CHALMERS

Anomaly Detection in Logged Sensor Data

Master's thesis in Complex Adaptive Systems

JOHAN FLORBÄCK

Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2015

MASTER'S THESIS IN COMPLEX ADAPTIVE SYSTEMS

Anomaly Detection in Logged Sensor Data

JOHAN FLORBÄCK

Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous Systems
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2015

Anomaly Detection in Logged Sensor Data
JOHAN FLORBÄCK

Master's thesis 2015:35
ISSN 1652-8557
Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers Reproservice
Göteborg, Sweden 2015

Anomaly Detection in Logged Sensor Data
Master's thesis in Complex Adaptive Systems
JOHAN FLORBÄCK
Department of Applied Mechanics
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

Anomaly detection methods are used in a wide variety of fields to extract important information (e.g. credit card fraud, presence of tumours or sensor malfunctions). Current anomaly detection methods are data- or application specific; a general anomaly detection method would be a useful tool in many situations.

In this thesis a general method based on statistics is developed and evaluated. The method includes well-known statistical tools as well as a novel algorithm (*sensor profiling*) which is introduced in this thesis. The general method makes use of correlations found in complex sensor systems, which consists of several sensor signals. The method is evaluated using real sensor data provided by *Volvo Car Corporation*. The sensor profiling can be used to find clusters of data with similar probability distributions. It is used to automatically determine the sensor performance across different external conditions.

Evaluating the anomaly detection method on a data set with known anomalies in one sensor signal results in 94 % of anomalies detected at 6 % false detection rate. Evaluating the method on additional sensor signals was not done. The sensor profiling revealed conditions where the sensor signal behaves qualitatively and quantitatively different. It is able to do this in data where other commonly used methods, such as regression analysis, fail to extract any information. Sensor profiling may have additional applications beyond anomaly detection as it is able to extract information when other methods can not.

To conclude, this thesis presents a seemingly natural method and tool chain to automatically detect anomalies in any sensor data that can be represented as a time series. The performance of this method is still to be proven on a large set of general sensor data, but it shows promise, mainly for sensor systems consisting of several sensor signals.

Keywords: Anomaly detection, Regression analysis, Sensor profile, Distribution clustering, Sensor validation, Sensor model, ROC-curve

Acknowledgements

First and foremost, I would like to thank my supervisor and examiner.
Lars Tornberg, for guidance, ideas, discussions and continuous feedback on all aspects of the project.
Krister Wolff, for feedback on the report and structure of the project.

Tony Heimdahl, for originating the project and introducing me to the subject.

Mikael Kjellgren and the rest of the **Sensor Validation Group at Volvo** for providing the data together with useful insights regarding anomalies.

Fredrik Elofsson for feedback on the project and the report.

Contents

1 Introduction

A recent trend in the automotive industry is the development of autonomous driving and active safety functions. Active safety functions can warn the driver or even take control of the vehicle when it senses that danger is imminent. Active safety taken to the extreme allows the vehicle to drive autonomously in most or all situations. In order to ensure the performance of the active safety functions the sensors which monitor the traffic environment need to be reliable. Testing and evaluating the sensors is an important step needed for the creation of these active safety functions. Especially as a false function intervention, caused by a sensor malfunction, may cause an accident instead of preventing one.

Testing of sensors for active safety or autonomous driving implies long test driving with extensive data collection. Failure of active safety functions may be an indication of the failure of the sensors. However, these failures may not occur during field tests, even when the test is extensive. If the sensor performance is degraded for some reason, this does not necessarily happen at the moment when danger is imminent and the functions intervene which causes possible sensor malfunctions to go unnoticed. These malfunctions can be seen as anomalies in the entire set of logged sensor data, at least if we use the general definition that *anomalies are points or patterns in a data set that differs from the expected, normal, behaviour*. Analysing anomalies in logged sensor data may lead to important insights in the behaviour of the sensors. Gaining such insights may, in turn, lead to more accurate risk assessments and improved performance of the active safety functions. Finding underlying factors and frequency of these anomalies can also be used for creating sensor models that more accurately simulate real sensor behaviour. Since the logged sensor data is extensive and these anomalies occur rarely, an automated anomaly detection method is needed to find them.

