Anomaly detection in recordings from in-vehicle networks

Andreas Theissler

IT-Designers GmbH, Esslingen, Germany
andreas.theissler@it-designers.de

published in proceedings of

BIG DATA APPLICATIONS AND PRINCIPLES
First International Workshop, BIGDAP 2014
Madrid, Spain, September 11-12 2014

Anomaly detection in recordings from in-vehicle networks

Andreas Theissler

IT-Designers GmbH, Esslingen, Germany
andreas.theissler@it-designers.de

Abstract. In the automotive industry test drives are being conducted during the development of new vehicle models or as a part of quality assurance of series-production vehicles. Modern vehicles have 40 to 80 electronic control units interconnected via the so-called in-vehicle network. The communication on this in-vehicle network is recorded during test drives for the use of fault analysis, which results in big data.
This paper proposes to use machine learning to support domain-experts by preventing them from contemplating irrelevant data and rather pointing them to the relevant parts in the recordings. The underlying idea is to learn the normal behaviour from the available multivariate time series and then to autonomously detect unexpected deviations and report them as anomalies.
The one-class support vector machine "support vector data description" is enhanced to work on multivariate time series. The approach allows to detect unexpected faults without modelling effort as is shown on recordings from test drives.
The proposed methodology could be applicable to multivariate time series from other sources, e.g. industrial plants or network traffic with fixed communication patterns.

Keywords: anomaly detection, data mining, fault detection, machine learning

1 Introduction

This paper proposes an approach to detect anomalies in multivariate times series in recordings from in-vehicle networks. The effectiveness of the approach is shown by applying it to big data recorded during vehicle tests.

Modern vehicles have 40 to 80 electronic control units (ECUs) interconnected via the so-called in-vehicle network. Those ECUs read data measured by sensors, calculate values and control actuators. The sensors' and actuators' values are being transmitted over the in-vehicle network to other ECUs. This results in a highly complex network of software and hardware subsystems.

In order to be able to locate faults or to evaluate the behaviour of vehicle subsystems, the communication on the in-vehicle network is being recorded during test drives. This kind of recordings are conducted by manufacturers with

prototype vehicles, before start of production, or with series-production vehicles as part of the end of line tests.

The big data resulting from recording test drives is in some cases searched for known fault patterns. Additionally, suspicious behaviour is reported by the test drivers. However, there are no systematic measures to detect unexpected faults. To address this shortcoming, this paper contributes by proposing an approach that

- uses available multivariate time series from in-vehicle networks and extracts the relevant knowledge
- autonomously points the expert to anomalies in the time series

From the reported anomalies, the expert can start investigating the data base of recordings in a goal-oriented way.

2 Related work

In [12], the authors propose to use visual analytics to explore data from automotive systems. In [8] a data-driven approach to classify the health state of an in-vehicle network based on the occurrences of error frames on the CAN bus is proposed. In contrast to the underlying paper, [8] bases on a training set of recordings from fault-free and faulty mode.

In [2] anomaly detection is used on vehicle data in the field of road condition monitoring. Based on a training set of recordings from drives in normal operation mode, potholes are identified as anomalies.

In [6] intrusion detection based on recordings from in-vehicle network communication is presented. The underlying assumption is, that the communication on the in-vehicle network has a certain degree of randomness, i.e. entropy. From data recorded in normal operation mode, the normal value of entropy is learnt. An attack, like increasing the frequency of specific messages or message flooding, appears less random and is thereby detected as an anomaly.

In [3] classification between sober and drunk drivers based on ARMA models is proposed. The determination of the order and the coefficients of the models is a great challenge with that approach.

3 Anomaly detection using one-class support vector machines

Detecting anomalies can be automated by teaching an anomaly detection system normal and abnormal behaviour by the means of a labelled training set and have the system classify unseen data. This corresponds to a two-class classification problem. The task is to assign an unclassified instance to either the normal class ω_n or the abnormal class ω_a based on a set of features f. For fault-detection two major drawbacks of such a traditional classification approach were identified:

1. Often no abnormal data sets exist beforehand. On the other hand normal data can be obtained by recording data from a system in normal operation mode.
2. Even if abnormal data exists, it is highly likely that it is not representative, because many faults in a system are not known. Using a non-representative training data set of anomalies, an incorrect decision function is learned.

