An Anomaly Detection Approach to Detect Unexpected Faults in Recordings from Test Drives

Andreas Theissler, Ian Dear

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. The communication on the in-vehicle network, data from external sensors, or internal data from the electronic control units is recorded by automotive data loggers during the test drives. The recordings are used for fault analysis. Since the resulting data volume is tremendous, manually analysing each recording in great detail is not feasible.

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 available recordings, i.e. a training set, and then to autonomously detect unexpected deviations and report them as anomalies.

The one-class support vector machine “support vector data description” is utilised to calculate distances of feature vectors. SVDDsubseq is proposed as a novel approach, allowing to classify subsequences in multivariate time series data. The approach allows to detect unexpected faults without modelling effort as is shown with experimental results on recordings from test drives.

Keywords—Anomaly detection, fault detection, test drive analysis, machine learning.

I. Introduction

EVEN though various test phases are conducted for each electronic control unit (ECU) and for each vehicle function [1], the integration of all ECUs inside a vehicle, with real sensors, actuators and the real in-vehicle network, is challenging – often unexpected problems occur. Hence, the conduction of test drives is inevitable.

The importance of test drives as a measure for quality assurance is widely accepted, in [2] a vehicle manufacturer stated, that before one of its premium class models came to market in 2009, 34 million kilometres of test drives were conducted.

The in-vehicle network communication, data from external sensors, or internal ECU data is recorded by automotive data loggers during the test drives. During fault analysis the recordings allow to reconstruct the situation the vehicle was in, e.g. steering manoeuvres, the vehicle’s velocity, its yaw rate or the battery voltage can be determined from the data. 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 authors identified the following approaches to be currently followed by vehicle manufacturers in order to detect abnormal behaviour occurring in test drives:

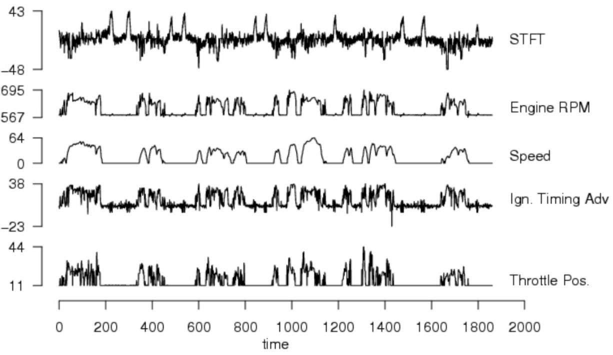

Fig. 1. Excerpt of a recording of a test drive with faults, showing five signals over 2000 seconds.

1) Illegal constellations of the signals are pre-configured and the measured data is observed during the test drive.
2) Diagnostic testers are used to read out the ECUs’ diagnostic trouble codes [3].
3) After the test drive, the recordings are searched for known, pre-configured fault patterns.
4) A test driver manually flags erroneous vehicle behaviour during the test drive.
5) A fraction of the recordings is manually investigated by an expert after the test drive by random inspection.

The first three points rely on pre-configured knowledge, meaning that only those faults are detected, that were thought of during the definition of the test strategy. The fourth and fifth point solely rely on the experience of human beings. None of the listed measures handles unexpected faults.

A. Motivation

A vehicle is a high-technology product with a high fraction of vehicle electronics. The gross of innovations is achieved by the means of vehicle electronics. Only by effective means of analysing the recordings, can one make sure that the effort put in the conduction of test drives pays off.

The amount of data to be analysed is huge, depending on the logging device a recording can contain hundreds of signals. The chance to manually find unexpected or unmodelled faults is low. An example of a recording of a test drive with faults is shown in Fig. 1. Visually detecting the errors is almost infeasible.