Anomaly detection methods are already used for detecting everything from network intrusion [1] and credit card fraud [2] to the presence of malignant tumours [3] or sensor failure in space crafts [4]. The reason anomaly detection methods are used in many fields is that anomalies, independent of cause, contain valuable information and anomaly detection, both in logged signals and in real time, is needed to extract and utilise this information.

Since anomalies are present in many different types of signals and applications there is no exact notion of an anomaly. Converting the general and abstract definition of an anomaly to a stringent definition that captures all anomalies even within a single application may prove difficult. This is one of the reasons that most anomaly detection methods used are based on machine learning. Another issue is that since anomalies are in general rare, manually creating a data set large enough so that it captures all types of anomalies and normal behaviour is both time consuming and difficult. Hence many anomaly detection methods have to operate in a so-called *unsupervised* setting [5].

1.1 Purpose

The aim of this project is to develop scientifically sound methods to identify "normal working mode" and "failure mode", in large amounts of logged sensor data from field tests. Sensor failure can be seen as anomalous sensor behaviour. The logged data only contains the sensor data and no external knowledge of the monitored system is obtainable. The methods should be general enough to be used on any type of sensor signal, even if some modifications has to be made.

1.2 Scope

After a literature study a general anomaly detection method is chosen and implemented. The chosen method is evaluated using a single sensor signal. Comparing several methods using several sensor signals to draw general conclusions would be interesting but is beyond the scope of this project.

1.3 Outline

This thesis is organised as follows. Chapter 2 contains background to anomaly detection as well as an introduction to the concepts used and related work. It concludes with a short description of the method proposed in this thesis and motives behind the selection of the method. Chapter 3 describes the selected method and the needed components. Chapter 4 provides a deeper understanding of the first part of the anomaly detection method, namely regression analysis. Chapter 5 describes the novel method i.e. sensor profiling,

introduced in this thesis. Chapter 5 also includes motives behind the methodology, an extensive example and a description of the algorithms used. In Chapter 6 the results from evaluating the method using a single sensor signal are presented. These results, the general method and further uses of some of the individual methods used are discussed in Chapter 7. Finally, future work and possible applications are presented in Chapter 8.

1.4 Contributions

This thesis provides a structured framework for a general anomaly detection method for time series data. The method is general and can be applied to any type of time series data and is not limited to sensor data. In this thesis a novel method for establishing relationships between variables and the distribution of a separate stochastic variable is also introduced. This method can be viewed as a distribution clustering algorithm. The applications of this method may extend beyond the field of anomaly detection and can, for example, be used to determine the performance of a sensor under different conditions.

2 Background and Related Work

The goal of this chapter is to establish a foundation for the understanding of the concepts and methods used throughout the thesis, as well as giving the reader an understanding of the motive behind the selected method. This is done by discussing the concepts and creating an overview of the methods used for detecting anomalies in time series in general.

2.1 Concept Introduction

In this section concepts related to anomaly detection and sensor evaluation will be introduced and discussed. This provides the basic understanding needed for remainder of the thesis.

2.1.1 What is an Anomaly?

Anomalies are points or patterns in a data set that differs from the expected, "normal", behaviour. The cause and appearance of an anomaly varies depending on the data and application and there is no exact notion of an anomaly. Converting the general and abstract definition of an anomaly to a stringent definition that captures all anomalies even within a single application may prove to be too difficult. What can be done is to classify anomalies into three different types as follows [5]:

Point anomalies

A point anomaly is a data point that lies outside the boundary of the normal regions. Hence it can be considered anomalous with regard to all the other data points. An example would be measuring the outside temperature in Sweden to be 50°C.