An alternative is to only learn the normal behaviour and classify deviations as abnormal referred to as one-class classification. Support vector machines (SVM)[11, 1] have shown to yield good results on classification tasks and have been widely used. In [9] the one-class SVM "support vector data description" (SVDD) was introduced to cope with the problem of one-class classification. SVDD finds a closed decision boundary, a hypersphere, around the normal instances in the training data set using a so-called kernel function. It is therefore ideal for anomaly detection.

The hypersphere is determined by the radius R and the center a, as illustrated in Fig. 1, and is found by solving the optimisation problem of minimising the error on the normal class and the chance of misclassifying data from the abnormal class.

The error on the normal class is minimised by adjusting R and a in a way that all instances of the training data set are contained in the hypersphere. Minimising the chance of misclassifying data from the abnormal class is done by minimising the hypersphere's volume. The trade-off F between the number of misclassified normal instances and the volume of the normal region is optimised by minimising

$$F(R, a) = R^2 \tag{1}$$

subject to

$$\|x_i - a\|^2 \leq R^2 \quad \forall i \qquad i = 1, .., M \tag{2}$$

where x_i denotes the instances and M the number of instances in the training data set, a is the hypersphere's center, and $\|x_i - a\|$ is the distance between x_i and a.

The hypersphere is described by selected instances from the training data set, so-called support vectors. The center a is implicitly described by a linear combination of the support vectors. The remaining instances are discarded.

If all instances are contained in the hypersphere, outliers contained in the training data set massively influence the decision boundary, which is not desired. Slack variables ξ_i are introduced, which allow for some instances x_i in the training data set to be outside the hypersphere. The parameter C is introduced controlling the influence of the slack variables and thereby the error on the normal class and the hypersphere's volume. So the optimisation problem of eq. (1) and eq. (2) changes into minimising

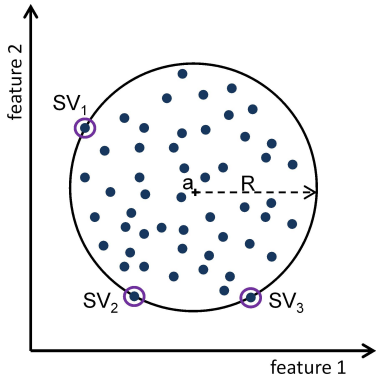

Fig. 1. A hypersphere in a 2-dimensional feature space with radius R and center a is described by the three support vectors $SV_1 \dots SV_3$.

$$F(R, a, \xi_i) = R^2 + C \sum_{i=1}^{M} \xi_i \tag{3}$$

subject to

$$\|x_i - a\|^2 \leq R^2 + \xi_i \quad \forall i \quad \text{and} \quad \xi_i \geq 0 \quad \forall i \tag{4}$$

As described in [10], the constrained optimisation problem is transformed into an unconstrained one by integrating the constraints into the equation using the method of Lagrange [4]. The partial derivatives w.r.t. R, a, ξ are set to 0 and the resulting equations are resubstituted, yielding the following optimisation problem to be maximised:

$$L(\alpha) = \sum_{i=1}^{M} \alpha_i (x_i \cdot x_i) - \sum_{i,j=1}^{M} \alpha_i \alpha_j (x_i \cdot x_j) \tag{5}$$

subject to

$$0 \leq \alpha_i \leq C \quad \forall i \tag{6}$$

Finally, since strictly spherical-shaped decision boundaries are not appropriate for most data sets, non-spherical decision boundaries are introduced by mapping the data into a higher-dimensional space by the so-called kernel trick [11].

As indicated by eq. (5), x_i and x_j are solely incorporated as the inner products $(x_i \cdot x_i)$ and $(x_i \cdot x_j)$ respectively. Instead of actually mapping each instance to a higher-dimensional space using a mapping function $\phi()$, the so-called kernel trick is used to replace the inner products $(\phi(x_i) \cdot \phi(x_j))$ by a kernel function $K(x_i, x_j)$. The radial basis function (RBF) kernel is used, because it is reported to be most suitable to be used with SVDD in [10]. The RBF kernel is given by

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{\sigma^2}} \tag{7}$$

Incorporating the RBF kernel eq. (5) becomes:

$$L(\alpha) = 1 - \sum_{i,j=1}^{M} \alpha_i \alpha_j K(x_i, x_j) \tag{8}$$

A major challenge in one-class classification problems is having to adjust parameters, in this case the parameters C and σ. The approach proposed in [13] was used to solve this problem.

4 Enhancing SVDD to multivariate time series

Based on SVDD, in this section an enhancement is shown, that makes SVDD applicable to multivariate time series. The approach was proposed by this paper's author in [14].

