Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection

Andreas Theissler – ORCID https://orcid.org/0000-0003-0746-0424

Esslingen University of Applied Sciences

Abstract

The massive growth of data produced in the automotive industry by acquiring data during production and test of vehicles requires effective and intelligent ways of analysing these recordings. In order to detect potential faults, data from the in-vehicle network interconnecting vehicle subsystems is recorded during road trials. The complexity and volume of this data keeps increasing since the degree of interconnection between the vehicle subsystems and the amount of data transmitted over the in-vehicle network is augmented with each functionality added to modern vehicles. In this paper, an anomaly detection approach is proposed that (a) is capable of detecting faults of known and previously unknown fault types, (b) functions as an out-of-the-box approach not requiring the setting of expert-parameters and (c) is robust against different driving scenarios and fault types. To achieve this, an ensemble classifier is used consisting of two-class and one-class classifiers. Without modelling effort and user parameterisation the approach reports anomalies in the multivariate time series which point the expert to potential faults. The approach is validated on recordings from road trials and it could be shown that the ensemble-anomaly detector is robust against different driving scenarios and fault types.

Keywords: anomaly detection, fault detection, one-class classification, ensemble methods, vehicle electronics

Publication details

Accepted manuscript

Published by Elsevier's Knowledge-Based Systems.
https://doi.org/10.1016/j.knosys.2017.02.023

1. Introduction

The automotive industry has experienced a massive growth of the data recorded during production and test of products. Vehicles have turned from mechanical machines into highly complex products of interconnected subsystems dominated by software and electronics. The complexity keeps increasing since the degree of interconnection between the subsystems and the amount of data transmitted over the in-vehicle network is augmented with each functionality added to modern vehicles. The number of electronic control units (ECUs) that communicate over the in-vehicle network is up to 80. As a consequence, there is a high chance of faults either caused by individual subsystems or by the interconnection of those subsystems within the vehicle.

In order to detect potential faults, road trials are conducted by vehicle manufacturers and suppliers and the multivariate time series data are recorded, resulting in mass data. Only by effective and intelligent ways of analysing the recordings, one can make sure that the effort put into the vehicle tests pays off. Consequently, effective ways of data analysis can become a competitive advantage.

Based on a training set of recordings from vehicles, the aim of this work is to classify subsequences in a test set as either normal ω_n or anomaly ω_a, i.e. as a potential fault. This can be achieved 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 two-class classification or supervised anomaly detection [1] and has been reported to work for autmotive data [2].

For fault-detection such an approach has major drawbacks. Obtaining or creating abnormal training data is very intricate. If abnormal data can be obtained, it is highly likely that it is not representative, because many faults in a vehicle are not known or cannot easily be injected. However, using a non-representative training data set of anomalies yields an incorrect decision function.

On the other hand, normal data can easily be obtained by recording data from a vehicle in normal operation mode. So an alternative is to only learn the normal behaviour and classify deviations as anomalies referred to as one-class classification [3] or semi-supervised anomaly detection [1]. This was successfully applied to automotive data in [4, 5, 6]. The accuracy of such approaches are however lower compared to two-class classification on a representative data set.

1.1. Proposed approach

For the detection of faults in automotive recordings there are two challenges that are addressed in this paper:

a) An approach is needed that is capable of detecting known fault types as well as previously unknown fault types. In [7] the capability to detect previously unknown faults is identified as one of the most important properties of a fault detection system.

b) Since recordings from vehicle tests are highly variable, an approach is required that is robust w.r.t. different driving scenarios. While a classifier might yield good results on a specific data set, the same classifier may fail to work well on different data sets [8].

The challenge to detect known and unknown fault types is addressed in this paper by incorporating one-class and two-class classifiers into an ensemble [9, 10].

The robustness w.r.t. different driving scenarios and different fault types is tackled by using a set of diverse classifiers in the ensemble. Therefore different types of anomaly detectors are used: one-class and two-class classifiers for univariate and multivariate anomalies. The ensemble's output is postprocessed in order to report subsequences of anomalies.

The proposed approach has the following main objectives:

1. detection of known and unknown fault types in automotive multivariate time series
2. functioning as an out-of-the-box approach that does not require expert knowledge for configuration
3. robustness against different driving scenarios and fault types

The results on recordings from four different scenarios show the effectiveness of the proposed ensemble method. The ensemble is capable to detect known and unknown fault types and is robust against the variability of recordings from a constrained environment (vehicle in idle mode) and an unconstrained environment (recordings from overland drives).

1.2. Contributions

The following contributions are made in this paper:

1. the range of potential faults in a vehicle is carefully investigated and an overview is given
2. a categorization of anomalies in automotive recordings is given
3. the performance of two-class and one-class classifiers are evaluated for the detection of known and unknown fault types
4. an out-of-the-box ensemble method is created that can detect known and unknown fault types without the need for expert knowledge
5. the ideas are validated on real data

This paper is organized as follows. The next section surveys related work. Section 3 categorizes potential faults in vehicles, introduces ensemble methods and briefly discusses the used base classifiers. Section 4 introduces the anomaly detection approach, followed by section 5 reporting the experimental results and discussing the outcomes. Section 6 concludes this paper.

2. Related work

This section surveys related work from the fields of fault and anomaly detection in automotive systems and ensemble-based anomaly detection and contrasts it to this paper.

In [7] a general survey of fault detection is given. The detection of faults or anomalies in automotive systems was addressed in various publications. The authors of [5] used anomaly detection 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 contrast to this work, the approach in [5] detects a specific type of anomaly which differs from the detection of faults, where different types of potentially unknown anomaly types can occur.

The author of [2] discusses a data-driven approach to classify the health state of an in-vehicle network based on the occurrences of so-called error frames using a labelled training set of recordings from fault-free mode and faults. In [11] predictive maintenance of commercial vehicles and buses is addressed using classification with known fault types included in the training set. These approaches rely on a representative training set of faults, in contrast to the present paper.

In [4] automotive security is addressed. From data recorded from the in-vehicle network communication in normal operation mode, the normal value of entropy is learnt. Deviations from that entropy are reported as potential intrusions. In [6] and [12] the author of this paper used one-class classification to detect potential faults in recordings from road trials. In contrast to the present paper, these papers do not exploit knowledge about occurred anomalies, i.e. intrusions or faults.

The authors of [13] and [14] propose fault detection for predictive maintenance of commercial vehicles. Data from different vehicles are compared and anomalies are detected in an unsupervised manner being those vehicles deviating from the others. The approach is unsupervised, i.e. it does not incorporate knowledge about the normal or abnormal operation mode, which differs from the proposed approach in the present paper.

