presents a method which is based on Byttner et al.'s earlier work [Byttner et al., 2007, 2008, 2009, 2011] on self-organised models and consensus based deviation detection. Both linear relations and histograms are used to model the data streams to find the deviations. This paper is the first attempt to use this method in a real world setting with buses in daily operation. The paper further expands the methods with ways to measure the deviation distance and matching them to interesting vehicle maintenance events. The method is distinguished from the works of Quevedo et.al and Angelov et.al by being specifically crafted for systems of low resources such as bandwidth and computational. The deviations are found on the data collected over

a week or day while the methods proposed by Quevedo and Angelov monitor the systems more closely.

Paper III is an early attempt of using supervised classification to predict vehicle maintenance. Large databases of historical data are mined to find patterns and match them to subsequent repairs. The method uses already existing data which were and still are collected for other purposes. This makes the method interesting as the economical thresholds of implementation are low, but technical problems with data quality arise.

Paper IV develops the ideas of paper III and takes it all the way from raw data to maintenance prediction. The concept of paper III is extended with feature selection, methods to handle unbalanced datasets and ways to properly train and test data. The method is complete in the sense that it makes predictions of repairs and not just deviation detection. Moreover, paper IV makes the process of finding the predictive models automatic and without any human in the loop which is in contrast to Zhang's expert rules. Further, the method uses both usage and warranty history whereas Buddhakulsomsiri and Zakarian [2009] only rely on warranty data. This makes the prediction method in paper IV susceptible to usage differences and not only to prior repairs.

Chapter 3
Methodology

3.1 Learning from historical data

3.1.1 Motivation

This method is an approach based on supervised classification and historical, off-board, sources of data. These data sources already exist and are cheap to explore as no investments are needed for data collection, IT-infrastructure or in-vehicle hardware. The quality of the data limits the precision in predictions and the results shall be judged considering the implementation costs and speed of deployment.

This off-board data has a very low and irregular update frequency. Figure 3.1 illustrates the update frequency (right panel) and how the amount of data varies with the vehicle age. A lot of repair data are missing from older vehicles. This is unfortunate since it would have been very useful.

The historical data enables ubiquitous implementation of predictive maintenance in vehicles while still being cost effective as the data is already available. Further, doing classification instead of predicting RUL is likely to require less data as no continuous and monotonically decreasing life-time prediction is necessary. Instead, the classification approach can be seen as an advanced threshold which, when crossed, triggers a repair instruction. The threshold can be dynamic and vary with input conditions such as vehicle usage.

RUL models are desirable if the predictions are accurate and with a low spread. Otherwise, a wide safety margin must taken into account which leads to shorter actual predicted life. A classification approach can then be beneficial as it can make better binary decisions on the same data. This results in lower safety margins and more accurate actual predictions.

As mentioned earlier, the LVD and VSR databases are combined, pre-processed and data mined for interesting patterns linked to vehicle air compressors. The method as such is a general supervised approach with some domain specific features and easily extended to cover other components.

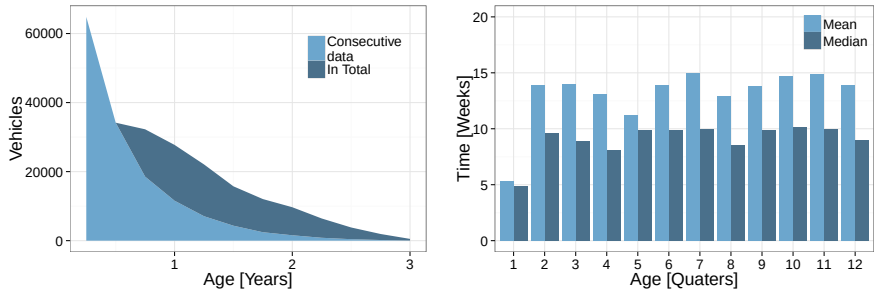


Figure 3.1: Data availability statistics

3.1.2 Pre-processing of data

The LVD and VSR databases contain many of the general problems of real-world databases as well as more specific problems related to how data is entered and structured in these databases. Typical problems are missing values and duplicate entries and more specific problems related to these sources are varying parameter sets among vehicles and uncertain repair dates. Furthermore, the LVD database only consists of aggregated data, which smoothens the trends in the data.

Both the LVD and VSR databases need pre-processing to become useful. Missing values are interpolated and duplicate entries removed. The LVD parameters are differentiated to amplify changes and restore some of the information lost in the aggregation. The date of repair entered in the VSR database is almost always wrong. This is fixed by looking at the closest prior LVD readout of the same vehicle. The date of the readout is more likely to be correct as it is automatically stored when the technician connects the workshop PC to the vehicle.

3.1.3 Dataset

The data from the LVD and VSR databases are merged into one dataset. The data from the LVD is extended with time to failure variables based on the repair dates in the VSR. The data is organised in a large matrix conceptually similar to table 3.1. The data is right censored, meaning that it contains time to failure data of vehicles which have not yet failed.

The label, which denotes the expected outcome by a predictive algorithm, is derived from the column time to failure. Examples, or readouts, with time to failure below a given threshold, called prediction horizon (PH), are labelled *Faulty* while readouts farther in the past are labelled *Normal*. The label is used to discriminate the examples into the two groups of imminent failures and safely operating vehicles. This is necessary in the supervised classification and sometimes in feature selection. Figure 3.2 illustrates how the readouts, denoted with x are labelled either *Normal* or *Faulty*. The readouts just prior to the failure (*time to failure* = *0*) are labelled *Repair*

Vehicle	Date	Mileage	LVD1	LVD2	...	Time to Failure	Label
A-12345	2010-06-01	1030023	100	35	...	125	Normal
A-12345	2010-08-15	1123001	101	25	...	50	Normal
A-87654	2010-04-20	9040223	120	29	...	110	Normal
A-87654	2010-01-21	9110223	121	26	...	21	Faulty
A-34567	2010-11-05	1330033	90	23	...	>301	Normal
A-34567	2011-03-11	1390033	121	26	...	>175	Normal

Table 3.1: Conceptual illustration of a dataset

and excluded from the experiment since these readouts are likely to belong to the same workshop visit as the air compressor repair.

3.1.4 Feature selection

Feature selection is a research topic of its own within machine learning and control theory. The methods used in machine learning are usually categorised in three categories; Filters, Embedded methods and Wrappers. Bolón-Canedo et al. [2013] has a recent overview of the current state-of-art while the book by Guyon et al. [2006] is more comprehensive.

Heterogeneous datasets, or datasets with a large percentage of missing values, cause problems when doing feature selection. Backward and forward selection methods, which are the basis of the majority of wrappers, require consistent datasets with no missing values. Filter methods can be adapted to handle or neglect missing values in various ways. *RELIEFF* [Kononenko et al., 1997] e.g. imputes missing values with their statistically most likely value.

Missing values are typically imputed in the pre-processing of the data. In that case both wrappers and embedded methods can be used on datasets with missing values. The method works well if the frequency of missing values is low, as shown by Lou and Obradovic [2012].

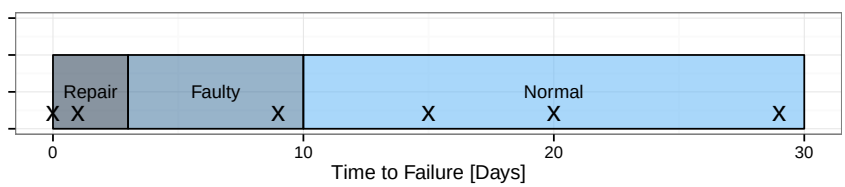


Figure 3.2: The Prediction Horizon

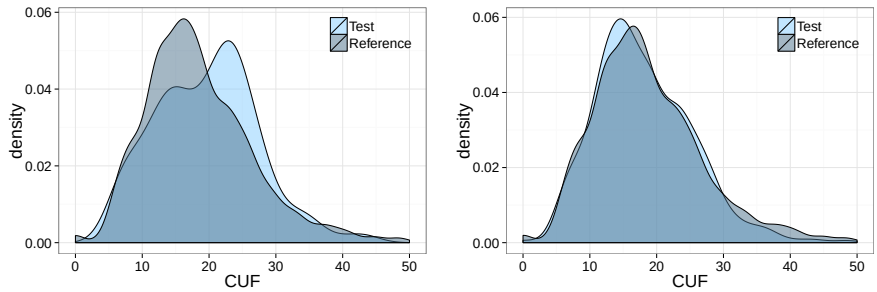


Figure 3.3: Differences in the PDF of the parameter CUF due to wear.

The LVD database contains more than 25% of missing values, which is too much to impute and this motivates the use of a filter based method over wrappers and embedded methods. Further, both a filter and a wrapper based method is proposed for finding the interesting parameters. The wrapper relies on a method for selecting vehicles such that the dataset stays consistent and avoids imputing missing values.

3.1.5 The Filter method

A filter based method finds the most important parameters by analysing them individually. The proposed method splits the data of one parameter into two groups based on the label. The Probability Density Functions (PDF) of the groups are compared with the Kolmogorov-Smirnov test [Hazewinkel, 2001]. The likelihood that the two groups come from the same distribution is calculated and expressed with a p-value. Parameters where the two groups are distinctively unique (the p-value is close to zero) are associated with the wear prior to failure and included in the dataset.

Figure 3.3 illustrates how wear affects the *Compressor Duty Cycle* (CUF parameter). The test and reference data are selected from vehicles with a documented air compressor repair and the groups *test* and *reference* are based on how the readouts were labelled. *Test* includes all data labelled as *faulty* while *reference* includes all data labelled *Normal*. The prediction horizon is set to 5 weeks in the left figure and to 25 weeks in the right figure. No sign of wear is present when the PH is set to 25 weeks prior to the repair in contrast to when the PH is set to 5 weeks.

3.1.6 Wrapper based method

The original wrapper method of Kohavi and John [1997] uses the best-first-approach with respect to accuracy or any other evaluation criterion. This requires the algorithm to be able to freely expand features without changing the dataset size. This is not the case in the data at hand as the feature sets vary between vehicles. The net feature set is the intersection of available parameters across all vehicles included in the dataset. By adding a new feature to the dataset all vehicles without this are excluded. Adding

rare features causes the dataset to drop to very few examples. This makes it important to take the reduction in dataset size into account when choosing features.

The proposed wrapper method uses beam search to find new features to include. It is a greedy graph search over the power set of all the features, looking for the subset that maximises the classification accuracy. In addition, we only expand nodes that maintain the data set size above the given threshold. The threshold is reduced with the number of parameters as shown in equation 3.1. $n_{dataset}$ denotes the minimum dataset size required and n_{all} is the number of available readouts. Further, the constraint factor is between 0 and 1 and n_{params} is the number of parameters included in the selected dataset. Each new parameter is allowed to reduce the dataset with a small fraction. This ensures a lower bound on the data set size. The top five nodes, with respect to accuracy, are stored for next iteration. This increases the likelihood to find the global optimum. The search is stopped when a fixed number of features are found.

$$n_{dataset} = n_{all} \times \mathrm{constraintFactor}^{n_{params}}. \tag{3.1}$$

3.1.7 Unbalanced datasets

Unbalanced datasets can be categorized into two groups; relatively imbalanced and imbalanced due to rare instances. Relatively imbalanced datasets have numerous examples of the minority class while this is still outnumbered by the majority class by a factor of 100 or more. These are common in many real-world problems such as the present. The main problem is not lack of examples but rather the overlap between classes, e.g. no clear boundaries between the classes. Imbalance due to rare instances, on the other hand, is hard to analyse as there is little data to learn from.

Applying a machine learning algorithm to an unbalanced dataset typically results in a high accuracy of predicting the majority class whereas the accuracy of the minority class lingers. The overall accuracy is good as the influence of the poorly predicted minority class is small. A dataset with an imbalance of 1:100 typically yields an accuracy of 95-100% of the majority class and 0-10% in the minority class. The overall accuracy is around 99%. This looks great at a first glance but it could be devastating since the classification of minority examples can be most important and misclassifications can be very costly; e.g. in medical diagnostics or fraud detection where an undetected diagnosis or attempt of fraud is costly.

Learning from an unbalanced dataset can be done in a couple of ways. The first and foremost approach is to combine a pre-processing method with a traditional machine learning algorithm. The pre-processing method restores the class balance by various sampling techniques so it can be used with an out-of-the-box classifier.

Synthetic Minority Oversampling Technique (SMOTE) [Chawla et al., 2002] is used to re-balance our dataset. Instead of undersampling the majority class the minority class is boosted with new, generated, examples. SMOTE works as follows: The K-nearest examples, S, of a given minority example, x, are found. A synthetic exam-

ple is generated along the line between x and a randomly selected member, $\hat{x}$, of S. x_{new} is placed randomly in between $\hat{x}$ and x.

3.1.8 Classification

A central assumption in machine learning (and statistics) is that of independent and identically distributed (iid) data. Most common algorithms work quite well also in cases when this assumption is not fully fulfilled.However, the iid assumption is still important, especially when evaluating and comparing different solutions as results with non-iid data tends to yield stronger patterns and misleadingly high accuracy.

The readouts of the LVD database consist of aggregated data that have been sampled at different times. Subsequent values from any given truck are highly correlated to each other. Readouts further in time from the same vehicle are also correlated as there are individual patterns that are specific to each truck. Hence, to get iid data the test and train datasets are split on a per vehicle basis and not randomly among all readouts. The data sets for training and testing contain unique, non-overlapping, sets of vehicles in order to guarantee that patterns learned are linked to wear or usage instead of specific usage patterns of individual vehicles.

3.2 Learning from real-time data streams

3.2.1 Motivation

The proposed method in paper II uses an unsupervised and self-organised approach to detect deviations. A database is built of deviations known to later cause unplanned repairs by matching the disappearance of deviations to a documented repair. The database can later be used to predict failures as already known deviations reappear in different vehicles. The method enables a life-long learning as the database gradually evolves over time and new deviations are learned.

Automatic and life-long learning enables a predictive maintenance to evolve over time and react to unforeseen (in the product development phase) phenomena or failures. The method is at the same time cost effective as it requires little manual resources, and easily expandable to new vehicle components.

3.2.2 The COSMO approach

Consensus Self-Organizing Models (COSMO), first presented in Byttner et al. [2008], are used to represent data streams on-board machines. A set of models are built to describe the relationships in the data which in return describe the physics of the system. The models enable comparisons of the machines, or vehicles in this case, through differences in their parametrisation. This is particularly useful if the vehicles are similar with respect to hardware, usage and ambient conditions. Then, wear, failures and noise are the only phenomena left to differentiate among the models.

Further, the concept of COSMO includes a self-discovery of the local model representations. Each vehicle, or agent on-board the vehicle, is set out to automatically and independently of the other members of the fleet find the best data representations. Good models, can identify failures or wear prior to a repair. They have consistent parameters over time, i.e. linear correlations close to 1, far from uniform histograms or dense clusters. A sudden parameter change is then more likely to relate to change of the underlying physical dynamic rather than to noise or modelling error.

The model exploration can be distributed on the fleet. Not all vehicles need to find all interesting models as the number of possible models can be very large. Each on-board agent is in contact with a back-office system that coordinates the search and collects the most interesting models found by each vehicle.

Each agent of the fleet uploads the model parameters of the most interesting models to the back-office. A distance measure, operating in the model parameter space, is used to compare and quantify the similarity among the models. Models, and thus vehicles, that are found to be outliers compared to the rest are flagged as deviating and indicating a potential fault. As the model family determines the parametrisation and thus the distance measure, no distance measure can be decided upon before the model family is chosen. Ideally, the distance measure supports proper statistical handling such that a hypothesis test can be performed.

So far the COSMO method only supports deviation detection. Diagnostic and fault isolation can be achieved by building a back-office database of repairs associated with prior observed deviations. This is then used to issue diagnostics statements as soon new and already known deviations are reported to the back-office system.

3.2.3 Reducing the ambient effects in on-board data streams

Paper one presents the concept of ambient filtering where each on-board signal is categorised as either measuring an ambient condition or an observation of a physical signal. Each physical signal is then modelled by using the signals in the ambient category. The modelled value is the projection, or influence, of ambient conditions to the measurement of the actual physical state. The ambient influence can then be removed by subtracting the modelled, $\hat{y}$, from the actual measured y. The resulting residual ($r = y - \hat{y}$) reflects a more stable signal with less external influences which could trigger a false deviation.

Chapter 4
Results

4.1 Paper I

Paper I presents the first steps towards an unsupervised method for discovering useful relations between measured signals in a Volvo truck, both during normal operations and when a fault has occurred. The interesting relationships are found in a two-step procedure. In the first step all valid models, defined by a MSE threshold, are found. In the second step the model parameters are studies over time to determine which are significant.

Moreover, a method for doing ambient filtering is presented. It reduces the influences of the ambient conditions which give more stable (over time) signal relations. The method is evaluated on a dataset from a controlled fault injection experiment with four different faults. One out of the four faults was clearly found while the others werer mixed up.

4.2 Paper II

Paper II applies the COSMO algorithm in a real-world setting with 19 buses. The algorithm is demonstrated to be useful to detect failures related the to cooling fan and heat load of the engine. Erroneous fan control and coolant leaks were detected at several occasions.

Eleven cases of deviations related to the coolant fan gauge were found using histograms as on-board models. It means that the coolant fan was controlled in an abnormal way. Four occurrences could be linked to fan runaways where the cooling fan is stuck at full speed. This was the result of a failure in the control of the fan, e.g. short circuit in the ECU and non-functional cooling fan valve. Further, two occurrences of coolant leaks where found as well as one jammed cylinder causing higher engine compartment temperatures that required an extraordinary amount of cooling. Three occurrences of deviations were impossible to link to any repair and they were left unexplained.

The failures discovered with the linear relations were related to the wheel speeds sensors. The sensors are crucial for safety critical systems such as Anti-Lock Brake

Systems and traction control and are thus under supervision of an on-board diagnostic algorithm. However, the proposed algorithm found the upcoming failures before the existing on-board algorithms warned. This failure was found on four of the 19 buses.

Moreover, an uptime versus downtime analysis was done on the 19 buses. The amount of downtime was measured by studying the VSR entries combined with GPS data. The downtime is defined as time spent in workshop, in transportation to and from workshop and in a broken state on the road. It was measured to 11% which, compared to operator's goal of 5%, is too high. However, much of the downtime is spent waiting at the workshop while the actual wrench time is much less.

4.3 Paper III

Paper III introduces the off-board data sources LVD and VSR and presents early results of predicting air compressor failures. Three different classifiers are evaluated and both F-score and a cost function is used as evaluation criteria. Moreover the paper discusses the problem of the dataset not being *iid* and classifiers learning individual truck behaviour in contrast to signs of wear. The paper concludes that using these off-board data sources is viable as input data for predicting vehicle maintenance, albeit it will require a lot more work.

4.4 Paper IV

Paper IV introduces the presented off-board method that uses supervised machine learning to find patterns of wear. The method is evaluated on the air compressor of a Volvo FH13 vehicles. The method evaluates a vehicle just prior to an already scheduled workshop visit. The vehicle is flagged as *Faulty* in case the vehicle's air compressor is predicted to fail before the next planned workshop visit. This results in an extra air compressor check-up at the upcoming workshop visit.

The approach was evaluated using a cost function to determine the benefit of implementing this as a predictive maintenance algorithm. The cost function considers the costs for the vehicle operator associated with an unplanned road-side stop along with the cost of replacing the air compressor during a regular workshop visit. The method was successful in achieving an economical benefit, albeit at the cost of a rather high level of false repair claims.

Figure 4.1 illustrates the performance of the classifier based on sensitivity and specificity. This figure shows the trade-off of making the classifier more sensitive to detect vehicles with an imminent failure (*sensitivity*). It casuses the rate of miss-classified normal vehicles (*specificity*) to increase as well.

The red line indicates the break-even of the business case, according to the cost function. The approach is beneficial to the vehicle owners as long as the blue line is to the right of the red. The optimum case is when the classifier is tuned to the sensitivity of 0.4 resulting in 0.9 specificity. This means that 40% of the failing air compressors were correctly predicted and 10% of the normal readouts where wrongly classified as failing.

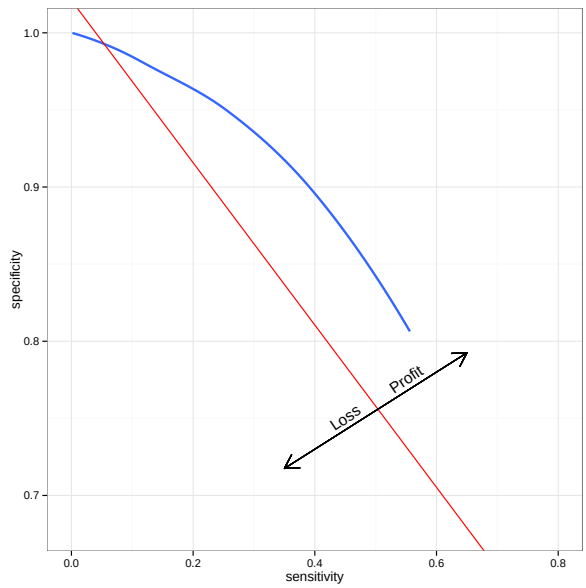


Figure 4.1: The specificity vs. sensitivity curve of the classifier for predicting air compressor replacements on Volvo FH13 trucks

Chapter 5
Discussion

The COSMO method presented in Paper II can be described as bottom-up where the on-board data source defines how the data is compressed, compared and transmitted. This is a deviation detection focused approach designed to find deviations and later associate them with a diagnosis or repair. The approach is buttom-up as it takes off from the source of the problem and then links the deviation to a likely repair.

All deviations are interesting since they give a clue of what could be wrong. However, to make an automatic repair recommendation algorithm deviations must be linked to repairs. A repair just prior to the disappearance of a deviation is interesting since it is likely the repair restored the vehicle status to normal.

The matching of all disappearing deviations and repairs is a potential large work. Deviations seen for the first time are linked to whatever repair just prior to it. While finding new occurrences of deviations already known to the systems either confirms the already known deviation-repair association rule or introduces ambiguities to what repair actually fixed the root cause.

There are several problems and limitations to this approach. Firstly, i case more than one repair has been conducted the problem of deciding which to merit the deviation needs to be addressed. In worst case this leads to miss-assigned deviation causing noisy labels and lower prediction performance.

Secondly, not all deviations occur due to wear or vehicle failures. The vehicles produced constantly change as they are updated to deal with quality issues such as on-board software bugs or hardware issues. Further, suppliers and vehicle manufacturers constantly introduce changes to reduce vehicle production cost, e.g. replacing sensors with software estimates and replacing complete components with cheaper alternatives from a different supplier.

Thirdly, the method requires time for introduction and evaluation. The targeted vehicle models need to run for years at customers before the knowledge based is large enough. Accurate statistics and good prediction accuracy is not possible to obtain before a statistically significant number of vehicles have failed and been saved by the method.

Thus, the bottom-up approach enables good deviation detection while the accuracy of the prediction is limited to the precision of matching deviations to known repairs.

The maintenance classification takes off from the maintenance records and is thus a top-down approach in comparison to Paper II . Here *deviations* are sought and linked to a specific repair that has already happened.

The LVD data data originates from the same sensors and networks as the approach in paper II, but has been packaged and transmitted differently. The data compression method is not optimised for deviation detection but rather for visualisation or human interpretation. This, in combination with the data being sampled in an unknown and changing interval, reduces the deviation detection capabilities. New parameter updates are not sent as new, potentially interesting, deviations are available on-board. Nor is it guaranteed that there exists any deviation in the recorded data just prior to a repair which introduces noise to the prediction labels as well.

The strength in the top-down approach lies in the initial amount of data and the fact that the data originate from customer owned vehicles in daily operation. Moreover, the performance is easily obtained prior to full scale implementation, which is a valuable feature in any kind of product development, where new features are implemented in the order of their priority.

The clear limitations of this method is the deviation detection. It is limited by how the data is represented in the database and the low and unknown update sampling frequency. The aggregation introduces smoothing effects that removes valuable information and the low, and independent, update frequency results in little data on a per vehicle basis. This makes it hard to make individual deviation detection and identify anything but large and stable trends linked to usage.

Further, the quality of the maintenance records is crucial. Misslabelled repairs and repairs that have not been reported causes noisy class labels and lower accuracy. However, the most important limitation of them all, and this limitation is valid to both approaches, is if enough parameters which are sensitive to failures are available. One way of handling this is to include more data, both in on- and off-board sources. Feature selection and distributed methods gets more and more important as the dimensionality of the data increases.