4.1 Transforming time series to feature vectors

A recording contains multiple time-stamped signals, i.e. it corresponds to multivariate time series data [5]. Transforming the multivariate time series to feature vectors is done by transforming the values at each time point T_i to one feature vector x_i. Thereby, a $N \times M$ multivariate time series is transformed to N feature vectors of length M.

4.2 Working with subsequences

Time series data from from technical systems can be considered noisy. As a consequence, it is very likely that a fraction of individual data points of previously unseen data lies outside the decision boundary without actually being abnormal, which is confirmed by experiments on the recordings from vehicles. Instead of classifying feature vectors, subsequences in the original time series are formed using a fixed-width non-overlapping window of length W. While this approach ignores the order of the subsequences, it takes into account the local neighbourhood of the data points.

In order to classify subsequences, a distance measure for the subsequences has to be defined. Informally spoken, the distance measure should yield a big distance for a subsequence if many data points lie outside the decision boundary or if few data points lie far outside the decision boundary.

As a first step, for every feature vector x_{t_k}, the distance to the center is calculated by

$$dist_{x_{t_k}} = \|x_{t_k} - a\| \tag{9}$$

which is squared to be able to apply the RBF kernel

$$dist^2_{x_{t_k}} = \|x_{t_k} - a\|^2 \tag{10}$$

Solving the binomial, replacing a by its linear combination of support vectors, and replacing the inner products by the RBF kernel function yields:

$$dist^2_{x_{t_k}} = 1 - 2\sum_{i=1}^{M} \alpha_i K(x_k, x_i) + \sum_{i,j=1}^{M} \alpha_i \alpha_j K(x_i, x_j) \tag{11}$$

The distance of a subsequence is now calculated by averaging the distances of the window's feature vectors.

$$dist_{subseq} = \frac{1}{W} \sum_{k=1}^{W} dist_{x_{t_k}} \tag{12}$$

The proposed measure does not indicate the distance between two arbitrary subsequences, but indicates how abnormal a subsequence is. The formation of a subsequence is illustrated in Fig. 2 for a contrived multivariate time series containing two univariate time series.

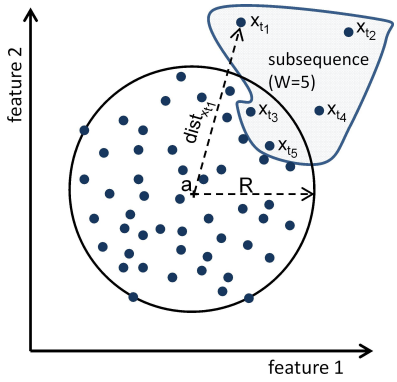

Fig. 2. Subsequence with window length 5 formed from a multivariate time series with M = 2. The highlighted feature vectors $x_{t_1} \ldots x_{t_5}$ belong to one subsequence.

4.3 Determining the classification threshold

Being able to calculate distances for subsequences allows to classify them. The procedure during training is as follows:

1. train SVDD with feature vectors in training set
2. calculate the distances $dist_{x_{t_k}}$ of the feature vectors

3. form subsequences of length W
4. calculcate $dist_{subseq}$ for all subsequences
5. from all $dist_{subseq}$ determine a threshold thr_{subseq}

A first approach to determine the threshold thr_{subseq} could be to use the maximum distance in the training set as the threshold for classifying subsequences. However, this is highly sensitive to outliers in the training set since the threshold would be determined solely by the most distant subsequence.

It is proposed to not necessarily include all subsequences in the determination of the threshold, and thereby be robust against outliers. The threshold is determined using box plots known from statistics (see e.g. [7]). For a box plot the first and the third quartile (Q_1 and Q_3) of the data are calculated. The margin between Q_1 and Q_3 is referred to as the inter-quartile range, which holds 50% of the data. Based on the inter-quartile range, the so-called whiskers are calculated by $Q_3 + 1.5(Q_3 - Q_1)$ and $Q_1 - 1.5(Q_3 - Q_1)$. The data points outside the whiskers are regarded as outliers.

In this work, outlier distances are the ones that are greater than the upper whisker. Those distances are discarded according to

$$dist_{outlier} > 1.5(Q_3 - Q_1) + Q3 \tag{13}$$

The maximum of the remaining distances is used as the threshold for classification.

5 Experimental results

The approach was validated on data sets from a real vehicle. Test drives were conducted in different traffic situations ranging from urban traffic to motorways over a time span of one year to capture recordings from different weather conditions. Different types of faults were injected into the vehicle. Preliminary results were previously presented in [14].

