# Multi-class Novelty Detection in Diagnostic Trouble Codes from Repair Shops

Andreas Theissler
Robert Bosch GmbH, Automotive Aftermarket
Plochingen, Germany
ORCID: https://orcid.org/0000-0003-0746-0424

***Abstract*—The complexity of vehicles has increased over the last years and will continue to do so. Hence, repairs in repair shops become more and more complex and thereby time-consuming, where time to repair is a competitive factor. During repairs and servicing of vehicles in the independent aftermarket the data read-out using diagnostic testers is transferred back to a common back-end. This "Big Data" contains millions of diagnostic trouble codes (DTCs) and freeze frames, where DTCs point to potential fault causes and freeze frames describe the vehicle's condition when the DTC was stored, e.g. the current engine RPM.**

**This paper proposes an approach to benefit from this "Big Data" in order to (a) acquire knowledge for the development of new vehicles and components and (b) to speed up the time for future fault analyses and repairs. The aim is to detect the vehicle operation modes where a specific fault code was previously not or rarely observed, referred to as novelty detection (anomaly detection). This can point to rare fault causes, which are likely to have a longer time to repair since they are not common. From the field of machine learning, one-class classifiers are applied in two setups: (1) one-class novelty detection, where the data items from all operation modes are combined to one training set and (2) multi-class novelty detection, where an individual classifier is trained for each operation mode.**

**On real data it is shown that novelties can be successfully detected. While it was found that many faults predominantly occur while the vehicle is in a specific operation mode, e.g. when the engine is warm and is in idle, the most interesting novelties were faults when the engine was off or cold. It was found that the multi-class case is superior for the underlying problem using autoSVDD to yield the best results.**

## Publication Details

accepted paper, published at: IEEE 15th International Conference on Industrial Informatics (INDIN), 2017.
DOI: https://doi.org/10.1109/INDIN.2017.8104917

## I. Introduction

Modern automobiles and trucks contain a high number of sensors, actuators and electronic control units (ECUs) communicating over the so-called in-vehicle network. Each of these up to 80 ECUs has one or more responsibilities, e.g. engine control, chassis control, or the brake system. This results in a highly complex, heterogeneous network of software and hardware subsystems delivered by a variety of suppliers. The gross of innovations in vehicles is achieved by means of vehicle electronics, which will further increase the complexity of vehicles. As an effect, repairs in repair shops become more and more complex.

The electronic control units have on-board diagnosis software modules monitoring the vehicle functions during operation. If pre-defined fault setting conditions are met, a fault is detected and a so-called diagnostic trouble code (DTC) is stored in the ECU's local error memory [1]. Examples of conditions that trigger a DTC are implausible or erroneous signal values or signals that were not received.

In repair shops or during inspections the DTCs can be read-out using diagnostic testers and give an indication about potential faults causes. Due to the complexity of vehicles, there is no 1:1 mapping from a DTC to the underlying fault. A DTC can be viewed as a symptom, rather than as the actual fault. An example of a DTC is "P0300", which indicates misfiring of the engine. Problems that triggered that DTC could be related to the spark plugs, the wiring, or the injector nozzles.

In addition to the DTC, an ECU stores so-called freeze frames, which describe the conditions that were present when the DTC was detected. Examples are the current value of the engine revolutions, the coolant temperature, or the vehicle speed.

During the diagnosis of vehicles in a repair shop, this data is read-out from vehicles using diagnostic testers. The technical data is transferred back to vehicle manufacturers, suppliers, or workshop chains. The acquired data can be of enormous value in order to understand how faults occur in the field. These insights can be used (a) in the development of new vehicles and components and (b) to speed up repair of vehicles in the field.

The underlying data in this work are diagnostic sessions from thousands of repair shops from the independent aftermarket (IAM), i.e. vehicles of different brands are present in the data set. For vehicle manufacturers this data can be essential to understand faults in the field after the manufacturer's warranty period has exceeded and many vehicle owners have decided to have their vehicles serviced in repair shops of the independent aftermarket.

Currently many millions of diagnostic sessions are available conducted in repair shops from all over the world over a period of several years. Handling this "Big Data" is done on a Hadoop cluster [2] with Apache Hive to manage, pre-process and access the data. For data analytics Apache Spark [3], the statistical software R [4], and an adopted version of lib-SVM [5] were used.

A subset of the diagnostic trouble codes are standardized OBD-codes, the remaining ones are manufacturer-specific. In this work, the standardized OBD-codes are analysed, making them comparable between different vehicle models and brands. The same type of analyses would be possible for the manufacturer-specific codes.