5.1 Future work

The future work will focus on improving the off-board as industrialisation of the on-board and fleet based COSMO approach is farther away in time due to the requirement of advanced on-board logging equipment. An implementation of the proposed off-board will require higher accuracy and especially higher specificity, thus correctly classify normal vehicles as normal. There are several ways to go about this and the following directions will be the future work.

Firstly, ways of combining the on- and off-board approaches will be investigated. The aim is to find a set of specially crafted LVD parameters which is small and yet sensitive to as many faults as possible. The COSMO approach can be introduced

in closely monitored test fleet to learn upcoming failures. A search over different model representations, which fit the limitations of LVD, will be used to optimise the detection and fault isolation performance.

Further, a deeper analysis of repair information in the VSR will be conducted. The aim is to increase the precision of the repair recommendations taking previous repairs into account. The idea of combining an IF-THEN rule approach, much like Buddhakulsomsiri and Zakarian [2009], with the off-board repair recommendations is intriguing and will be the second topic of my future work.

Lastly, little effort has been made in optimising the off-board classification model and this needs to be addressed. Further a comparison between different RUL-methods and classification algorithm will be conducted to conclude which way to go.

References

P. Angelov, A Kordon, and Xiaowei Zhou. Evolving fuzzy inferential sensors for process industry. In *Genetic and Evolving Systems, 2008. GEFS 2008. 3rd International Workshop on*, pages 41–46, March 2008. doi: 10.1109/GEFS.2008.4484565.

A Atamer. Comparison of fmea and field-experience for a turbofan engine with application to case based reasoning. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 5, pages 3354–3360 Vol.5, March 2004. doi: 10.1109/AERO.2004.1368142.

Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3):483–519, 2013.

Jirachai Buddhakulsomsiri and Armen Zakarian. Sequential pattern mining algorithm for automotive warranty data. *Computers & Industrial Engineering*, 57(1):137–147, 2009. ISSN 0360-8352. doi: 10.1016/j.cie.2008.11.006.

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

S. Byttner, T. Rögnvaldsson, and M. Svensson. Self-organized modeling for vehicle fleet based fault detection. Technical paper 2008-01-1297, Society of Automotive Engineers (SAE), 2008.

Stefan Byttner, Thorsteinn Rögnvaldsson, Magnus Svensson, George Bitar, and W. Chominsky. Networked vehicles for automated fault detection. In *Proceedings of IEEE International Symposium on Circuits and Systems*, 2009.

Stefan Byttner, Thorsteinn Rögnvaldsson, and Magnus Svensson. Consensus self-organized models for fault detection (COSMO). *Engineering Applications of Artificial Intelligence*, 24:833–839, 2011.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

Daimler. Mercedes fleetboard vehicle management. https://www.fleetboard.info/fileadmin/content/international/Brochures/VM.pdf, 2014. Accessed: 2014-08-23.

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

Dimitar P. Filev, Ratna Babu Chinnam, Finn Tseng, and Pundarikaksha Baruah. An industrial strength novelty detection framework for autonomous equipment monitoring and diagnostics. *IEEE Transactions on Industrial Informatics*, 6:767–779, 2010.

Erik Frisk, Mattias Krysander, and Emil Larsson. Data-driven lead-acid battery prognostics using random survival forest. 2014.

Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lotfi A. Zadeh. *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 3540354875.

Michiel Hazewinkel, editor. *Encyclopedia of Mathematics*. Springer, 2001.

Rolf Isermann. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer-Verlag, Heidelberg, 2006.

Hillol Kargupta, Michael Gilligan, Vasundhara Puttagunta, Kakali Sarkar, Martin Klein, Nick Lenzi, and Derek Johnson. *MineFleet®: The Vehicle Data Stream Mining System for Ubiquitous Environments*, volume 6202 of *Lecture Notes in Computer Science*, pages 235–254. Springer, 2010.

Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1âĂŞ2):273 – 324, 1997. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(97)00043-X. URL http://www.sciencedirect.com/science/article/pii/S000437029700043X. Relevance.

Igor Kononenko, Edvard Šimec, and Marko Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 1997.

Qiang Lou and Zoran Obradovic, editors. *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*, 2012. AAAI Press.

MAN. MAN TeleMatics efficient operation. http://www.truck.man.eu/global/en/services-and-parts/efficient-operation/man-telematics/overview/MAN-TeleMatics.html, 2014. Accessed: 2014-08-23.

A. Mosallam, K. Medjaher, and N. Zerhouni. Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, pages 1–12, 2014. ISSN 0956-5515. doi: 10.1007/s10845-014-0933-4. URL http://dx.doi.org/10.1007/s10845-014-0933-4.

OnStar. On Star on-star services. https://www.onstar.com, 2014. Accessed: 2014-08-23.

J. Quevedo, H. Chen, M. í. Cugueró, P. Tino, V. Puig, D. García, R. Sarrate, and X. Yao. Combining learning in model space fault diagnosis with data validation/reconstruction: Application to the barcelona water network. *Eng. Appl. Artif. Intell.*, 30:18–29, April 2014. ISSN 0952-1976. doi: 10.1016/j.engappai.2014.01.008. URL http://dx.doi.org/10.1016/j.engappai.2014.01.008.

Michael Reimer. Service relationship management – driving uptime in commercial vehicle maintenance and repair. White paper, DECISIV, 2013a.

Michael Reimer. Days out of service: The silent profit-killer – why fleet financial and executive management should care more about service & repair. White paper, DECISIV, 2013b.

Cirrus TMS. Manage your transportation needs from the cloud. Brouchure, Cirrus, 2013.

Gancho Vachkov. Intelligent data analysis for performance evaluation and fault diagnosis in complex systems. In *Proceedings of the IEEE International conference on fuzzy systems, July 2006*, pages 6322–6329. IEEE Press, 2006.

Volkswagen. Volkswagen on the road to big data with predictive marketing in aftermarket. http://www.csc.com/auto/insights/101101-volkswagen_on_the_road_to_big_data_with_predictive_marketing_in_aftermarket, 2014. Accessed: 2014-08-23.

AB Volvo. Remote Diagnostics remote diagnostics : Volvo trucks. http://www.volvotrucks.com/trucks/na/en-us/business_tools/uptime/remote_diagnostics/Pages/Remote_Diagnostics.aspx, 2014. Accessed: 2014-08-23.

Von Harald Weiss. Ingenieur.de predictive maintenance: Vorhersagemodelle krempeln die wartung um. http://www.ingenieur.de/Themen/Forschung/Predictive-Maintenance-Vorhersagemodelle-krempeln-Wartung-um, 2014. Accessed: 2014-08-23.

Yilu Zhang, Gary W. Gantt Jr., Mark J. Rychlinski, Ryan M. Edwards, John J. Correia, and Calvin E. Wolf. Connected vehicle diagnostics and prognostics, concept, and initial practice. *IEEE Transactions on Reliability*, 58:286–294, 2009.

Appendix A
Paper I - Towards Relation Discovery for Diagnostics

Towards Relation Discovery for Diagnostics

Rune Prytz
Volvo Technology
Götaverksgatan 10
405 08 Gï£¡teborg
Rune.Prytz@volvo.com

Sławomir Nowaczyk
Halmstad University
Box 823
301 18 Halmstad
Slawomir.Nowaczyk@hh.se

Stefan Byttner
Halmstad University
Box 823
301 18 Halmstad
Stefan.Byttner@hh.se

ABSTRACT

It is difficult to implement predictive maintenance in the automotive industry as it looks today, since the sensor capabilities and engineering effort available for diagnostic purposes is limited. It is, in practice, impossible to develop diagnostic algorithms capable of detecting many different kinds of faults that would be applicable to a wide range of vehicle configurations and usage patterns.

However, it is now becoming feasible to obtain and analyse on-board data on real vehicles while they are being used. This makes automatic data-mining methods an attractive alternative, since they are capable of adapting themselves to specific vehicle configurations and usage. In order to be useful, however, such methods need to be able to automatically detect interesting relations between large number of available signal.

This paper presents the first steps towards an unsupervised method for discovering useful relations between measured signals in a Volvo truck, both during normal operations and when a fault has occurred. The interesting relationships are found in a two step procedure. In the first step all valid, defined by MSE threshold, models are found. In the second step the model parameters are studies over time to determine which are significant. We use two different approaches towards estimating model parameters, the LASSO method and the recursive least squares filter. The usefulness of obtained relations is then evaluated using supervised learning and classification. The method presented is unsupervised with the exception of some thresholds set by the user and trivial ambient versus system signal clustering.

Categories and Subject Descriptors

I.5.4 [**Pattern recognition**]: Applications — Signal Processing

General Terms

Algorithms and Reliability

Keywords

Fault detection, Vehicle diagnostics, Machine learning

1. INTRODUCTION

Mechatronic systems of today are typically associated with a high software and system complexity, making it a challenging task to both develop and, especially, maintain those systems. For commercial ground vehicle operators (such as bus and truck fleet owners), the maintenance strategy is typically reactive, meaning that a fault is fixed only after it has occurred. In the vehicle industry it is difficult to move towards predictive maintenance (i.e. telling that there is a need for maintenance before something breaks down) because of limited budget for on-board sensors and the amount of engineering time it takes to develop algorithms that can handle several different kinds of faults, but also work in a wide range of vehicle configurations and for many different types of operation.

If the current trend of increasing number of components in vehicles continues (alongside with increased requirements on component reliability and efficiency), then the only solution will be to move towards automated data analysis to cope with increasing costs. At the same time, with the introduction of low-cost wireless communication, it is now possible to do data-mining on-board real vehicles while they are being used. This paper presents an approach that allows discovery of relations between various signals that available on the internal vehicle network. It is based on the assumption that while it is difficult to detect faults by looking at signals (such as *road speed*) in isolation, the interrelations of connected signals are more likely to be indicative of abnormal conditions.

One requirement for our approach is to be able to perform relation discovery in a fully unsupervised way. This is important since, while an engineer may be able to predict a large number of "good" relations between various signals, her knowledge will in most cases be *global*, i.e. general enough to hold for all or almost all vehicles. An automated system, on the other hand, can be more tightly coupled to the specifics of a particular *fleet* of vehicles — for example, it is likely that some of the signal relations that hold in Alaska do not hold in Mexico, or that some of the relations that are useful for detecting faults in long-haul trucks would be inadequate for delivery trucks.

This paper is organised as follows. In the following section we briefly highlight some of the related research ideas. After that, in section 3, we describe the data we have been working with. Section 4 presents our approach towards dis-

covering relations, in three steps: data preprocessing, signal selection and model parameter estimation. Finally, we evaluate obtained results in section 5 using supervised learning, and we close with some conclusions in section 6.

2. RELATED RESEARCH

Automated data mining for vehicle applications has previously been the topic of several papers. An early paper by Kargupta et. al. [4] shows a system architecture for distributed data-mining in vehicles, and discusses the challenges in automating vehicle data analysis. In Zhang et al. [10], the benefits of being able to do cross-fleet analysis (comparing properties of different vehicles) is shown to benefit root-cause analysis for pre-production diagnostics. In Byttner et. al. [1], a method called COSMO is proposed for distributed search of "interesting relations" (e.g. strong linear correlations) among on-board signals in a fleet of vehicles. The interesting relations can then be monitored over time to enable e.g. deviation detection in specific components. A method based on a similar concept of monitoring correlations (but for a single vehicle instead of a fleet) is shown in D'Silva [2]. In Vachkov [6], the neural gas algorithm is used to model interesting relations for diagnostic of hydraulic excavators. Contrary to our work, however, both the papers by D'Silva and Vachkov assume that the signals which contain the interesting relations are known *apriori*. In [5] a method for monitoring relations between signals in air craft engines is presented. Relations are compared across a fleet of planes and flights. In contrary to our approach they monitor relationships evaluated by experts.

3. DESCRIPTION OF DATA

Measurements have been done on a Volvo VN780 truck with a D12D diesel engine. We have analysed a total of 14 driving runs, four of which were performed under normal operating conditions and the other ten were performed under a number of faulty conditions. The truck was equipped with a logging system which collected data from the internal vehicle network as well as from a set of extra sensors. In total, 21 signals were recorded with a sampling frequency of 1 Hz. Each driving run was approximately four hours in length, and consisted of a variety of driving conditions.

We have specifically targeted the air-intake system, since it is prone to wear and needs regular maintenance during the lifetime of a vehicle. Four different faults have been injected into the truck. The first two were clogging of air filter (AF) and grill. AF change and grill cleaning are routine maintenance operations that are performed regularly. The third fault was charge air cooler (CAC) leak. Such leaks occur in the joints between CAC and connecting air pipes and are considered faults that are hard to find manually. Finally, exhaust pipe was partially congested, which is a rather uncommon fault.

4. RELATION DISCOVERY

The method we use for discovering relations consists of three steps. We start with data preprocessing and removing influence of ambient conditions. Then, we proceed to select the most interesting signals to model, as well as which signals should be used to model them. Finally, we proceed to estimate model parameters. In this last step we use two different approached, the LASSO (Least Absolute Shrinkage and Selection Operator) method and RLS (Recursive Least Squares) method.

4.1 Preprocessing

First, we have performed normalisation of all signals, and removed obvious outliers. Then, available signals were divided into system and ambient signals (in the obvious way, for example *engine speed* is a system signal while *ambient air temp* and *altitude* are ambient signals).

In order to improve the regularity of signals, we begin by filtering out the effects of ambient conditions on the system, using a (slightly simplified) procedure introduced in [**?**]. Namely, we attempt to model each system signal y_k as a linear combination of all ambient signals:

$$\Theta_k = \arg\min_{\Theta \in \mathbb{R}^a} \left(\sum_{t=1}^{n} \left(y_k(t) - \Theta^\top \varphi_{amb}(t) \right)^2 \right)$$

$$\hat{y}_k(t) = \Theta_k^\top \varphi_{amb}(t)$$

$$y_k^{norm}(t) = y_k(t) - \hat{y}_k(t)$$

where a is number of ambient signals, $y_k(t)$ is the value of a system signal k at time t, Θ_k is a vector of parameter estimates for the model of y_k and φ_k is the regressor for the model of y_k. Intuitively, y_k^{norm} is this part of signal y_k that cannot be explained by external conditions.

Figure 1: Signal normalisation

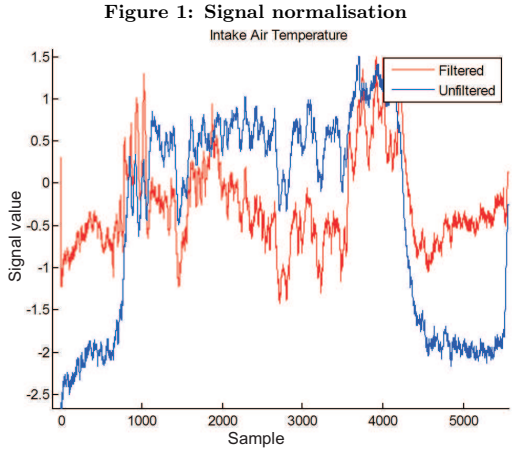


Figure 1 above illustrates how the *intake air temperature* is affected by the ambient conditions (mainly *ambient air temperature*). After ambient filtering, the signal has significantly less variance.

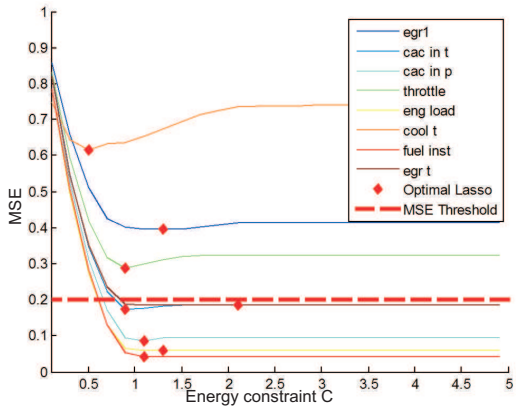
4.2 Signal selection

The next step is to find out which relations between signals are interesting. We perform this step in two stages. In the first stage we attempt to model each system signal using all other systems signals as regressors:

$$\Psi_k = \arg\min_{\Psi \in \mathbb{R}^{s-1}} \left(\sum_{t=1}^{n} \left(y_k(t) - \Psi^\top \varphi_k(t) \right)^2 \right), \ \sum_{i=0}^{s-1} \Psi_{k,i} < C_k$$

where s is number of system signals, Ψ_k is a vector of parameter estimates for the model of y_k and φ_k is the regressor for model of y_k (i.e. the set of all other system signals). The LASSO constraint C_k provides an upper bound on the sum of absolute value of all estimates for signal y_k. We progressively increase its value, performing a cross-validation test after each run. Initially, the mean squared error of the model keeps decreasing, but at some point it begins to increase, as seen in figure 2. By introducing the lasso constraint small parameter representing insignificant relations will go to zero while signficant relations will be prioritized. This gives models with less non zero parameters in comparison to standard least squares.

Figure 2: LASSO parameter estimation



Not all system signals can be modelled in a good way. Some of them have very low correlation to any other system signal and are thus of no interest since any relationship with other system signals are insignificant. These signals are found by studying the MSE of each model. When we increase C_k, the MSE initially decreases, until the model starts to overfit the training data and the MSE goes up. This allows us to find a good C_k value for each signal, by repeating this procedure over a set of time slices and choosing C_k which results in the lowest average MSE. Moreover, all system signals with the MSE above a given threshold (we have used value 0.2 for our experiment) are disregarded.

The second stage consists of finding and removing insignificant model parameters, namely those which are unstable and with low values. To find the relations that are actually important, a sequence of estimates for each regressor within a model is collected over a series of time slices. We perform a *t-test* to find which of those estimates are significant, i.e. which are non-zero. This allows us to remove artificial signal dependencies, leaving only strong relationships.

The end result of the signal selection is a set of system signals that are worthwhile to model, and, for each of them, a unique regression vector containing signals that should be used to model them.

4.3 Model parameter estimation

To estimate parameters for the models we have used two different approaches. The first one is the LASSO method, as explained in previous sections. We have split available data into a number of time slices, and calculated optimal model parameters for each slice. This allows an estimator to easily adapt to changing models (for example, it is easy to imagine that some relations look differently when truck is going downhill than when it is going uphill). On the other hand, when there are two (or more) different models that are similarly plausible, LASSO estimator is likely to oscillate between them in a nearly random way.

A second method is a Recursive Least Square filter [3], which recursively calculates the estimates over a sliding window defined by the forgetting factor. It aims to minimise a weighted linear least squares cost function, thus exhibiting very fast convergence at the cost of poor tracking performance (i.e. when the "true relation" to be estimated changes, it takes a long time for RLS to catch up with this change). The specification of RLS is as follows:

$$P(0) = \delta_{init}^{-1} I$$

$$\Theta(0) = \Theta_{init}$$

$$e(n) = y(n) - \Theta^{\top}(n-1)\varphi(n)$$

$$g(n) = \frac{P(n-1)\varphi(n)}{\lambda + \varphi^{\top}(n)P(n-1)\varphi(n)}$$

$$P(n) = \lambda^{-1}P(n-1) - g(n)\varphi^{\top}(n)\lambda^{-1}P(n-1)$$

$$\Theta(n) = \Theta(n-1) + e(n)g(n)$$

The estimates from all RLS-estimators are collected into an array, W. At each new sample from the system there will be new estimates and thus a new W vector:

$$W(t) = [\Theta_1(t) \ \cdots \ \Theta_s(t)]$$

Using the LASSO method, we obtain one set of model parameters for each time slice. With RLS, we get significantly more data, but there is also a lot more interdependencies within this data. While model parameters from LASSO method are all calculated from different parts of input time series, RLS models are all evolutions of the same predecessor.

Figures 3 and 4 show parameters used for modelling the signal *fuel inst.* The X and Y axis each represents one of the model parameters (i.e. dimensions in the input space of the classifier, as explained in the following section). The dots in the figures each correspond to a single estimate from the RLS estimator and from the LASSO estimator, respectively. As can be seen, our method has autonomously discovered that *fuel inst* (instantaneous fuel consumption) can be approximated using *cac in p* (charge air cooler input pressure) and *in manif t* (input manifold temperature). In other words, there exists a relation

$$\text{fuel inst} = x * \text{cac in p} + y * \text{in manif t}$$

The actual values of x and y parameters, of course, depend on the exact values of the signals in question, but as can be seen in figures 3 and 4, they show an interesting regularities. There are some differences between the two methods, but the general pattern is the same. It appears that some

Figure 3: Model parameters (Lasso method)

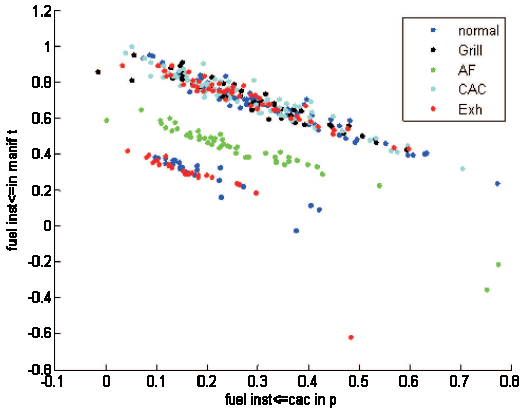


Figure 4: Model parameters (RLS method)

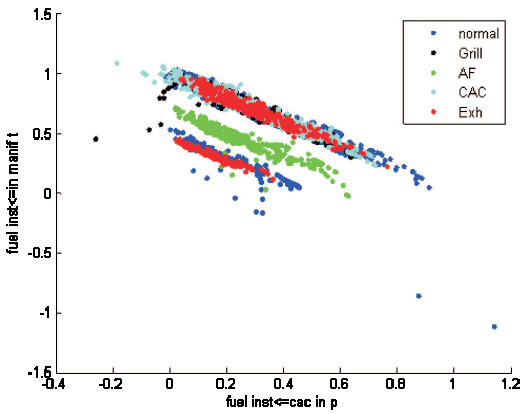


Figure 5: Model parameters (Lasso method)

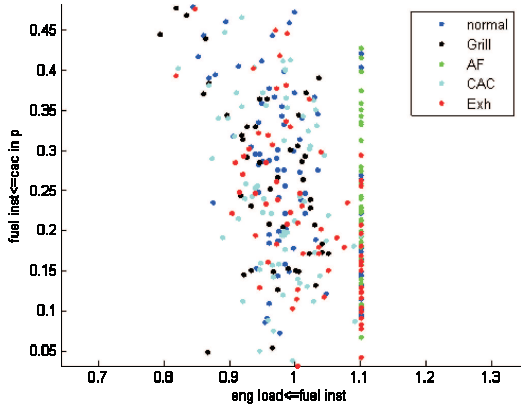
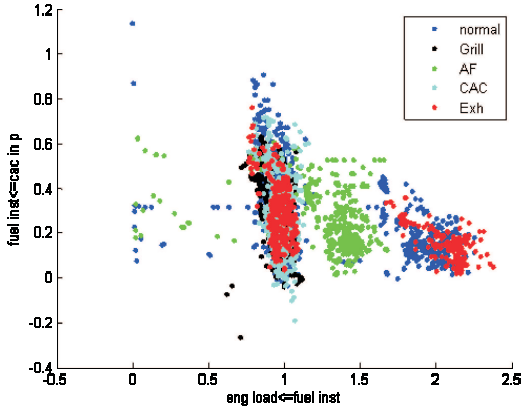


Figure 6: Model parameters (RLS method)



of the faults (in particular, *clogged air filter*) can be quite easily differentiated from normal operation, but some faults (especially CAC leakages) are impossible to detect. This is mainly due to two reasons. Firstly, the injected CAC leaks was very small in comparison to what todays diagnotic algorthms can find and secondly, there is little measurements from system closely related to the fault.

In a similar fashion, figures 5 and 6 show model parameters for estimating *engine load*. As can be seen, in this case the results of RLS method are significantly better than those of the LASSO method.

Overall, though, it is rather difficult to evaluate the quality of discovered relations. Some of them make sense from the domain expert point of view in the general sense, but actual parameter values tend to vary greatly between different time slices. Therefore, we have decided to use supervised learning in order to analyse how much useful information is there in those relations.

5. EVALUATION

We have used three different classifiers: linear regression [8], support vector machine (SVM) [9] and random forest [7]. Each classifier has been used for multi-class classification using the model parameter space generated during the system monitoring step, both from LASSO and from RLS estimators.

The array W contains all the estimates for all found models over time. First half of the matrix was used for training data while the latter was used for test data to evaluate the performance of the classifiers. All model parameters are treated independently of which model they belong to and just as a data stream.