This paper contributes by proposing an approach that

- uses available recordings and extracts the relevant knowledge to be able to find abnormal deviations in new recordings

Andreas Theissler is with IT-Designers, Germany and Brunel University, West London, UK

Ian Dear is with Brunel University, West London, UK

- autonomously points the expert to potential errors by reporting anomalies in recordings from test drives

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

II. From fault detection to anomaly detection

Detecting unmodelled or unexpected faults in the recordings is in this paper done by means of anomaly detection. In this paper, the terms *fault*, *error*, and *failure* are used as defined by ISO 26262-1 [4]. A *fault* is an "abnormal condition that can cause an element or an item to fail". An *error* is the "discrepancy between a computed, observed or measured value or condition, and the true, specified or theoretically correct value or condition" [4]. A *failure* is the "termination of the ability of an element, to perform a function as required" [4]. A fault may manifest itself as an error, which in turn may cause a failure [5].

Fault detection is one part of fault diagnosis which comprises fault detection and fault isolation [6]. The authors in [7] subdivide fault locations in vehicle electronics into sensors/actuators, electric, buses, and ECUs.

In [8] two important properties of a fault diagnostic system are identified: the modelling effort should be as minimal as possible, and the detection system should be able to identify novelties, i.e. faults, that were not modelled. It is further stated that the identification of novelties is challenging, because typically no data set exists, that includes all faults in order to fully model the abnormal region. On the other hand, data sets with normal behaviour are usually available [8].

These assessments substantiate using anomaly detection based on a training set, as done in this paper.

In [4] the term anomaly is defined as a "condition that deviates from expectations, based, for example, on requirements, specifications, design documents, user documents, standards, or on experience". [9] defines an anomaly more general as a deviation from expected behaviour, other terms are novelty or outlier [10]. An anomaly is considered a potential error in this paper.

The detection of 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. The decision boundary is heavily influenced by the choice of the abnormal data. 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. In other words, the training period is exclusively conducted using a training set of normal instances.

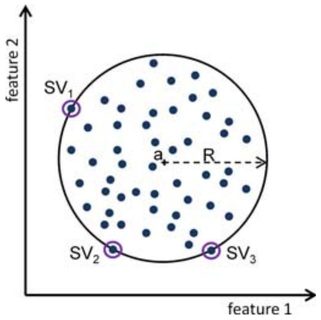

Fig. 2. A hypersphere in a 2-dimensional feature space with radius R and center $\vec{a}$ is described by the three support vectors $SV_1 \dots SV_3$.

Support vector machines (SVM)[11], [12] have shown to yield good results on classification tasks and have widely been used. In [13] 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. In [14], the authors of this paper have applied SVDD to analysing data from a DC motor.

III. The one-class support vector machine SVDD

In [13] the one-class support vector machine "support vector data description" (SVDD) was introduced. As a decision function, SVDD forms a hypersphere around the normal instances in the training data set. The hypersphere is determined by the radius R and the center $\vec{a}$, as illustrated in Fig. 2, 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 $\vec{a}$ 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

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

subject to

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

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

The hypersphere is described by selected instances from the training data set, so-called support vectors. The center $\vec{a}$ 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. So SVDD in this form is very sensitive to outliers, which is not desired. Slack variables ξ_i are introduced, which allow for some instances $\vec{x_i}$ 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

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

subject to

$$\|\vec{x_i} - \vec{a}\|^2 \leq R^2 + \xi_i \quad \forall i \tag{4}$$

and

$$\xi_i \geq 0 \quad \forall i \tag{5}$$

As described in [15], the constrained optimisation problem is transformed into an unconstrained one by integrating the constraints into the equation using the method of Lagrange [16]. The partial derivatives w.r.t. R, $\vec{a}$, $\vec{\xi}$ are set to 0 and the resulting equations are resubstituted, yielding the following optimisation problem to be maximised:

$$L(\vec{\alpha}) = \sum_{i=1}^{M} \alpha_i (\vec{x_i} \cdot \vec{x_i}) - \sum_{i,j=1}^{M} \alpha_i \alpha_j (\vec{x_i} \cdot \vec{x_j}) \tag{6}$$

subject to

$$0 \leq \alpha_i \leq C \quad \forall i \tag{7}$$

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. (6), $\vec{x_i}$ and $\vec{x_j}$ are solely incorporated as the inner products $(\vec{x_i} \cdot \vec{x_i})$ and $(\vec{x_i} \cdot \vec{x_j})$ 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 $(\phi(\vec{x_i}) \cdot \phi(\vec{x_j}))$ by a kernel function $K(\vec{x_i}, \vec{x_j})$. The mapping is implicitly done by solving $K(\vec{x_i}, \vec{x_j})$. So eq. (6) becomes:

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

The radial basis function (RBF) kernel is used, because it is reported to be most suitable to be used with SVDD in [17]. The RBF kernel is given by

$$K(\vec{x_i}, \vec{x_j}) = e^{-\frac{\|\vec{x_i} - \vec{x_j}\|^2}{\sigma^2}} \tag{9}$$

The kernel function can take on values from the interval $(0, 1]$ and converges to 0 for high distances $\|\vec{x_i} - \vec{x_j}\|$. Since $K(\vec{x_i}, \vec{x_i}) = e^{-\frac{\|\vec{x_i} - \vec{x_i}\|^2}{\sigma^2}} = 1$ and $\sum_{i=1}^{M} \alpha_i = 1$, eq. (8) can be simplified for the RBF kernel:

$$L(\vec{\alpha}) = 1 - \sum_{i,j=1}^{M} \alpha_i \alpha_j K(\vec{x_i} \cdot \vec{x_j}) \tag{10}$$

subject to

$$0 \leq \alpha_i \leq C \quad \forall i \tag{11}$$

An approach to autonomously tune the parameters C and σ in the absence of faults in the training set was proposed by this paper's authors in [18].

IV. Enhancing SVDD to multivariate time series

Based on SVDD, in this section a novel approach is proposed, that enhances SVDD to work with multivariate time series. The approach shall be named SVDDsubseq.

A. Feature extraction

A recording is a list of time-stamped signal values recorded during tests. A recording may contain multiple signals, i.e. it corresponds to multivariate time series data [19].

SVDD is trained with instances in feature space, i.e. feature vectors. So from the time series, features need to be extracted. Transforming the multivariate time series to feature vectors is conducted by transforming the values at each time point T_i to one feature vector $\vec{x_i}$. Thereby, a $N \times M$ multivariate time series

$$Y_T = \begin{pmatrix} x_{1,t_1} & x_{1,t_2} & \dots & x_{1,t_N} \\ x_{2,t_1} & \dots & \dots & x_{2,t_N} \\ \dots & \dots & \dots & \dots \\ x_{M,t_1} & x_{M,t_2} & \dots & x_{M,t_N} \end{pmatrix} \tag{12}$$

is transformed to N feature vectors of length M

$$\vec{x_i} = (x_{1,t_i}, \ x_{2,t_i}, \ \dots \ , \ x_{M,t_i}) \quad \forall i \tag{13}$$

B. Forming subsequences

Since time series data from vehicles are recordings from technical systems, the time series data can be considered noisy. Measuring identical situations, it is likely to observe similar but not identical values. As a consequence, it is very likely that a fraction of individual data points of previously unseen data lie outside the decision boundary without actually being abnormal, which is confirmed by experiments.

The aim of this work is pointing domain-experts to abnormal subsequences in the recordings. Instead of classifying feature vectors, subsequences $Y_{t_j \dots t_{(j+W-1)}}$ in the original time series are formed using a fixed-width non-overlapping window of length W.

$$Y_{t_j \dots t_{(j+W-1)}} = \begin{pmatrix} x_{1,t_j} & \dots & x_{1,t_{j+W-1}} \\ \dots & \dots & \dots \\ x_{M,t_j} & \dots & x_{M,t_{j+W-1}} \end{pmatrix} \tag{14}$$

Working with feature vectors, the order of the data is ignored. In other words, shuffling the feature vectors (the columns in (12)) prior to applying SVDD yields the same results. By combining neighbouring values to subsequences, the local order of the data is taken into account.