### A. Aims of this work

Based on faults in the vehicles, which are observable by the read-out diagnostic trouble codes (DTCs), and the corresponding freeze frames, this work has the following aims, where the focus is on the second aim:

1) Understand the circumstances under which faults occur, by identifying under which operation modes faults predominantly occur. This has the potential to reason about potential causes, and thereby helping to speed up repairs.
2) Detect vehicle operation modes where a fault was previsouly not or rarely observed, i.e. novelty detection (anomaly detection). This could point to rare fault causes, which are likely to have a higher time to repair since they are not common.

The first aim, the identification of common vehicle operation modes for a given fault code, is achieved by a combination of cluster analysis and visual analytics. An example of a vehicle operation mode is the mode: engine is warm and vehicle is moving.

The second aim is achieved using machine learning. Classifiers are trained on data from the previously identified operation modes. Data items deviating from these operation modes are reported as novelties.

In other words, the task is not to detect faults in a vehicle. The faults have already been detected by the vehicle's ECUs. The goal is to understand conditions when a fault occurred and to detect unexpected conditions.

The following four one-class classifiers were utilized in this work: Mahalanobis distance classifier, linear $\nu$-SVM, $\nu$-SVM with RBF kernel and parameters set by heuristics, and SVDD with autonomous parameter tuning referred to as autoSVDD.

This paper is organized as follows. Section II introduces novelty detection and section III briefly reviews related work. The methodology is presented in section IV and the results are reported in section V.

## II. Background

Novelty detection refers to the identification of data items with previously unseen characteristics. This could be a simple outlier value in measured data, an intruder in a surveillance system, or an attack in a intrusion detection system. In this paper it is a vehicle operation mode that was previously not or rarely encountered for a given fault code. Terms related to novelty are anomaly [6] or outlier [7].

In this work classifiers from the field of machine learning [8] are used for novelty detection. From a training set of normal data items a classifier is trained without using data items from the novelty class. New data items that do not resemble the normal class are reported as novelties.

This is illustrated for a two-dimensional feature space in Fig. 1, where the blue circles correspond to the normal class and the red squares indicate novelties. The classifier is trained on a subset of the blue circles and the task is to distinguish between the blue circles and the red squares. For this task, one-class classifiers or adoptions of two-class classifiers can be used to learn the decision boundary between the normal class and novelties. Apart from novelty detection, this approach is referred to as one-class classification [9] or semi-supervised anomaly detection in [6].

In two-class classification the trade-off between the two classes is optimized in order to find the optimal decision boundary [10]. A major challenge for one-class classification is that optimising the trade-off between normal data items misclassified as novelties and novelties misclassified as normal is not possible, since no novelties are contained in the training set. This is solved by applying a threshold or by introducing an optimization criterion.

While in Fig. 1 one compact cluster of data items is shown, for the problem discussed in this paper multiple of these clusters are expected, all of them representing data from the normal class (as shown in Fig. 2). These clusters of data items correspond to the operation modes of the vehicle, e.g. (1) engine warm and in idle mode or (2) engine warm and vehicle moving. The task is then to determine whether new data items belong to any of the previsouly seen clusters.

In that case, linear classifiers cannot effectively separate the normal instances from the novelties. More advanced classifiers are capable to learn non-linear decision functions and are hence applicable to such a scenario.

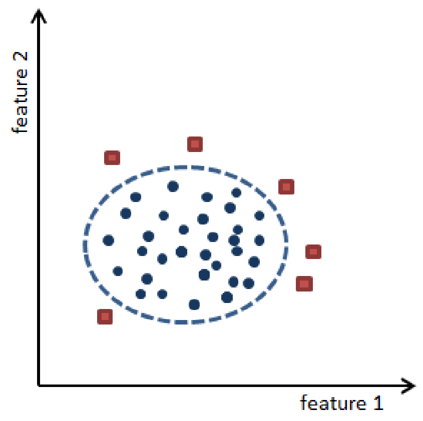

Fig. 1. Illustration of novelty detection in a two-dimensional feature space, where the red squares correspond to the novelties. For classification, the task is to determine whether a data item is inside or outside the decision boundary, depicted by the dashed line.