A hybrid of model-based diagnosis and one-class classification is proposed in [15] to detect and isolate faults in vehicles. The approach is reported to be capable to detect known and previously unknown fault types, but requires to build a model.

Ensembles of classifiers have been successfully used to detect anomalies in various applications. In [16] unreliable sensors in wireless sensor networks are detected using an ensemble of five anomaly detectors. The diversity required for effective ensemble methods is modelled by using different classifiers exploiting different aspects of the data: spatial redundancy of sensors located close to each other, temporal redundancy of consecutive sensor readings and the combination of both. In the same research field, the authors of [17] proposed to use an ensemble of classifiers to detect anomalies in a decentralized manner under the constraints of limited resources on the embedded systems of sensor

networks. The authors of [18] proposed an ensemble anomaly detection method for streaming data. In [19] ensemble-based anomaly detection using Hidden Markov models was applied to intrusion detection based on system calls and a method was proposed to prune the number of base classifiers. Ensembles of unsupervised anomaly detectors were applied on data sets for intrusion detection and for breast cancer detection in [8]. In [20] an ensemble of one-class classifiers is proposed that uses dynamic selection of the most appropriate base classifer for a given input feature vector based on the classifiers' accuracies over the feature space.

The authors in [21] contrasted the performance of two-class and one-class classifiers for the task of keystroke dynamics authentication, but did not combine the classifiers to an ensemble.

3. Background

In this section potential fault locations in vehicles are surveyed, followed by a categorization of anomaly types that can be present in recordings from vehicles. Following that, ensemble-based anomaly detection is introduced and the two-class and one-class classifiers that are used in the proposed ensemble-based anomaly detector are described.

3.1. Fault detection in automotive recordings

To emphasize the scale of the problem, potential fault locations in a vehicle were identified and are presented in Fig. 1. The tree-like structure shows how manifold faults can be. At the top level the locations are categorised into function specifications, in-vehicle network, sensors, actuators, ECUs, gateways, power supply, vehicle subsystems, and the data acquisition system. Fault locations in ECUs can be further subdivided into software and hardware. For a more detailled discussion the reader is referred to [6].

A recording of a road trial corresponds to time series data or can be resampled equidistantly to become time series data. Time series can be univariate or multivariate [22], where observations of one variable are referred to as univariate and observations of multiple variables are referred to as multivariate. An example of a univariate time series is the recording of the vehicle's velocity signal over some period of time. A multivariate time series would be the vehicle's velocity together with further signals like yaw rate, steering wheel angle and engine speed.

In this paper, faults are detected using an anomaly detection approach. The term anomaly is defined as a condition that deviates from expectations in ISO-26262 [23]. Other terms used in literature are novelty, outlier [24] and discord [25].

In [1], anomalies are categorised as point, contextual, and collective anomalies. The idea of point and contextual anomalies was borrowed for this work and applied to recordings from automotive systems.

A multivariate time series consists of multiple univariate time series. Therefore in a multivariate time series, anomalies of a univariate time series can occur.

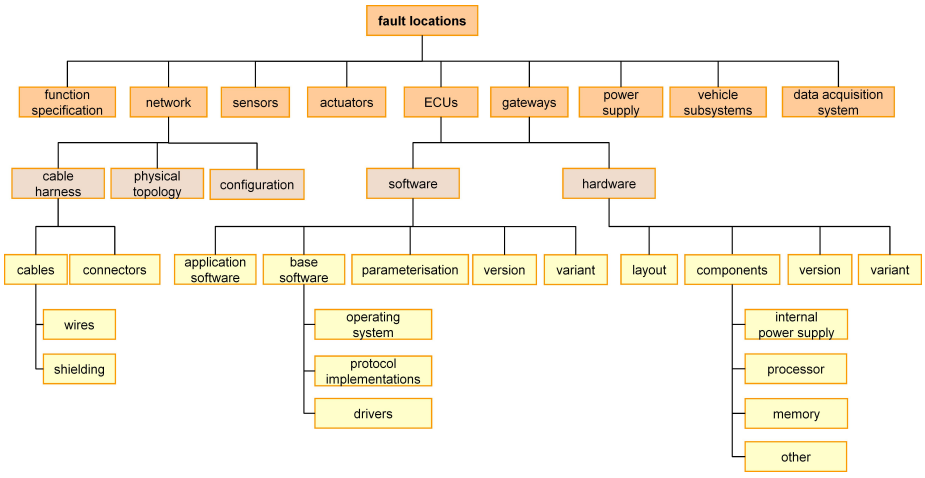

Figure 1: Categorisation of potential fault locations in a vehicle.

Additionally, anomalies in the relationship between the contained univariate time series can occur. The following three types of anomalies are distinguished in this work:

1. *Type I: subsequence anomaly in univariate time series.* An individual subsequence can be classified as normal or abnormal without konwledge about further subsequences.
2. *Type II: contextual anomaly in univariate time series.* Classification of an individual subsequence requires knowledge about the context, i.e. about preceeding subsequences within the same univariate time series.
3. *Type III: contextual anomaly in multivariate time series.* Classification of an individual subsequence in a multivariate time series requires knowledge about subsequences in additional univariate time series.

This work will focus on anomalies of type I and type III. Anomalies of type II could for example be detected using Hidden Markov models [26].

3.2. Ensemble-based anomaly detection

In this work an ensemble of classifiers is used for anomaly detection. In ensemble-based classification, a number of base classifiers are combined and their outputs are used to create a single classification result. There are different methods to combine the outputs of the base classifiers. Many methods use the crisp classifier outputs and apply majority voting, weighted voting or select the best subset of base classifiers. Other methods weight a base classifier's output w.r.t. the class. An alternative approach referred to as stacking uses continuous classifier outputs as features to train an independent classifier [9, 27].

The base classifiers in the ensemble are desired to be diverse [27], i.e. to yield different classification results. Only by disagreeing on some feature vectors can

ensembles become more effective compared to the use of individual classifiers. One diversity measure is the entropy measure [28], which is given by

$$E = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|C| - \lceil |C| - 2 \rceil} min(c(\mathbf{x_i}), |C| - c(\mathbf{x_i})) \tag{1}$$

where N is the number of feature vectors, $|C|$ is the number of base classifiers and $c(\mathbf{x_i})$ is the number of classifiers that classified the feature vector $\mathbf{x_i}$ correctly. In this work the requirement of having diverse classifiers is met in several ways: one-class and two-class classifiers are incorporated as well as classifiers for the detection of univariate and multivariate anomalies.