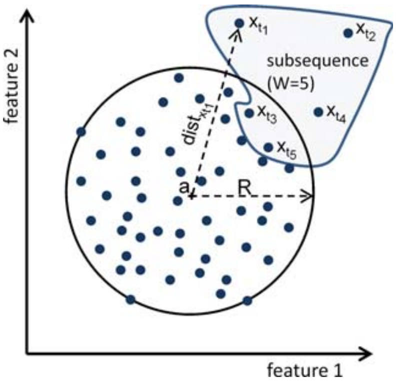

Fig. 3. Subsequence with window length 5 formed from a multivariate time series with M = 2. The highlighted feature vectors $x_{t_1} \ldots x_{t_5}$ belong to one subsequence.

C. Assigning distances to subsequences

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 $\vec{x_{t_k}}$, the distance to the center is calculated by

$$dist_{\vec{x_{t_k}}} = \|\vec{x_{t_k}} - \vec{a}\| \tag{15}$$

which is squared to be able to apply the RBF kernel

$$dist^2_{\vec{x_{t_k}}} = \|\vec{x_{t_k}} - \vec{a}\|^2 \tag{16}$$

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

$$dist^2_{\vec{x_{t_k}}} = 1 - 2\sum_{i=1}^{M} \alpha_i K(\vec{x_k}, \vec{x_i}) + \sum_{i,j=1}^{M} \alpha_i \alpha_j K(\vec{x_i}, \vec{x_j}) \tag{17}$$

As can be seen from eq. (17), classification is very fast. It involves basic vector algebra and the application of the kernel function. Based on this distance measure, a distance is assigned to each subsequence. The distance of a subsequence is calculated by averaging the distances of the window's feature vectors.

$$dist_{subseq} = \frac{1}{W} \sum_{k=1}^{W} dist_{\vec{x_{t_k}}} \tag{18}$$

The proposed distance 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. 3 for a contrived multivariate time series containing two univariate time series.

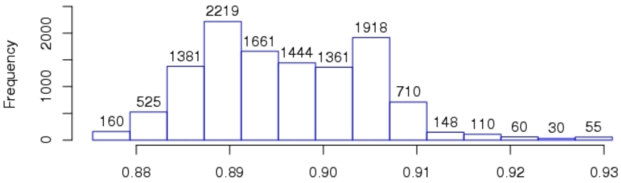

Fig. 4. Histogram of subsequence distances in a training set from recordings of test drives.

D. Training and test

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 $dist_{\vec{x_{t_k}}}$ of the feature vectors
3) form subsequences according to (14)
4) calculcate $dist_{subseq}$ for all subsequences as given by (18)
5) from all $dist_{subseq}$ determine the 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 distances of all subsequences in the training set are calculated and those that are considered outliers are not used to determine the threshold.

The outliers in the training set could be identified based on the statistical distribution of the distances by cutting off at the upper tail of the distribution. Experiments have shown that the distribution does not correspond to a normal distribution (see Fig. 4), so the type of the distribution has to be determined, the parameters have to be estimated and a threshold has to be specified.

It is proposed to use a more applicable way of determining the threshold using box plots known from statistics (see e.g. [20]). 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 and corresponds to the box in Fig. 5. 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 outside the whiskers are regarded as outliers. This has been successfully applied on real-world data in [21] to identify outliers in medical data.

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

$$dist_{outlier} > 1.5(Q_3 - Q_1) + Q3 \tag{19}$$

The maximum of the remaining distances is used as the threshold for classification. Fig. 5 shows the box plot for the distances shown in Fig. 4, where the distances above the upper

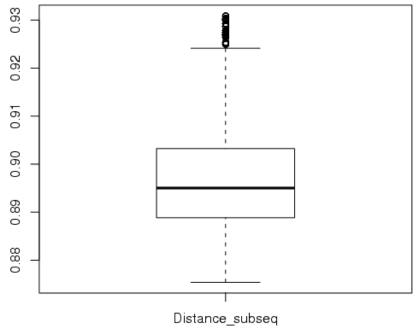

Fig. 5. Box plot of subsequence distances in training set. The distances above the upper horizontal line are regarded as outliers and are not included for the determination of the threshold.

horizontal line (whisker) are discarded resulting in a threshold of 0.9214 for the investigated test drives.