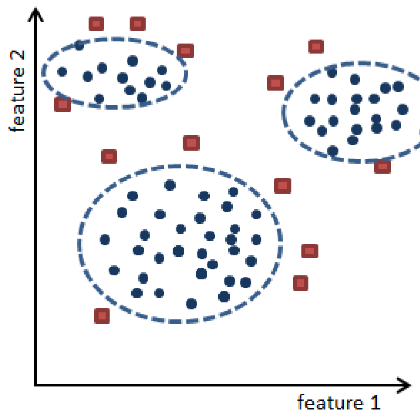

Fig. 2. Illustration of a two-dimensional feature space with several clusters of normal data items, where the red squares correspond to the novelties. For classification, the task is to determine whether a data item belongs to any of the clusters of normal data.

There are two approaches to cope with the problem of multiple clusters of normal data items:

1) *One-class novelty detection*: The data items from all operation modes are combined to one training set and a non-linear classifier is trained on that. For one data item $\vec{x_i}$ classification is done as follows, where $c_{OC}()$ refers to the one-class classifier:

$$\vec{x_i} = \begin{cases} \text{normal if} & c_{OC}(\vec{x_i}) = 0 \\ \text{novelty} & otherwise \end{cases} \tag{1}$$

2) *Multi-class novelty detection*: The knowledge about the different operation modes is incorporated. An individual one-class classifier is trained for each operation mode. In other words, for $n$ operation modes, $n$ one-class classifiers are used. A new data item is checked by each of the base classifiers and *preliminarily* classified as normal if the data item corresponds to the operation mode the classifier was trained on or as novelty, otherwise.
This combination of base classifiers can be viewed as an ensemble [11]. In a final step, a voter uses the base classifiers' results to classify the data item in the following way: If none

of the base classifiers classified a data item as normal, it is reported as novelty. If a data item $\vec{x_i}$ was assigned to the normal class by at least one base classifier $c_{MC_j}()$, the data item is classifed as normal:

$$\vec{x_i} = \begin{cases} \text{normal if} \quad \exists \quad c_{MC_j}(\vec{x_i}) = 0 \quad \forall j \\ \text{novelty} \quad \textit{otherwise} \end{cases} \tag{2}$$

## III. Related work

An overview on novelty detection (outlier/anomaly detection) can be found in [7], [12] and [13]. An experimental comparison of common techniques was presented in [14]. Learning from a training set with normal data items only, i.e. one-class classification, is discussed in [9].

In [15] data from repair shops is used for predictive maintenance of commercial vehicles and buses using classification with known fault types included in the training set. In [16] and [17] one-class classification is used to detect potential faults in recordings from road trials. In [18] road condition monitoring is addressed. The authors of [19] and [20] addressed fault detection for predictive maintenance of commercial vehicles with an unsupervised approach. A hybrid of model-based diagnosis and multiple one-class classifiers is proposed in [21] to detect and isolate different faults in vehicles.

Multi-class novelty detection has been addressed in [22] and [23]. It can be viewed as a special case of ensemble-classification, with each base classifier trained on a different subset of the training set. In [24] an ensemble of one-class classifiers is proposed using dynamic selection of the most appropriate base classifer w.r.t. locality in the feature space. Ensembles of classifiers have been successfully used to detect anomalies in various applications, e.g. detection of unreliable sensors in wireless sensor networks [25], fault detection in road trials [26] and on streaming data [27].

## IV. Methodology

In order to understand why a fault occurred, the circumstances under which the DTC was stored are analysed, thereby trying to reason about conditions that potentially caused the fault or were predominantly observed when the DTC was stored. For each DTC the corresponding values of the freeze frames are used as feature vectors. In a first step the data set is reduced to the standardized OBD DTCs and the most frequent DTCs (top-50) are used.

### A. Aim 1: Identifying conditions under which faults occur

Using the freeze frames, more refined investigations can be conducted, for example analysing if faults occur during specific operation modes of the vehicle. This was achieved with cluster analysis [28]. The identification of clusters was done with DBSCAN [29], due to its ability to cope with outliers. An alternative approach could be to use clustering based on mutual neighbors [30].

With the support of domain-experts and using visual analytics the clusters were validated, adapted if necessary and undesired outliers were removed. Scatter plot matrices [28] and parallel coordinates [31] were used for this purpose, which have previously been applied to automotive data [32]. In addition, the DTCs are enriched with information about the vehicle's brand and model and the time of year, which could be used for further analysis.

### B. Aim 2: Detecting novelties in fault occurrences

The detection of unexpected vehicle operation modes can point to rare fault causes, which are likely to have a higher time to repair.