Contextual anomalies

A contextual anomaly is a data point that is considered anomalous in a specific context, but not otherwise. In order to define the context the data has to be structured in some way. A common example is time series, where points close to each other in time are expected to have similar behaviour. An example of a contextual anomaly would be measuring the outside temperature in Sweden to be 20°C in December.

Collective anomalies

A subset of related data points that can be considered anomalous with regards to the entire set of points is called a collective anomaly. An example would be measuring the outside temperature in Sweden to be 16°C constantly (day and night) during the entire month of May. Even though this temperature during May is not considered anomalous by itself.

2.1.2 Handling Contextual Anomalies

Most anomaly detection methods found in the literature are focused on detecting point anomalies, the research on contextual anomaly detection methods has been limited [5]. However, there are still several methods used to handle contextual anomalies found in the literature, these can be crudely divided into two approaches.

The first approach is to define the possible contexts and apply point anomaly detection methods on each context. Using the example with outside temperature, the data could be grouped by month and analysed individually to find both point and contextual anomalies.

The second approach is to model the normal structure in the data, this model is then used to reduce contextual anomalies to point anomalies. For example when finding anomalies in temperature readings a model is learned from training data, which can predict expected temperatures, maybe based on previous temperatures and month. If there is a large difference between the expected and the predicted reading, an anomaly is declared.

The second approach is more suitable for time-series data, and other cases, where breaking up the data into contexts is non-trivial [5]. As this is true for sensor data, the methods used in this thesis for anomaly detection all fit into this second category.

2.1.3 Sensor Data Characteristics

As different anomaly detection methods are used in widely different applications it is natural that these detection methods have different prerequisites. The choice of the anomaly detection method depend on both the intended use and the characteristics of the data that is analysed. Besides the possible restriction of detecting anomalies in real-time or in logged data, the conditions can described using *features* and *labels* [6].

A single data instance, record, may consist of one or several features e.g. sensor data that includes the sensor reading, a confidence value and the location of the sensor. In this example, the sensor data at a single time step is the record. The sensor reading, confidence and location are the features of a record. Features used for predicting future sensor readings are in this thesis called predictors [7], this terminology originates from regression analysis, described in Chapter 4. Features can also be used to find regions in the feature space where the sensor exhibits similar behaviour, these regions are, in this thesis, called sensor profiles.

A record can also have labels, an example would be an e-mail being labelled as "spam" or "non-spam". A more relevant example would be sensor data labelled as "anomalous" or "normal", perhaps along with either the label "simulated" or "logged" depending on the origin of the data. Certain anomaly detection methods require labelled data to train a classifier to distinguish between anomalous and normal data instances. This is called supervised learning and is often done using a classifier algorithm. The case with missing labels is called unsupervised learning and is often achieved through clustering algorithms. There are also algorithms that operates in a semi-supervised setting where some of the records are labelled. Labelling sensor data as "anomalous" or "normal" is most often a manual task, and since the data sets are in many cases extensive, most anomaly detection methods used are unsupervised [5].

2.2 The Lack of Ground Truth

Finding anomalies in the sensor output by comparing the output to the true reading, ground truth, is of course an accurate and intuitive method. In field tests, access to ground truth can be limited or missing. In this case, other methods must be used. In this thesis it is proposed that the sensor output is compared to a predicted sensor output. The accuracy of the model that creates the predicted output, the predictor, is of great importance to correctly evaluate the sensor performance. In general, any predictor could be used as long as it predicts normal sensor behaviour correctly. The choice of possible predictors ranges from the simple assumption-based predictors, such as the naïve predictor or a constant velocity model, to more complex predictors, like an extended Kalman filter or predictors based on artificial neural networks [5]. The most suitable "predictor" for logged sensor data may include future readings, an example is using a moving average as "predictor". This type of predictor may be the best suited for logged data, but is unable to predict sensor readings in real time. Hence, the best choice of predictor will depend not only on the normal behaviour of the sensor but also its applications.