It is recommended to partition the data according to the vehicle's subsystems like e.g. the engine control. In a first step, the relevant signals were selected using visual analytics as proposed by this paper's author in [12]. Eight signals were taken into account, e.g. engine rpm, vehicle speed, ignition timing advance, and the throttle position. The majority of signals on an in-vehicle network are transmitted in a cyclic manner with typical cycle times in the range of 20ms to 1s. In a pre-processing step, the data was resampled to a sample rate of 1s.

In the absence of abnormal data it is recommended to start by testing with normal data. If the number of false negatives (FN), i.e. falsely detected anomalies, is too high, the detection system will not be useful.

The size of the training set was varied with a fixed test set size as shown in Fig. 3. While for very small training sets the number of false negatives acts non-deterministically between very low and very high values, for larger training sets, the training set becomes more representative and the number of false negatives

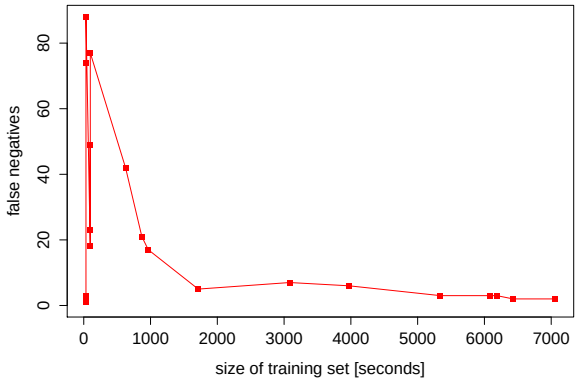

Fig. 3. False negatives for varied size of training set and fixed size of test set (2423 seconds) with normal data only.

stabilises at low values. This type of experiment can be used as an indicator of how representative the training set is.

The approach is now tested with recordings from different driving conditions. The size of the test sets used for the experiments is relatively small due to the availability of labelled data. Since classification involves basic vector algebra in combination with the RBF kernel it is rather fast. Even for large vehicle fleets, classification is orders of magnitude faster than the production of the data. So test data of the size that one would refer to as "big data" can easily be classified sufficiently fast.

As a first step, the system was trained and tested with recordings from one driving condition, i.e. motorway, overland, or urban traffic. The results are given in the first three rows in Table 1.

As can be seen, between 42.9% and 76.9% of the faults were detected (TNR). For the subsequent experiments the system was trained on the recordings from all driving conditions. The last four rows in Table 1 show the results for these experiments. With the combined training set, the number of falsely reported anomalies per hour has significantly decreased compared to the experiments with individual training sets shown in the first three rows, since the training set has become more representative.

In Fig. 4 the results on a one hour test drive are shown, where the spark plug lead was temporarily disconnected while the vehicle was standing still. From the 10 injected faults, 8 were detected. None of the signals is out of the valid value range, the faults were detected solely due to violations of learnt relationships between signals. No anomalies were falsely reported for this recording.

The percentage of detected anomalies is reasonably high, taking into account that classification is done solely on the information of the normal class. The percentage of correctly detected anomalies in the set of reported anomalies is

training,test	training set	test set	FN/h	TN	TNR	precision
motorway,motorway	20843s	4845s	6.7	9	42.9%	50.0%
overland,overland	24604s	12076s	4.8	31	73.8%	66.0%
urban,urban	21336s	7224s	10.5	10	76.9%	32.3%
all,motorway	63631s	4845s	0.0	10	47.6%	100%
all,overland	63631s	12076s	3.0	27	64.3%	73.0%
all,urban	63631s	7224s	2.0	9	69.2%	69.2%
all,all	63631s	24145s	2.1	45	59.2%	76.3%

Table 1. Results for motorway, overland, and urban test drives. (FN/h: falsely reported anomalies per hour test drive, TN: correctly detected anomalies, TNR: true negative rate, precision: percentage of correctly detected anomalies in the set of reported anomalies)

high as well, which means that a domain-expert analysing the output of the classification system will not have to spend a large amount of time for fault analysis of reported anomalies that turn out to be normal occurrences.

6 Conclusion

This paper addressed the problem of having to cope with big data resulting from vehicle tests. The aim was to report potential errors in the recordings. The key point was to be able to detect unexpected faults without modelling effort. This was achieved by learning from a training set of error-free recordings, and then autonomously reporting deviations in the test set as anomalies.