Testing instances from a test set works by applying the threshold determined during training:

1) calculate the distances $dist_{\vec{x_{i_k}}}$ of the feature vectors in test set
2) form subsequences according to (14)
3) calculcate $dist_{subseq}$ for all subsequences
4) classify subsequences as abnormal if $dist_{subseq} > thr_{subseq}$

Based on (18), an anomaly score ϵ for subsequences s_i is defined as

$$\epsilon_{s_i} = \begin{cases} dist_{subseq_{s_i}} - thr_{subseq} : \text{if } s_i \text{ is abnormal} \\ 0 : \text{if } s_i \text{ is normal} \end{cases}$$

This anomaly score allows to rank the reported anomalies.

E. Determining the classification results

In the test set, consecutive fixed-length subsequences with the same label are grouped together as variable-length subsequences, $s_{lab_{\omega_a}}$ for abnormal subsequences and $s_{lab_{\omega_n}}$ for normal ones respectively.

The aim is to point the expert to abnormal subsequences $s_{lab_{\omega_a}}$. If some or all feature vectors contained in $s_{lab_{\omega_a}}$ are reported as abnormal, the anomaly is detected, i.e. the system detected one true negative.

The classification results are determined as follows, where $s_{class_{\omega_a}}$ is a fixed-length subsequence classified as abnormal and $s_{class_{\omega_n}}$ a subsequence classified as normal.

- true negative (TN), i.e. anomaly detected:
  if $s_{lab_{\omega_a}}$ contains at least one $s_{class_{\omega_a}}$
- false positive (FP), i.e. anomaly not detected:
  if $s_{lab_{\omega_a}}$ does not contain a $s_{class_{\omega_a}}$
- false negative (FN), i.e. falsely reported as anomaly:
  for each $s_{class_{\omega_a}}$ contained in a $s_{lab_{\omega_n}}$
- true positive (TP):
  for each group of consecutive $s_{class_{\omega_n}}$ contained in $s_{lab_{\omega_n}}$

TABLE I
VEHICLES USED FOR THE EXPERIMENTS: FOUR DIFFERENT RENAULT TWINGOS.

Id	Date of manufacture	Engine displacement	Engine power	Weight
Tw1	2002	1149 ccm	43 kW	895 kg
Tw2	2002	1149 ccm	43 kW	895 kg
Tw3	2011	1149 ccm	55 kW	1019 kg
Tw4	2012	1149 ccm	55 kW	994 kg

TABLE II
OBD-SIGNALS RECORDED FOR THE EXPERIMENTS.

Signal	OBD PID
Short Term Fuel Trim (Bank 1)	06 hex
Engine RPM	0C hex
Vehicle Speed	0D hex
Ignition Timing Advance (Cylinder 1)	0E hex
Absolute Throttle Position	11 hex

V. EXPERIMENTAL RESULTS

The approach was validated on real data sets from vehicles. Over 250 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. This data acquisition phase started with one vehicle and one driver and was later extended to ten drivers and four vehicles (see Table I) to become more representative. The reason for choosing “Renault Twingo” as test vehicles was just the availability. The results are transferable to other vehicles as well.

The data was recorded using the on-board diagnostics (OBD-II or EOBD) interface, which allows to approximately read 10-15 emission-related signals in a standardised way. The signals in Table II were analysed.

Recordings from in-vehicle networks are highly confidential and could therefore not be used. However, recordings from in-vehicle networks were available to the author but not for publishing purposes. So the authors could assure that the data is comparable, has a lower sample rate though. The results presented here on data from OBD apply to recordings from an in-vehicle network as well.

The first experiments were conducted with the vehicle in idle mode to gain confidence in the approach. Then recordings from test drives were used.