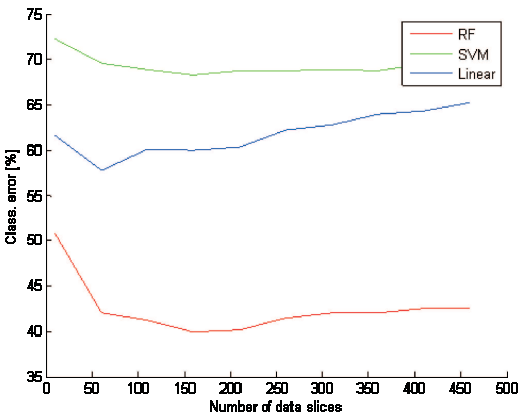
Both the forgetting factor (for RLS) and the data slice size (for LASSO) are parameters for tuning. Larger slices and forgetting factor gives better signal to noise ratio and a more robust solution. On the other hand, they are less sensitive to faults that only appear under certain conditions. In our

case, a partially clogged air filter will only have a visible effect if the engine is running at high power, since this is the only situation when a large air flow is required.

In an approach to find the best data slice size and forgetting factor the classification task were run a number of times with different time slices and forgetting factors. Figures 7 and 8 present the result of that experiment. It is easily seen that choosing too small forgetting factor for RLS is detrimental. On the other hand, the effect of choosing too small data slices is not visible, which indicates that we stopped to early in our exploration.

In general, the random forest classifier outperforms both SVM and linear classifier by a pretty large margin. Besides that, RLS estimator appears to be a little better than the LASSO estimator, but the difference is not huge (it is not clear if this difference is worth the significantly higher computational complexity). An interesting observation is that the number of data slices does not have a big impact on the classification accuracy, but there is a definite sweet point for the forgetting factor at 0.001.

Figure 7: Classification error (Lasso method)

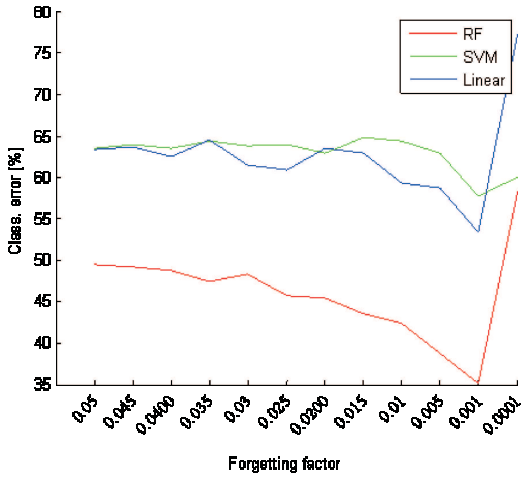


As a final comment, the resulting classification error appears to be rather high, but it is important to take into account that this data set is a pretty difficult one. There is a lot of different external influences that disturb the "normal" operation of a truck, and the low quality of many of the sensors result in high levels of noise in the data. Also, for the predictive maintenance needs, it is not necessary to achieve 100% or close accuracy — it is usually enough to detect faults some proportion of the time, since we are often more interested in following gradual wear rather than abrupt failures.

6. CONCLUSIONS

In this paper we have presented a method for automatic discovery of interesting relations between time series of vehicle signal data. We have evaluated it on the data logged from a Volvo truck, and we have shown that resulting models can be used for diagnostics using supervised learning. This is an important step towards a system that would be able to analyse on-board data on real vehicles and detect anomalies in an autonomous way.

Figure 8: Classification error (RLS method)



This is very much work in progress and there are numerous directions to extend those results. An obvious thing is to look into ways of improving classification accuracy: we have used three well-known learning algorithms with defaults settings, but there is room for improvement in both the learning phase itself, as well as in the estimation of model parameters. We have implemented two methods (LASSO and RLS), but there are many other potential solutions. Also, we have identified advantages and flaws of both of those methods, so it would be interesting to look into possibility of developing some kind of hybrid approach.

It is also not quite clear if the supervised classification is the best way of evaluating usefulness of discovered relations. We intend to explore other possibilities.

Finally, all the data we have access to at the moment comes from a single vehicle. The major benefit of unsupervised relation discovery lies in the possibility of generalising knowledge across multiple similar vehicles in a larger fleet. We are currently in the process of gathering data from a number of trucks and buses, so in the near future we should be able to evaluate our approach in that setting.

7. ACKNOWLEDGEMENTS

The work presented in this paper has been partially funded by grants from VINNOVA and from the Knowledge Foundation.

8. REFERENCES

[1] S. Byttner, T. Rögnvaldsson, and M. Svensson. Consensus self-organized models for fault detection (COSMO). *Engineering Applications of Artificial Intelligence*, 24(5):833–839, 2011.

[2] S. D'Silva. Diagnostics based on the statistical correlation of sensors. Technical Report 2008-01-0129, Society of Automotive Engineers (SAE), 2008.

[3] M. H. Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, Inc., 1996.

[4] H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, S. Bushra, J. Dull, K. Sarkar, M. Klein, M. Vasa, and D. Handy. VEDAS: A mobile and distributed data stream mining system for real-time vehicle monitoring. In *International SIAM Data Mining Conference*, 2003.

[5] J. Lacaille and E. Come. Visual mining and statistics for a turbofan engine fleet. In *IEEE Aerospace Conference*, pages 1–8, March 2011.

[6] G. Vachkov. Intelligent data analysis for performance evaluation and fault diagnosis in complex systems. In *IEEE International Conference on Fuzzy Systems*, pages 6322–Ű6329, July 2006.

[7] WWW. http://code.google.com/p/randomforest-matlab/.

[8] WWW. http://www.csie.ntu.edu.tw/∼cjlin/liblinear/.

[9] WWW. http://www.csie.ntu.edu.tw/∼cjlin/libsvm/.

[10] Y. Zhang, G. Gantt, M. Rychlinski, R. Edwards, J. Correia, and C. Wolf. Connected vehicle diagnostics and prognostics, concept, and initial practice. *IEEE Transactions on Reliability*, 58(2):286–Ű294, 2009.

Appendix B
Paper II - Wisdom of Crowds for Self-organized Intelligent Monitoring of Vehicle Fleets

Wisdom of Crowds for Self-organized Intelligent Monitoring of Vehicle Fleets

Thorsteinn Rögnvaldsson, *Member, IEEE*, Stefan Byttner, Rune Prytz, Sławomir Nowaczyk and Magnus Svensson

Abstract—An approach is presented and experimentally demonstrated where consensus among distributed self-organized agents is used for intelligent monitoring of mobile cyberphysical systems (in this case vehicles). The demonstration is done on test data from a 30 month long field test with a city bus fleet under real operating conditions. The self-organized models operate on-board the systems, like embedded agents, communicate their states over a wireless communication link, and their states are compared off-line to find systems that deviate from the consensus. In this way is the group (the fleet) of systems used to detect errors that actually occur. This can be used to build up a knowledge base that can be accumulated over the life-time of the systems.

Index Terms—Industrial automation, intelligent agents, machine learning.

1 Introduction

Current approaches for equipment monitoring all build on using some form of model, see e.g. [1], [2], [3], [4], [5], [6], [7] for reviews. These approaches have been quite successful but they require a significant amount of "manual" expert work, e.g. trying out different model structures, different feature sets, collecting data, formulating suitable residuals to monitor, et cetera. These approaches are not easily scaleable and it is problematic to handle complex mass-produced mechatronic systems where the profit margins are slim. It requires considerable engineering effort to construct models prior to market release and one must select carefully which systems to monitor.

It is necessary to look at new approaches for equipment health monitoring for modern mechatronic systems; approaches that to a large extent are self-organizing and do not require interaction with a human for tuning and design, that do not require thinking of all possible faults beforehand (of which many will never happen), that can do the best possible with the sensors and signals that are available, that can easily handle "one more system", and that can improve over the lifetime of the equipment.

We have, in only a few years, become used to living in a world of mass communication and collaborative production and to using "the wisdom of crowds" [8] to filter, sort, improve and automate decision making. Surely, it should be possible to design self-organizing (self-learning) equipment monitoring systems based on these concepts, and shift the focus from the isolated individual equipment, and the limitations associated with this view, to the fleet of equipments and build systems that discover and learn from the group in an autonomous way.

Modern mechatronic systems are "cyberphysical" systems; they are equipped with many sensors and control systems, they have standardized communication protocols, and they typically have communication equipment that allows interaction with back-office systems. A fleet of such systems are a (possibly mobile) database with data streams where data can be collected and observed in large quantities and at a low cost. We present in this paper some ideas on how to move towards self-learning and self-monitoring systems for the case of vehicle fleets and provide concrete examples of how this has worked in a long term field test.

2 Previous and related work

The idea to use consensus among self-organized models to detect deviations and to define an "interestingness" measure to automatically select useful features was suggested by Byttner et al. in 2007 [9], [10] and demonstrated on simulated data. Some experimental results were later presented for a heavy duty truck on a test route with different injected faults [11], bus on a test route and for computer hard-disk data [12]. The results presented in this paper are the first that in considerable detail describe the usefulness of this approach on "off-the-shelf" vehicles (city buses) that are driven under normal traffic conditions, by different drivers, over a long time. It is also the first time it is shown how service records can be used to match against observed deviations.

Filev et al. [13], [14] have presented a framework for using novelty detection to build an autonomous

- *T. Rögnvaldsson, S. Byttner and S. Nowaczyk are with CAISR, Halmstad University, Box 823, 301 18 Halmstad, Sweden. E-mail: denni@hh.se, stefan@hh.se and slno@hh.se*
- *R. Prytz and M. Svensson are with Volvo Group Trucks Technology, 405 08 Göteborg, Sweden. E-mail: rune.prytz@volvo.com and magnus.svensson@volvo.com*

Manuscript received May 25, 2014.

system for equipment monitoring and diagnostics. Their framework builds on dynamically updated Gaussian mixture model fuzzy clusters and they assume access to an external "expert" that provides relevant features. The clusters capture different operating modes (e.g. startup, normal, or idle), clusters are updated to account for drift and new clusters are created if the equipment operates 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 the monitored system to run the Gaussian mixture model fuzzy clusters but does not require any off-board analysis.

The idea that models of relationships between signals can be used as indicators for fault detection was also suggested 2008 by D'Silva [15], who used correlations between signals to detect deviations. Similar ideas with linear correlations have been used in the Vedas and MineFleet® systems developed by Kargupta et al. [16], [17], [18]. Kargupta et al. focus on monitoring correlations between on-board signals for vehicles and they use a supervised paradigm to detect fault behaviors. Vachkov [19] has presented work where self-organized neural gas models are used to capture the relationships between signals for diagnostic purposes. All these works assume that features are provided by experts.

The idea that groups of systems with similar usage profiles can be used define "normal" behavior during operation was recently used by Lapira et al. in their work on cluster-based fault detection for fleets of similar machines (wind farms and manufacturing robots) [20], [21] . Zhang et al. [22] also used fleets of vehicles for detecting and isolating unexpected faults in the production stage. Zhang et al. and Lapira et al. use expert selected features for the fault detection.

3 THE COSMO APPROACH

The approach we suggest is called COSMO, short for Consensus Self-organizing Models, since it is based on self-organization and measuring consensus (or absense of consensus) among self-organizing models. The idea is that models are used to represent the streams of data on-board the systems and fleets of similar systems provide a normalization of the models.

A "model" is a representation of a stream of data, consisting of one or more signals. This can be, e.g., means, averages, correlations, distributions, or functional relationships between signals (and signals at different time shifts). Since there are endless possible model configurations, and hierarchies of models of increasing complexity, it is necessary to study ways to automatically determine if a particular configuration is interesting for further study. The framework should include methods for automatically selecting models that are useful for detecting deviations and communicating system status to human experts.

The setting where the COSMO approach is applicable is where one has access to a fleet of systems that do similar things, where there are on-board data streams on the systems, but where it is expensive (difficult/unpractical) to collect huge amounts of raw data to an off-board server, and where there is information available about historical maintenance and repair done to the systems.

The approach can be split into three parts: searching for clues (models); determining deviations; and determining causes. The first step is done on-board the systems and the two latter are done off-board.

3.1 Looking for clues

The first thing needed is to equip the monitored vehicles with a self-organizing technology that can be used to find potential clues about the state of subsystems on the buses, clues that can be communicated with other vehicles and a supervisory system. This can be done by software agents embedded on-board the vehicles; agents that search for interesting relationships among the signals that are communicated on the internal field buses (or inside the internal electronic control units). Such relationships have different "arities"; singular when they describe a single signal, binary when they express relations between two signals, and so on and so forth. Relationships can be encoded with histograms, probability density models, principal components, with linear correlations, with auto-encoders, with self-organizing maps or other clustering methods, et cetera. For example, Filev et al. [13], [14] use the Gaussian mixture model fuzzy clustering and principal components. Vachkov [19] uses a neural gas model to model the relationships.

Relationships that look far from random are marked as possibly useful (interesting) clues about the state of the system. If the found relationship is far from a random relationship, e.g. if a histogram is far from being uniform, or a linear correlation is close to one, or if the cluster distortion measure is low, then this is a sign that the found relationship can be useful for describing the state of the vehicle and then possibly also for detecting faulty operation. This is illustrated in Figs. 1 and 2.

It is not necessary for all systems to check all possible models. The search for interesting relationships can be done in parallel on a fleet where a subset of the vehicles check a subset of models. The back-office application then ranks the models from the full fleet and, after having determined the most interesting relations, request that all systems in the fleet compute and return the parameters for the most interesting model configurations.

3.2 Oddities in parameter space

When all vehicles have computed the parameters for the most interesting model configurations, these parameters are sent (wirelessly) to the back-office application (the central server). The back-office application then performs a test to see if the parameters from the different vehicles are in consensus with each other. If one or a few vehicles disagree with the other vehicles, see Fig. 3, then these vehicles, and the corresponding subsystems, are flagged as potentially faulty.

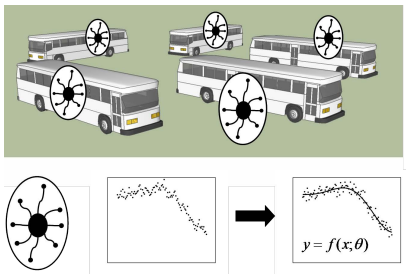


Fig. 1. Looking for clues: On-board the fleet of vehicles are embedded self-organizing agents that listen to the signals on the internal communication network. The agents hunt, much like "sniffers" on local networks, for interesting relationships between signals, i.e. relationships that do not look random (exemplified here with two signals x and y). These are encoded in functions, f, whose parameters, θ, are transmitted over a wireless link to a back-office application.

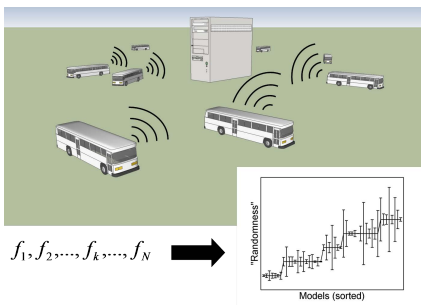


Fig. 2. Ranking the clues: Each vehicle reports back to a server (a back-office application) the models that the embedded agents on-board have found interesting, together with an interestingness measure. The models are ranked by "interestingness" and the back-office application then issues a command to every vehicle to return the parameters for the most interesting models.

There are many ways this test could be performed. It is an outlier/anomaly detection problem and there is a huge body of literature on this subject. Some recent reviews on this topic are: Chandola, Banerjee and Kumar [23]; Gogoi et al. [24]; Gupta et al. [25]; Patcha and Park [26]; Pimentel et al. [27]; Sodemann, Ross and Borghetti [28]; Xie et al. [29]; Zhang [30]; and Zimek, Schubert and Kriegel [31]. Furthermore, Laxhammar[32] recently introduced the conformal anomaly predictor based on the work by Vovk and Gammerman [33], which is not covered in previous reviews.

The test requires a suitable metric for the distance between models, and a method for estimating distributions of models. It is desireable with a test that can produce a measure of the probability for the null hypothesis, that all models have been drawn from the same distribution,

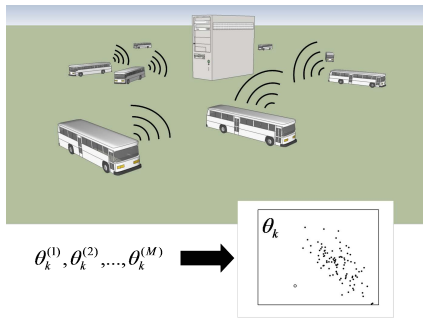


Fig. 3. Finding the oddity: Each vehicle reports the parameters for the most interesting models back to the server. The figure illustrates the case for the k^{th} model and M systems. The parameters are then compared and outlying vehicles and subsystems are detected (open circle) and flagged for a further check.

since this allows proper statistical handling when several samples are drawn from one and the same system. That is, the test should produce a p-value.

3.3 Fault isolation and diagnosis

When a deviation is observed in the parameter space then a potential fault is flagged. An attempt at diagnosing the reason for the deviation can be to compare with previously observed deviations and associated repairs, i.e. a supervised case-based reasoning approach. This requires that a somewhat large corpus of labelled fault observations exists. This labelling should, if possible, be done automatically with the use of maintenance databases, which exists for most modern systems.

Another approach could be to use a simulation model off-board to reproduce the deviation. This allows running advanced simulation models off-line to compare with on-line performance and thus isolate the fault while the vehicle is still driving, which decreases the need for high-performance hardware on-board the vehicle. The simulator is thus used to do a directed search for the fault. This is illustrated in Fig. 4.

There are unfortunately problems with both these approaches currently. There are quality issues with maintenance databases, which are discussed later in this paper, and simulators are designed (parameter tuned) to be good at replicating the operation of normally functioning systems but not for systems that are faulty. Improved quality control of maintenance databases and improved design of system simulators are required to change this. It seems quite easy, with currently available technology, to improve the quality of maintenance databases so that it is possible to design a "life-long learning" fault monitoring system for vehicles. It was, however, necessary for the work in this paper to resort essentially to manual analysis of the maintenance databases to demonstrate how this would work.

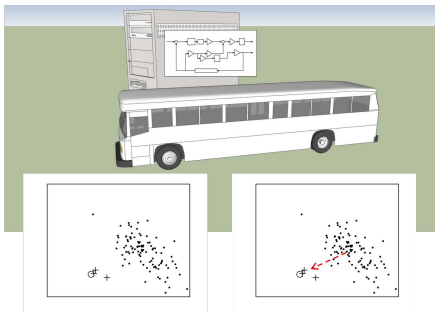


Fig. 4. Isolating the fault: If a deviation is detected then it can be compared to reference parameter values stored in a database (lower left) or a simulation model can be used by the back-office application to replicate the deviation (lower right). Reference parameter values can come from, e.g., maintenance histories, test vehicles or simulation studies.

4 Data models

We here provide implementation details for two examples of the COSMO approach that we have successfully applied in our field test. One where unary relationships are used in the form of histograms, and another were binary relationships are used in the form of linear functions.

4.1 Histograms

A one-dimensional histogram describes one signal. Histograms are more expressive than other single signal representations like the mean and average or statistical moments. 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).

Essentially all on-board signals have special values that denote "out of range" or "error". These can be handled in different ways; they can be removed as "bad data", or they can be kept track of. For the histograms we chose the first option, to remove them, 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 and how much the histograms vary between two consecutive times. The entropy of histogram $\mathbf{P}$ is defined as

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

where P_i is the normalized 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. To enable comparison of two histograms with different number of bins (two different signals) a normalized 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 ot many empty, unused, bins.

The normalized entropy difference is bounded so that $0 \leq \mathrm{NE} \leq 1$. A high value of NE indicates that most of the data are concentrated in few bins, indicating an almost constant signal. An NE value of one corresponds to a constant signal (i.e. only one bin occupied all the time). Signals with NE = 1 are considered "uninteresting" and the others are considered to have an "interestingness" that decreases with NE. The variation in the histograms is also considered in the "interestingness'". Histograms that have a large variation, i.e. shift a lot from one time to another, will be difficult to use for detecting changes and thus less interesting.

Obviously, that a histogram is "interesting" does not mean that it is guaranteed to 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 fault detection and diagnostics.

Many measures have been suggested for quantifying the difference between two histograms (discrete probability densities) and it is quite clear that there is no "best" method for doing this. Measuring histogram distances is still an active research field. Pele [34] recently introduced some new distance measures and presented a good overview of desired characteristics of histogram distance measures, emphasizing the need for cross-bin distances in image retrieval tasks. A review by Cha [35] covers many measures, although not the earth mover's distance [36], and discusses the relationships between them.

We used the Hellinger distance in this work since it has several attractive features. It is an f-divergence measure [37], it is a proper metric, it is quick to compute, and has no problem with empty bins. It is not a cross-bin distance but this should not present a problem since the histogram domains are well aligned. The Hellinger distance between two probability histograms $\mathbf{P}$ and $\mathbf{Q}$ with N bins is defined as

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

where P_i and Q_i are the bin values for $\mathbf{P}$ and $\mathbf{Q}$, respectively. The two histograms are normalized, i.e.

$$\sum_{i=1}^{N} P_i = \sum_{i=1}^{N} Q_i = 1, \tag{4}$$

so that the Hellinger distance is bounded: $0 \leq H(\mathbf{P}, \mathbf{Q}) \leq 1$.

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 over reasonable times each day of the week and we have M vehicles, then this gives a set of size $L = 7 \times M$ histograms for the week. There must be a minimum number of samples required for a valid histogram, which can correspond to (e.g.) a couple of hours of operation. The pairwise Hellinger distances between the L histograms are computed, yielding a symmetric ($H_{ij} = H_{ji}$) distance matrix

$$\mathbf{H} = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1L} \\ H_{21} & H_{22} & \cdots & H_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ H_{L1} & H_{L2} & \cdots & H_{LL} \end{pmatrix}. \quad (5)$$

The statistics for finding outliers in this set of histograms are then computed in a leave-one-out fashion. All histograms relating to one bus, the "test bus", are removed from the set and the distance matrix for the remaining histograms is computed. The most central histogram is found by finding the row with the lowest row sum; this is the histogram with the smallest summed distance to the other histograms. The distances from all other histograms in the matrix to the central histogram are then used as the empirical distribution. This distribution is typically built from about one hundred distance values when using daily histograms collected over a week. The empirical tail frequency is then computed for each histogram for the test bus, i.e. the number of distances in the empirical distribution that are larger than the test bus histogram's distance to the central histogram. We denote this tail frequency by z (it is an estimate of the p-value for the null hypothesis).

This simple method is motivated by the observation that with a Euclidean distance is the pattern with the lowest row sum in the distance matrix also the pattern that is closest to the average pattern. It is very close to a conformal anomaly detection [32] method with the distance to the most central pattern as conformity measure.

The null hypothesis is that all distances H_{ij} are drawn from the same distribution. Under this hypothesis should repeated samplings of z values be uniformly distributed between zero and one [38], which is the idea behind the *transformation method* for generating random numbers from any continuous distribution. We use the mean of the "test bus" z values as statistic and the one-sided p-value for this mean is calculated by assuming that the z values were drawn from a uniform distribution $U(0, 1)$.

Bland [39] recently reported that the z values do not follow a uniform distribution under the null hypothesis if tests are not independent and our tests are not completely independent (comparison data in two different tests may overlap). This can mean that our test would end up being a bit off. However, we stick with this assumption and let the proof of the puddding be in the eating.

If the mean of the z values, over some suitable time horizon, for one bus has a p-value below a pre-specified threshold then that signal and bus are marked as outliers.

This approach for computing z values assumes that the distribution of histograms is spherically symmetric in the particular metric that is used, and unimodal. We believe that assuming a single mode in the "normal" distribution is reasonable given the nature of the problem, but the symmetry is probably not. However, the method is quick and quite sufficient in this demonstration.

4.2 Linear functions

For the case of using linear functions, the first step employs an exhaustive search strategy, generating all pairwise linear signal combinations between two signals x_i and x_j, where $i \neq j$, in the form of

$$\hat{x}_i = a_{ij}x_j + b_{ij} = f(\theta_k, x_j). \quad (6)$$

The total number of signals with nonzero entropy (i.e. not constant) is K and the index $k = 1, 2, ..., K(K-1)$ is used to label each model (each pair of i and j, where order is important). The hat denotes that the x_i on the left hand side is a model of x_i and not the measured value. The model parameters are $\theta = (a, b)$ with the notation used in Figures 1–3. Note that each vehicle and each time window can have one model for each signal pair and a full notation for θ would be $\theta_{k,v}(t)$, where the index v runs over the vehicles, but indices are supressed when possible for notational simplicity.