2.3 Predicting Sensor Output

Predicting the sensor output and computing the prediction error, i.e. the difference between the predicted sensor output and the measured sensor output, is not a necessary step for all anomaly detection methods. It is, however, a way to compute an additional feature that is useful for many anomaly detection methods. Note that several predictors could be used to compute several additional features as well.

2.3.1 Assumption Based Predictors

Some simple predictors can be constructed from assumptions regarding the signal to be analysed. Assuming a stationary reading, a constant change (velocity) or a constant acceleration results in one of the following prediction models.

The Naïve Predictor

One of the simplest predictors imaginable is the naïve predictor, where the predicted sensor output, $\hat{Y}_i$ is equal to the previous measured sensor output,Y_{i-1}, i.e. $\hat{Y}_i = Y_{i-1}$. For a sensor measuring stationary entities this predictor is sufficient since using this predictor is equivalent to looking at the difference between two sequential

readings to find anomalies. If the end goal is to find anomalies magnitudes above normal variations, this is probably a good choice for the predictor.

The Constant Velocity Model

As the name suggests the constant velocity model works under the assumption that the velocity will remain constant between two sequential readings, i.e $\hat{Y}_i = 2Y_{i-1} - Y_{i-2}$. The accuracy of this predictor will stand or fall with the validity of the assumption. Hence this predictor may be suitable for certain sensor readings while being too inaccurate for others.

The Constant Acceleration Model

The final assumption-based model is the constant acceleration model, $\hat{Y}_i = \frac{3}{2}Y_{i-1} - Y_{i-2} + \frac{1}{2}Y_{i-3}$. It is more complex, for better or worse, than the other models and may also be suitable for certain sensor readings.

2.3.2 Regression Analysis

Regression analysis is the most commonly used method for predicting time series [5]. Regression analysis is a general statistical set of techniques used for finding relationships among variables, more specifically the relationships between a set of predictor variables, $X_i^1, X_i^2, \ldots, X_i^m$, and one or more output variables, Y_i. The result is a regression model for expressing the predicted sensor reading at single time step as a function of the predictor variables, $\hat{Y}_i = f(X_i^1, X_i^2, \ldots, X_i^m)$. Worth noticing at this point is that if the previous sensor reading, the discrete derivative and the discrete acceleration is included in the predictor variables the regression analysis will result in the most suitable of the three assumption-based models as long as there is no other predictive model that is more statistically accurate.

2.3.3 Other Predictor Models

Other predictive models are used in the litterature as well. A Kalman Filter is a recursive filter that approximates the state of a system based on noisy measurements [8]. Dynamic Bayesian Networks can be seen as generalized Kalman filters or Markov models and estimates the probability of an output based on the state of a system. It can also predict the most likely output based on the estimated state [9]. Artificial Neural Networks can also be used to predict time series when trained using historical data. Such networks are able to capture non-linear relationships and are used for predicting financial time series [10]. Simpler methods for approximating the true value based on a noisy signal are smoothing filters. Using the moving average or spline interpolation will not predict future output, but can still be used to estimate the expected sensor output in logged data [11].

2.4 Detecting Anomalies

In this section, various anomaly detection techniques will be presented and their strengths and weaknesses will be discussed. This will stand as a foundation for the choices made in this thesis. For a more thorough review of the different anomaly detection methods and their uses, consider one of the survey articles available within this field, for instance [5].

2.4.1 Classification Based

Methods based on classification tries to directly classify a record as either "normal" or "anomalous". Examples of classifiers are Artifical Neural Networks, most often Feed-Forward Perceptron Networks, Bayesian Networks, Support Vector Machines and rule based classifiers, where the rules may be set using either machine learning or human expertise. Most often the classifiers train on a data set that represents all possible normal behaviour, without containing any anomalies. If new records can not be classified into any one class of normal behaviour it is classified as an anomaly. A classifier can also be trained using a data set consisting of normal and anomalous records labeled accordingly to create a classifier able to accurately classify future record as either normal or anomalous directly. The main drawback of these methods is that they all, except from the "human-decided-rules"-based, need large labelled data sets. Since these labelled data sets often needs to be created manually they are often non-existing.