A. Vehicle in idle mode

The first experiment was conducted with recordings of the vehicle in idle mode. The signals speed and throttle position were discarded for this experiment because they remain unchanged. The vehicle “Tw1” was used as the test vehicle and the following errors were provoked:

In order to simulate a cable break in the spark plug lead or a damaged spark plug, the spark plug lead was removed several times. One recording is shown in Fig. 6.

- The sensor measuring the engine coolant temperature was manipulated. A negative temperature coefficient (NTC)

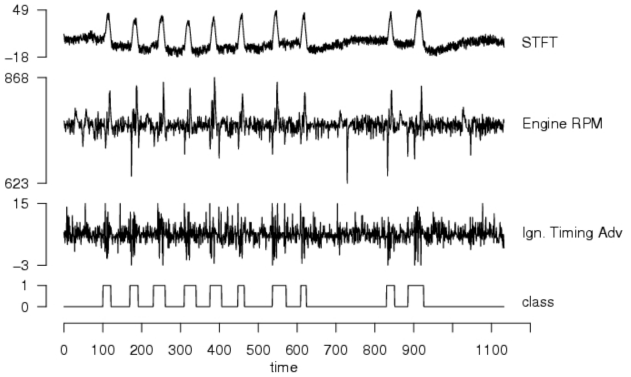

Fig. 6. Recordings of vehicle in idle mode, with 10 injected faults (class=1).

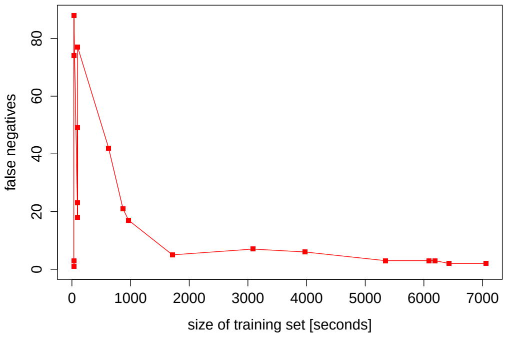

Fig. 7. False negatives for varied size of training set and fixed size of test set (2423 seconds) with normal data only.

thermistor is used as the sensor in the test vehicles, i.e. low resistance corresponds to high temperature [22]. By means of a potentiometer, an erroneous sensor, a loose contact, and a short circuit were simulated.

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

To investigate the affect of the size of the training set, the size was varied with a fixed test set size as shown in Fig. 7. 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 stabilises at low values. This type of experiment can be used as an indicator of how representative the training set is.

A test set with faults was used in Fig. 8. It can be seen that, as the ratio between the size of the training set and the size of the test set increases, the false negative rate and the false positive rate decreases.

After the initial experiments, the maximal training set from Fig. 8 was selected and tested with a test set containing faults. As shown in Table III, the results are very good, 92.9% of the faults were detected. This was expected, as the faults could easily be visually identified in the plots in Fig. 6.

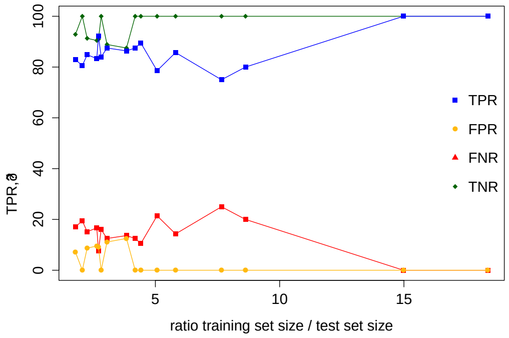

Fig. 8. TPR, FPR, FNR, TNR w.r.t. the ratio between the size of the training set and the size of the test set.

TABLE III
RESULTS ON RECORDINGS FROM VEHICLE IN IDLE MODE WITH FAULTS.

index	training	test	TN FN	prec
0	7056 s vehicle: Tw1	3958 s vehicle: Tw1	26 (92.9%) 7	78.8%

B. Test drives

For the further experiments recordings from test drives were used. As a next step, the affect of different drivers and different vehicles were investigated.

In Table IV the number of false negatives are shown for different constitutions of fault-free test sets with training on recordings of approximately 6 hours from various test drives with vehicle “Tw1” and driver “dr1”.