In the second step all the models are subject to an interestingness metric that aims at estimating each models' potential usefulness for monitoring. The metric has two components that are determined for each model: the α_k and the β_k value.

The α_k value measures the accuracy of each model (i.e. the strength of each pairwise signal relation). It is based on computing the Normalized Mean Square Error for each model k;

$$\text{NMSE}_{v,k} = \frac{1}{N\sigma^2}\sum_{n=1}^{N}[x(n) - \hat{x}(n)]^2 \quad (7)$$

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

$$\alpha_k = \frac{1}{V}\sum_{v=1}^{V}\text{NMSE}_{v,k} \quad (8)$$

where V is the total number of vehicles in the fleet. This can also be over a subset of the vehicles if parallel searches for relations are performed.

The second component, the β_k value, corresponds to how much the model parameters vary across the fleet. For each vehicle and model is the maximum Euclidean

distance to the other vehicles' model parameters computed:

$$d_{v,k} = \max_{v'} \left(\|\theta_{k,v} - \theta_{k,v'}\| \right) \quad (9)$$

where $v' \neq v$. The β_k value is then defined as

$$\beta_k = \sqrt{\frac{1}{V} \sum_{v=1}^{V} (d_{v,k} - \bar{d})^2} \quad (10)$$

where $\bar{d}$ is the average d. Again, this can be done over a subset of vehicles if the interestingness search is done in parallel.

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_k 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_k value. The β_k value quantifies the variation in model parameters among the vehicles in the fleet. A large β_k 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 characterized by a small α_k value and a large β_k value.

It is possible that a useful model shows a small α_k and a small β_k since a fault has not occured yet, which is why the search should be repeated over time in order to discover those models as interesting.

Once a model has been identified as interesting by a user, it is possible to do an automatic search for a potential explanation in the vehicle service records. This was performed in two steps; first by automatically determining the points in time where there are deviations for a vehicles model parameter θ_k relative the fleet. The deviations were determined by applying a leave-one-out test to the set of model parameters from the fleet; fitting a gaussian mixture model on the set of model parameters while leaving one model out in order to compute its p-value. If the p-value of a model was larger than a certain threshold, then a deviation exist. The second step involves identifying the point in time where the deviations end and then querying the service record database with a window around this timepoint, counting all the part replacements that were performed on the deviating vehicles at workshop visits that occured in the window. The total number of part replacements were then summarized to the user with a bar chart, where the user can see what is the most frequently occuring repair and thus serve as an explanation to the observed deviation.

5 Description of the data

The on board data used in this study were collected between August 2011 and December 2013 on a bus fleet with 19 buses in traffic around a city on the west coast of Sweden. The data were collected during normal operation of the buses. The buses in the fleet were year models 2007, 2008 and 2009: four were from 2009, one from 2008, and the remaining 14 from 2007. Each bus was driven approximately 100,000 km per year.

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 CAN). 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 listens to the data traffic on the network and does not affect the traffic 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 snapshots and 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 Volvo Service Record (VSR) database. This data base 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, which means that there are significant quality issues with it. Furthermore, the VSR database is primarily designed for keeping track of invoicing, which means that the date and mileage information are much less than perfect, whereas the parts and operations information is quite accurate. The VSR data were partly curated by comparing them to GPS data for the vehicles and information from the bus operator's notebooks where service dates were (sometimes but far from always) noted.

The bus operator has a maintenance solution that may be typical for a medium sized Scandinavian 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 subcontractor repair shop down the road from the bus fleet garage is therefore often used for maintenance and repairs, which saves a lot of transportation time. Unfortunately, this decreases the VSR information quality. Subcontractors' operations are seldomly 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.

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

6 Uptime and downtime for the bus fleet

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

We did not have information of how large the effective downtime was 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 analyzing the VSR entries and the GPS signals for the bus fleet during the 30 months we observed it. The buses spent on average 11-12% of the time in or at workshops or in transportation to and from workshops. The number varied a lot between the vehicles; the lowest was below 8% and the highest was above 18%.

The buses in the fleet have a maintenance schedule with four planned services per year; one large in association with the annual compulsory Swedish motor vehicle test, another of medium size, plus two minor. The large maintenance is scheduled to take four days and the other less. Additional repairs were almost always done in connection to the service operations. The buses had, on average, about eleven visits to workshops per year, of which more than half were unplanned visits, and spent an average of four days in or at the workshop, or in transportation to or from the workshop, each visit. The variation in visit length was also significant: the shortest was about half a day (we did not count visits shorter than half a day) and the longest was 35 days. A number of on-road assistance turn-outs were needed in addition to this. It was verified in oral communication with the bus operator that they had to keep more "spare" vehicles than desired to guarantee undisturbed transportation.

It should be noted that the numbers are very similar to those for US heavy duty trucks [40], [41] 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 waiting time; the bus is in the workshop but no work is being done on it. This means that the workshops optimize their operation, 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 the time lost for not planning. Unplanned repairs often leads to long waiting times while resources are being allocated for the repair in the workshop.

Thus, an ability to detect beginning problems before they become problems and deal with them during a planned maintenance stop has the potential to decrease the downtime very much. The buses in the field study spend on average more than one month per year in workshops, including transportation time there and back again. It is not unrealistic that this time could be shrunk by half with the techniques described in this paper and by decreasing the waiting time, e.g. by having fewer unplanned workshop visits.

7 Application of COSMO to the city bus fleet

7.1 Histograms

The process described in Section 4.1 was used. Figure 5 shows the the average normalized entropy (NE) plotted versus average Hellinger distance between two consecutive histograms on the same vehicle and 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 times 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 6 shows example histograms from different parts in Fig. 5. 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) but occasionally has other values too. 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*, which is the measured temperature for the transmission oil. The latter has a high entropy and also a high variation between consecutive histograms (not shown). The two other histograms represent points that are between the previous two. The signal *Boost Pressure* is located at (0.07, 0.13). The signal *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 more in relation to linear relations below. One interesting signal close to the upper left corner is the *Coolant Gauge %*, located at (0.10, 0.43) in Fig. 5. This signal is the coolant gauge on the dashboard, which shows a filtered version of the coolant liquid temperature. It is equal to 50% most of the time during normal operation.

Figure 7 shows the z statistics, which is supposed to be uniformly distributed between 0 and 1 under normal conditions (the null hypothesis), and the p-value for the arithmetic mean for the z statistic when computed over the previous 30 days for one of the buses, for the signal *Coolant Gauge %*. Sometimes, especially in the beginning

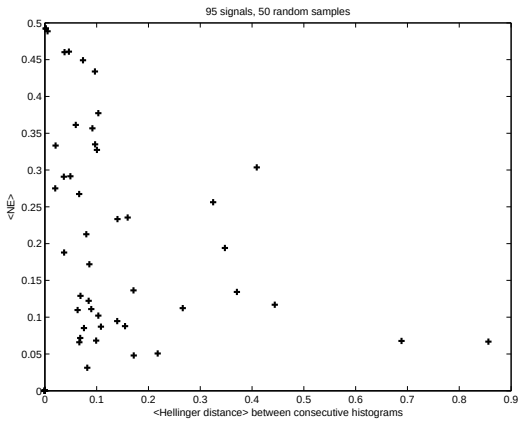


Fig. 5. "Interestingness" for the signal histograms. Angle brackets denote averages. The upper left corner corresponds to histograms that are peaked around a few bins and that are fairly stable with time. The lower left corner corresponds to flat distribution histograms that are stable. The lower right corner corresponds to flat distributions that are non-stable. The upper left corner histograms are most interesting and the lower right corner histograms are least interesting.

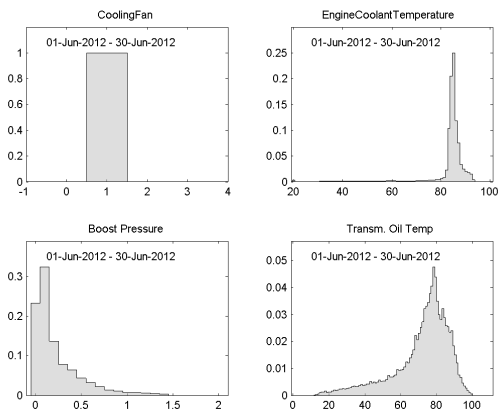


Fig. 6. Examples of histograms in different parts of the "interestingness" graph (Fig. 5). The shown histograms are average histograms for all vehicles in the bus fleet during one summer month 2012. The upper left plot shows the *Cooling Fan* signal, which is a discrete control signal to the cooling 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.

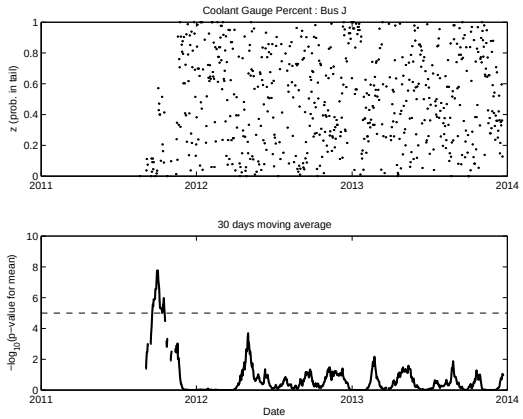


Fig. 7. The z statistic (upper plot) and the p-value for the arithmetic mean over a 30 day moving average (lower plot) for bus J. Results are based on daily histograms, where fleet data (for comparison) are collected over a week. The dashed line in the lower plot marks p-value $= 10^{-5}$.

of the time period, are data 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 (the limit for a significant deviation is set at a p-value below 10^{-5}). The deviation disappeared in late October – early November 2011, during a repair visit that lasted 28 days. One of the repairs that were done during this visit was a broken Electronic Control Unit (ECU); oil had leaked into the ECU and shorted a circuit to the engine cooling fan, 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 not to have been noticed by the customer. It is questionable if it had 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. Figures 8–10 show the p-value for the other buses for the field test period.

Bus A (Fig. 8 top) deviated from the fleet in the second half of February 2012 (from Feb. 10 to 22). During this period tended the *Coolant Gauge %* to be lower for bus A than for the fleet on average. This bus drove considerable shorter daily distances than the other buses in the fleet during this period, less than half of the daily average for the other buses, and significantly shorter daily distances than normally for this bus. The period ended with a minor planned maintenance service after which the bus ran normal daily distances again and the deviation disappeared. There was no repair done during the service that explained the deviation.

Bus B (Fig. 8 second from top) deviated in *Coolant*

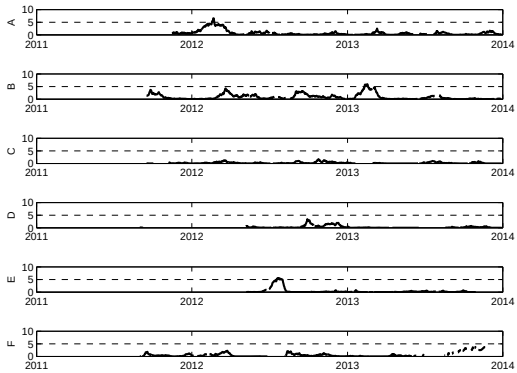


Fig. 8. The p-value for the arithmetic mean over a 30 day moving average for the signal *Coolant Gauge %* for buses A, B, C, D, E and F. Results are based on daily histograms, where fleet data are collected over a week.

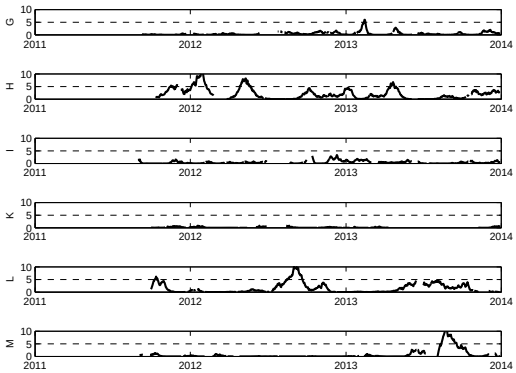


Fig. 9. The p-value for the arithmetic mean over a 30 day moving average for the signal *Coolant Gauge %* for buses G, H, I, K, L and M. Results are based on daily histograms, where fleet data are collected over a week.

Gauge % exactly one year later. The 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 the engine cooling fan running at full speed all the time. The reason was found to be oil that had leaked into the ECU and created a short circuit. As a consequence the cooling fan ran at full speed all the time. This was the same fault as seen earlier on bus J.

Bus E (Fig. 8 second from bottom) deviated from the fleet during the second half of July 2012. The deviation disappeared after the last day of July 2012. July is a month when many people in Sweden take vacation and the load on the public transport system is low. Bus E had many days during this period when it stood still in the bus operator's parking lot. Furthermore, the *Coolant Gauge %* histograms for the buses were very narrow (almost all values located in a few bins around 50%) during this time, which meant that small variations gave large deviations. However, there was no repair done to it at the end of July that explained the (disappearance of the) deviation.

Bus G (Fig. 9 top) deviated briefly during four days in mid February 2012, when the *Coolant Gauge %* signal tended to be higher for this bus than the fleet. The bus was in repair, mostly indoors, during these four days.

Bus H (Fig. 9 second from top) deviated from the fleet several times. This particular bus 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 (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" then bus H was the only bus that the majority of bus drivers agreed on . The colder engine could be a result of how this vehicle was operated.

Bus L (Fig. 9 second from bottom) deviated for about five days in mid October 2011, and for about 40 days from mid August to mid September 2012. The *Coolant Gauge %* signal was higher than the fleet during the first deviation, which ended with a stop in the road, on-road action service and towing of the bus due to a coolant leak. The same was true for the second deviation, which ended with a coolant leak that had to be repaired on-site at the bus depot.

Bus M (Fig. 9 bottom) deviated from mid August to mid September 2013, and is high (although not above the threshold we set) until early October 2013. The bus was sent for repair in early October 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 seen earlier on buses J and B.

Bus P (Fig. 10 third from top) had deviations twice, once in April 2012 and once in December 2012 to January 2013. Both were short deviations, lasting 1–2 weeks. The *Coolant Gauge %* signal for this bus was higher than the fleet during the first period. That period ended with one of the engine cylinders jamming, resulting in a four week long unplanned stop with engine renovation. The *Coolant Gauge %* signal tended to be lower than for the fleet during the second period. This deviation disappeared quickly after the first week of January 2013 without any explanation why; there was no repair done at this time.

Bus Q (Fig. 10 third from bottom) had several deviations, all during 2013. The *Coolant Gauge %* signal on this bus became more and more different from the fleet during the second half of the observation period. There was no repair done that removed the deviation.

Bus R (Fig. 10 second from the top) deviated for a long time, between August 2012 and May 2013. The bus was in maintenance service on August 27, 2012, with a

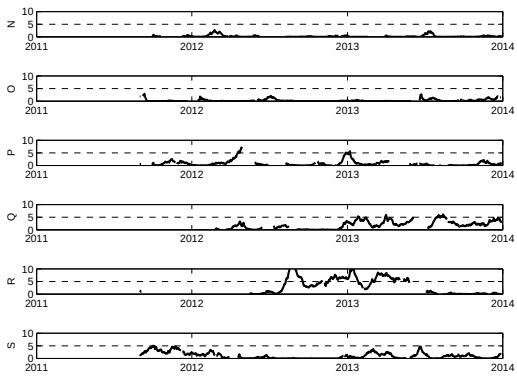


Fig. 10. The p-value for the arithmetic mean over a 30 day moving average for the signal *Coolant Gauge %* for buses N, O, P, Q, R and S. Results are based on daily histograms, where fleet data are collected over a week.

Bus code	Deviation	Explained	Cause
A	Yes	No	Used less (guess)
B	Yes	Yes	Runaway cooling fan (ECU)
C	No	No	
D	No	No	
E	Yes	No	Unknown
F	No	No	
G	Yes	Yes	Indoors in repair
H	Yes	No	Unknown
I	No	No	
J	Yes	Yes	Runaway cooling fan (ECU)
K	No	No	
L	Yes	Yes	Coolant leaks
M	Yes	Yes	Runaway cooling fan (ECU)
N	No	No	
O	No	No	
P	Yes	Yes	Jammed cylinder (1)
Q	Yes	No	Unknown
R	Yes	Yes	Runaway cooling fan (valve)
S	No	No	

TABLE 1
Summary of deviations for the *Coolant Gauge %* signal for the bus fleet during the period August 2011 to December 2013.

comment from the customer that there were warnings about high engine temperature. A fault search was done and an oil leak by the oil filter was repaired. However, the *Coolant Gauge %* signal tended to be lower than for the fleet during the deviation period. The deviation disappeared after a repair of the cooling fan control in connection with the mandatory Swedish annual vehicle inspection. The customer had complained about the cooling fan running at full speed all the time. The fault was a broken magnetic valve. This fault was similar to the faults on buses B, J and M.

The results from observing the *Coolant Gauge %* and matching it to maintenance operations are summarized in Table 1. Deviations were observed in 11 out of 19 vehicles. Of these could at least 7 be explained from the maintenance records and 4 were caused by the same problem, a runaway cooling fan. This is an example where there is no implemented on-board fault detection (diagnostic trouble code) since this is neither a critical nor a frequent fault over the total vehicle lifetime. It is therefore not economically motivated, from the original equipment manufacturer's (OEM's) point of view, to invest in a diagnostic function specifically for this fault. However, it is very uneconomical for the vehicle operator to have a runaway cooling fan. 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 maximum total engine power. The cooling fan runs at maximum speed less than 1% of the time under normal circumstances, as is evident from the lower left plate in Fig. 6. A runaway cooling fan is thus a significant waste of power (and money) for the particular vehicle that is experiencing it.

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 or 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. 5 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. There are other signals, with lower values of the average NE in Fig. 5, that show stronger responses to different faults. One example is the signal *Temperature Upstream Catalyst*. Figure 11 shows the statistics for this signal for bus P, which experienced a jammed cylinder in May 2012 that resulted in a one month long unplanned stop. This signal indicates an abnormality already six months before the engine breakdown.

The threshold for a significant deviation was set at 10^{-5} for the p-value. This value corresponds, if measurements are independent and our null hypothesis holds, to one false alarm per ten years per signal if one monitors the signals daily (for the whole fleet). We monitored 100 signals daily.

7.2 Linear relations

The procedure described in Section 4.2 was followed and the search was applied to data from a one week long window starting on January 12, 2012. The window contained 100,000 samples from all signals. The algorithm built all pairwise linear combinations of 48 signals (the 48 signals that had non-zero entropy, i.e. were non-constant) resulting in a total of 2,256 models. These models were generated on each of the 19 vehicles in the fleet, and the α and β values were computed for each model. The result of this is shown in Figure 12. The most interesting models, from a monitoring point of view, are models that have small α values and

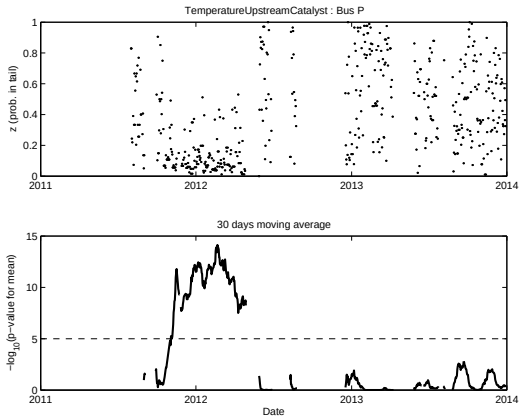


Fig. 11. The z statistic (upper plot) and the p-value for the arithmetic mean over a 30 day moving average (lower plot), for the signal *Temperature Upstream Catalyst* on bus P. Results are based on daily histograms, where fleet data (for comparison) are collected over a week. The dashed line in the lower plot marks p-value $= 10^{-5}$.

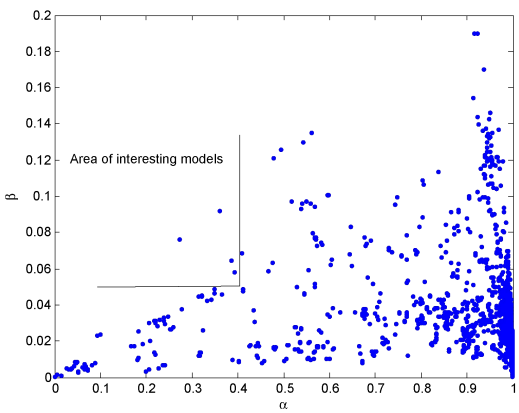


Fig. 12. Interestingness metrics α and β. The α measures the modelling accuracy of the models that were fitted onboard the vehicles. The β value measures how large the variation is between the models (of the same type) in the fleet. The most interesting models are in the upper left corner because they model strong relations that exhibit large variation.

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 $\beta > 0.05$.

Seven models were returned with this setting of maximum α and minimum β. They are listed in Table 2. Several models are (unsurprisingly) mirror images of each other. Two of the models, #273 and #414, are between ambient condition signals and are discarded as indicators of vehicle health. The models #24 and #1,952 were also ignored since relations where at least one signal is discrete (*ACCstatus* and *RetLevelPos* can both be either "on" or off") should not be measured with metrics designed for relations between continuous signals, e.g. NMSE.

TABLE 2
The seven "interesting" relations from Figure 12 (with the structure $x_1 = ax_2 + b$).

Model #	Signal x_1	Signal x_2
24	ACCstatus	RetLevelPos
273	BarometricPressure	CabInteriorTemperature
414	CabInteriorTemperature	BarometricPressure
1676	RelSpdFrontLeft	RelSpdFrontRight
1723	RelSpdFrontRight	RelSpdFrontLeft
1952	RetLevelPos	ACCstatus
2116	Transm. Oil Temp	Coolant Gauge %

We focus here on models #1,676 and #1,723, which model the relationship between the relative speeds of the left and right front wheels. The intercept term b, from Eq. 6, is shown over time in Figures 13 and 14. There are vehicles that deviate significantly in the b model parameter compared to the "normal" fleet vehicles (b should be close to zero). It was found, when these cases were compared with entries in the VSR data base, that a broken wheel speed sensor had been replaced at the time when the deviation disappeared. The front right and left wheel speed sensors operate independently so the b parameter uniquely identifies which sensor is deviating in this case. This is why mirrored relations of two signals should be included in the search.

Figures 13 and 14 show clearly how the linear relationship has captured problems with the wheel speed sensors, which in turn lead to problems with the brakes. The wheel speed sensor problems are visible months before they are fixed. One of the four repairs in Figures 13 and 14 was an on-road service due to a brake system warning, which turned out to be caused by the faulty wheel speed sensor. The remaining three were in the planned maintenance schedule and the wheel speed sensors were replaced because diagnostic trouble codes on-board the vehicles had been triggered.

7.3 Associating deviations with repair events

For the examples described in the previous sections, finding the repair operation or the key replaced component (the wheel speed sensor) to explain an end-of-deviation event was performed manually by looking in the service records. It is possible to automate the search for an explanation to an end-of-deviation event if there are several examples of the same repair (same codes/identifiers) in the VSR data base and if the VSR entry dates are fairly correct.

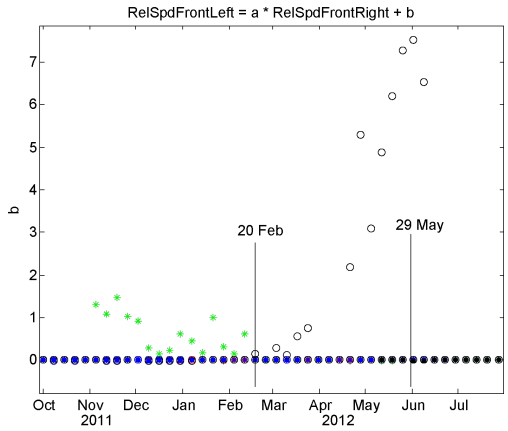


Fig. 13. The b model parameter over time for all vehicles and model #1,676 (see Table 2), between Oct. 2011 and Aug. 2012. Most of the 19 vehicles in the fleet have values close to zero, except two vehicles: bus C ($*$) and bus Q ($\circ$). A wheel speed sensor was replaced on bus C in a workshop visit between Feb. 16 and Feb. 23, 2012, after which the deviation disappeared. For bus Q there was a wheel speed sensor replacement in a maintenance service from May 29 to early June, 2012.