Strengths

- Fast testing phase
- Can be used in real-time
- Many powerful algorithms

Weaknesses

- Requires labelled data
- Computationally heavy training phase
- Anomaly detection is secondary goal
- Most determines the class of the record in a binary fashion, without any certainty score attached

2.4.2 Nearest Neighbour Based

Anomaly detection methods based on Nearest Neighbour analysis relies on the following assumption: *normal records occurs in dense neighbourhoods, while anomalies occur far from their neighbours.*

The main issue, that needs to be resolved, is to determine a suitable distance (similarity) measure. Simple Euclidean distance in a sub-space of features is often used but other measures exists as well. Almost any similarity measure can be used, however it is usually positive-definite and symmetric even if it does not satisfy the triangle inequality. The analysis often consists of either analysing the distance to the k nearest neighbours or analyse the number of records within a specific distance d of the record that is to be tested. This is usually a computationally heavy task and the performance relies on the distance measure. Defining the distance measure may also be challenging, especially when the data is large or complex.

Strengths

- Unsupervised
- Data driven (no assumptions regarding distribution of the points)
- Easily implemented

Weaknesses

- Computationally heavy testing phase
- Defining distance between points may be challenging
- Some methods needs large and representative training set without anomalies

2.4.3 Clustering Methods

Clustering-based anomaly detection methods defines clusters for the normal data instances and compares new records to these clusters. There are two main approaches for finding anomalies using cluster analysis. The first is to cluster the entire data set and analyse the density of each cluster, normal records are assumed to belong to large and dense clusters while anomalous records belong to sparse or small clusters. The second approach is based on the assumption that anomalous records are located further from the center of the cluster they belong to than normal records.

Clustering techniques often need a distance measure in the same way as Nearest Neighbour analysis, the main difference between the two is that in Cluster analysis a record is compared to the different clusters and not each record. This makes Cluster analysis faster when it comes to the testing phase.

Strengths

- Unsupervised
- Fast testing
- Can be used in real-time

Weaknesses

- Optimised to find clusters, not anomaly detection
- Computationally heavy to train
- Same challenges with the distance measure as nearest neighbour analysis
- Significant anomaly cluster may be undetected

2.4.4 Statistical Methods

The key assumption behind any statistical anomaly detection technique is: *Normal records are located in high probability regions of a stochastic model and anomalies occur in the low probability regions of the model* [5].

Statistical anomaly detection methods consists of two steps, first a statistical model (usually only for the normal behaviour) is fitted to the data. In the second step new data instances are tested against this model in order to determine the probability of the occurrence of that particular record. In time-series, the statistical model used is often a regression-based model, and anomaly detection methods has been applied to a number of regression models: *Autoregressive*, AR, [12, 13] *Autoregressive Moving Average*, ARMA, [14] and *Autoregressive Integrated Moving Average*, ARIMA [15, 16]. However, models using external predictors, ARX, ARMAX and so on, has not been as thoroughly investigated in the litterature.

A (Dynamic) Bayesian Network can be used for finding anomalies in time-series as well [9]. Bayesian Networks, often claimed by machine learning, are statistical models that can be used both for predicting outputs or determining the probability of a specific output. They can be seen as generalised Kalman Filters or even generalised Hidden Markov Models and is a way to apply Bayes' Rule [17] on complex problems [18].

Other statistical methods are more suitable for detecting point anomalies, such as those that estimate the distribution of normal records directly in some way and then compare the record to the distribution, examples of this is Gaussian Models or histogram-based techniques.