There are different variations of ensemble classifiers. The authors in [27] distinguish between the application of the same classifier on different subsets of the training set and the application of different classifiers on one common training set. The selection of base classifiers can be static, where a fixed set of classifiers is used or dynamic, where the selection is based on the previous classifications. In [29] ensemble classifiers are categorized into independent and sequential ensembles, where the first refers to the application of different classifiers on the complete or a subset of the training set. The latter refers to the sequential application of one or more classifiers, where the classification is influenced by the previous results. In this work the selection of base classifiers is done statically.

3.3. Anomaly detection using two-class classifiers

Detecting faults in test drive recordings can be solved by means of classification using a training set with labelled data from normal operation mode and faults, which is referred to as supervised anomaly detection [1]. In the case of normal and abnormal corresponding to one class each, two-class classification can be used. Two-class classification is a long-established research field and classifiers from a broad range can be used. Four two-class classifiers were selected to be incorporated into the ensemble: mixture of Gaussians classifier, Naive Bayes classifier, random forest, and support vector machine.

MOG. The mixture of Gaussians (MOG) classifier is used in a univariate way addressing anomalies in univariate time series (type I anomalies, as defined in Section 3.1). For each feature the one-dimensional probability density function is determined from the training set, one for each of the two classes ω_n and ω_a. If for one feature, an instance is classified as anomaly, the entire instance is classified as an anomaly. The classifier thereby detects if e.g. one signal is out of the valid value range.

Naive Bayes. The naive Bayes classifier [30] estimates each class' probability density function for each feature individually and determines thresholds that minimize the probability of misclassification. Following that, the decision function is determined by combining the thresholds of each dimension. The classifier assumes independence between the features but has been reported to work well for cases where this assumption does not hold.

Random forest. A random forest [31] is an ensemble method that grows a high number of decision trees on different subsets of the feature space and of the training set. The classification result is based on the combination of the trees' outputs.

Support vector machine. A support vector machine (SVM) [32, 33] is essentially a two-class classifier and can thereby be used to distinguish between the two classes ω_n and ω_a. A separating hyperplane is determined from the training data set by demanding a class separation with maximum margin, allowing some instances to be outside the decision boundary controlled by the regularization parameter C. An SVM can be enhanced to function as a non-linear classifier by using the kernel trick to map the input feature space to a higher-dimensional feature space, where the data can be linearly separated by a hyperplane. In this work, the RBF kernel is used adding the kernel parameter γ. No assumptions about probability distributions are made by an SVM, classification is exclusively done based on the instances at the boundaries of the classes.

3.4. Anomaly detection using one-class classifiers

Obtaining or creating abnormal training data is very intricate. Recordings from normal operation mode on the other hand can be obtained easily. In practice it is unlikely to have a representative training set of all possible faults, making one-class classification [3, 34] especially valuable to detect unknown or unmodelled faults.

The goal is to learn the normal behaviour based on a training set that exclusively contains instances from the normal class, which is also referred to as semi-supervised anomaly detection [1]. Test instances are classified as normal if they are similar in some way to the training set or as anomalies otherwise.

As opposed to two-class classifiation, for one-class classification optimising the trade-off between true and false positives is not possible, since no anomalies are present in the training set. The challenge in one-class classification is to learn the boundaries of the normal region. It requires the setting or tuning of parameters which can be pre-defined, defined based on heuristics, or determined from the training set.

On the one hand, one-class classifiers have the potential to detect unexpected faults that were not covered by the test process or not modelled. On the other hand, the accuracies of one-class classifiers will typically be lower compared to two-class classification with a respresentative training set.

Some two-class classifiers can be adopted to become one-class classifiers. In addition there are classifiers specifically designed for one-class classification. The following four one-class classifiers are used in the ensemble in this work: extreme value analysis, Mahalanobis classifier, one-class support vector machine, and support vector data description.

Extreme value analysis. In extreme value analysis [29] a normal distribution is assumed and the mean value and standard deviation are determined. A threshold is required for extreme value analysis to act as a classifier. It is used

as a univariate classifier, i.e. each feature is treated individually equivalently to the univariate MOG classifier.

Mahalanobis classifier. The Mahalanobis distance uses the covariance matrix of the data set to incorporate the correlations between features into the distance calculation. For a given feature vector, it is the distance of the feature vector to the mean value of a multi-dimensional distribution. In order for the Mahalanobis distance to be used as as classifier, a threshold has to be set.

One-class support vector machine. The one-class support vector machine (OC-SVM) proposed in [35] uses a hyperplane to separate normal and anomalous instances, where the position of the hyperplane is controlled by the parameter ν. An RBF kernel is used in this work introducing the kernel parameter γ.

Support vector data description. In [36] the one-class support vector machine "support vector data description" (SVDD) was introduced. While other support vector machines separate the data by a hyperplane, SVDD forms a hypersphere around the normal instances in the training data set. The hypersphere is found by solving the optimisation problem of minimising both, the error on the normal class and the chance of misclassifying data from the abnormal class. The soft-margin SVDD with an RBF kernel is used, introducing the regularization parameter C and the RBF Kernel parameter γ.

4. The methodology: ensembled-based anomaly detection

In this section an approach for the detection of anomalies in automotive recordings is proposed and its components are discussed.

Fault detection in automotive recordings puts special requirements on the task of anomaly detection. Faults should be detected, regardless if the fault types are included in the training set or not. Reasons for the latter could be that the faults are unknown, unexpected or recordings are unavailable. Furthermore, the approach is required to be robust against the high variablity of the recordings caused by different driving scenarios and environmental conditions. In addition anomalies that are present in a univariate time series as well as in the relation between the individual time series should be detected.

A data-driven approach is proposed that is trained on a data set of recordings from vehicles. The aim is to classify subsequences as either normal ω_n or anomaly ω_a, where an anomaly points to a potential fault. The aim is not to classify the type of faults, but rather to detect whether a fault was present at a given point in time. Reported anomalies could be faults, but could also be data from a system's operation mode or driver behaviour that were not contained in the training set. In this work an anomaly is referred to as a positive, so a detected anomaly that points to a fault is referred to as a true positive (TP).

The process from the input of raw data to the reporting of anomalous subsequences is described in this section. The process is subdivided into the following steps: data selection, data transformation, ensemble classification, filtering and the creation of subsequences as shown by the flowchart in Fig. 2.