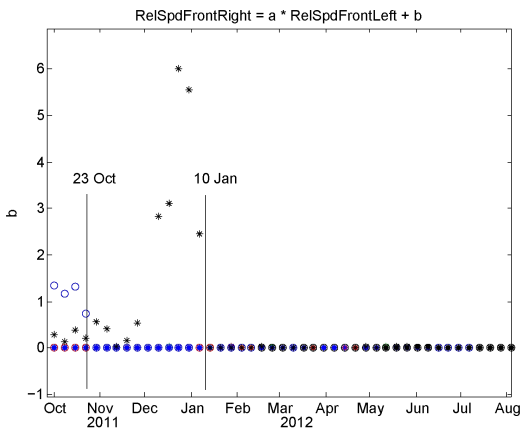


Fig. 14. The b model parameter over time for for all vehicles and model #1,723 (see Table 2), between October 2011 and August 2012. Most fleet vehicles have b-values close to zero. Two vehicles, bus L ($\circ$) and bus S ($*$), show deviations from this. A wheel speed sensor was replaced during a workshop visit Oct. 23–27, 2011, on bus L, after which the deviation disappeared. Similarly, a wheel speed sensor was replaced on bus S during a workshop visit Jan. 10–11, 2012.

The rear modulator, which is related to the rear wheel speed signals, was a component that had several replacements during our field test and there were also several deviations seen in the rear wheel speed sensor signals. By applying the procedure described in Section 4.2, deviations could automatically be determined. This allowed computing when the end-of-deviation event occured for a model, and automatically querying the VSR database for entries in a time window around that date (remembering that the VSR times can be very uncertain). The result of this is shown in Figure 15. For this experiment, a fortnight long time window was used, after the end-of-deviation event, to count all the parts that were replaced at that time. This worked since the modulator replacements were done by the OEM service shops, whose dates were seldomly off by more than two weeks. Figure 15 shows the frequency of the parts that were replaced. Two parts occurred more frequently than others for the repairs on the vehicles that had the deviations: "cable tie" (english translation for "bandklamma") and "modulator". This was later confirmed by experts to be a reasonable explanation to the deviations in the signal relation. This is therefore one possible method in how this can be used for automatically producing an explanation to engineers that work with developing products and services.

The experiment with linear models shows that the detected deviations, for linear models that are selected based on "interestingness", correspond to real faults on the vehicles; faults that require planned service. 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 front wheel speed sensors, which was detected with our method, resulted in an on-road emergency repair.

8 Conclusion and Discussion

We have presented and demonstrated an approach, consensus self-organizing models (COSMO), that builds on using embedded software agents, self-organizing algorithms, fleet normalization and deviation detection for condition monitoring of mechatronic equipment. The potential of the approach was illustrated on data from a long-term field study of seasoned city buses in normal operation. As such, the work is a significant contribution towards self-learning self-monitoring systems in cases when the product cost picture does not allow dedicating human expert efforts to building fault detection and diagnostic models.

The COSMO approach is generally formulated, allowing for embedded agents that can vary both model types and variables. It is also generic in the sense that it does not require much knowledge about the monitored system; it has been demonstrated on hard-disk drives, heavy duty truck data, bus wheel speed sensors and bus engine cooling systems alike. Two examples were described in detail: a singular based on histograms and a

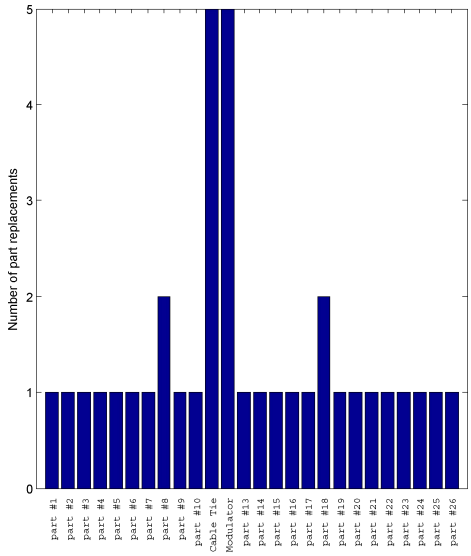


Fig. 15. Number of part replacements found in the VSR data base for deviations in the linear model based on signals *RelSpdRear1Right* and *RelSpdRear2Right*. The histogram was generated by counting the number of part replacements that appears in a 14 day long VSR window after the deviations disappeared. The two most frequent parts in the histogram, #11 and #12, are "cable tie" and "modulator".

binary based on linear regression models. Both produced clear deviations that were related to vehicle maintenance issues, in some cases resulting in on-road stops and in other cases in extended unplanned stops at workshops. It is important that the monitoring was not done with expert defined features, instead the method tries to come up with features, rank them and then observe what happens. This is a new step towards autonomous knowledge discovery.

An overview analysis of the maintenance statistics for the vehicle fleet indicates that it should be possible to decrease the number of workshop days per vehicle significantly, probably by a factor of two, with the implementation of predictive maintenance for catching problems early and better planning of workshop visits.

It is, after this extensive field study on a fleet of vehicles in normal operation, clear that it is feasible to construct self-learning and self-monitoring systems. This does necessarily require curating databases and improving how data are entered and matched to vehicle and dates, in order to have a fully working system. This is, however, not difficult to do. It is also necessary. Future condition monitoring systems must to a large part be self-learning and rely much less, or not at all, on human expertize to define suitable features, model structures and provide labeled training data.

ACKNOWLEDGMENT

The authors thank Vinnova (Swedish Governmental Agency for Innovation Systems), Volvo AB, Halmstad University, and the Swedish Knowledge Foundation for financial support for doing this research.

REFERENCES

[1] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Heidelberg: Springer-Verlag, 2006.

[2] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, pp. 1483–1510, 2006.

[3] J. Hines and R. Seibert, "Technical review of on-line monitoring techniques for performance assessment. volume 1: State-of-the-art," U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, DC 20555-0001, Technical review NUREG/CR-6895, 2006.

[4] J. Hines, D. Garvey, R. Seibert, , and A. Usynin, "Technical review of on-line monitoring techniques for performance assessment. volume 2: Theoretical issues," U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, DC 20555-0001, Technical review NUREG/CR-6895, Vol. 2, 2008.

[5] J. Hines, J. Garvey, D. R. Garvey, and R. Seibert, "Technical review of on-line monitoring techniques for performance assessment. volume 3: Limiting case studies," U.S. Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, Washington, DC 20555-0001, Technical review NUREG/CR-6895, Vol. 3, 2008.

[6] Y. Peng, M. Dong, and M. J. Zuo, "Current status of machine prognostics in condition-based maintenance: a review," *International Journal of Advanced Manufacturing Technology*, vol. 50, pp. 297–313, 2010.

[7] J. Ma and J. Jiang, "Applications of fault detection and diagnosis methods in nuclear power plants: A review," *Progress in Nuclear Energy*, vol. 53, pp. 255–266, 2011.

[8] J. Surowiecki, *The wisdom of crowds*. Doubleday, 2004.

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

[10] J. Hansson, M. Svensson, T. Rögnvaldsson, and S. Byttner, "Remote diagnosis modelling," U.S. Patent 8,543,282 B2, 2013 (filed 2008).

[11] S. Byttner, T. Rögnvaldsson, M. Svensson, G. Bitar, and W. Chominsky, "Networked vehicles for automated fault detection," in *Proceedings of IEEE International Symposium on Circuits and Systems*, 2009.

[12] S. Byttner, T. Rögnvaldsson, and M. Svensson, "Consensus self-organized models for fault detection (COSMO)," *Engineering Applications of Artificial Intelligence*, vol. 24, pp. 833–839, 2011.

[13] D. P. Filev and F. Tseng, "Real time novelty detection modeling for machine health prognostics," in *Proceedings of the Annual meeting of the North American Fuzzy Information Processing Society, June 3–6, Montreal, Canada, 2006 (NAFIPS 2006)*. IEEE Press, 2006, pp. 529–534.

[14] D. P. Filev, R. B. Chinnam, F. Tseng, and P. Baruah, "An industrial strength novelty detection framework for autonomous equipment monitoring and diagnostics," *IEEE Transactions on Industrial Informatics*, vol. 6, pp. 767–779, 2010.

[15] S. H. D'Silva, "Diagnostics based on the statistical correlation of sensors," Society of Automotive Engineers (SAE), Technical paper 2008-01-0129, 2008.

[16] H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, S. Bushra, and J. Dull, "Vedas: a mobile and distributed data stream mining system for real-time vehicle monitoring," in *Proceedings of the Fourth SIAM International Conference on Data Mining (SDM), Lake Buena Vista, Florida, USA, April 22-24*, M. W. Berry, U. Dayal, C. Kamath, and D. B. Skillicorn, Eds. SIAM, 2004.

[17] H. Kargupta, M. Gilligan, V. Puttagunta, K. Sarkar, M. Klein, N. Lenzi, and D. Johnson, *MineFleet®: The Vehicle Data Stream Mining System for Ubiquitous Environments*, ser. Lecture Notes in Computer Science. Springer, 2010, vol. 6202, pp. 235–254.

[18] H. Kargupta, V. Puttagunta, M. Klein, and K. Sarkar, "On-board vehicle data stream monitoring using mine-fleet and fast resource constrained monitoring of correlation matrices," *New Generation Computing*, vol. 25, pp. 5–32, 2007.
[19] G. Vachkov, "Intelligent data analysis for performance evaluation and fault diagnosis in complex systems," in *Proceedings of the IEEE International conference on fuzzy systems, July 2006*. IEEE Press, 2006, pp. 6322–6329.
[20] E. R. Lapira, H. Al-Atat, and J. Lee, "Turbine-to-turbine prognostics technique for wind farms," U.S. Patent WO 2011/143 531 A2, 2011 (filed 2010).
[21] E. R. Lapira, "Fault detection in a network of similar machines using clustering approach," Ph.D. dissertation, University of Cincinnati, 2012.
[22] Y. Zhang, G. W. G. Jr., M. J. Rychlinski, R. M. Edwards, J. J. Correia, and C. E. Wolf, "Connected vehicle diagnostics and prognostics, concept, and initial practice," *IEEE Transactions on Reliability*, vol. 58, pp. 286–294, 2009.
[23] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, pp. 15:1–15:58, 2009.
[24] P. Gogoi, D. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *The Computer Journal*, vol. 54, pp. 570–588, 2011.
[25] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, pp. 1–20, 2013.
[26] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, pp. 3448–3470, 2007.
[27] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.
[28] A. A. Sodemann, M. P. Ross, and B. J. Borghetti, "A review of anomaly detection in automated surveillance," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, pp. 1257–1272, 2012.
[29] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 34, pp. 1302–1325, 2011.
[30] J. Zhang, "Advancements of outlier detection: A survey," *ICST Transaction on Scalable Information Systems*, vol. 13, pp. e2:1–e2:26, 2013.
[31] A. Zimek, E. Schubert, and H. P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining*, vol. 5, pp. 363–378, 2012.
[32] R. Laxhammar, "Conformal anomaly detection," Ph.D. dissertation, University of Skövde, 2014.
[33] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. New York: Springer-Verlag, 2005.
[34] O. Pele, "Distance functions: Theory, algorithms and applications," Ph.D. dissertation, The Hebrew University of Jerusalem, 2011.
[35] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, pp. 300–307, 2007.
[36] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, 2000.
[37] I. Csiszár and P. C. Shields, "Information theory and statistics: A tutorial," *Foundations and Trends™ in Communications and Information Theory*, vol. 1, pp. 417–528, 2004.
[38] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++ (2nd ed.)*. Cambridge University Press, 2002.
[39] M. Bland, "Do baseline p-values follow a uniform distribution in randomised trials?" *PLOS ONE*, vol. 8, no. e76010, pp. 1–5, 2013.
[40] M. Reimer, "Service relationship management – driving uptime in commercial vehicle maintenance and repair," DECISIV, White paper, 2013. [Online]. Available: http://www.decisiv.com/wp-content/uploads/2013/07/SRM-Driving-Uptime-in-Commercial-Fleet-Maintenance-and-Repair-FINAL-7-11-13MP.pdf
[41] ——, "Days out of service: The silent profit-killer – why fleet financial and executive management should care more about service & repair," DECISIV, White paper, 2013. [Online]. Available: http://www.decisiv.com/wp-content/uploads/2013/07/DOS-White-Paper-vfinal.pdf

PLACE PHOTO HERE

Thorsteinn Rögnvaldsson is professor of computer science at Halmstad University, Sweden. He has a PhD in theoretical physics from Lund University 1994 and did his post-doc at the Oregon Graduate Institute 1994–1996. He has previously been Professor of Mechatronics at Örebro University, Sweden. He is the manager for the CAISR research center at Halmstad University. His research interests are in autonomic computing, machine learning and self-organizing algorithms.

PLACE PHOTO HERE

Stefan Byttner is a senior lecturer in information technology at Halmstad University, Sweden. He has a PhD in electrical engineering from Chalmers University of Technology 2005 and did his post-doc at Halmstad University 2006. His research interests are in machine learning and data mining.

PLACE PHOTO HERE

Rune Prytz is a research engineer and PhD candidate in Uptime systems and services within Global Trucks Technology in AB Volvo. His research interests lie within in predictive maintenance and advanced data analysis, especially machine learning techniques for predicting maintenance needs.

PLACE PHOTO HERE

Sławomir Nowaczyk is assistant professor in Information Technology at Halmstad University, Sweden. He has a MSc degree from Poznan University of Technology (Poland, 2002) and a PhD degree from Lund University of Technology (Sweden, 2008), both in Computer Science. He has done his postdoc at AGH University of Science and Technology in Cracow, Poland. His research interests are knowledge representation, data mining and self-organizing systems.

PLACE PHOTO HERE

Magnus Svensson is a senior specialist in Uptime systems and services within Global Trucks Technology in AB Volvo. He holds a Master of Science degree from Royal Institute of Technology (Stockholm, Sweden). He has developed controller software for embedded systems on trucks and has experience of innovation process development. His research interests are mainly data mining and self-organizing systems; especially on distributed systems.

Appendix C

Paper III - Analysis of Truck Compressor Failures Based on Logged Vehicle Data

Analysis of Truck Compressor Failures Based on Logged Vehicle Data

Rune Prytz, Sławomir Nowaczyk, Thorsteinn Rögnvaldsson, *Member, IEEE*, and Stefan Byttner

***Abstract*— In multiple industries, including automotive one, predictive maintenance is becoming more and more important, especially since the focus shifts from product to service-based operation. It requires, among other, being able to provide customers with uptime guarantees. It is natural to investigate the use of data mining techniques, especially since the same shift of focus, as well as technological advancements in the telecommunication solutions, makes long-term data collection more widespread.**

In this paper we describe our experiences in predicting compressor faults using data that is logged on-board Volvo trucks. We discuss unique challenges that are posed by the specifics of the automotive domain. We show that predictive maintenance is possible and can result in significant cost savings, despite the relatively low amount of data available. We also discuss some of the problems we have encountered by employing out-of-the-box machine learning solutions, and identify areas where our task diverges from common assumptions underlying the majority of data mining research.

***Index Terms*— Data Mining, Machine Learning, Fault Prediction, Automotive Diagnostics, Logged Vehicle Data**

I. Introduction

With modern vehicles becoming more and more sophisticated cyber-physical systems, increased software and system complexity poses new development and maintenance challenges. For commercial ground fleet operators, including bus and truck companies, the maintenance strategy is typically reactive, meaning that a fault is fixed only after it has become an issue affecting vehicle's performance.

Currently, there is a desire for truck manufacturers to offer uptime guarantees to their customers, which obviously requires a shift in the paradigm. New ways of thinking about component maintenance, scheduling and replacement need to be introduced. Statistical lifetime predictions are no longer sufficient, and workshop operations need to be planned and their results analysed at the level of individual vehicles.

At the same time, it is slowly becoming feasible to analyse large amounts of data on-board trucks and buses in a timely manner. This enables approaches based on data mining and pattern recognition techniques to augment existing, hand crafted algorithms. Such technologies, however, are not yet in the product stage, and even once they are deployed, a significant time will be required to gather enough data to obtain consistently good results.

In the meantime, it is necessary to explore existing data sources. One example of that is Volvo's "Logged Vehicle Database" (LVD), that collects statistics about usage and internal workings of every vehicle. This data is stored on-board Electronic Control Units during regular operation, and uploaded to a central system during visits in authorised workshops.

The LVD is just one database among many that are of interest for predictive maintenance purposes. Others that are being currently used in related projects include "Vehicle Data Administration" (VDA) and "Vehicle Service Records" (VSR). These databases each contain different, but complementary information: usage statistics and ambient conditions, up-to-date information regarding vehicle equipment, design and configuration specifications, as well as history of all maintenance and repair actions conducted at Volvo Authorised Workshops.

In a typical data mining study, the underlying assumption is that a lot of information is available. For example, it is common in fault prediction research to be able to continuously monitor the device in question. In this regard, the automotive domain is much more restrictive. We are only able to observe any given truck a couple of times per year, at intervals that are unknown *a priori* and difficult to predict even during operation.

In this project we have decided to focus on analysing two components: compressor and turbocharger. Due to lack of space, in this work we only present results related to the compressor, but most of our discussions are valid for both subsystems. The main motivation of predictive maintenance is the possibility to reduce the unplanned stops at the road side. They can be very costly, both for the customer and for the OEM.

If the truck is under warranty or service contract the following expenses could typically be incurred: towing, disruption of garage workflow, actual repair, rent of replacement truck and loss of OEM reputation. During a service contract all maintenance and service costs are covered by a fixed monthly fee. A secondary motivation is to minimise the amount of maintenance that is done on trucks under service contract while still guaranteeing required level of uptime towards the customer.

Additionally, certain components, such as the turbocharger or timing belt, cause significant collateral damage to the vehicle when they fail. Such components are often already either designed to last the full lifetime of the vehicle or scheduled for planned maintenance. In practice, however, this is not enough to prevent all unexpected failures. In these cases predictive maintenance would also be very effective in reducing the excess cost, even though the number of

Rune Prytz is with the Volvo Group Trucks Technology, Advanced Technology & Research Göteborg, Sweden (email: rune.prytz@volvo.com).

Sławomir Nowaczyk, Thorsteinn Rögnvaldsson and Stefan Byttner are with the Center for Applied Intelligent Systems Research, Halmstad University, Sweden (emails follow firstname.lastname@hh.se pattern).

breakdowns is low.

Obviously, predictive maintenance not only saves money, it also introduces additional expenses in terms of unnecessary repairs for the wrongly diagnosed vehicles as well as wasted component life. The latter comes from the fact that the still working component gets exchanged.

The importance of this factor varies greatly depending on particular application. In this study we disregard it completely, since both turbocharger and compressor are exchanged at most once during a vehicles lifetime.

The other cost factor, incorrectly diagnosed failures, can never be completely avoided, but is expected to be surpassed by the savings obtained from finding vehicles before they have an unexpected breakdown. This expense will be the major focus of our discussions in this work.

From classification point view, this can be directly linked to the ratio between True Positive examples and False Positive ones. As mentioned previously, the cost of one on-the-road breakdown is far greater than the cost of one unnecessary component replacement. It is also important to notice that the number of False Negatives is almost irrelevant in this application. They represent "wasted opportunity," i.e. money that could potentially be saved but was not, however they do not incur any direct expenses.

The predictive maintenance solution we are proposing in this paper is designed to be used as an aid in the garage. Whenever a truck is in the workshop for whatever reason, logged data is collected and analysed. The classification algorithm then marks the vehicle as either normal or in need of compressor replacement (within a specified prediction horizon). The workshop will then either exchange the compressor right away, perform additional diagnostics, or schedule another visit in the near future.

This paper is organised as follows. In the next section we describe in more detail the type of data we are working with, as well as present the business constraints that dictate how we state the problem and how are we trying to solve it. We follow by a discussion of related research in Section III. We present our approach in Section IV and results of experiments we have conducted in Section V. We close with conclusions in Section VI.

II. Data and constraints

A typical quality measurement in the automotive industry is the fault frequency of a component. It's percentage of components that fail within a given time: most typically, either a warranty or service contract period. However, that is not a representative measure for our case. Our data consists of a number of data readouts from each truck, spread over long time, but compressor or turbocharger gets replaced at most once.

Most of the vehicles never have a failure of the components we are interested in. Even for those that do, many of the readouts come from the time when the compressor is in good condition, and only in some cases there is a readout from the workshop visit when it is exchanged.

In order to get representative data, we need to select our examples from three scenarios: some of the data should come from trucks on which compressor never failed, some should come from readouts shortly before compressor failure, and some should come from trucks on which the compressor failed far in the future. In order to ensure that, we also consider the number of readouts that is available from each vehicle. Trucks that have too few readouts or do not contain all the data parameters we are interested in are discarded at this stage.

One of the topics of our analysis is to investigate how does the relative ratio of positive and negative examples in train and test datasets influence machine learning results. It is obvious that component failures are an exception rather than a norm. However, there are different ways of measuring the precise ratio between "faulty" and "good" cases. Nevertheless, the fault frequency in the vehicle population does not necessarily translate directly into exactly the same level of imbalance between examples.

We are not able to disclose any real fault frequency data. However, as a guidance, high fault frequency is between 5-10% while a good components may have fault frequency in the range of 0 to 3%. In this paper we will construct the dataset in such way that the baseline fault frequency is 5%. It is important to be aware, however, that there are many factors affecting this and under different circumstances, the data can look very different. Examples include truck configuration and age, usage patterns, geographical location and many more

As a simple example, we can easily imagine a predictive maintenance system being deployed and not applied to all vehicles, but only to those that service technicians consider "high risk". Similarly, while compressor is an important component to monitor, the methodology itself is fully general, and there are other parts that could be targeted. Some of them are designed to be replaced regularly, and thus could have failures that occur on almost all trucks. Therefore, in several places in this paper, we will discuss how different fault frequencies affect classification results.

The vehicles in our dataset are all Volvo trucks, from the same year model, but equipped with three different compressor types. They also vary with respect to geographical location, owner, and type of operation, for instance long-haul, delivery or construction.

We have selected 80 trucks which had compressor failures and at least 10 LVD readouts, with the right number of parameters available. In addition we have chosen 1440 trucks on which, so far at least, no compressor had failed. They all fulfil the same requirements on LVD data. We could easily obtain more "non-faulty" vehicles, but it is the ones with compressor failures that are the limiting factor.

A. Logged Vehicle Data

Logged Vehicle Data is a Volvo internal database which gathers usage and ambient statistics collected from Volvo vehicles. The data is downloaded from the truck when it is serviced at an authorised Volvo workshop, or wirelessly through a telematics gateway. The database is used for

various tasks during product development, after market and even sales support.

A typical task for product development would be to support a simulation or validate an assumption with real usage statistics from the field. For instance, such questions could concern the relationship between average fuel economy and weight, altitude or engine type. During the sales process the database can provide usage statistics for already existing customers, which is helpful in configuring the right truck for a particular purpose.

This database contains data of varying types and has high number of dimensions. Typically a vehicle record contains hundreds of parameters and at most tens of readouts. The number of readouts directly depends on the availability of telematics equipment and on whether the vehicle has been regularly maintained at a Volvo workshop. For example, in our dataset the average number of readouts per vehicle is 4 per year. However, the variance is very high and many trucks have one or less readouts per.

There is also a problem with missing values, typically caused by connectivity issues or software updates. Modern on-board software versions log more parameters, which means that older readouts tend to include less data than newer ones.

Finally, the stored parameters are typically of cumulative nature. This means that the readouts are highly correlated and not *independently identically distributed*, as is usually assumed in machine learning. It could be interested to analyse, instead of the LVD data itself, the changes between subsequent readouts — but it can be complicated because there is a number of different aggregation schemes employed (for example, averages, accumulators and histograms).

B. VSR and VDA

The Volvo Service Records a database that keeps track of all maintenance and repair operations done on a particular vehicle. The database is mainly used by the workshop personnel for invoicing purposes, as well as for diagnostics, allowing to check previously carried out repairs.

A typical repair event contains date, current mileage, and a list of unique maintenance operation codes and exchanged part numbers. In addition to that there may be a text note added by the technician. For the purposes of this work, we are using VSR to find out whether and when a compressor was replaced on a given truck.

The VDA database contains vehicle specification for all vehicles produced by Volvo. It lists the included components such as gearbox model, wheel size, cab version, or engine and compressor type. All options have a unique label which makes it easy to use for classification.

III. Related Work

In a survey of Artificial Intelligence solutions being used within automotive industry, [1] discusses, among other things, both fault prognostics and after-sales service and warranty claims. An representative example of work being done in this area are [2] and [3], where authors present two data mining algorithms that extracts associative and sequential patterns from a large automotive warranty database, capturing relationships among occurrences of warranty claims over time. Employing a simple IF-THEN rules representation, the algorithm allows filtering out insignificant patterns using a number of rule strength parameters. In that work, however, no information about vehicle usage is available, and the discovered knowledge is of a statistical nature concerning relations between common faults, rather than describing concrete individual.