Machine learning is used to learn data from a training set representing the vehicle operation modes predominantly encountered for each DTC, i.e. the "normal" class. Data items deviating from this class are reported as novelties. The following steps are taken:

Listing 1. Steps for multi-class novelty detection on DTCs

```
// data preparation
$A$ := get standardized OBD DTCs
$B$ := group $A$ by DTC
$B'$ := get most frequent DTCs in $B$

// training
for each DTC $\in B'$:
  $D_{(DTC,\vec{x_i})}$ := get freeze frames $\vec{x_i}$
    for each DTC $\in D_{(DTC,\vec{x_i})}$:
      $C_{DTC}$ := find clusters in $D_{(DTC,\vec{x_i})}$ $\forall i$
        for each cluster $C_{DTC_j} \subseteq C_{DTC}$:
          train one $c_{MC_{DTC_j}}()$ on $C_{DTC_j}$

// test
for each new data $\vec{x_i}$ in test set $E$:
  classify $\vec{x_i}$ using $vote(c_{MC_j}(\vec{x_i}) \quad \forall j)$
```

The following one-class classifiers are used for novelty detection:

*1) Mahalanobis distance classifier:* The Mahalanobis distance can be used as a one-class classifier by applying a threshold. The Mahalanobis distance calculates the covariance matrix of the data set to incorporate the correlations between the features into the distance calculation. For a $D$-dimensional data item, it is the distance of that data item to the mean value of the $D$-dimensional distribution of the data set. In order to use the Mahalanobis distance as a one-class classifier, the threshold for the distance is set to $3\sigma$, i.e. a data item is classified as novelty if its Mahalanobis distance to the mean value is $> 3\sigma$.

*2) Linear one-class support vector machine $\nu$-SVM:* The one-class support vector machine $\nu$-SVM was proposed in [33] and uses a hyperplane to separate normal data items and novelties, where the position of the hyperplane is controlled by the parameter $\nu$. The parameter $\nu$ was set to $\frac{1}{N}$, where $N$ is the number of data items in the training set. This corresponds to the case, where no novelties are expected in the training set.

*3) Non-linear one-class support vector machine $\nu$-SVM:* As for two-class SVMs [10], $\nu$-SVM can be enhanced by kernels to allow for non-linear decision functions. In this work the RBF kernel is used introducing the kernel parameter $\gamma$ (or $\sigma$). The parameter $\nu$ was set to $\frac{1}{N}$ and the kernel parameter $\gamma$ was set to $\frac{1}{D}$, where $D$ is the number of features, as done by [5].

*4) One-class support vector machine autoSVDD:* In [34] support vector data description (SVDD) was introduced. While two-class support vector machines (see [10] and [35]) and $\nu$-SVM [33] separate the data by a hyperplane, 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. 3, and is found by solving the optimisation problem of minimising

1) the error on the normal class (normal data reported as novelties)
2) the chance of misclassifying data from the abnormal class (novelties misclassified as normal data)

Minimising the error on the normal class is achieved by tuning $R$ and $\vec{a}$ in a way that all data items of the training data set are contained in the hypersphere. Minimising the chance of misclassifying data from the novelty class cannot be achieved straightforward, since in the absence of novelties in the training data, misclassified novelties cannot occur during training. It is achieved by minimising the hypersphere's volume, assuming this reduces the risk of misclassification. This trade-off between the number of misclassified normal data items and the volume of the hypersphere is optimised by minimising

$$F(R, \vec{a}) = R^2 \tag{3}$$

subject to

$$\|\vec{x_i} - \vec{a}\|^2 \leq R^2 \quad \forall i \qquad i = 1,..,M \tag{4}$$

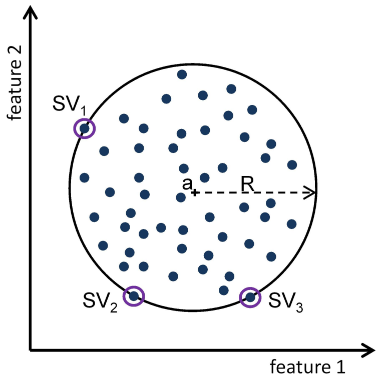

Fig. 3. Illustration of a SVDD decision boundary in two-dimensional feature space with radius $R$ and center $\vec{a}$. The hypersphere is described by support vectors (here: $SV_1 \dots SV_3$).

where $\vec{x_i}$ denotes the data items and $M$ the number of data items 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 $\vec{a}$.