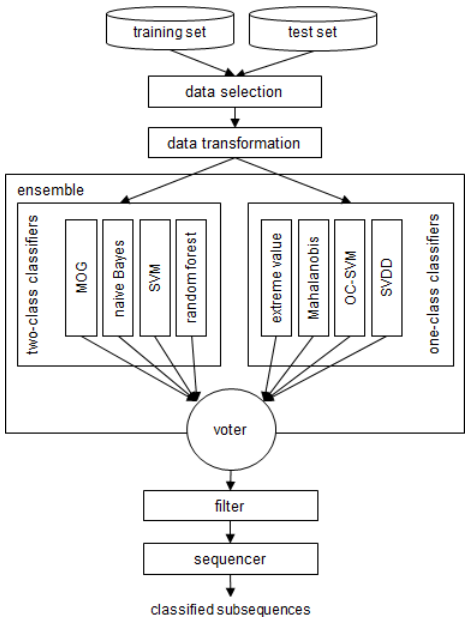

Figure 2: Flowchart showing the steps of the approach, where the data flow is illustrated by arrows. The training and test data are passed to the ensemble and hence to the individual base classifiers. The classifiers' outputs are combined by majority voting, followed by a filter and a sequencer.

4.1. Data selection

The data selection step is used during training mode. The full training set is passed to the two-class classifiers, while only the recordings containing no faults are selected and passed to the one-class classifiers.

4.2. Data transformation

In the data transformation step the multivariate time series are transformed to feature vectors by transforming the values at each time point T_i to a feature vector F_i. An $M \times N$ multivariate time series is thereby transformed to N feature vectors $F_i = (x_{1,t_i}, x_{2,t_i}, \ldots, x_{M,t_i})$ of length M.

Following that, the feature vectors are normalized feature-wise. Dimensionality reduction like PCA or the handling of the imbalanced classes e.g. using SMOTE [37] did not have a positive effect on the results and was therefore not implemented in the final solution, but should generally be considered for an anomaly detection approach.

4.3. The ensemble

To address the requirements (a) of being robust against the data variability from different driving scenarios and (b) to detect known and unknown fault types, it is proposed to use ensemble classification [27]. The challenge to detect known and unknown fault types is addressed by incorporating two-class and one-class classifiers, where the latter ones have their strengths for the case of unknown fault types. The robustness w.r.t. different driving scenarios and different fault types is addressed by using a set of diverse classifiers. Different types of anomaly detectors are used: MOG and extreme value analysis focus on the detection of faults in the univariate time series, while the remaining six classifiers are capable to detect multivariate anomalies.

An independent ensemble is used with different base classifiers trained on the same training set, where the one-class classifers are trained on the subset of fault-free recordings. The selection of the base classifiers was done statically, based on considerations of the diversity of classifiers. The following four two-class classifiers are incorporated into the ensemble:

MOG. The mixture of Gaussians classifier is used since it is a simple method for the detection of univariate anomalies.

Naive Bayes. The naive Bayes classifier has been successfully used in many applications. It is used since it bases on the estimation of the data's distribution, in contrast to the used random forest or support vector machines, ensuring diversity between the base classifiers.

Random forest. A random forest was selected as a base classifier, since it is an ensemble method itself. By using decision trees it is entirely different from the other classifiers. Random forests have been used on automotive data, e.g. in [38]. The parameters were set as given in Table 1

Support vector machine. In order to be diverse to the previous classifiers, a support vector machine is used as a base classifier. SVMs find the decision function by determining the maximum margin without assumptions about the distribution of the data. They have been used for fault detection in many publications, e.g. in [39]. A two-class soft-margin SVM with RBF kernel is used. The parameters are tuned using grid search.

In order to address previously unseen faults, the following four one-class classifiers are used as base classifiers in the ensemble:

Extreme value analysis. This classifier was selected since it is a simple method to detect univariate anomalies. The required threshold is set to mean value $\pm 3\sigma$ per feature, i.e. feature values that deviate from the feature's mean value by more than 3σ are classified as anomaly.

Mahalanobis classifier. The Mahalanobis classifier was selected since it is a standard method for the detection of anomalies in multidimensional feature space. In order to use the Mahalanobis distance as a one-class classifier, the threshold for the distance is set to 3σ.

One-class support vector machine. One-class support vector machines have been successfully used for anomaly detection in many publications, therefore the SVM proposed in [35] is used as a base classifier in its variant with an RBF kernel. The required parameter ν was set to $\frac{1}{N}$, where N is the number of instances in the training set, which corresponds to the case, where no anomalies are expected in the training set. The kernel parameter γ was set to $\frac{1}{D}$, where D is the number of features, as done by [40].

Support vector data description. The one-class support vector machine SVDD with RBF kernel was successfully used for anomaly detection in automotive data of the same kind in [6] and was hence incorporated into the ensemble. Having to manually adjust the parameters would make SVDD non-applicable for the problem discussed in this paper, therefore autonomous parameter tuning is desired. The author of this paper proposed an approach to tune the SVDD parameters solely on the training set [41]. The approach bases on the observation that for the radius of SVDD's hypersphere a value of 1 can be considered as optimal over the entire value range. The trade-off between the optimal radius and the error rate is optimised to find the optimal set of parameters. This approach was shown to yield good decision boundaries for a variety of data sets [6].

For the case where the RBF kernel is used, SVDD and one class-SVM are similar [3], i.e. the difference here is solely the parameter tuning.

In accordance with the goal defined in Section 1, to have an out-of-the-box method without the need for user-defined parameters, the aim was to determine the parameters from the training set or to set them based on heuristics. The base classifiers used in the ensemble anomaly detector and their parameters are shown in Table 1.

	base classifier	parameters
two-class	MOG	-
	naive Bayes	-
	random forest	features at split = $\lfloor\sqrt{D}\rfloor$
		number of trees = 500
	SVM	ν, γ found by grid search
one-class	extreme value	threshold = $\pm 3\sigma$
	Mahalanobis	threshold = 3σ
	one-class SVM	$\nu = \frac{1}{N}$, $\gamma = \frac{1}{D}$
	SVDD	ν, γ found by autonomous param. tuning

Table 1: Classifiers used as base classifiers in the proposed ensemble anomaly detector, together with the setting of parameters (where N is the number of instances in the training set, D is the number of features and σ is the standard deviation).

4.4. The voter: combining classifier outputs

Each base classifier of the ensemble classifies the individual feature vectors and passes its result to the voter. The voter combines these crisp outputs to one result using

$$C_e = \sum_{i=1}^{k} w_i C_i \tag{2}$$

where k is the number of base classifiers C_i and w_i is the weight for each base classifier. C_i can take on the values 0 if classified as ω_n and 1 if classified as ω_a and C_e takes on values of $[0, 1]$. In this paper majority voting is used, i.e. all w_i are set to $w_i = \frac{1}{k}$. Classification is then done as follows, where in the case of a draw, the feature vector is classified as anomaly:

$$C_E = \begin{cases} \text{normal if} & C_e < 0.5 \\ \text{anomaly if} & C_e \geq 0.5 \end{cases} \tag{3}$$

As an alternative to majority voting, the approach can be enhanced towards a weighted voting by manually adjusting the weights w_i in (2) if there is evidence that certain base classifiers perform better or to focus on previously known or unknown faults by weighting the two-class and one-class classifiers differently.