More recently [4] presented a survey of 150 papers related to the use of data mining in manufacturing. While their scope was broader than only diagnostics and fault prediction, including areas such as design, supply chain and customer relations, they have covered a large portion of literature related to the topic of this paper. The general conclusion is that the specifics of automotive domain make fault prediction a more challenging problem than in other domains: almost all research considers a case where continuous monitoring of devices is possible, e.g. [5] or [6].

It is more common to consider emergent solutions, where vehicles are able to communicate using telematic gateways. An early paper [7] shows a system architecture for distributed data-mining in vehicles, and discusses the challenges in automating vehicle data analysis. In [8] cross-fleet analysis, i.e. comparing properties of different vehicles, is shown to benefit root-cause analysis for pre-production diagnostics. In [9] and [10], a method called COSMO is proposed for distributed search of “interesting relations” among on-board signals in a fleet of vehicles, enabling deviation detection in specific components.

A method based on a similar concept of monitoring correlations, but for a single vehicle instead of a fleet, is shown in D’Silva [11]. In Vachkov [12], the neural gas algorithm is used to model interesting relations for diagnostic of hydraulic excavators. Contrary to our work, however, both the papers by D’Silva and Vachkov assume that the signals which contain the interesting relations are known *a priori*. In [13], a method for monitoring relations between signals in aircraft engines is presented. Relations are compared across a fleet of planes and flights. Unlike us, however, they focus on discovering relationships that are later evaluated by domain experts.

Even though not particularly recent, [14] and [15] are still excellent introductions to more general machine learning and artificial intelligence topics. In this paper we are also facing many challenges related to the imbalanced nature of diagnostics data. In order to make our initial investigations more widely accessible we have decided not to use any specialised solutions, but an overview of research on this area can be found, for example, in [16], [17] or [18].

IV. Approach

We have decided to base our initial analysis on using out-of-the-box supervised classification algorithms. From among the available attributes, 4 interesting VDA parameters and 8 LVD interesting parameters were chosen by experts within

Volvo. Those include, for example: compressor model, engine type, vehicle mileage, average compressed air usage per kilometre, etc.

At this stage of our research, we have decided to consider each data readout as a single learning example. Even though they definitely do not satisfy the basic *independent and identically distributed* assumption, this gives us flexibility in both the classifier choice and in deciding how to analyse actual faults.

When constructing the dataset we need to merge data from the three databases. First we find, in the VSR, all truck that had the compressor exchanged. To do that we use the unique maintenance code for compressor replacement. After that we find all the LVD and VDA data for the faulty vehicles, up to and until the aforementioned repair occurred. At this stage we discard some vehicles, either because they do not have sufficient number of readouts or because not all the interesting parameters selected by Volvo experts are available. After that we also select some number of "non-faulty" trucks.

For each LVD readout, we also create a new parameter denoting time to repair. It uses the timestamp of repair entry in VSR and this particular readout's date. In the case of non-faulty trucks we are assuming that they may break just after the latest readout available, so that the *time to repair* parameter can be calculated for all trucks. This parameter is later used for labelling examples as either positive or negative, based on the prediction horizon, but is of course not used for classification. This step is one of the areas where there is definitive room for improvement, since it is definitely not clear, however, when – if at all – the symptoms for the imminent failure become visible in the data.

When selecting examples for classification a prediction horizon and the desired fault rate must first be defined. The *time to repair* parameter is used to determine which readouts are considered as positive: those that fall within the prediction horizon. After that, at most two examples per vehicle are drawn to form the training and test datasets.

For the trucks marked as faulty, we select exactly one positive and one negative example, at random. Finally, we add one negative example from the remaining trucks until the desired fault frequency is archived. By selecting an equal (and small) number of positive and negative examples from each truck we avoid the problem of classifiers learning characteristics of individual vehicles rather than those of failing compressors.

The reason for choosing random readouts as examples is twofold. First of all, it is not entirely clear how to choose which data readout is the best one to use. It is important that there is sufficient distance between corresponding positive and negative example, in order for the data to be changed significantly. The further apart the two examples are, the larger the chance that symptoms of failing compressor are present in the positive example and are missing from the negative one. On the other hand, selecting dates close to the cutoff boundary would allow more precision in estimating *when* the components is likely to break.

The random approach avoids any systematic bias in either direction, but it means that actual training dataset only depends on the prediction horizon to a limited degree. It also means that we have no real control over how similar positive and negative examples actually are. It is an interesting question of how to find the appropriate cutoff point automatically, preferable on an individual basis.

In the final step, we remove 10% of the dataset, to be used as the test data, and use the rest as train data. Since we have few examples available, we use both out-of-bag evaluation on the training dataset, as well as the separate evaluation on the test data. In section V we sometimes present both evaluations, and sometimes only one of them, depending on which one is more appropriate for a particular purpose.

One of the issues with out-of-bag evaluations is that it is computationally intense. To speed up the processing, each classifier is only evaluated on a subset of the train data. The out-of-bag evaluation subset contains all the positive examples, but only a portion of negative examples. The resulting confusion matrix is then up-scaled for the *true negatives* and *false positives*.

As an evaluation of the business case for the predictive maintenance solution, we introduce measure of cost savings:

$$C_{save} = TP \cdot (C_u - C_p) - FP \cdot C_p$$

The method will be profitable if the correctly classified faulty trucks (i.e. *true positives TP*) save more money than the non-faulty trucks wrongly classified as faulty (i.e. *false positive FP* waste. Because an on-road *unplanned* breakdown costs (C_u) is much higher than the *planned* component replacement (C_p), every TP reduces costs.

A. Learning algorithms

In this work we have used the KNN, C5.0 and Random Forest learning algorithms. Each of them is evaluated in R using the Caret package as described in [19]. By default, the Caret package tunes the parameters of each classifier.

V. Experiments

In this section we present the results of early experiments we have performed. Throughout this presentation we have two main goals. First, we argue that those initial results are encouraging and promise a tangible business benefits, thus warranting further work, and hopefully inspiring others to investigate similar approaches in other applications. Second, we demonstrate difficulties we have encountered due to the type of data available and specifics of the domain.

As the first step towards familiarising the reader with our data, we present how the dataset size affects quality of classification. In Figure 1 we have plotted the classification accuracy, both using out-of-bag evaluation and a separate test set, for all three classifiers.

This figure is mainly useful to show the level of variance in classifier behaviour, since — even though it looks impressive — accuracy is not a particularly suitable measure for this

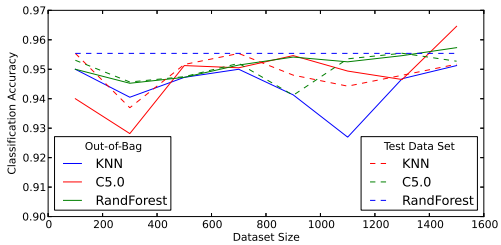


Fig. 1. Impact of dataset size on classification accuracy

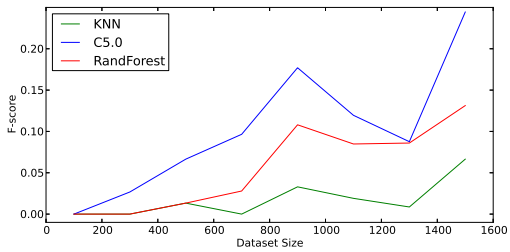


Fig. 2. Impact of dataset size on $F_{0.5}$-score

problem. As explained before, the baseline for our analysis is to assume 5% fault frequency, and this is the ratio between positive and negative examples in both training and test datasets.

Therefore, accuracy of 95% can be achieved in a very simple manner, by doing no generalisation whatsoever and simply answering "No" to every query. As can be seen from the plot, classification algorithms we are using are employing more complicated schemes, but only Random Forests consistently beats that simplest strategy, and only on the test data set — which in itself is not entirely conclusive, due to the limited size of the data we are working with.

Finally, this plot also shows that there is no significant difference in results between out-of-bag and test data evaluations. Therefore, in some of the subsequent plots we will limit ourselves to only presenting one of them, unless particular scenario makes both interesting.

In figure 2 we are presenting the F-score:

$$F = (1+\beta^2)\frac{\text{precision} * \text{recall}}{\beta^2 * \text{precision} + \text{recall}},$$

as this is one of the most popular measures that is actually suitable for highly imbalanced data sets. In our case we have decided to use parameter $\beta = 0.5$, because in this application, precision is significantly more important than recall: every compressor that we do not flag as needing replacement simply maintains *status quo*, while every unnecessary repair costs money.

By analysing this plot it is clearly visible that the dataset we have currently access to is very small, only barely sufficient for the analysis. Even when using all the data as the training set, the F-score of the best classifier barely exceeds 0.2. On the other hand, this plot clearly shows that we have not yet reached saturation levels, and it is reasonable to assume that as more data becomes available, the quality of classification will continue to increase. This also means that most of the results presented subsequently can be expected to improve in the future.

One of the most interesting questions with regard to predictive maintenance is how early in advance can faults be detected. In order to answer that, we have performed an experiment where we were interested in evaluating the influence of prediction horizon on the classification quality.

In this case we have decided to present the results in Figure 3 for three different values of fault frequency (colours correspond to different classifiers, while line styles denote 5%, 20% or 50% class distribution). The imbalanced nature of the data is obviously a problem, but as we have discussed in section II, there is significant flexibility in how the final product will be deployed, and that allows us some freedom. Therefore, it is interesting to see prediction quality in a number of settings. That said, the performance on highly skewed data sets is still the most important one, because other solutions typically involve various kinds of cost-incurring tradeoffs. In order to not clutter the figure, we only include F-score evaluated using out-of-bag method.

In most diagnostic applications the prediction horizon is a very, if not the most, important measure. In out case, however, it is both less critical and more difficult to define precisely. The former comes from the fact that one is only expected to exchange compressor once in a lifetime of a vehicle. Therefore, the precise time of when is it done, as long as it is reasonable, does not directly influence the costs. There are, of course, some benefits of minimising wasted remaining useful life, but they are difficult to measure since they mainly relate to customer satisfaction.

The difficulty in defining the prediction horizon, however, is definitely something we are interested in investigating further. One idea would be to take into account individual usage

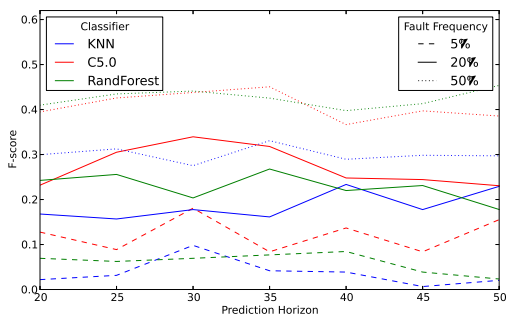


Fig. 3. $F_{0.5}$-score as a function of prediction horizon, for three different levels of fault frequency in vehicle population

patterns of trucks, for example by assuming that vehicles that are rarely in the workshop should have longer advance notice, while those that are maintained more regularly can wait until the failure is more imminent.

At the moment, however, we are treating all data readouts as individual and independent examples, and therefore each of them has to be marked as either positive or negative one. We use a very simple scheme of assuming that all examples closer to the failure than the prediction horizon are positive, and all examples further away are negative. This, however, makes analysing influence of prediction horizon on the classification quality more difficult, especially taking into account the irregular intervals at which we obtain vehicle data.

Moreover, during our first attempts of analysing the data (which we are not presenting here due to space constraints), we have encountered a situation that all machine learning algorithms learned to almost exclusively consider characteristics of particular trucks, instead of indicators of failing compressor. They would provide, for most of the vehicles, predictions that never changed over time. This resulted in classifiers that achieved good accuracy and F-score, but were completely useless from business point of view.

To this end we have decided to use exactly two data readouts from each vehicle on which we have observed compressor replacement: one positive and one negative example. This solves the aforementioned problem, since now there is no benefit to distinguishing individual, but it even further reduces the size of available data. In addition, it is not entirely clear how to choose which data readout to use, if we can only use one of them.

On the one hand, one would want to use readouts as close to the prediction horizon boundary as possible, to be highly precise in predicting wasted life of the components. On the other hand, it is not good to choose positive and negative examples that are too close in time, since it is very likely that the difference in logged data between those two points does not contain any new information about state of the compressor.

To this end, we have decided to choose one example from each side of the prediction horizon boundary at random. It means, however, that varying the prediction horizon only introduces small changes in the actual training and test datasets. It may even happen that for two significantly different values of the horizon, we end up with the same data. This explains the results that can be seen in Figure 3: prediction horizon has very little influence on the F-score.

Accuracy and F-score are important measures from research point of view. The inspiration for our work, however, arises from practical needs of automotive industry, and the major measure from the business perspective is clearly cost reduction. It is very expensive to have components fail during transport missions, because not only does it introduce disruptions in the workshop operations, it also incurs other costs, like towing, collateral damage, and customer dissatisfaction. Therefore, it is significantly cheaper to replace components during scheduled maintenance. The exact degree to which this is the case varies, of course, from component to component, and also depends on which factors are taken into account — reputation, for example, is notoriously difficult to appraise.

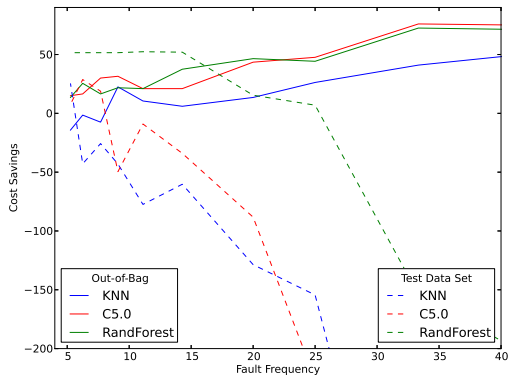


Fig. 4. Maintenance cost savings that can be achieved for varying fault frequency in training dataset (test set always has 5% of positive examples).

Therefore, in order to be on the safe side, we have decided to use a factor of 2.5 to measure cost savings that can be provided by our solution. In other words, it costs on average two and a half as much to repair a truck in which compressor failed on the road, as it would cost to replace this component as a scheduled operation.

Figure 4 shows how the benefits of introducing our predictive maintenance solution depend on the fault rate in the vehicle population. The most interesting is, of course, the left side of the plot, because it shows that even the low quality classification results that we are able to obtain from our 1600 data samples are enough to offer tangible benefits. Both Random Forest and C5.0 classifiers are accurate enough to save expenses.

It is interesting to see how cost savings (at least looking at out-of-bag data) grow as the imbalance in the data decreases. This is consistent with results from Figure 2 and can be easily explained by the higher quality of classification.

On the other hand, the cost when measured on the test set drops very rapidly (except for the Random Forest classifier, the result which we are not able to explain just yet). The reason for this behaviour is that the test data always contains 95%–5% split of negative and positive examples. As the distribution of data in the training set become more and more different from the distribution in test set, the quality of classification drops.

Finally, in Figure 5 we present the relation between True Positives and False Positives, again as a function of fault frequency. We are only using out-of-bag evaluation here. This is the plot that actually contains the most information, since those are the two factors that directly affect the economical viability of our solution. As mentioned earlier, presence of

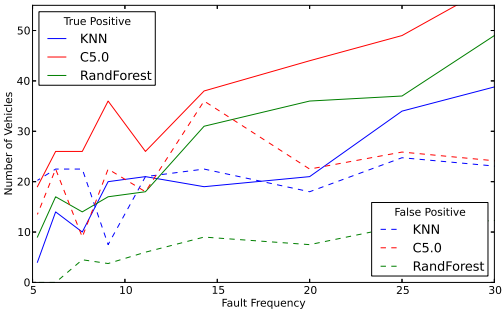


Fig. 5. True Positives and True Negatives

False Negatives does not affect the cost in any direct way. It is interesting to look at the differences between the three classifiers, and the potential tradeoffs that may be important from business perspective.

It is clear that KNN is not well-suited for this particular problem, although it can possibly be explained by the fact that we have not performed any data normalisation, and the large differences in absolute values of various parameters may be difficult for it to handle. Even for more balanced data sets, this classifier is struggling to obtain more True Positives than False Positives.

From the pure cost perspective, Random Forest seems to be better than C5.0, because the difference between True Positives and True Negatives is larger. On the other hand, C5.0 actually detects more faulty compressors, in simply makes more FP mistakes as well. In Figure 4 those two classifiers score very close, but if we would assume another relative costs for planned and unplanned component replacements, the difference between them could be significant. It would be interesting to investigate what is the reason for this difference, and possibly to identify parameters that would allow us to control this tradeoff.

VI. Conclusions and Future Work

The most important conclusion of this work is that using data mining based on Logged Vehicle Data as predictive maintenance solution in automotive industry is a viable approach. We will continue the work in this area, investigating more complex machine learning approaches. Current classification quality and cost avoidance is not great, but it is expected to increase as we get access to more data and as we replace generic algorithms with more specialised ones.

It is known that data availability will dramatically increase as the new Volvo truck reaches the customers. It is equipped with new and enhanced telematics platform, enabling larger and more frequent LVD readouts.

The second contribution of this paper is identifying a number of distinctive features of automotive industry, and discussion regarding to what degree do they fit typical machine learning and data mining research paradigms.

Ideas for future work include extending this analysis to other components, especially the ones where “exchange once in a lifetime” assumption does not hold, as well as evaluating known methods of dealing with imbalanced data sets.

It is also necessary to define the notion of prediction horizon in a better way, preferably allowing learning algorithm to choose the threshold in an individualised manner. Another approach to investigate is to use regression to predict *time to repair*. One possible solution would be to look at the differences between readouts, as this may decrease the correlation between examples and enhance classification performance.

Finally, we would like to use rich data representations, combining all readouts of a single truck together.

References

[1] O. Gusikhin, N. Rychtyckyj, and D. Filev, “Intelligent systems in the automotive industry: applications and trends,” *Knowledge and Information Systems*, vol. 12, pp. 147–168, 2007.

[2] J. Buddhakulsomsiri, Y. Siradeghyan, A. Zakarian, and X. Li, “Association rule-generation algorithm for mining automotive warranty data,” *International Journal of Production Research*, vol. 44, no. 14, pp. 2749–2770, 2006.

[3] J. Buddhakulsomsiri and A. Zakarian, “Sequential pattern mining algorithm for automotive warranty data,” *Computers & Industrial Engineering*, vol. 57, no. 1, pp. 137 – 147, 2009.

[4] A. Choudhary, J. Harding, and M. Tiwari, “Data mining in manufacturing: a review based on the kind of knowledge,” *Journal of Intelligent Manufacturing*, vol. 20, pp. 501–521, 2009.

[5] A. Kusiak and A. Verma, “Analyzing bearing faults in wind turbines: A data-mining approach,” *Renewable Energy*, vol. 48, pp. 110–116, 2012.

[6] A. Alzghoul, M. Löfstrand, and B. Backe, “Data stream forecasting for system fault prediction,” *Computers & Industrial Engineering*, vol. 62, no. 4, pp. 972–978, May 2012.

[7] H. Kargupta *et al.*, “VEDAS: A mobile and distributed data stream mining system for real-time vehicle monitoring,” in *Int. SIAM Data Mining Conference*, 2003.

[8] Y. Zhang, G. Gantt *et al.*, “Connected vehicle diagnostics and prognostics, concept, and initial practice,” *IEEE Transactions on Reliability*, vol. 58, no. 2, 2009.

[9] S. Byttner, T. Rögnvaldsson, and M. Svensson, “Consensus self-organized models for fault detection (COSMO),” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 5, pp. 833–839, 2011.

[10] R. Prytz, S. Nowaczyk, and S. Byttner, “Towards relation discovery for diagnostics,” in *Proceedings of the First International Workshop on Data Mining for Service and Maintenance*. ACM, 2011, pp. 23–27.

[11] S. D’Silva, “Diagnostics based on the statistical correlation of sensors,” Society of Automotive Engineers (SAE), Tech. Rep. 2008-01-0129, 2008.

[12] G. Vachkov, “Intelligent data analysis for performance evaluation and fault diagnosis in complex systems,” in *IEEE International Conference on Fuzzy Systems*, July 2006, pp. 6322–6329.

[13] J. Lacaille and E. Come, “Visual mining and statistics for turbofan engine fleet,” in *IEEE Aerospace Conf.*, 2011.

[14] T. M. Mitchell, *Machine Learning*. McGraw-Hill Higher Education, 1997.

[15] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice Hall Series in AI, 2003.

[16] G. M. Weiss, “Mining with rarity: a unifying framework,” *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 7–19, Jun. 2004. [Online]. Available: http://doi.acm.org/10.1145/1007730.1007734

[17] K. Napierala and J. Stefanowski, “Bracid: a comprehensive approach to learning rules from imbalanced data,” *Journal of Intelligent Information Systems*, vol. 39, no. 2, pp. 335–373, 2012.

[18] J. Stefanowski, “Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data,” in *Emerging Paradigms in Machine Learning*, vol. 13. Springer, 2013, pp. 277–306.

[19] M. Kuhn, “Building predictive models in R using the `caret` package,” *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26, 2008.

Appendix D
Paper IV - Predicting the Need for Vehicle Compressor Repairs Using Maintenance Records and Logged Vehicle Data

Predicting the Need for Vehicle Compressor Repairs Using Maintenance Records and Logged Vehicle Data

Rune Prytz[a], Sławomir Nowaczyk[b], Thorsteinn Rögnvaldsson[b], Stefan Byttner[b]

[a]*Volvo Group Trucks Technology, Advanced Technology & Research, Göteborg, Sweden.*
[b]*Center for Applied Intelligent Systems Research, Halmstad University, Sweden.*

Abstract

Methods and results are presented for applying supervised machine learning techniques to the task of predicting the need for repairs of air compressors in commercial trucks and buses. Prediction models are derived from logged on-board data that are downloaded during workshop visits and have been collected over three years on large number of vehicles. A number of issues are identified with the data sources, many of which originate from the fact that the data sources were not designed for data mining. Nevertheless, exploiting this available data is very important for the automotive industry as means to quickly introduce predictive maintenance solutions. It is shown on a large data set from heavy duty trucks in normal operation how this can be done and generate a profit.

Random forest is used as the classifier algorithm, together with two methods for feature selection whose results are compared to a human expert. The machine learning based features outperform the human expert features, wich supports the idea to use data mining to improve maintenance operations in this domain.

Keywords: Machine Learning, Diagnostics, Fault Detection, Automotive Industry, Air Compressor

1. Introduction

Today, Original Equipment Manufacturers (OEMs) of commercial transport vehicles typically design maintenance plans based on simple parameters such as calendar time or mileage. However, this is no longer sufficient in the market and there is a need for more advanced approaches that provide predictions of future maintenance needs of individual trucks. Instead of selling just vehicles, the sector is heading towards selling complete transport services; for example, a fleet of trucks, including maintenance, with a guaranteed level of availability. This moves some of the operational risk from the customer to the OEM but should lower the overall cost of ownership. The OEM

Email addresses: rune.prytz@volvo.com (Rune Prytz), slawomir.nowaczyk@hh.se (Sławomir Nowaczyk), thorsteinn.rognvaldsson@hh.se (Thorsteinn Rögnvaldsson), stefan.byttner@hh.se (Stefan Byttner)

has the benefit of scale and can exploit similarities in usage and wear between different vehicle operators.

Predicting future maintenance needs of equipment can be approached in many different ways. One approach is to monitor the equipment and detect patterns that signal an emerging fault, which is reviewed by Hines and Seibert (2006), Hines et al. (2008a), Hines et al. (2008b), and Ma and Jiang (2011). A more challenging one is to predict the Remaining Useful Life (RUL) for key systems, which is reviewed by Peng et al. (2010), Si et al. (2011), Sikorska et al. (2011) and Liao and Köttig (2014). For each of these approaches there are several options on how to do it: use physical models, expert rules, data-driven models, or hybrid combinations of these. The models can look for parameter changes that are linked to actual degradation of components, or they can look at vehicle usage patterns and indirectly infer the wear on the components. Data-driven solutions can be based on real-time data streamed during operation or collected historical data.