Strengths

- Results are given with a confidence
- If assumptions about data distribution holds,the results are statistically justifiable
- Can be unsupervised

Weaknesses

- Rely on assumptions about the data distribution that may not hold true
- Selecting the most appropriate statistical test is nontrivial
- Multi-variate contextual anomalies can slip through

2.5 Proposed Method for Detecting Anomalies

In this section a proposed general method for detecting sensor malfunction is introduced and the motives behind selecting this method is discussed. The work flow of the method is shown in Figure 2.1.

Using Linear Regression Analysis to find the Predictive Model

As regression analysis is a fast and general method that produces a model that can be statistically evaluated using proven methods [7, 19], it is proposed as a general prediction method that can be applied to any sensor reading. The predictive model is used to compute a predicted sensor output. The predicted sensor output is compared to the measured sensor output to compute the prediction error.

As sensors for active safety become more advanced the amount of available sensor data increases. There can be statistically significant correlations between sensor readings from one or several sensors that are unforeseen or even, in human eyes, illogical. These correlations can be found by regression analysis and be used to more accurately predict the sensor output. The resulting model can be analysed, understood and applied in other contexts leading to insights about the sensor system that may otherwise pass undetected.

Sensor Profiling

The evaluation of the sensor performance can also be extended beyond analysing the raw difference between the predicted and the measured sensor reading. An additional step is to determine so-called sensor profiles, that is feature regions where the sensor behaviour is similar within the sensor profile and different across sensor profiles. Different measuring conditions may result in different noise levels in the measurement of a sensor, for instance, a thermometer for outdoor use may be more accurate when measuring temperatures between $-15°$C and $+30°$C than when measuring temperatures below $-30°$C. Accuracy may also depend on the rate of change of the temperature or other conditions, such as the humidity or whether or not wind or sun light hits the sensor. Finding these sensor profiles can be done manually, using expertise in the specific sensor, or automatically using the novel method introduced in this thesis.