The weights could also be tuned based on the training data. This approach was however not followed since there is no reason to assume that the faults in the training set are representative and a tuning of the ensemble towards these faults is hence not desired. As the anomaly detector is in operation it will detect more previously unseen faults. By incorporating these faults in the training set, the tuning of the weights is then an option for further improvement. The alternative to use a base classifiers' outputs and train an independent classifier, i.e. stacking, requires a larger amount of labelled data from both classes.

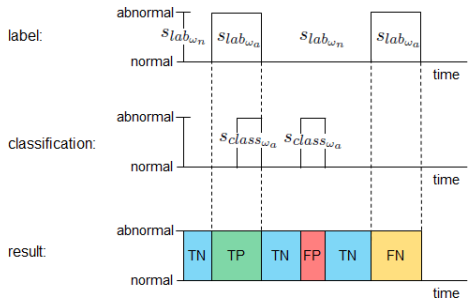

Figure 3: Classification of subsequences showing correctly and falsely detected anomalies (TP, FP), correctly detected normal subsequences (TN) and undetected anomalies (FN).

4.5. The filter

Since recordings from test drives are noisy, a filter is applied to the output of the ensemble. Recordings from similar situations show similar but not identical values. These values are likely to be normal, but were recorded in slightly different conditions regarding e.g. weather, road or driver behaviour. Classifying individual data points thereby yields a high number of falsely detected anomalies. An approach is needed, that compensates for small deviations of data points. The idea is to not classify individual data points, but to incorporate the local neighbourhood of the data points by working on subsequences. If the feature vectors of k subsequent time points are classified as anomaly, the current subsequence is classified as anomaly, where $k = 3$ in this work, which corresponds to a fault that is present for ≥ 3 seconds.

4.6. The sequencer: forming subsequences

Faults in the recordings can be of arbitrary length. Therefore the classification results are not based on individual time points but rather on variable-length subsequences that are formed by a sequencer. In the test set, consecutive data points of the same label are grouped together as variable-length subsequences and are denoted by $s_{lab_{\omega_n}}$ for normal subsequences and $s_{lab_{\omega_a}}$ for abnormal ones respectively.

The classification results are then determined as follows, where $s_{class_{\omega_a}}$ is a variable-length subsequence classified as abnormal and $s_{class_{\omega_n}}$ a subsequence classified as normal. An example is shown in Fig. 3.

$$\text{TP: } \exists s_{class_{\omega_a}} \subseteq s_{lab_{\omega_a}} \tag{4}$$

$$\text{FN: } \nexists s_{class_{\omega_a}} \subseteq s_{lab_{\omega_a}} \tag{5}$$

$$\text{FP: } \forall s_{class_{\omega_a}} \subseteq s_{lab_{\omega_n}} \tag{6}$$

$$\text{TN: } \forall s_{class_{\omega_n}} \subseteq s_{lab_{\omega_n}} \tag{7}$$

The described approach was applied to recordings from road trials, the results are presented in the next section.

Short name	Full OBD name
load	Calculated Load Value
coolant temp	Engine Coolant Temperature
STFT *	Short Term Fuel Trim (Bank 1)
MAP	Intake Manifold Absolute Pressure
rpm *	Engine RPM
speed	Vehicle Speed
ign timing adv *	Ignition Timing Advance
throttle	Absolute Throttle Position

Table 2: Vehicle signals used for the experiments on recordings from the vehicle in idle mode and overland drives, where the signals used for idle mode are marked by asterisks.

5. Experimental results

This section shows the detection of faults in recordings from vehicles. The results of the ensemble as well as of the individual one-class and two-class classifiers are given in the tables. The first two experiments work in a constrained environment, where data was recorded from the vehicle in idle mode. The third and fourth experiment use recordings from drives in overland traffic, i.e. drives within towns and on ordinary roads. The detetection of known and unknown fault types is investigated.

5.1. Data sets

A "Renault Twingo I" (model year 2002, 1149 ccm, 43 kW) was used as the test vehicle due to the easy accessibility of components in the engine bay. The data during the test drives were recorded using the on-board diagnostics interface according to ISO 15031 (OBD-II or EOBD).

The 8 recorded signals shown in Table 2 were used for the experiments, where the 3 signals marked by an asterisk were used for the experiments on idle mode. The signal *load* refers to the engine's load, *coolant temp* holds the temperature of the engine coolant, and *STFT* (short term fuel trim) is the injection pulse width, which keeps the air-fuel ratio optimal, i.e. the lambda value close to 1. The signal *MAP* is the manifold absolute pressure which is used to calculate the air mass flow rate, which in turn determines the fuel to be injected for optimal combustion. Furthermore, *rpm* is the engine revolution per minute, *speed* refers to the vehicle's speed and *ign timing adv* (ignition timing advance) measures the angle of the piston position where the ignition takes place. Finally the value of *throttle* holds the position of the throttle valve, which is directly proportional to the accelerator pedal position.

5.2. Injected faults

In order to validate the anomaly detection system, faults were injected into the test vehicle during drives in order to obtain recordings with errors. The faults were injected using a self-made device that allows for manipulations in the engine bay during a drive. This was done in two ways:

1. by interrupting connections in order to simulate the failure of a component or a cable break
2. by bypassing a sensor using a potentiometer in order to simulate an erroneous sensor

The focus was the detection of intermittent faults, so the faults were injected for a limited time span (several seconds to minutes). These intermittent faults are a challenge for fault diagnosis in practice, since the faults only occur under specific conditions and may have disappeared when the vehicle has returned to the plant or a repair shop. The simpler case of detecting permanent faults, i.e. faults that do not disappear after they have occurred, was not addressed.

The types of faults were chosen such that they manifest themselves as different types of anomalies as given in the categorization in Section 3.1.

Referring to the locations of potential faults identified in Fig. 1, the injected faults correspond to faults in the cable harness, actuators, or sensors. Four different types of faults were injected. One recorded drive is shown in Fig. 4.