The one-class support vector machine SVDD was enhanced to work on multivariate time series data and the effectiveness of the approach was shown on real data sets.

The proposed methodology could be applicable to multivariate time series from other domains as well, e.g. industrial plants or network traffic with fixed communication patterns.

References

1. Shigeo Abe. Support Vector Machines for Pattern Classification (Advances in Pattern Recognition). 2010. Springer-Verlag London Ltd.
2. Fengyu Cong, Hannu Hautakangas, Jukka Nieminen, Oleksiy Mazhelis, Mikko Perttunen, Jukka Riekki and Tapani Ristaniemi. Applying Wavelet Packet Decomposition and One-Class Support Vector Machine on Vehicle Acceleration Traces for Road Anomaly Detection. Advances in Neural Networks 2013
3. Kan Deng, Andrew Moore and Michael Nechyba. Learning to Recognize Time Series: Combining ARMA models with Memory-based Learning. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation. 1997.
4. Chris A. Jones. Lecture Notes: MATH2640 Introduction to Optimisation 4. 2005. University of Leeds, School of Mathematics

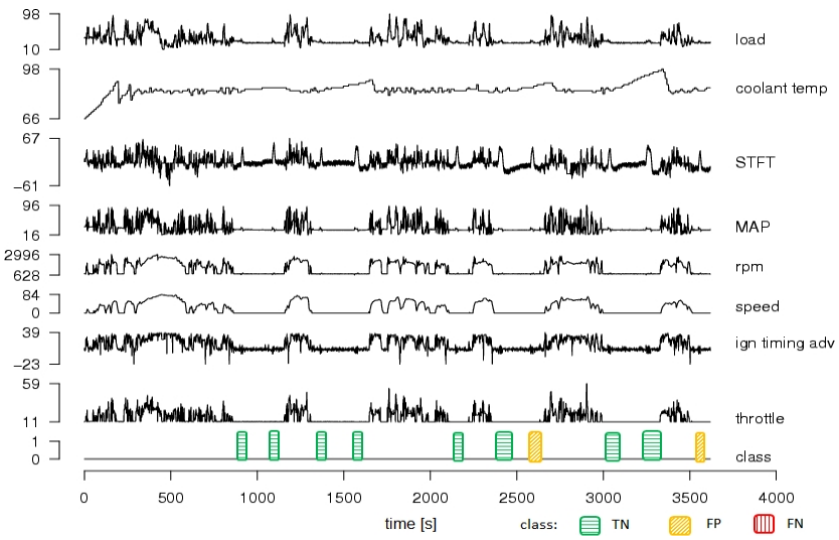

Fig. 4. Classification results for an overland drive of one hour where 10 faults were injected by temporarily interrupting the spark plug lead. The results are marked with frames (TN: green, FP: yellow) in the "class" row.

5. Theophano Mitsa. Temporal Data Mining. 2010. Chapman & Hall/CRC
6. Michael Mueter and Naim Asaj. Entropy-based anomaly detection for in-vehicle networks. IEEE Intelligent Vehicles Symposium. 2011
7. T. Raykov and G.A. Marcoulides Basic Statistics: An Introduction with R. 2012. Rowman & Littlefield Publishers
8. J. Suwatthikul and R. McMurran and R.P. Jones. In-vehicle network level fault diagnostics using fuzzy inference systems. Applied Soft Computing. vol. 11, nr. 4 2011
9. David M. Tax and Robert Duin. Data domain description using support vectors. Proceedings of the European Symposium on Artificial Neural Networks. 1999. pages 251–256
10. David M. Tax and Robert Duin. Support Vector Data Description. Machine Learning. January 2004. volume 54. pages 45–66. Kluwer Academic Publishers, MA, USA
11. Sergios Theodoridis and Konstantinos Koutroumbas. Pattern Recognition, Fourth Edition. 2009. Academic Press
12. Andreas Theissler, Daniel Ulmer and Ian Dear. Interactive Knowledge Discovery in recordings from vehicle tests. In Proc. 33rd FISITA World Automotive Congress. 2010
13. Andreas Theissler and Ian Dear. Autonomously determining the parameters for SVDD with RBF kernel from a one-class training set. In Proc. of the WASET International Conference on Machine Intelligence. 2013
14. Andreas Theissler and Ian Dear. An anomaly detection approach to detect unexpected faults in recordings from test drives. In Proc. of the WASET International Conference on Vehicular Electronics and Safety. 2013