We present a data-driven approach that combines pattern recognition with the RUL estimation, by classifying if the RUL is shorter or longer than the time to the next planned service visit. The model is based on combining collected (i.e. not real-time) data from two sources: data collected on-board the vehicles and service records collected from OEM certified maintenance workshops. This presents a number of challenges, since the data sources have been designed for purposes such as warranty analysis, variant handling and financial follow-up on workshops, not for data mining. The data come from a huge set of real vehicles in normal operation, with different operators. The challenges include, among others, highly unbalanced datasets, noisy class labels, uncertainty in the dates, irregular readouts and unpredictable number of readouts from individual vehicles. In addition, multiple readouts from the same truck are highly correlated, which puts constraints on how data for testing and training are selected. We specifically study air compressors on heavy duty trucks and the fault complexity is also a challenge; air compressors face many possible types of failures, but we need to consider them all as one since they are not differentiated in the data sources.

The paper is structured as follows. A survey of related works introduces the area of data mining of warranty data. This is followed by an overview of the data sets and then a methodology section where the problem is introduced and the employed methods are described. This is finally followed by a results section and a conclusion section.

Related Work

There are few publications where service records and logged data are used for predicting maintenance needs of equipment, especially in the automotive industry, where wear prediction almost universally done using models that are constructed before production.

In a survey of artificial intelligence solutions in the automotive industry, Gus (2007) discuss fault prognostics, after-sales service and warranty claims. Two representative examples of work in this area are Buddhakulsomsiri and Zakarian (2009) and Rajpathak (2013). Buddhakulsomsiri and Zakarian (2009) present a data mining algorithm that extracts associative and sequential patterns from a large automotive warranty database, capturing relationships among occurrences of warranty claims over time.

Employing a simple IF-THEN rule representation, the algorithm filters out insignificant patterns using a number of rule strength parameters. In their work, however, no information about vehicle usage is available, and the discovered knowledge is of a statistical nature concerning relations between common faults. Rajpathak (2013) presents an ontology based text mining system that clusters repairs with the purpose of identifying best-practice repairs and, perhaps more importantly, automatically identifying when claimed labour codes are inconsistent with the repairs. Related to the latter, but more advanced, is the work by Medina-Oliva et al. (2014) on ship equipment diagnosis. They use an ontology approach applied to mining fleet data bases and convincingly show how to use this to find the causes for observed sensor deviations.

Thus, data mining of maintenance data and logged data has mainly focused on finding relations between repairs and operations and to extract most likely root causes for faults. Few have used them for estimating RUL or to warn for upcoming faults. We presented preliminary results for the work in this paper in an earlier study (Prytz et al., 2013). Furthermore, Frisk et al. (2014) recently published a study where logged onboard vehicle data were used to model RUL for lead-acid batteries. Their approach is similar to ours in the way that they also use random forests and estimate the likelihood that the component survives a certain time after the last data download. Our work is different from theirs in two aspects. First, a compressor failure is more intricate than a battery failure; a compressor can fail in many ways and there are many possible causes. Secondly, they also attempt to model the full RUL curve whereas we only consider the probability for survival until the next service stop.

Recently Choudhary et al. (2009) presented a survey of 150 papers related to the use of data mining in manufacturing. While their scope was broader than only diagnostics and fault prediction, they covered a large portion of literature related to the topic of this paper. Their general conclusion is that the specifics of the automotive domain make fault prediction and condition based maintenance a more challenging problem than in other domains; almost all research considers the case where continuous monitoring of devices is possible.

Jardine et al. (2006) present an overview of condition-based maintenance (CBM) solutions for mechanical systems, with special focus on models, algorithms and technologies for data processing and maintenance decision-making. They emphasize the need for correct, accurate, information (especially event informaton) and working tools for extracting knowledge from maintenance databases. Peng et al. (2010) also review methods for prognostics in CBM and conclude that methods tend to require extensive historical records that include many failures, even "catastrophic" failures that destroy the equipment, and that few methods have been demonstrated in practical applications. Schwabacher (2005) surveys recent work in data-driven prognostics, fault detection and diagnostics. Si et al. (2011) and Sikorska et al. (2011) present overviews of methods for prognostic modelling of RUL and note that available on-board data are seldom tailored to the needs of making prognosis and that few case studies exist where algorithms are applied to real world problems in realistic operating environments.

When it comes to diagnostics specifically for compressors, it is common to use sensors that continuously monitor the health state, e.g. accelerometers for vibration statistics, see Ahmed et al. (2012), or temperature sensors to measure the compressor working temperature, see Jayanth (2010 (filed 2006). The standard off-board tests

for checking the health status of compressors require first discharging the compressor and then measuring the time it takes to reach certain pressure limits in a charging test, as described e.g. in a compressor trouble shooting manual Bendix (2004). All these are essentially model-based diagnostic approaches where the normal performance of a compressor has been defined and then compared to the field case. Similarly, there are patents that describe methods for on-board fault detection for air brake systems (compressors, air dryers, wet tanks, etc.) that build on setting reference values at installment or after repair, see e.g. Fogelstrom (2007 (filed 2006).

In summary, there exist very few published examples where equipment maintenance needs are estimated from logged vehicle data and maintenance data bases. Yet, given how common these data sources are and how central transportation vehicles are to the society, we claim it is a very important research field.

2. Presentation of Data

Companies that produce high value products necessarily have well-defined processes for product quality follow-up, which usually rely on large quantities of data stored in databases. Although these databases were designed for other purposes, e.g. analysing warranty issues, variant handling and workshop follow-up, it is possible to use them also to model and predict component wear. In this work we use two such databases: the *Logged Vehicle Data* (LVD) and the *Volvo Service Records* (VSR).

LVD

The LVD database contains aggregated information about vehicle usage patterns. During operation, a vehicle continuously aggregates and stores a number of parameters, such as average speed or total fuel consumption. The values are downloaded each time a vehicle visits an OEM authorised workshop for service and repair. This happens several times per year, but at intervals that are irregular and difficult to predict *a priori*. In this work we have used data from approximately 65000 European Volvo trucks, models FH13 and FM13, produced between 2010 and 2013.

The vehicles in our data set visit a workshop, on average, every 15 weeks. This means that the predictive horizon for the prognostic algorithm must be at least that long. The system needs to provide warnings about components with an increased risk for failing until the next expected workshop visit. However, if the closest readout prior to the failure is 3-4 months, then it is less likely that the wear has had visible effects on the data.

This time sparseness is a considerable problem with the LVD. The readout frequency varies a lot between vehicles and changes with vehicle age, and can be as low as one readout per year. They also become less frequent as the vehicle ages. Figure 1 illustrates how many vehicles have data in LVD at different ages. Some vehicles (dark blue in the Figure) have consecutive data, defined as at least one readout every three months. They are likely to have all their maintenance and repairs done at OEM authorised workshops. Many vehicles, however, only have sporadic readouts (light blue in the Figure).

For data mining purposes are the vehicles with consecutive data most useful. They have readouts in the LVD database and repairs documented in the VSR system. They

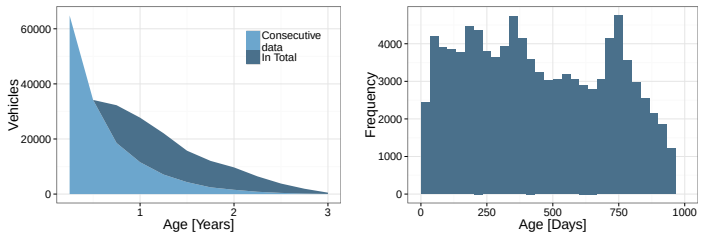


Figure 1: Vehicle age distribution, based on readouts in LVD. The left panel shows the number of vehicles, with data readouts, at different ages: dark blue are vehicles with any data; light blue are vehicles with consecutive data. The right panel shows the age of vehicles at data readouts for the subset of vehicles that have *any* readouts beyond age of two years (warranty period).

contribute with sequences of data that can be analysed for trends and patterns. On the other hand, from the business perspective it is important that as many trucks as possible are included in the analysis.

The right panel of Figure 1 illustrates two different maintenance strategies. The three peaks during the first year correspond to the typical times of scheduled maintenance. Repairs then get less frequent during the second year, with the exception of just before the end of it. This is probably the result of vehicles getting maintenance and repairs before the warranty period ends. In general, all vehicles visit the OEM authorised workshops often during the warranty period. After that, however, some vehicles disappear, while the remaining ones continue to be maintained as before, without any significant drop in visit frequency. This loss of data with time is a problematic issue. Plenty of valuable LVD data is never collected, even after the first year of vehicle operation. A future predictive maintenance solution must address this, either by collecting the logged data and the service information using telematics or by creating incentives for independent workshops to provide data.

Finally, the specification of parameters that are monitored varies from vehicle to vehicle. A core set of parameters, covering basic things like mileage, engine hours or fuel consumption, is available for all vehicles. Beyond that, however, the newer the vehicle is, the more LVD parameters are available, but it also depends on vehicle configuration. For instance, detailed gearbox parameters are only available for vehicles with automatic gearboxes. This makes it hard to get a consistent dataset across a large fleet of vehicles and complicates the analysis. One must either select a dataset with inconsistencies and deal with missing values, or limit the analysis to only vehicles that have the desired parameters. In this work we follow the latter approach and only consider parameter sets that are present across large enough vehicle fleets. Sometimes this means that we need to exclude individual parameters that most likely would have been useful.

VSR

The VSR database contains repair information collected from the OEM authorised workshops around the world. Each truck visit is recorded in a structured entry, labelled with date and mileage, detailing the parts exchanged and operations performed. Parts and operations are denoted with standardised identification codes. Unfortunately, however, there are no codes for reasons *why* operations are done. In some cases those can be deduced from the free text comments from the workshop personnel, but not always. The quality and level of detail of those comments vary greatly. This is a serious limitation since it introduces a lot of noise into the training data classification labels. In the worst case can a perfectly good part be replaced in the process of diagnosing an unrelated problem.

Undocumented repairs are also a problem. They rarely happen at authorised workshops since the VSR database is tightly coupled with the invoicing systems. On the other hand, there is seldom any information about repairs done in other workshops. Patterns preceding faults that suddenly disappear are an issue, both when training the classifier and later when evaluating it.

Much of the information in the VSR database is entered manually. This results in various human errors such as typos and missing values. A deeper problem, however, are incorrect dates and mileages, where information in the VSR database can be several weeks away from when the matching LVD data was read out. This is partly due to lack of understanding by workshop technicians; for the main purposes, invoicing and component failure statistics, exact dates are not overly important. In addition, the VSR date is poorly defined. In some cases the date can be thought of as the date of diagnosis, i.e. when the problem was discovered, and it may not be the same as the repair date, i.e. the date when the compressor was replaced.

3. Methods

Machine learning algorithm and software

All experimental results are averages over 10 runs using the Random Forest (Breiman, 2001) classifier, with 10-fold cross validation. We used the R language (R Core Team, 2014) including `caret`, `unbalanced`, `DMwR` and `ggplot2` libraries[1].

Evaluation criteria

Supervised machine learning algorithms are typically evaluated using measures like accuracy, area under the Receiver Operating Characteristic (ROC) curve or similar. Most of them, however, are suitable only for balanced datasets, i.e. ones with similar numbers of positive and negative examples. Measures that also work well for the unbalanced case include, e.g., the Positive Predictive Value (PPV) or F_1-score:

$$F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \qquad \mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{1}$$

[1] `http://cran.r-project.org/web/packages/`

where TP, FP and FN denote *true positives*, *false positives* and *false negatives*, respectively.

However, the prognostic performance must take business aspect into account, where the ultimate goal is to minimise costs and maximise revenue. In this perspective, there are three main components to consider: the initial investment cost, the financial gain from correct predictions, and the cost of false alarms.

The initial investment cost consists of designing and implementing the solution, as well as maintaining the necessary infrastructure. This is a fixed cost, independent of the performance of the method. It needs to be overcome by the profits from the maintenance predictions. In this paper we estimate it to be €150,000, which is approximately one year of full time work.

The financial gains come from correctly predicting failures before they happen and doing something about them. It is reported in a recent white paper by Reimer (2013) that *wrench time* (the repair time, from estimate approval to work completion), is on average about 16% of the time a truck spends at the workshop. Unexpected failures are one of the reasons for this since resources for repairs need to be allocated. All component replacements are associated with some *cost of repair*. However, unexpected breakdowns usually cause additional issues, such as *cost of delay* associated with not delivering the cargo in time. In some cases, there are additional costs like *towing*. Fixed *operational costs* correspond to the cost of owning and operating a vehicle without using it. This includes drivers wage, insurances and maintenance. A European long-haul truck costs on average €1,000 per day in fixed costs.

The cost of false alarms is the cost generated when good components are flagged by the system as needing replacement and an action is taken. At best this results in additional work for workshop personnel, and at worst it leads to unnecessary repairs.

It is important to note that *false negatives*, which correspond to actual component failures that were not detected, do not enter into the calculations. They are missed opportunities but they do not factor into the evaluation of a predictive maintenance solution; in comparison to the current maintenance scheme, where the vehicles run until failure, they maintain status quo.

In this respect is the predictive maintenance domain for the automotive industry quite different from many others. For instance, in the medical domain, *false negatives* correspond to patients who are not correctly diagnosed even though they carry the disease in question. This can have fatal consequences and be more costly than *false positives*, where patients get mistakenly diagnosed. It is also similar for the aircraft industry.

Among others, Sokolova et al. (2006) analyse a number of evaluation measures for assessing different characteristics of machine learning algorithms, while Saxena et al. (2008) specifically focus on validation of predictions. They note how lack of appropriate evaluation methods often renders prognostics meaningless in practice.

Ultimately, the criterion for evaluation of the model performance is a cost function based on the three components introduced at the beginning of this section. The function below captures the total cost of the implementation of the predictive maintenance system:

$$\text{Profit} = \text{TP} \times \text{ECUR} - \text{FP} \times \text{CPR} - \text{Investment}, \tag{2}$$

where ECUR stands for *extra cost of unplanned repair* and CPR stands for *cost of planned repair*. Each *true positive* avoids the additional costs of a breakdown and each *false positive* is a repair done in vain, which causes additional costs.

It is interesting to study the ratio between the cost of planned and unplanned repairs. It will vary depending on the component, fleet operator domain and business model, et cetera. On the other hand, the cost for a breakdown for vehicles with and without predictive maintenance can be used to determine the "break even" ratio required between *true positives* and *false positives*.

Prediction Horizon

We define the Prediction Horizon (PH) as the period of interest for the predictive algorithm. A replacement recommendation should be made for a vehicle for which the air compressor is expected to fail within that time frame into the future. As described earlier, the vehicles visit the workshop on average every 15 weeks and the PH needs to be at least that long. The system should provide warnings about components that are at risk of failing before the next expected workshop visit.

It is expected that the shorter the PH, the more likely it is that there is information in the data about upcoming faults. It is generally more difficult to make predictions the further into the future they extend, which calls for a short PH. However, from a business perspective it is desireable to have a good margin for planning, which calls for a long PH. We experiment with setting the PH up to a maximum of 50 weeks.

Independent data sets for training and testing

A central assumption in machine learning (and statistics) it that of *independent and identically distributed (IID)* data. There are methods that try to lift it to various degrees, and it is well known that most common algorithms work quite well also in cases when this assumption is not fully fulfilled, but it is still important, especially when evaluating and comparing different solutions.

The readouts consist of aggregated data that have been sampled at different times. Subsequent values from any given truck are highly correlated to each other. It is even more profound in case of cumulative values, such as total mileage, a single event of abnormal value will directly affect all subsequent readouts. Even without the aggregation effect, however, there are individual patterns that are specific to each truck, be it particular usage or individual idiosyncrasies of the complex cyber-physical system. It makes all readouts from a single vehicle dependent. This underlying pattern of the data is hard to visualize by analysing the parameter data as such. However, a classifier can learn these patterns and overfit.

A partial way of dealing with the problem is to ensure that the test and train dataset be split on a *per vehicle* basis and not randomly among all readouts. It means that if one or more readouts from a given vehicle belong to the test set, no readouts from the same vehicle can be used to train the classifier. The data sets for training and testing must contain unique, non-overlapping, sets of vehicles in order to guarantee that patterns that are linked to wear and usage are learned, instead of specific usage patterns for individual vehicles.

Feature selection

The data set contains 1,250 unique features and equally many differentiated features. However, only approximately 500 of them are available for the average vehicle. It is clear that not all features should be used as input to the classifier. It is important to find the subset of features that yields the highest classification performance. Additionally, the small overlap of common features between vehicles makes this a research challenge. It is hard to select large sets of vehicles that each share a common set of parameters. Every new feature that gets added to the dataset must be evaluated with respect to the gain in performance and the decrease in number of examples.

Feature selection is an active area of research, but our setting poses some specific challenges. Guyon and Elisseeff (2003) and Guyon et al. (2006) present a comprehensive and excellent, even if by now somewhat dated, overview of the feature selection concepts. Bolón-Canedo et al. (2013) present a more recent overview of methods. Molina et al. (2002) analyse performance of several fundamental algorithms found in the literature in a controlled scenario. A scoring measure ranks the algorithms by taking into account the amount of relevance, irrelevance and redundance on sample data sets. Saeys et al. (2007) provide a basic taxonomy of feature selection techniques, and discuss their use, variety and potential, from the bioinformatics perspective, but many of the issues they discuss are applicable to the data analysed in this paper.

We use two feature selection methods: a wrapper approach based on the beam search algorithm, as well as a new filter method based on the Kolmogorov-Smirnov test to search for the optimal feature set. The final feature sets are compared against an expert dataset, defined by an engineer with domain knowledge. The expert dataset contains four features, all of which have direct relevance to the age of the vehicle or the usage of the air compressor.

The beam search feature selection algorithm performs a greedy graph search over the powerset of all the features, looking for the subset that maximises the classification accuracy. However, at each iteration, we only expand nodes that maintain the data set size above the given threshold. The threshold is reduced with the number of parameters as shown in equation (3). Each new parameter is allowed to reduce the dataset with a small fraction. This ensures a lower bound on the data set size. The top five nodes, with respect to accuracy, are stored for next iteration. This increased the likelihood of finding the global optimum. The search is stopped when a fixed number of features is found:

$$n_{dataset} = n_{all} \times constraintFactor^{n_{params}}. \tag{3}$$

Many parameters in the LVD dataset are highly correlated and contain essentially the same information, which can potentially lower the efficiency of the beam search. Chances are that different beams may select different, but correlated, feature sets. In this way the requirement for diversity on the "syntactic" level is met, while the algorithm is still captured in the same local maximum. We have not found this to be a significant problem in practice.

With the Kolmogorov-Smirnov method, we are interested in features whose distributions vary in relation to oncoming failures of air compressors. There are two main reasons for such variations: they are either related to different usage patterns of the vehicle, or to early symptoms of component wear. To identify these features, we define

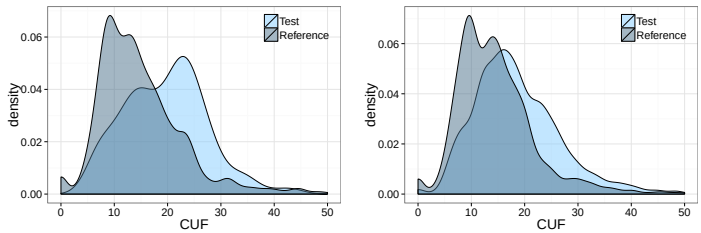


Figure 2: Differences due to usage. The left panel shows the normal and fault distributions for the feature *Compressor Duty Cycle* (CUF) when the fault sample (light blue) is drawn from periods 0–5 weeks prior to the compressor repair. The right panel shows the same thing but when the fault sample (light blue) is drawn from 0–25 weeks prior to the repair. The normal sample (grey) is in both cases selected from vehicles that have not had any air compressor repair.

normal and *fault* data sets, and compare their distributions using the Kolmogorov-Smirnov (KS) test (Hazewinkel, 2001).

The *normal* sample is a random sample of fault-free LVD readouts, while the *fault* sample are LVD readouts related to a compressor repair. The fault sample is drawn from vehicles with a compressor change and selected from times up to PH before the repair. The normal sample is drawn either from all vehicles that have not had a compressor change, or from vehicles with a compressor change but outside of the PH time window before the repair. These two different cases are referred to as *usage* difference and *wear* difference, respectively.

The two samples are compared using a two-sample KS test and a p-value is computed under the null hypothesis that the two samples are drawn from the same distribution. The p-value is a quantification of how likely it is to get the observed difference if the null hypothesis is true and a low p-value indicates that the null hypothesis may not be true. Features with low p-values are therefore considered interesting since the observed difference may indicate a fundamental underlying effect (wear or usage). The lower the p-value, the more interesting the feature. The KS filter search is terminated when a predetermined number of features has been reached.

Figure 2 illustrates the case with the feature Compressor Duty Cycle (CUF) when evaluated as relevant from the usage point of view. There is a clear difference 0–5 weeks before the repair and there is also a difference 0–25 weeks before the repair. Figure 3 illustrates the same case but when evaluated from the wear point of view. Also in the latter case is CUF selected as a very important feature.

In the previous description there was no consideration taken to the age of the vehicles, i.e. it was assumed that feature differences are independent of vehicle age. This may not always be correct. Wear features can be affected by the vehicle age and one might get spurious results in the feature selection because the age distribution in the fault set is different from the normal set. The risk for such spurious effects can be reduced by sampling the normal group so that the sample has the the same mileage or engine hour distribution as the fault group. We call this age normalisation. The

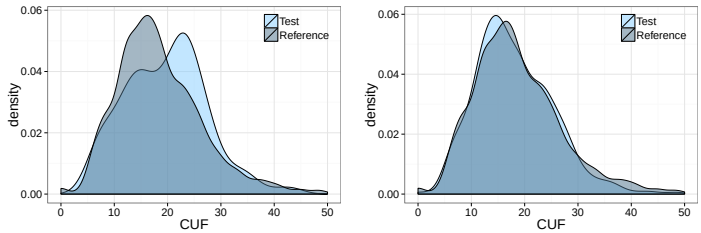


Figure 3: Differences due to wear. The left panel shows the normal and fault distributions for the feature *Compressor Duty Cycle* (CUF) when the fault sample (light blue) is drawn from periods 0–5 weeks prior to the compressor repair. The right panel shows the same thing but when the fault sample (light blue) is drawn from 0–25 weeks prior to the repair. The normal sample (grey) is in both cases selected from vehicles that have had an air compressor repair, but times that are before the PH fault data.

sampling is done in two steps. The first step is to resample the reference distribution uniformly. In the second step is the uniform reference distribution sampled again, this time weighted according to the distribution of the test set. In cases with a narrow age distribution for the fault set will only a fraction of the normal data be used. This requires a substantial amount of normal data which, in our case, is possible since the dataset is highly unbalanced and there is much more normal data than fault data. The effect of age normalisation is illustrated in Fig. 4.

Balancing the dataset

Machine learning methods usually assume a fairly balanced data distribution. If that is not fulfilled, then the results tend to be heavily biased towards the majority class. This is a substantial problem in our case, since only a small percentage of the vehicles experience compressor failure and, for any reasonable value of the PH, only a small subset of their readouts is classified as faulty.

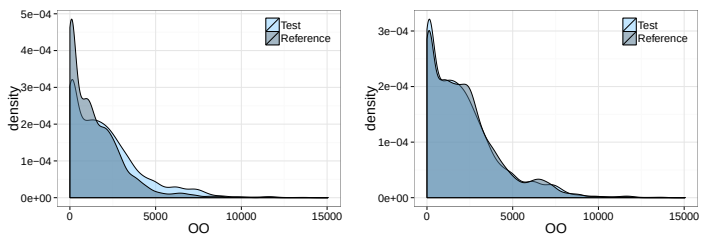


Figure 4: Illustration of the effect of age normalisation. The left panel shows the normal (grey) and the fault (light blue) distributions for a feature without age noralisation. The right panel shows the result after age normalisation. Here age normalisation removed the difference, which was a spurious effect caused by age.

Imbalanced datasets require either learning algorithms that handle this or data preprocessing steps that even out the imbalance. We chose to use the latter. There are many domains where class imbalance is an issue, and therefore a significant body of research is available concerning this. For example, He and Garcia (2009) provide a comprehensive review of the research concerning learning from imbalanced data. They provide a critical review of the nature of the problem and the state-of-the-art techniques. They also highlight the major opportunities and challenges, as well as potential important research directions for learning from imbalanced data. Van Hulse et al. (2007) present a comprehensive suite of experimentation on the subject of learning from imbalanced data. Sun et al. (2007) investigate meta-techniques applicable to most classifier learning algorithms, with the aim of advancing the classification of imbalanced data, exploring three cost-sensitive boosting algorithms, which are developed by introducing cost items into the learning framework of AdaBoost. Napierala and Stefanowski (2012) propose a comprehensive approach, called BRACID, that combines multiple different techniques for dealing with imbalanced data, and evaluate it experimentally on a number of well-known datasets.