Fault 1: erroneous injection. Misfiring by an erroneous or coked injector nozzle or a loose contact in wiring was simulated by switching off an injector nozzle for a short period of time, suppressing injection for one cylinder. As a countermeasure the engine control system adapts the injection pulse width, which is observable by a change in the signal STFT. For some of the injected faults this leads to values of STFT that are greater than the values in the training set, i.e. this leads to a type I anomaly. The remaining occurrences correspond to anomalies of type III (contextual anomaly in multivariate time series) and are only detectable by considering dependent signals.

Fault 2: erroneous ignition. A loose connection in the spark plug lead or a worn spark plug was simulated by interrupting the spark plug lead for a short period of time while the vehicle was standing still. This causes the engine control system to adapt, which is observable in the STFT signal. The signal increases but not to values that are out of the normal range, which makes this fault an anomaly of type III.

Fault 3: unavailable engine temperature. A loose contact in the wiring of the temperature sensor was simulated by interrupting the connecting wire, which leads to signal values out of the value range present in the training set. The fault manifests itself as an anomaly of type I (subsequence anomaly in univariate time series) in the categorization in Section 3.1.

Fault 4: erroneous engine temperature. An error in the sensor measuring the engine coolant temperature was introduced by adding positive or negative sensor offsets using a potentiometer. This fault corresponds to a type I anomaly.

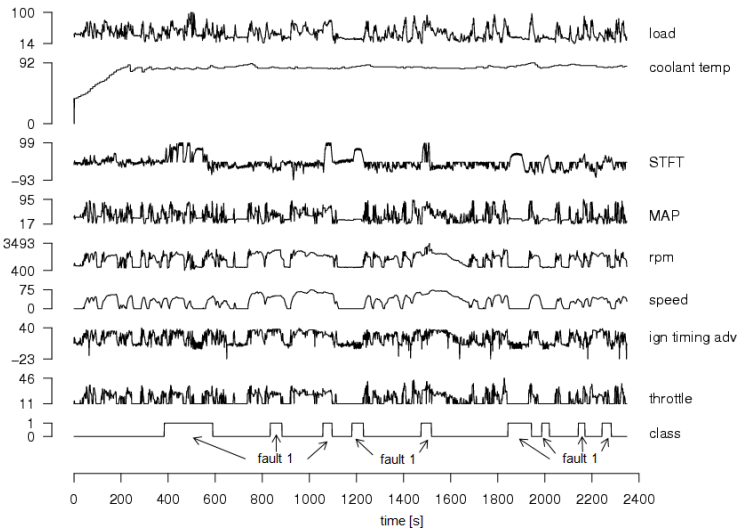

Figure 4: Faults of type 1, injected during an overland drive of approximately 40 minutes.

5.3. Setup of experiments

The two-class classifiers were trained on data sets containing normal data and faults, the one-class classifiers were trained on the same training set but without recordings containing faults. To be realistic, the splitting of training and test sets was not done on the basis of feature-vectors. An entire drive was assigned to either the training or test set, where no recording was part of both. Two scenarios were investigated: (a) the detection of known fault types, where recordings of the same fault types were included in the training and test set and (b) detection of unknown fault types, where recordings of previously unseen fault types were included in the test set.

Experiments for the following setups were conducted:

1. idle mode with known fault types
2. idle mode with unknown fault types
3. overland drives with known fault types
4. overland drives with unknown fault types

For the first and the third experiment the fault types 1 and 2 were included in the training and test set for the two-class classifiers. The second and fourth experiment address the problem of not being able to model all possible faults, since many faults are either unknown or there are no recordings available. Therefore the test set exclusively contains faults of the types 3 and 4, which were not included in the training set.

The results of the individual two-class- and one-class-classifiers, as well as the results of the ensemble are reported in one table for each of the four scenarios.

The following metrics are presented in the results: the true anomaly rate (TPR), which gives the percentage of detected anomalies and the precision

	classifier	TPR	prec	F2-score
two-class	MOG	100	100	**100**
	naive Bayes	100	42.9	78.9
	random forest	100	60.0	88.2
	SVM	100	92.3	98.4
one-class	extreme value	100	100	**100**
	Mahalanobis	100	100	**100**
	one-class SVM	100	34.3	72.3
	SVDD	100	92.3	98.4
ensemble	all	100	46.2	81.1

Table 3: Experiment 1: classification results on recordings from idle mode.

(*prec*), which expresses the percentage of true faults in the result set of reported anomalies.

Anomaly detection is the trade-off between missing faults and falsely reporting normal instances as anomalies. In addition to TPR and precision, as a single figure to evaluate the approach, the F2-score is used (eq. (8)). The F2-score incorporates TPR and precision while putting more emphasis on the detection rate. For all used metrics, 100% is the optimum.

$$\text{F2-score} = \frac{5 * prec * TPR}{4 * prec + TPR} \tag{8}$$

5.4. Experiments on recordings from idle mode

The first expmeriment is the detection of known fault types from a vehicle in idle mode. The results are shown in Table 3. All classifiers detected all injected faults, but the precision varies between the classifiers. MOG, extreme value analysis and Mahalanobis detected all faults and did not classify a single normal instance as anomaly. For the one-class SVM the precision and F2-score is the lowest, i.e. a high number of normal instances were falsely classified as anomaly. The ensemble's F2-score is below the F2-score of the best base classifiers.

The second experiment incorporates fault types that were not previously included in the training set. The results are given in Table 4. The F2-score of two of the two-class classifiers descreased to 34.1% and 44.4%. The one-class classifiers' results decreased but are still at a useful level, having detected the majority of the faults.

5.5. Experiments on recordings from overland drives

The third experiment bases on recordings from overland drives. Overland drives are much more variable than idle mode, which in general is expected to lead to more normal instances being falsely classified as faults. This is confirmed by the results shown in Table 5, where most of the classifiers show a lower precision. Naive Bayes, one-class SVM and SVDD yield reasonably high F2-scores, while detecting 89.5% of the faults. The Mahalanobis classifier performs poorly, which could point to the threshold not being appropriate for the data

	classifier	TPR	prec	F2-score
two-class	MOG	30.0	75.0	34.1
	naive Bayes	90.0	69.2	**84.9**
	random forest	80.0	100	**83.3**
	SVM	40.0	80	44.4
one-class	extreme value	90.0	75.0	**86.5**
	Mahalanobis	90.0	60.0	81.8
	one-class SVM	60.0	60.0	60.0
	SVDD	60.0	85.7	63.8
ensemble	all	80.0	57.1	74.1

Table 4: Experiment 2: classification results on recordings from idle mode, where the test set solely contains fault types that were not included in the training set.

	classifier	TPR	prec	F2-score
two-class	MOG	47.4	69.2	50.6
	naive Bayes	89.5	47.2	**75.9**
	random forest	94.7	25.4	61.2
	SVM	73.7	35.9	60.9
one-class	extreme value	42.1	25.8	37.4
	Mahalanobis	73.7	7.1	25.6
	one-class SVM	89.5	60.7	**81.7**
	SVDD	89.5	43.6	**73.9**
ensemble	all	89.5	35.4	68.5

Table 5: Experiment 3: classification results on recordings from overland drives.

set. The ensemble shows a high detection rate but misclassified a number of normal subsequences as faults, resulting in a medium F2-score.