The first row indicates, that testing on test drives from the same vehicle and the same driver yields good results. 10 falsely detected anomalies in recordings of approx. 3 hours is viewed as acceptable.

The affect of different drivers is shown in the second row, training on recordings from just one driver and testing on recordings from different drivers yields very poor results as indicated by the 84 false negatives.

The affect of different vehicles of the same model series is not as dramatic as shown in the third row.

Conclusively, an ideal training set should

1) be large enough to contain common driving situations and conditions, and different traffic situations
2) contain recordings from different test drivers
3) contain recordings from various vehicles

What is encountered in practice is that different drivers test different vehicles.

For different training sets, the accuracy was evaluated for one test drive with vehicle “Tw1”, driver “dr1”, and 10 faults, shown in Fig. 9. The following constitutions of the training set were used:

same vehicle and same driver (index 4 in Table V)
- same vehicle and various drivers including “dr1” (index 5 in Table V)

TABLE IV
RESULTS ON RECORDINGS FROM TEST DRIVES IN FAULT-FREE OPERATION MODE.

index	training	test	FN
1	22664 s vehicle: Tw1 driver: dr1	10242 s vehicle: Tw1 driver: dr1	10
2	22664 s vehicle: Tw1 driver: dr1	10981 s vehicle: Tw1 drivers: dr2-9	84
3	22664 s vehicle: Tw1 driver: dr1	10933 s vehicle: Tw2 driver: dr1	17

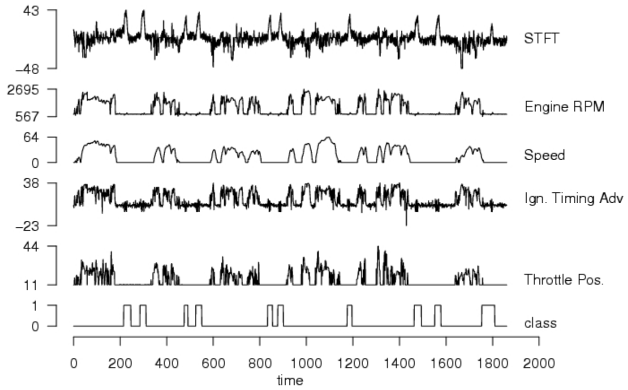

Fig. 9. Recording of a test drive with vehicle "Tw1" and driver "dr1". The 10 injected faults are difficult to manually detect, the detection system detected between 9 and 10.

- various vehicles including "Tw1" and different drivers not including "dr1" (index 6 in Table V)

All three scenarios show very good results as shown in Table V, between 90% and 100% of the faults were detected. This indicates that the proposed approach is applicable as the integral part of an anomaly detection system for test drive recordings.

In addition to the detection algorithm, data selection and pre-processing facilities, as well as intelligent ways to analyse the reported anomalies are required, which has not been discussed in this paper. The authors discussed efficient ways of user-driven data exploration in [23].

TABLE V
RESULTS ON RECORDINGS FROM TEST DRIVES WITH FAULTS (SEE FIG. 9).

index	training	test	TN FN	prec
4	22664 s vehicle: Tw1 driver: dr1	1997 s vehicle: Tw1 driver: dr1	9 (90%) 1	90.0%
5	11975 s vehicle: Tw1 drivers: dr1-9	1997 s vehicle: Tw1 driver: dr1	10 (100%) 1	90.9%
6	19443 s vehicles: Tw1-4 drivers: dr2-10	1997 s vehicle: Tw1 driver: dr1	10 (100%) 1	90.9%

VI. CONCLUSION

The paper addressed the problem of having to cope with huge data volumes 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 classification technique SVDD was enhanced to work on multivariate time series data and the approach was shown to work with the help of experimental results. Even if misclassifications occur, which is inevitable for classification systems based on learning from sample data, the detection system is very useful. Based on the reported anomalies the expert can conduct the analysis in a goal-oriented manner in contrast to random inspection.