After the training phase, the hypersphere is described by selected instances from the training data set, so-called support vectors (see Fig. 3). The center $\vec{a}$ is implicitly described by a linear combination of these support vectors. The remaining data items are discarded.

SVDD in this form is very sensitive to undesired outliers in the training set. Therefore slack variables $\xi_i$ are introduced, which allow for some data items $\vec{x_i}$ in the training 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 (3) and (4) changes into minimising

$$F(R, \vec{a}, \xi_i) = R^2 + C \sum_{i=1}^{M} \xi_i \tag{5}$$

subject to

$$\|\vec{x_i} - \vec{a}\|^2 \leq R^2 + \xi_i \quad \forall i \tag{6}$$

and

$$\xi_i \geq 0 \quad \forall i \tag{7}$$

As described in [9], the constrained optimisation problem is transformed into an unconstrained one by integrating the constraints into the equation using the method of Lagrange [36]. 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, where $\alpha_i$ and $\alpha_j$ are introduced by the Lagrange method and the data items with $\alpha_i > 0$ are support vectors:

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

subject to

$$0 \leq \alpha_i \leq C \quad \forall i \tag{9}$$

At this point, SVDD is capable of surrounding the normal data by a strict hypersphere. Analogous to $\nu$-SVM, SVDD uses the kernel trick [10] to overcome this inflexibility. The data items are mapped to a higher-dimensional space where they can be surrounded by a hypersphere, using a mapping function $\phi()$.

As can be seen from (8) $\vec{x_i}$ 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, the kernel trick is to replace the inner products of the mapped feature vectors $\phi(\vec{x_i}) \cdot \phi(\vec{x_j})$ by a kernel function $K(\vec{x_i}, \vec{x_j})$. The mapping is then implicitly done by applying this kernel function which can be viewed as a highly non-linear decision function.

Different kernel functions can be utilized to achieve flexible decision boundaries. In this work the radial basis function (RBF) kernel, is used

$$K(\vec{x_i}, \vec{x_j}) = e^{-\frac{\|\vec{x_i} - \vec{x_j}\|^2}{\sigma^2}} \tag{10}$$

where $\sigma$ is referred to as the kernel width, which in some literature is reformulated and denoted by $\gamma$.

The two parameters $C$ and $\sigma$ massively influence the classification accuracy. While SVDD yields very good classification results for the optimal parameter set, having to manually adjust the parameters makes it non-applicable for the problem discussed in this paper. Using an approach for autonomous tuning of the hyperparameters proposed by this paper's author in [16], the optimal parameter set is determined solely from the training set.

For each optimisation step, the error rate and the radius $R$ are determined using k-fold cross validation. Grid search is run over the parameter space and that set of parameters is selected where the radius is close to 1 while the error $e_i$ on the normal class is minimal, given by the following optimisation criterion. Find $(C_i, \sigma_i)$ where $\lambda_i$ is minimal:

$$\lambda_i = \sqrt{e_i^2 + |1 - R_i|^2} \quad \forall i \tag{11}$$

It has been shown that for the RBF kernel, SVDD and $\nu$-SVM yield equivalent solutions [9]. The differences in the results reported in this paper are solely caused by the way of tuning the hyperparameters for SVDD. The resulting ”out-of-the-box” SVDD will be denoted as autoSVDD.

## V. Experimental results

The underlying data set consists of millions of diagnostic sessions recorded in repair shops from all over the world over a period of several years. The data is stored on a Hadoop cluster [2] and Apache Hive is used to access the data. For data analytics Apache Spark [3], the statistical software R [4], and an adopted version of lib-SVM [5] were used.

The data contains hundreds of millions of fault codes. The analysis reported in this paper was done on the 50 most frequent DTCs. In this section the results for one representative DTC are reported: P0171 (”System Too Lean, Bank 1”), which is a commonly encountered emission-related DTC that is triggered when there is too much oxygen in the exhaust, i.e. the ratio of oxygene and fuel is not optimal. Some potential causes are a dirty sensor, a dirty oil-filter or a vacuum leak.

### A. *Identifying conditions under which faults occur*

Using a DTC's freeze frames, more refined investigations can be conducted, for example analysing if faults occur during specific operation modes of the vehicle. It was found that many faults predominantly occur while the vehicle is in a specific operation mode, while other faults do not depend on the operation mode. For those DTCs where the freeze frames reveal groups, cluster analysis is used for further analysis, in order to isolate these groups from different operation modes. Using DBSCAN [29] and visual analytics to refine and validate the found clusters, the following vehicle operations modes were identified for many of the DTCs (see Fig. 4): [1]

[1]DTCs are detected by built-in test routines running on the vehicle's electronic control units. In some cases, these test routines are only run under specific conditions, e.g. engine in idle. In that case the found clusters are misleading. Such patterns were found for a small number of the investigated DTCs, but are not reported here.