The fourth experiment is a scenario that is highly relevant in practice. The test set of the recordings of overland drives solely contains fault types that are not included in the training set. The results of the two-class classifiers become useless as shown in Table 6. None of the unknown faults were detected by three of the classifiers.

The one-class classifiers detected the majority of the unknown faults with a reasonably high F2-score. For this experiment, that is viewed as the most relevant one, the ensemble's F2-score is better than any of the individual classifiers.

In addition to the classification results, the ensemble's diversity for the four

	classifier	TPR	prec	F2-score
two-class	MOG	0	0	0
	naive Bayes	50.0	30.0	44.1
	random forest	0	0	0
	SVM	0	0	0
one-class	extreme value	83.3	35.7	65.8
	Mahalanobis	100	14.6	46.2
	one-class SVM	83.3	71.4	**80.6**
	SVDD	83.3	71.4	**80.6**
ensemble	all	83.3	83.3	**83.3**

Table 6: Experiment 4: classification results on recordings from overland drives, where the test set solely contains fault types that were not used during training.

experiment	diversity E
idle mode with known fault types	0.093
idle mode with unknown fault types	0.139
overland drives with known fault types	0.123
overland drives with unknown fault types	0.223

Table 7: The ensemble's diversity measure for the different experiments. The diversity's value range is 0...1, where higher values indicate higher diversities.

experiments is investigated. The diversity measure entropy (1) is used showing how much the base classifiers disagreed (see Table 7).

5.6. Discussion of results

In this section the results of the base classifiers and the ensemble are discussed on basis of the F2-scores. The F2-scores of all classifiers for all four experiments are shown in Fig. 5. There is no one classifier that performs best in all experiments. Some of the classifiers behave extremely volatile w.r.t. the different experiments, e.g. MOG and SVM, confirming the challenge regarding variability of the data from different setups (Section 1).

From the F2-scores a robustness measure is derived by

$$\text{robustness} = 100 - \text{range}(\text{F2-score}) \tag{9}$$

yielding values in the range of 0..100%, where high values indicate high robustness over the different experiments. The robustness w.r.t. the F2-score mean values over all experiments is shown in Fig. 6. SVDD has the best F2-score averaged over all setups (79%), indicating that the autonomous parameter tuning [41] yields good parameters. The robustness however is at a value of only 65%. The ensemble has the second best average F2-score (77%) and has the major benefit of a high robustness over all experiments (85%), i.e. the variance of the results is the lowest. This shows the effectiveness of the proposed ensemble method to tackle the variability problem introduced in Section 1.

Not surprisingly, the two-class classifiers performed best for the experiments with known fault types in a constrained environment (idle mode). The setup regularly encountered in practice, where fault types occur that were previously not included in the training set shows the limitations of the two-class classifiers. Three of the four two-class classifiers failed to detect any previously unseen fault types in overland drives.

On the other hand, One-class classifiers make no assumptions about potential faults, all instances that deviate from the training set are reported as anomalies. For that reason one-class classifiers performed well for unknown fault types. In specific, the one-class SVM and SVDD were stable for all four setups.

As shown in Table 7, the ensemble has a higher diversity for the experiments with previously unseen fault types. The shortcomings of the two-class classifiers are compensated by the one-class classifiers in that setup. This shows the effectiveness of the approach of combining one-class and two-class classifiers for the challenge to detect known and unknown fault types, as introduced in Section 1.

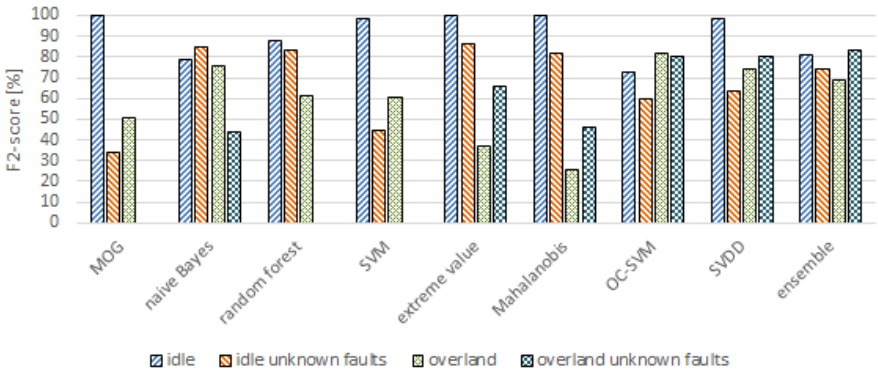

Figure 5: F2-scores for all classifiers in all of the four experiments: (1) idle mode with known fault types, (2) idle mode with unknown fault types, (3) overland drives with known fault types, and (4) overland drives with unknown fault types.

The major benefit of the ensemble is the robustness against the different setups of known and unknown fault types as well as its robust performance on recordings from a constrained environment (idle mode) or from an unconstrained environment (overland drives). The results of the individiual classifiers are volatile. While for specific setups, some of the individual classifers outperform the ensemble, the ensemble supplies the most stable results over all experiments making it most appropriate for the problem addressed in this paper.

6. Conclusion

In this paper anomaly detection in multivariate time series from vehicle tests was addressed. It was shown how manifold the potential faults in vehicles can be and which types of anomalies can be present in the data.

An anomaly detection approach was proposed that detects different faults of known and unknown fault types in various driving conditions and works without setting of expert-parameters. An ensemble anomaly detector was created consisting of two-class and one-class classifiers in order to detect both, fault types that were included in the training set and previously unseen fault types. The base classifiers' parameters were either pre-defined or determined from the training set making the ensemble and out-of-the-box approach.

The approach was validated on recordings from road trials. The results show that the individual base classifiers are sensitive to the given scenario. In general, the two-class classifiers yield good results for known fault types while the one-class classifiers perform best for previously unseen fault types. The individual classifier performing best over all tested scenarios is the one-class support vector machine SVDD with an autonomous parameter tuning approach. The ensemble anomaly detector yielded a high F2-score with the major benefit that the results were stable over all experiments with known and unknown fault types in idle mode and for overland drives.