We use the Synthetic Minority Over-sampling TEchnique (SMOTE), introduced by Chawla et al. (2002). It identifies, for any given positive example, the k nearest neighbours belonging to the same class. It then creates new, synthetic, examples randomly placed in between the original example and the k neighbours. It uses two design parameters: number of neighbours to take into consideration (k) and the percentage of synthetic examples to create. The first parameter, intuitively, determines how similar new examples should be to existing ones, and the other how balanced the data should be afterwards. SMOTE can be combined with several preprocessing techniques, e.g. introduced by Batista et al. (2004) and some others, aggregated and implemented in a R library by Dal Pozzolo et al.. We tried and evaluated four of them: The Edited Nearest Neighbour (ENN), the Neighbourhood Cleaning Rule (NCL), the Tomek Links (TL), and the Condensed Nearest Neighbour (CNN).

4. Results

Cost function

The *cost of planned repair* CPR, *cost of unplanned repair* CUR, and *extra cost of unplanned repair* ECUR can be split up into the following terms:

$$\begin{aligned} \text{CPR} &= \text{C}_{part} + \text{C}^{P}_{work} + \text{C}^{P}_{downtime} && (4)\\ \text{CUR} &= \text{C}_{part} + \text{C}^{U}_{work} + \text{C}^{U}_{downtime} + \text{C}_{extra} && (5)\\ \text{ECUR} &= \text{CUR} - \text{CPR} && (6) \end{aligned}$$

Here, C_{part} is the cost of the physical component, the air compressor, that needs to be exchanged. We set this to €1000. It is the same for both planned and unplanned repairs. C_{work} is the labour cost of replacing the air compressor, which takes approximately three hours. We set C^{P}_{work} to €500 for planned repairs and C^{U}_{work} to €1,000 for unplanned repairs. If the operation is unplanned, then one needs to account for diagnosis, disruptions to the workflow, extra planning, and so on.

$C_{downtime}$ is the cost for vehicle downtime. Planned component exchanges can be done together with regular maintenance; $C^P_{downtime}$ is therefore set to zero. It is included in equation (4) since it will become significant in the future, once predictive maintenance becomes common and multiple components can be repaired at the same time. The downtime is a crucial issue for unplanned failures, however, especially roadside breakdown scenarios. Commonly at least half a day is lost immediately, before the vehicle is transported to the workshop and diagnosed. After that comes waiting for spare parts. The actual repair may take place on the third day. The resulting 2–3 days of downtime plus a possible cost of towing, $C^U_{downtime}$, is estimated to cost a total of €3,500.

The additional costs, C_{extra}, are things like the delivery delay, the cost for damaged goods, fines for late arrival, and so on. This is hard to estimate, since it is highly dependent on the cargo, as well as on the vehicle operator's business model. The *just in time* principle is becoming more widespread in the logistics industry and the additional costs are therefore becoming larger. We set C_{extra} to €11,000.

Inserting those estimates into equations (4), (5) and (6) yields CPR = €1,500, CUR = €16,500 and ECUR = €15,000. The final Profit function, eq. (2), becomes (in Euros):

$$\text{Profit}(\text{TP}, \text{FP}) = \text{TP} \times 15,000 - \text{FP} \times 1,500 - 150,000. \tag{7}$$

Obviously, the Profit function (7) is an estimate and the numbers have been chosen so that there is a simple relationship between the gain you get from true positives and the loss you take from false positives (here the ratio is 10:1). A more exact ratio is hard to calculate since it is difficult to get access to the data required for estimating it (this type of information is usually considered confidential). Whether the predictive maintenance solution has a profit or loss depends much on the extra cost C_{extra} .

The importance of data independence

The importance of selecting independent data sets for training and testing cannot be overstated. Using dependent data sets will lead to overly optimistic results that never hold in the real application. Figure 5 shows the effects from selecting training and test data sets in three different ways.

The *random* method refers to when samples for training and testing are chosen completely randomly, i.e. when examples from the same vehicle can end up both in the training and the test data set. These data sets are not independent and the out-of-sample accuracy is consequently overestimated.

The *one sample* method refers to when each vehicle provides one positive and one negative example to the training and test data, and there is no overlap of vehicles in the training and test data. This leads to independent data sets that are too limited in size. The out-of-sample performance is correctly estimated but the data set cannot be made large. The *all sample* method refers to the case when each vehicle can contribute with any number of examples but there is no overlap of vehicles between the training and test data. This also yields a correct out-of-sample accuracy but the training data set can be made larger.

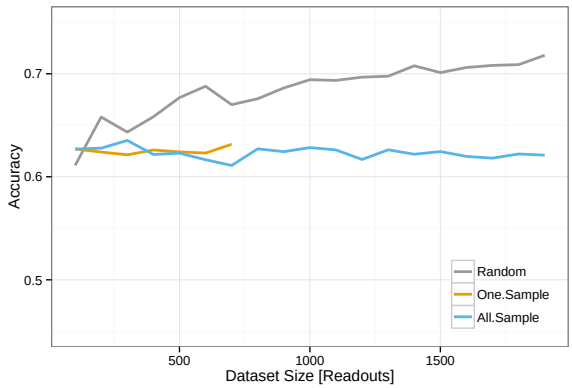


Figure 5: Comparison of strategies for selecting training and test data. The *Expert* feature set was used for all experiments and the data sets were balanced. The x-axis shows the size of the training data set.

Feature selection

The different feature selection approaches, and the age normalisation of the data, described in the Methods section produced six different feature sets in addition to the *Expert* feature set.

The beam search wrapper method was performed with five beams and a size reduction constraint of 10%. The search gave five different results, one from each beam, but four of them were almost identical, differing only by the last included feature. The four almost identical feature sets where therefore reduced to a single one, by including only the 14 common features. The fifth result was significantly different and was kept without any modifications. The two feature sets from the beam search are denoted *Beam search set 1* and *Beam search set 2*, respectively; they each had 14 features (the limit set for the method). Three out of the four features selected by the expert were also found by the beam search.

The KS filter method was used four times, with different combinations of *wear* features, *usage* features, and age normalisation (the reader is referred to the Methods section for details). This gave four feature sets: *Wear*, *Usage*, *Wear with age normalisation*, and *Usage with age normalisation*.

Figure 6 show the results when using the seven different feature sets. The overly optimistic result from using randomly selected data sets is shown for pedagogical reasons, reiterating the importance of selecting independent data sets. The data sets were balanced in the experiments. The *Usage* features performed best and the *Expert* features were second (except when an erroneous method for selecting data was used).

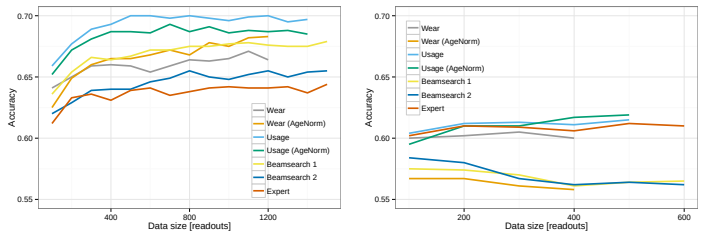


Figure 6: Comparison of feature selection methods when measuring the accuracy of the predictor. The left panel shows the result when training and test data are chosen randomly, i.e. with dependence. The right panel shows the result when the training and test data are chosen with the one sample method, i.e. without dependence.

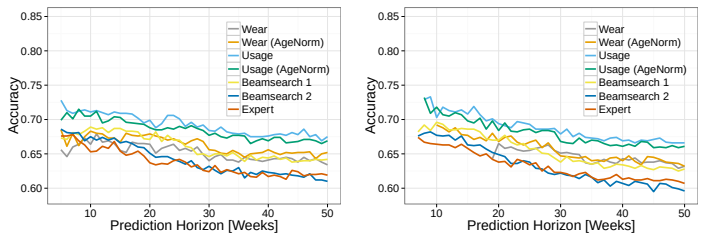


Figure 7: Prediction accuracy vs. prediction horizon. The left panel shows how the prediction accuracy decreases with PH when the training data set size is not limited. The right panel shows how the prediction accuracy decreases with PH when the training data set size is limited to 600 samples.

Accuracy vs. Prediction Horizon

Two experiments were done to gauge how the Prediction Horizon (PH) affects the classification results. In the first experiment were all available readouts used, while in the second experiment was the training set size fixed at 600 samples. Balanced data sets were used throughout why the number of available fault readouts were the limiting factor. As PH increased could more samples be used since more readouts were available for inclusion in the fault data set.

Figures 7 and 8 show the result of the two experiments. The accuracy (Fig. 7) is best at lower PH and decreases as the PH increases. This is probably due to the training labels being more reliable closer to the fault. Accuracy decreases somewhat less rapidly in the first experiment with unlimited training data set size (left panel of Fig. 7). Figure 8 shows the result when evaluated with the profit measure.

SMOTE

The SMOTE oversampling method depends on two parameters: the percentage of synthetic examples to create and the number of neighbours to consider when generating new examples.

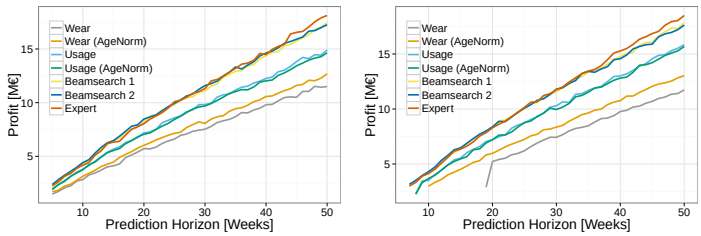


Figure 8: Profit vs. prediction horizon. The left panel shows how the profit increases with PH when the training data is not limited. The right panel shows how the profit increases with PH when the training data is limited to 600 samples. The PH required to achieve 600 samples varies between the datasets, which explains the differences in starting positions of individual lines.

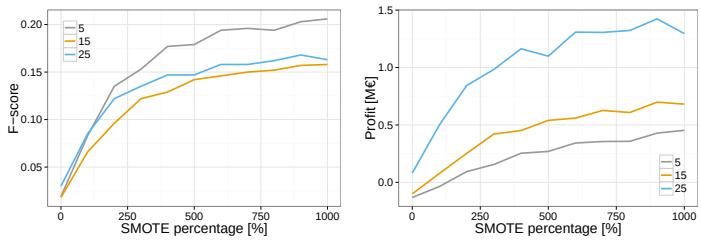


Figure 9: Evaluation of SMOTE percentage settings, using the *Expert* dataset. The number of SMOTE neighbours is fixed to 20.

Figure 9 shows F-score and Profit when the SMOTE percentage is varied but k is kept constant at 20, for three different values of the PH. All results improve significantly with the percentage of synthetic examples, all the way up to a ten-fold oversampling of synthetic examples. A lower PH is better from the F-score perspective but worse from the Profit perspective. Figure 10 shows the effect of varying k, the number of SMOTE neighbours, when the SMOTE percentage is kept fixed at 900%. The results are not very sensitive to k although a weak increase in performance comes with higher k.

The four SMOTE preprocessing methods mentioned in section 3 were evaluated using a PH of 25 weeks and a SMOTE percentage of 900% (the best average settings found). Nearly all feature sets benefitted from preprocessing but there was no single best method.

Final evaluation

A final experiment was done, using the best settings found for each feature set, in order to evaluate the whole approach. The best SMOTE settings were determined by first keeping k fixed at 20 and finding the best SMOTE%. Then the SMOTE% was

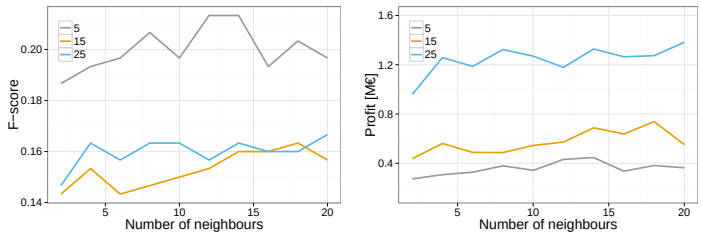


Figure 10: Evaluation of the number of SMOTE neighbours (k) using the *Expert* dataset and with the SMOTE% fixed at 900.

Feature set	Samples	Features	%	k	Prepr	Profit	nProfit
Wear	10,660	20	700	14	TL	1.59	86
Wear AN	10,520	20	1000	12	ENN	0.62	22
Usage	12,440	20	1000	16	TL	1.94	114
Usage AN	12,440	20	1000	20	CNN	1.60	110
Beam search 1	14,500	14	800	20	NCL	1.66	116
Beam search 2	14,500	15	800	16	TL	0.75	54
Expert	14,960	4	900	20	ENN	0.84	64

Table 1: The best settings for each of the feature sets (AN denotes age normalised). The total number of samples (second column) depends on the method used for selecting the data. The columns marked with % and k show the SMOTE parameter settings and the column labelled Prepr shows the SMOTE preprocessing method used. The Profit is evaluated for a PH of 15 weeks and the optimal training data set size (see Fig. 11 and the discussion in the text). The Profit (M€) depends on the test data set size, which depends on the method used for selecting the data. The rightmost column, labeled nProfit, shows per vehicle Profit (in €) which is the Profit normalised with respect to the number of vehicles in the testsets.

kept fixed at the best value and the k value varied between 1 and 20 and the value that produced the best cross-validation Profit was kept. The best SMOTE preprocessing determined in the previous experiments was used for each feature set. The final best settings for each feature set are summarised in Table 1, together with the basic data for each feature set.

The left panel of Fig. 11 shows how varying the training data set size affects the Profit. The PH was set to 15 weeks, which is a practical PH, even though many of the feature sets perform better at higher values of PH. From a business perspective is a PH of 30 weeks considered too long, since it leads to premature warnings when the vehicle is likely to survive one more maintenance period. The ordering of feature selection algorithms is mostly consistent; *Usage* is best, with the exception of very small data sizes where it is beaten by *Beam search 1*.

The left panel of Fig. 11 also shows an interesting phenomenon where profit grows and then drops as the data set size increases. This is unexpected, and we are unable to explain it. It may be related, for example, to the k parameter of the SMOTE algorithm. The right panel of Fig. 11 illustrates how varying the prediction horizon affects the

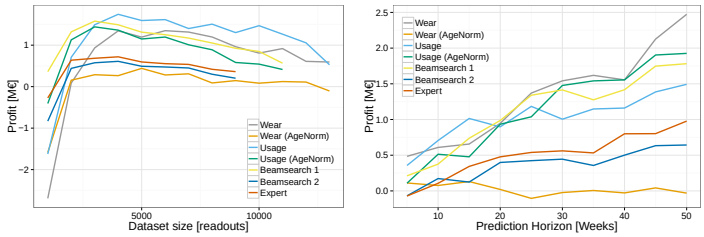


Figure 11: Final evaluation of all feature sets. The left panel shows how the Profit varies with the training data set size using a prediction horizon of 15 weeks. The right panel shows how the Profit changes with PH. The settings for each feature set are listed in Table 1.

Profit, using all available data for each feature set. In general, the longer the PH the better the Profit. The relative ordering among feature sets is quite consistent, which indicates that neither of them focus solely on patterns of wear. Such features would be expected to perform better at lower PH when the wear is more prominent.

The performances listed in Table 1 are for one decision threshold. However, the classifiers can be made to be more or less restrictive when taking their decision to recommend a repair, which will produce different numbers of true positives, true negatives, false positives and false negatives. Figure 12 shows the sensitivity–specificity relationships for each feature set (i.e. each classifier using each feature set). The perfect classifier, which certainly is unachievable in this case, would have both sensitivity and specificity equal to one. It is, from Fig. 12, clear that the feature sets *Beam search 1* and *Usage*, with or without age normalisation, are the best from the perspective of sensitivity and specificity. All three are better than the *Expert* feature set. Profit is not uniquely defined by specificity and sensitivity; it depends on the data set size and the mix of positive and negative examples. However, Profit increases from low values of specificity and sensitivity to high values.

5. Conclusions

Transportation is a low margin business where unplanned stops quickly turn profit to loss. A properly maintained vehicle reduces the risk of failures and keeps the vehicle operating and generating profit. Predictive maintenance introduces dynamic maintenance recommendations which react to usage and signs of wear.

We have presented a data driven method for predicting upcoming failures of the air compressor of a commercial vehicle. The predictive model is derived from currently available warranty and logged vehicle data. These data sources are in-production data that are designed for and normally used for other purposes. This imposes challenges which are presented, discussed and handled in order to build predictive models. The research contribution is twofold: a practical demonstration on these practical data, which are of a type that is abundant in the vehicle industry, and the techniques developed and

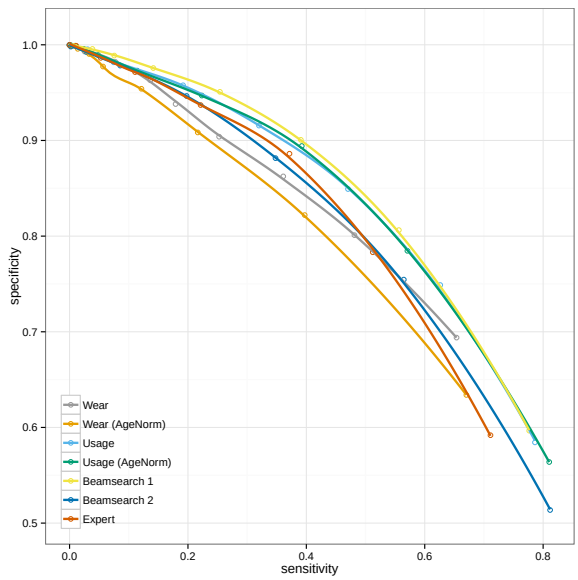


Figure 12: Sensitivity and specifity for the classifiers based on each feature set using the optimal settings in Table 1. Profit increases from lower left towards the upper right.

tested to handle; feature selection with inconsistent data sets, imbalanced and noisy class labels and multiple examples per vehicle.

The method generalises to repairs of various vehicle components but it is evaluated on one component: the air compressor. The air compressor is a challenge since a failing air compressor can be due to many things and can be a secondary fault caused by other problems (e.g. oil leaks in the engine that cause coal deposits in the air pipes). Many fault modes are grouped into one label. Components with clearer or fewer fault causes should be easier to predict, given that the information needed to predict them is available in the data sources, and given that the fault progresses slow enough. We have not tested it on other components but plan to do so in the near future.

The best features are the *Beam search 1* and the *Usage* sets, with or without age normalisation. All three outperform the *Expert* feature set, which strengthens the arguments for using data driven machine learning algorithms within this domain. There is an interesting difference between the *Wear* and *Usage* feature sets. In the latter, there is little effect of doing age normalisation while on the first the age normalisation removes a lot of the information. This indicates that important wear patterns are linked to age, which in turn is not particularly interesting since age is easily measured using mileage or engine hours. It is possible that trends due to wear are faster than what is detectable given the readout frequency. This could partly explain the low performance of the wear features.

All feature sets show a positive Profit in the final evaluation. However, this depends on the estimated costs for planned and unplanned repair. There are large uncertainties in those numbers and one must view the profits from that perspective. The investment cost can probably be neglected and the important factor is the ratio in cost between unplanned and planned repair.

Acknowledgment

The authors thank Vinnova (Swedish Governmental Agency for Innovation Systems), AB Volvo, Halmstad University, and the Swedish Knowledge Foundation for financial support for doing this research.

, 2007. Knowledge and Information Systems 12.

Ahmed, M., Baqqar, M., Gu, F., Ball, A.D., 2012. Fault detection and diagnosis using principal component analysis of vibration data from a reciprocating compressor, in: Proceedings of the UKACC International Conference on Control, 3-5 September 2012, IEEE Press.

Batista, G.E.A.P.A., Prati, R.C., Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations Newsletter 6, 20–29.

Bendix, 2004. Advanced Troubleshooting Guide for Air Brake Compressors. Bendix Commercial Vehicle Systems LLC.

Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., 2013. A review of feature selection methods on synthetic data. Knowledge and Information Systems 34, 483–519.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Buddhakulsomsiri, J., Zakarian, A., 2009. Sequential pattern mining algorithm for automotive warranty data. Computers & Industrial Engineering 57, 137–147.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357.

Choudhary, A.K., Harding, J.A., Tiwari, M.K., 2009. Data mining in manufacturing: a review based on the kind of knowledge. Journal of Intelligent Manufacturing 20, 501–521.

Dal Pozzolo, A., Caelen, O., Bontempi, G., . Comparison of balancing techniques for unbalanced datasets.

Fogelstrom, K.A., 2007 (filed 2006). Prognostic and diagnostic system for air brakes.

Frisk, E., Krysander, M., Larsson, E., 2014. Data-driven lead-acid battery prognostics using random survival forests, in: Proceedings of the 2:nd European Conference of the PHM Society (PHME14).

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182.

Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A., 2006. Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing). Springer-Verlag New York, Inc.

Hazewinkel, M. (Ed.), 2001. Encyclopedia of Mathematics. Springer.

He, H., Garcia, E., 2009. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21, 1263–1284.

Hines, J., Garvey, D., Seibert, R., , Usynin, A., 2008a. Technical Review of On-Line Monitoring Techniques for Performance Assessment. 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, D.R., Seibert, R., 2008b. Technical Review of On-Line Monitoring Techniques for Performance Assessment. 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. 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.

Jardine, A.K., Lin, D., Banjevic, D., 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing 20, 1483–1510.

Jayanth, N., 2010 (filed 2006). Compressor protection and diagnostic system.

Liao, L., Köttig, F., 2014. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. IEEE Transactions on Reliability 63, 191–207.

Ma, J., Jiang, J., 2011. Applications of fault detection and diagnosis methods in nuclear power plants: A review. Progress in Nuclear Energy 53, 255–266.

Medina-Oliva, G., Voisin, A., Monnin, M., Léger, J.B., 2014. Predictive diagnosis based on a fleet-wide ontology approach. Knowledge-Based Systems .

Molina, L., Belanche, L., Nebot, A., 2002. Feature selection algorithms: a survey and experimental evaluation, in: Proceedings of IEEE International Conference on Data Mining, pp. 306–313.

Napierala, K., Stefanowski, J., 2012. BRACID: a comprehensive approach to learning rules from imbalanced data. Journal of Intelligent Information Systems 39, 335–373.

Peng, Y., Dong, M., Zuo, M.J., 2010. Current status of machine prognostics in condition-based maintenance: a review. International Journal of Advanced Manufacturing Technology 50, 297–313.

Prytz, R., Nowaczyk, S., Rögnvaldsson, T., Byttner, S., 2013. Analysis of truck compressor failures based on logged vehicle data, in: Proceedings of the 2013 International Conference on Data Mining (DMIN13). URL: http://worldcomp-proceedings.com/proc/p2013/DMI.html.

R Core Team, 2014. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. URL: http://www.R-project.org.

Rajpathak, D.G., 2013. An ontology based text mining system for knowledge discovery from the diagnosis data in the automotive domain. Computers in Industry 64, 565–580.

Reimer, M., 2013. Service Relationship Management – Driving Uptime in Commercial Vehicle Maintenance and Repair. White paper. DECISIV.

Saeys, Y., Inza, I., Larraaga, P., 2007. A review of feature selection techniques in bioinformatics. Bioinformatics 23, 2507–2517.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., Schwabacher, M., 2008. Metrics for evaluating performance of prognostic techniques, in: Proceedings of the 2008 International Conference on Prognostics and Health Management, IEEE Press.

Schwabacher, M., 2005. A survey of data-driven prognostics, in: Infotech@Aerospace.

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

Sikorska, J.Z., Hodkiewicz, M., Ma, L., 2011. Prognostic modelling options for remaining useful life estimation by industry. Mechanical Systems and Signal Processing 25, 1803–1836.

Sokolova, M., Japkowicz, N., Szpakowicz, S., 2006. Beyond accuracy, f-score and ROC: A family of discriminant measures for performance evaluation, in: AI 2006: Advances in Artificial Intelligence. Springer Berlin Heidelberg. volume 4304 of *Lecture Notes in Computer Science*, pp. 1015–1021.

Sun, Y., Kamel, M.S., Wong, A.K., Wang, Y., 2007. Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition 40, 3358–3378.

Van Hulse, J., Khoshgoftaar, T.M., Napolitano, A., 2007. Experimental perspectives on learning from imbalanced data, in: Proceedings of the 24th International Conference on Machine Learning, ACM, New York, NY, USA. pp. 935–942.