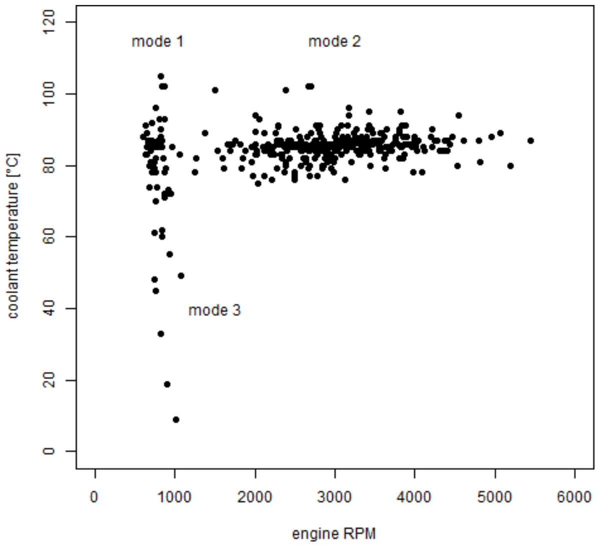

Fig. 4. Training data for DTC P0171 with two freeze frames. Three vehicle operation modes were identified: mode 1: engine warm and in idle, mode 2: engine warm and vehicle moving, mode 3: engine cold and in idle.

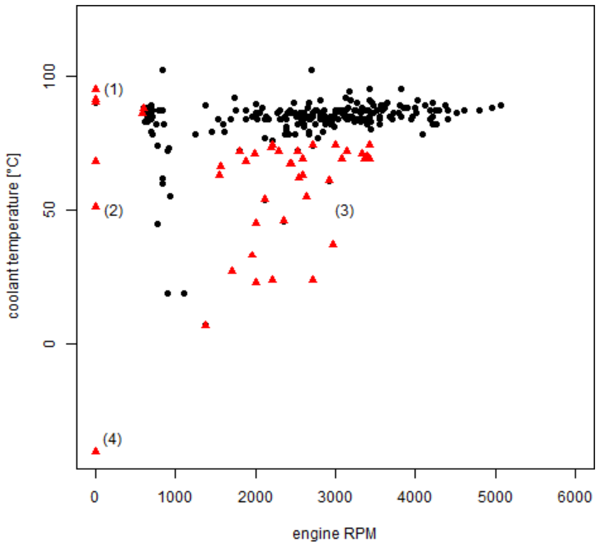

Fig. 5. Test data for DTC P0171 with two freeze frames. Four types of novelties were detected: (1) engine warm + engine off, (2) engine cold + engine off, (3) engine cold + vehicle moving, (4) default value.

- mode 1: engine warm and in idle
- mode 2: engine warm and vehicle is moving
- mode 3: engine cold and in idle

### B. Detecting novelties in fault occurrences

Analysing a DTC's freeze frames, rare or unexpected fault conditions were autonomously identified using one-class classifiers. The four classifiers described in Section IV were applied in both setups: one-class and multi-class novelty detection.

In order to compare and validate the accuracy of the different setups, a test set with data from all the previously identified normal operation modes was used enhanced by data items from the novelty class (see Fig. 5). The following accuracy measures were used:

1) *Weighted accuracy (balanced accuracy)*: Novelty detection is an imbalanced class problem, i.e. data items from the novelty class are rare. Hence, using the standard overall accuracy would be highly biased towards the majority class, i.e. the normal class. Therefore the weighted accuracy is used, which calculates the accuracies per class and combines the results considering the ratio of the classes.
2) *Novelty detection rate (recall)*: The novelty detection rate $d_{nov}$ represents the percentage of detected novelties, i.e. $d_{nov} = \frac{\text{detected novelties}}{\text{all novelties}}$.
3) *Precision on the novelty class*: The precision $p_{nov}$ is the percentage of true novelties in the subset of data items reported as novelties, i.e. $p_{nov} = \frac{\text{detected novelties}}{\text{items reported as novelties}}$.
4) *F-score*: The f-score (f-measure) incorporates the detection rate and the precision by $f - score = 2 * \frac{d_{nov} * p_{nov}}{d_{nov} + p_{nov}}$.