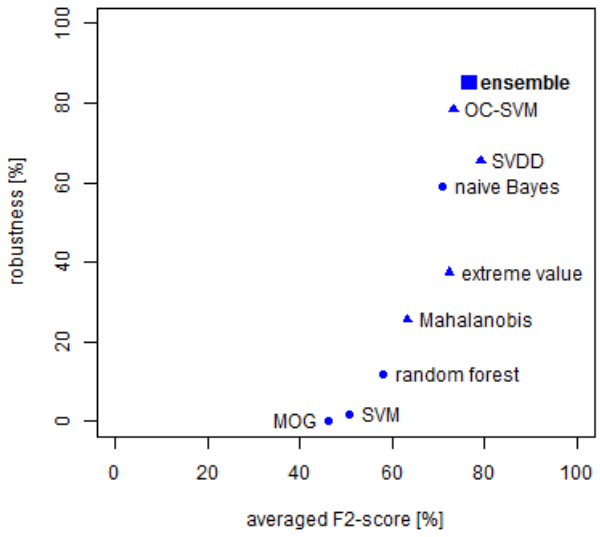

Figure 6: Robustness w.r.t. F2-score averaged over all of the four experiments for the 8 base classifiers and the ensemble. With 79% SVDD has the highest F2-score, with a robustness of 65%. The ensemble has a high F2-score of 77% with the benefit of the highest robustness of 85%, showing the effectiveness of the ensemble. Two-class classifiers are shown by circles, one-class classifiers by triangles.

While the approach was designed for offline-analysis of recordings from road trials, applications like predictive maintenance or condition monitoring can base on this work.

The approach in this paper was designed for the offline-analysis of recordings from road trials. However, applications requiring anomaly detection in an online-manner can also benefit from this approach. Examples are predictive maintenance and condition monitoring. In addition to automotive systems, the ideas discussed in this paper are applicable to related domains like fault detection in industrial applications.

References

References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Computing Surveys 41 (2009) 15:1–15:58.

[2] J. Suwatthikul, R. McMurran, R. Jones, In-vehicle network level fault diagnostics using fuzzy inference systems, Applied Soft Computing 11 (2011) 3709 – 3719.

[3] D. M. Tax, One-class classification. Concept-learning in the absence of counter-examples, Ph.D. thesis, Delft University of Technology, 2001.

[4] M. Mueter, N. Asaj, Entropy-based anomaly detection for in-vehicle networks, in: Intelligent Vehicles Symposium, IEEE, 2011, pp. 1110–1115.

[5] F. Cong, H. Hautakangas, J. Nieminen, O. Mazhelis, M. Perttunen, J. Riekki, 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.

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

[7] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S. N. Kavuri, A review of process fault detection and diagnosis: Part I: Quantitative model-based methods, Computers and Chemical Engineering 27 (2003) 293–311.

[8] Z. Zhao, K. G. Mehrotra, C. K. Mohan, Ensemble algorithms for unsupervised anomaly detection, in: Current Approaches in Applied Artificial Intelligence: 28th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer International Publishing, 2015, pp. 514–525.

[9] R. Polikar, Ensemble based systems in decision making, IEEE Circuits and Systems Magazine (2006).

[10] J. Han, M. Kamber, J. Pei, Data Mining - Concepts and Techniques, 3 ed., Morgan Kaufmann Publishers, 2011.

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

[12] A. Theissler, Anomaly detection in recordings from in-vehicle networks, in: Big Data Applications and Principles (BIGDAP 2014), 2014.

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

[14] S. Byttner, T. Rgnvaldsson, M. Svensson, Consensus self-organized models for fault detection (COSMO), Engineering Applications of Artificial Intelligence 24 (2011) 833 – 839.

[15] D. Jung, K. Y. Ng, E. Frisk, 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.

[16] D.-I. Curiac, C. Volosencu, Ensemble based sensing anomaly detection in wireless sensor networks, Expert Systems with Applications 39 (2012) 9087–9096.

[17] H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, A. Liotta, Ensembles of incremental learners to detect anomalies in ad hoc sensor networks, Ad Hoc Networks 35 (2015) 14–36.

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

[19] A. Soudi, W. Khreich, A. Hamou-Lhadj, An anomaly detection system based on ensemble of detectors with effective pruning techniques, in: IEEE International Conference on Software Quality, Reliability and Security, IEEE Computer Society, 2015, pp. 109–118.

[20] B. Krawczyk, M. Woźniak, Dynamic classifier selection for one-class classification, Knowledge-Based Systems 107 (2016) 43–53.

[21] M. Antal, L. Z. Szabo, An evaluation of one-class and two-class classification algorithms for keystroke dynamics authentication on mobile devices, in: 20th International Conference on Control Systems and Science, 2015.

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

[23] ISO 26262-1, ISO 26262: Road vehicles – Functional safety – Part 1: Vocabulary. Final Draft., 2011.

[24] V. J. Hodge, J. Austin, A survey of outlier detection methodologies, Artificial Intelligence Review 22 (2004) 2004.

[25] E. Keogh, J. Lin, S.-H. Lee, H. V. Verle, HOT SAX: finding the most unusual time series subsequence: algorithms and applications, Knowledge and Information Systems 11 (2006) 1–27.

[26] S. Laxman, P. Sastry, A survey of temporal data mining, Sadhana 31 (2006) 173–198.

[27] A. Jurek, Y. Bi, S. Wu, C. Nugent, A survey of commonly used ensemble-based classification techniques, The Knowledge Engineering Review 29 (2013) 551–581.

[28] L. I. Kuncheva, C. J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine Learning 51 (2003) 181–207.

[29] C. C. Aggarwal, Outlier Analysis, Springer, 2013.

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

[31] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

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

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

[34] M. M. Moya, D. R. Hush, Network constraints and multi-objective optimization for one-class classification, Neural Networks 9 (1996) 463–474.

[35] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (2001) 1443–1471.

[36] D. Tax, R. Duin, Support vector data description, Machine Learning 54 (2004) 45–66.

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

[38] P. Bergmeir, C. Nitsche, J. Nonnast, M. Bargende, Classifying component failures of a hybrid electric vehicle fleet based on load spectrum data, Neural Computing and Applications 27 (2016) 2289–2304.

[39] J. Liang, R. Du, Model-based fault detection and diagnosis of {HVAC} systems using support vector machine method, International Journal of Refrigeration 30 (2007) 1104 – 1114.

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

[41] A. Theissler, I. Dear, Autonomously determining the parameters for SVDD with RBF kernel from a one-class training set, in: WASET International Conference on Machine Intelligence 2013, Stockholm., 2013, pp. 1135–1143.