The proposed approach offers benefits in many steps of a vehicle's life cycle ranging from the development phase to the after sales service period. During test drives conducted before start of production the vehicle's behaviour on the road is evaluated. Utilising the proposed approach, the analysis becomes more thorough with a smaller chance of overseeing abnormal behaviour. After start of production sporadic test drives are being conducted with chosen vehicles to ensure the vehicles' quality. At that point in time many recordings with that type of vehicle exist from earlier phases. Therefore the system shall be able to offer good results for this step of the vehicle's life cycle. Even after a vehicle has gone through all of the manufacturer's steps, such a system can offer benefits, during the analysis of field data.

The approach was developed and successfully used in the authors' research project on detecting anomalies in recordings from test drives [14].

REFERENCES

[1] K. Lamberg, "Model-based testing of automotive electronics," *Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, p. 28, 2006.

[2] M. Krämer, A. Röhringer, W. Kirchner, T. Vogel, and J. Rochlitzer, "Programme Management and Project Control," *ATZ extra. The New E-Class by Mercedes-Benz*, 2009.

[3] C. Marscholik and P. Subke, *Road vehicles – Diagnostic communication.* Hüthig GmbH und Co. KG, 2008.

[4] "ISO 26262: Road vehicles – Functional safety – Part 1: Vocabulary. Final Draft." 2011.

[5] I. Sommerville, *Software Engineering (6th Edition).* Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2001.

[6] R. Isermann, *Fault-Diagnosis Systems – An Introduction from Fault Detection to Fault Tolerance*, 1st ed. Springer, 2006.

[7] B. Stein, O. Niggemann, and H. Balzer, "Diagnosis in Automotive Applications: A Case Study with the Model Compilation Approach," in *Third Monet Workshop on Model-Based Systems at the ECAI*, 2006, pp. 34–40.

[8] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[9] V. Chandola, "Anomaly detection for symbolic sequences and time series data," Ph.D. dissertation, Computer Science Department, University of Minnesota, 2009.

[10] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, p. 2004, 2004.

[11] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*, 4th ed. Academic Press, 2009.

[12] S. Abe, *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition)*, 2nd ed. Springer-Verlag London Ltd., 2010.

[13] D. M. Tax and R. P. Duin, "Data domain description using support vectors," in *Proceedings of the European Symposium on Artificial Neural Networks*, 1999, pp. 251–256.

[14] A. Theissler and I. Dear, "Detecting anomalies in recordings from test drives based on a training set of normal instances," in *Proceedings of the IADIS International Conference Intelligent Systems and Agents 2012 and European Conference Data Mining 2012. IADIS Press, Lisbon.*, 2012, pp. 124–132.

[15] D. M. Tax, "One-class classification. concept-learning in the absence of counter-examples," Ph.D. dissertation, Delft University of Technology, 2001.

[16] C. A. Jones, "Lecture notes: Math2640 introduction to optimisation 4," University of Leeds, School of Mathematics, Tech. Rep., 2005.

[17] D. Tax and R. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[18] A. Theissler and I. Dear, "Autonomously determining the parameters for SVDD with RBF kernel from a one-class training set," in *Proceedings of the WASET International Conference on Machine Intelligence 2013, Stockholm (to be published).*, 2013.

[19] T. Mitsa, *Temporal Data Mining*. Chapman & Hall/CRC, 2010.

[20] T. Raykov and G. Marcoulides, *Basic Statistics: An Introduction with R*. Rowman & Littlefield Publishers, 2012.

[21] J. Laurikkala, M. Juhola, and E. Kentala, "Informal identification of outliers in medical data," in *14th European Conference on Artificial Intelligence ECAI-2000. Berlin*, 2000.

[22] R. Etzold, *So wird's gemacht. Pflegen - warten - reparieren: Renault Twingo von 6/93 bis 12/06*, 8th ed. Delius Klasing, 2011.

[23] A. Theissler, D. Ulmer, and I. Dear, "Interactive knowledge discovery in recordings from vehicle tests," in *33rd FISITA World Automotive Congress*. FISITA, 2010.