The test data for DTC P0171 with two freeze frames is shown in Fig. 5. Four types of novelties were detected:

- (1) engine warm + engine off: this indicates that the fault was detected after the engine was switched off after a drive
- (2) engine cold + engine off: this constellation corresponds to faults that were detected during start of the engine, possibly the engine could not be started
- (3) engine cold + vehicle moving: this points to faults that occurred during the first minutes of a drive, when the engine was not warm
- (4) default values: sensor values could not be read or sensor was erroneous

| | classifier | acc | $d_{nov}$ | $p_{nov}$ | f-score |
|---|---|---|---|---|---|
| one-class | Mahalanobis | 64.4 | 31.1 | 77.8 | 44.4 |
| | linear $\nu$-SVM | 54.2 | 28.9 | 27.1 | 28.0 |
| | $\nu$-SVM RBF | 97.1 | **100** | 81.8 | 90.0 |
| | autoSVDD | 97.1 | **100** | 81.8 | 90.0 |
| multi-class | Mahalanobis | 96.6 | 95.6 | 91.5 | 93.5 |
| | linear $\nu$-SVM | 50.0 | 0 | – | – |
| | $\nu$-SVM RBF | 90.6 | **100** | 58.4 | 73.8 |
| | autoSVDD | **98.9** | 97.8 | **100** | **98.9** |

TABLE I
RESULTS FOR ONE-CLASS AND MULTI-CLASS NOVELTY DETECTION. FOUR DIFFERENT ONE-CLASS CLASSIFIERS WERE USED, WHERE THE BEST RESULTS WERE ACHIEVED WITH AUTOSVDD TRAINED ON THE DIFFERENT OPERATION MODES, I.E. MULTI-CLASS NOVELTY DETECTION.

The results are shown in Table I. As can be seen, the Mahalanobis classifier is not appropriate for the one-class case if the data is scattered in multiple clusters, since it works with a fixed threshold on the distance to the data set's mean value. The linear $\nu$-SVM misclassifies many data items since it cannot seperate the novelties with one linear decision function. The $\nu$-SVM and autoSVDD can cope with the multiple clusters due to the use of the RBF kernel and detected 100% of the present novelties while misclassifying some normal data items.

The Mahalanobis classifier benefits from the multi-class setup, where one classifier is trained per cluster. In this setup the mean value and threshold are determined per cluster, making the Mahalanobis classifier adapt to local differences in the feature space [37]. The linear $\nu$-SVM fails to detect novelties due to the inflexible decision functions. In general, as the number of normal clusters increases in a multi-class setup, the chance of novelties being assigned to any of the normal classes increases. The best results were achieved with autoSVDD in the multi-class novelty detection setup with an f-score of 98.9%.

In Fig. 6 the classifiers' f-scores, i.e. combined values for the detection rate $d_{nov}$ and the precision $p_{nov}$, are depicted. As can be seen, the Mahalanobis distance yields a good accuracy with multiple classifiers trained on each cluster. The kernel-based methods work well for both cases, where autoSVDD performs best in the multi-class setup.

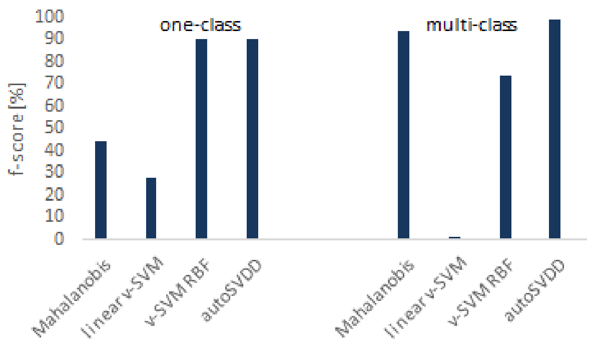

Fig. 6. Results for the four classifiers with (a) training of one classifier on the entire training set (one-class) and (b) training of one classifier per cluster in the training set (multi-class).