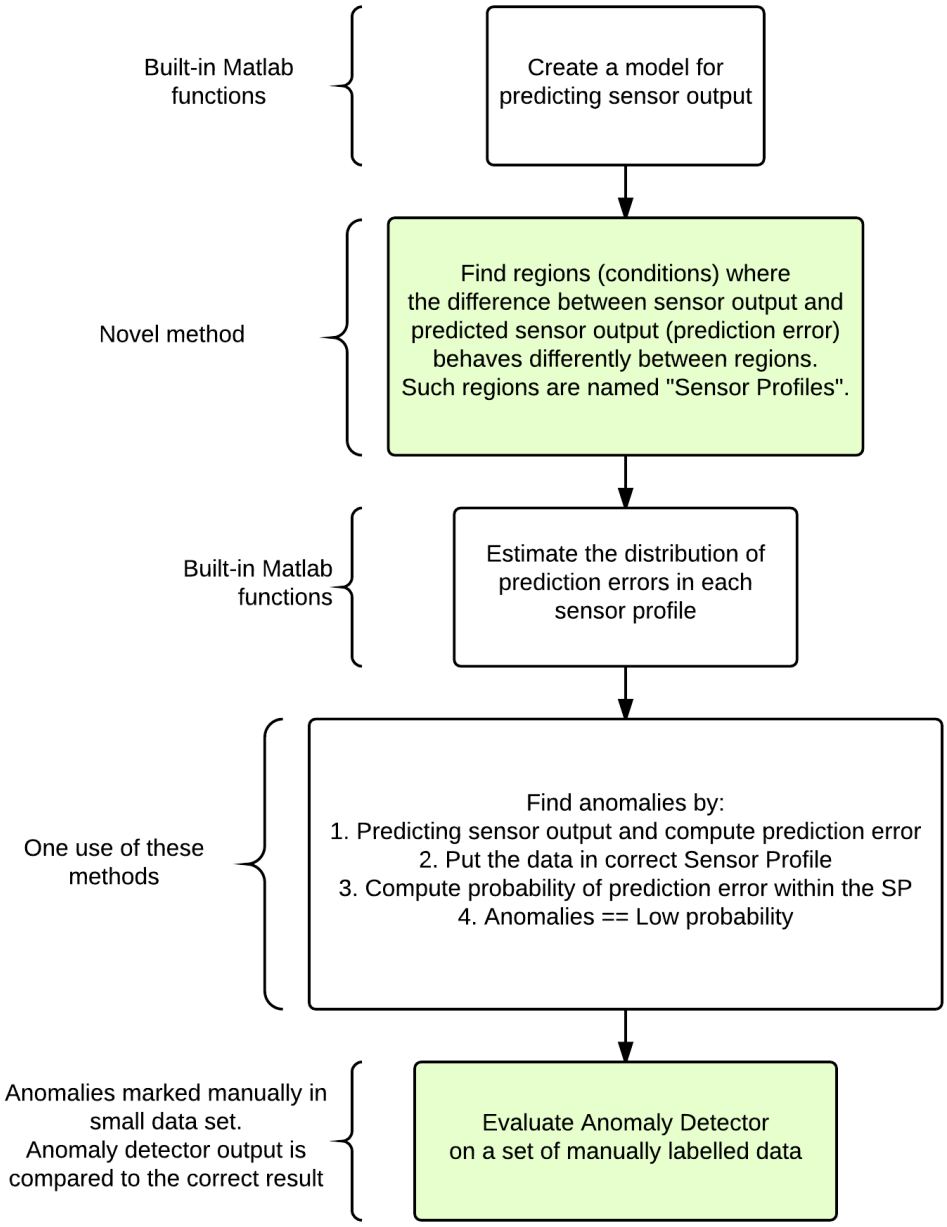


Figure 2.1: *The general method for detecting anomalies. The sensor model is used to predict sensor output. This predicted output is compared to the measured output to compute the prediction error. Sensor profiles are used to find regions where the normal prediction error behaves differently. The probabilities of the prediction errors is then computed within each profile and if the probability is lower than a certain threshold that record corresponds to an anomaly, otherwise it is considered normal.*

Finding Anomalies

The predictive model is used to compute the prediction error. Sensor profiling is used to determine the expected distribution of the prediction error under the specific set of conditions. This is used to compute the probability of the occurrence of the particular prediction error under a certain set of conditions. Anomalies will result in a low probability compared to normal data instances. In this thesis each anomaly detection method will assign an anomaly score to each record, where a larger anomaly score should correspond to a more anomalous record. The anomaly score for this method is directly proportional to the computed probability of the occurrence of the prediction error. If the anomaly score is larger than a certain threshold the corresponding record is automatically classified as an anomaly, otherwise it is classified as a normal record.

2.6 Comparing Anomaly Detection Methods

Comparing and evaluating results from anomaly detection methods can be done in the same way as for any binary classifier. A data set consisting of data points accurately labelled as either normal or anomalous can be compared to the output of the anomaly detection method. The result of this comparison is defined as follows:

True Label	**Anomaly Detection Output**	**Result**
Anomaly	Anomaly	True Positive
Normal	Normal	True Negative
Normal	Anomaly	False Positive
Anomaly	Normal	False Negative

From this the following measures are defined as [20]:

$$\begin{aligned}
\text{True Positive Rate,} \quad & TPR = \frac{\#\text{ True Positive}}{\text{Total \# of anomalies}} \\
\text{True Negative Rate,} \quad & TNR = \frac{\#\text{ True Negative}}{\text{Total \# of normal data points}} \\
\text{False Positive Rate,} \quad & FPR = \frac{\#\text{ False Positive}}{\text{Total \# of normal data points}} = 1 - TNR \\
\text{False Negative Rate,} \quad & FNR = \frac{\#\text{ False Negative}}{\text{Total \# of anomalies}} = 1 - TPR
\end{aligned} \tag{2.1}$$

The anomaly detection methods evaluated in this thesis all assign some form of anomaly score to each data point. If this score is higher than a specific threshold, α, the data point is classified as an anomaly. The *true positive rate* and the *false positive rate* can be computed for a range of thresholds to create a parametric curve that visualises the dependance between TPR and FPR. This curve is called a *receiver operating characteristic* (ROC)-curve and is typically used to visualise the performance of a binary classifier as the threshold varies [20]. In this thesis the anomaly detection methods will be compared to each other and evaluated on a data set with known anomalies using ROC-curves.