## VI. Conclusion

In this paper vehicle diagnoses from repair shops were analysed with the following two aims: (1) identification of conditions under which faults predominantly occur and (2) detection of vehicle operation modes that where previsouly not or rarely observed for a fault code. In a first step, the normal data was subdivided into clusters corresponding to vehicle operation modes. Following that, one-class classifiers were used to detect novelties. Two setups were investigated: (a) training one classifier on the entire training set and (b) incorporating the domain-knowledge of vehicle operation modes and training of one classifier per operation mode, i.e. per cluster. This multi-class novelty detection yielded the best results, with autoSVDD performing best. These results were reported for one representative DTC and can be generalized for most other DTCs. An example of new findings are novelties where the fault occurred when the engine was cold and not running, pointing to potential problems during start of the engine. The obtained insights can be used in the development of new vehicles and components and to speed up the time for future fault analyses and repairs.

## Acknowledgment

The author would like to thank the business unit Automotive Aftermarket of Robert Bosch GmbH for granting this research and Anne Meyen and Dennis Gross for support during data analytics.

## References

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

[2] T. White, *Hadoop: The Definitive Guide*, 4th ed. O'Reilly Media, Inc., 2015.

[3] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark: Lightning-Fast Big Data Analytics*, 1st ed. O'Reilly Media, Inc., 2015.

[4] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2015. [Online]. Available: https://www.R-project.org/

[5] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

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

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

[8] T. M. Mitchell, *Machine Learning*. McGraw-Hill Education (ISE Editions), October 1997.

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

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

[11] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, 2006.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[13] C. C. Aggarwal, *Outlier Analysis*. Springer, 2013.

[14] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313 – 327, 2014.

[15] R. Prytz, S. Nowaczyk, T. S. Rgnvaldsson, and S. Byttner, "Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data." *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 139–150, 2015.

[16] A. Theissler, "Detecting anomalies in multivariate time series from automotive systems," Ph.D. dissertation, Brunel University London, 2013.

[17] ——, "Anomaly detection in recordings from in-vehicle networks," in *Big Data Applications and Principles (BIGDAP 2014)*, 2014.

[18] F. Cong, H. Hautakangas, J. Nieminen, O. Mazhelis, M. Perttunen, J. Riekki, and T. Ristaniemi, "Applying wavelet packet decomposition and one-class support vector machine on vehicle acceleration traces for road anomaly detection," in *Advances in Neural Networks ISNN 2013*, 2013.

[19] M. Svensson, S. Byttner, and T. Rognvaldsson, "Self-organizing maps for automatic fault detection in a vehicle cooling system," in *Intelligent Systems, 4th International IEEE Conference*, vol. 3, 2008.

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

[21] D. Jung, K. Y. Ng, E. Frisk, and M. Krysander, "A combined diagnosis system design using model-based and data-driven methods," in *3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, 2016.

[22] G. Blanchard, G. Lee, and C. Scott, "Semi-supervised novelty detection," *J. Mach. Learn. Res.*, vol. 11, pp. 2973–3009, Dec. 2010.

[23] A. E. Lazzaretti, D. M. J. Tax, H. V. Neto, and V. H. Ferreira, "Novelty detection and multi-class classification in power distribution voltage waveforms," *Expert Systems with Applications*, vol. 45, pp. 322 – 330, 2016.

[24] B. Krawczyk and M. Woźniak, "Dynamic classifier selection for one-class classification," *Knowledge-Based Systems*, vol. 107, no. C, pp. 43–53, Sep. 2016.

[25] D.-I. Curiac and C. Volosencu, "Ensemble based sensing anomaly detection in wireless sensor networks," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9087–9096, Aug. 2012.

[26] A. Theissler, "Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection," *Knowledge-Based Systems*, vol. 123, no. C, pp. 163–173, May 2017.

[27] Z. Ding, M. Fei, D. Du, and F. Yang, "Streaming data anomaly detection method based on hyper-grid structure and online ensemble learning," *Soft Computing*, pp. 1–13, 2016.

[28] J. Han, M. Kamber, and J. Pei, *Data Mining - Concepts and Techniques*, 3rd ed. Morgan Kaufmann Publishers, 2012.

[29] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*. AAAI Press, 1996, pp. 226–231.

[30] J. Huang, Q. Zhu, L. Yang, D. Cheng, and Q. Wu, "A novel outlier cluster detection algorithm without top-n parameter," *Knowledge-Based Systems*, vol. 121, pp. 32 – 40, 2017.

[31] A. Inselberg, "The plane with parallel coordinates," *The Visual Computer*, vol. 1, no. 2, pp. 69–91, 1985.

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

[33] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1471, July 2001.

[34] 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.

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

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

[37] B. Krawczyk, “Learning from imbalanced data: open challenges and future directions,” *Progress in Artificial Intelligence*, pp. 1–12, 2016.