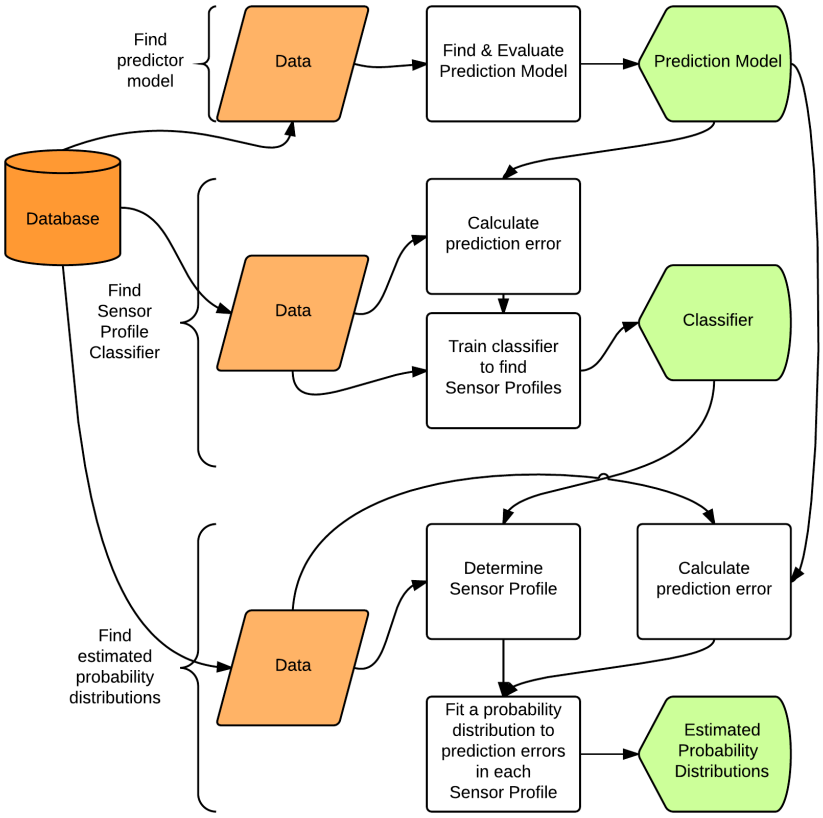


Figure 3.1: *A flow chart of the general method for finding and training the components needed for the suggested anomaly detection technique.*

3 Method

This chapter aims to provide an overview of the anomaly detection method proposed in this thesis. The implementation specific details in thesis are also stated here, such as names of high level native functions and parameter choices. A deeper understanding of the parts that make up the method is found in Chapter 4 and Chapter 5.

Introduction

An anomaly is defined as an unexpected sensor reading, with the assumptions that they are rare and significantly different from normal behaviour. Using this simple definition, anomaly detection can be divided into two parts, first find prediction model that define expected, normal, sensor behaviour and secondly an analysis of the difference between the predicted sensor reading and the actual sensor reading: the prediction error. The second part can be extended further by dividing the data into different sensor profiles. This will allow for natural differences in the prediction error between sensor profiles, which can be expected. A summarised description of the method for finding the components needed for this anomaly detection technique is visualised as a flow chart in Figure 3.1.

3.1 Finding Normal Sensor Behaviour

Finding expected sensor readings, Y_i, can be done in many ways. Assuming $Y_i' = 0$ or $Y_i'' = 0$ are simple approaches, using different types of filtering, e.g. Kalman filters or moving average, can also be effective [8].

But the goal was to find a general method that can be used for any sensor reading, both with or without physical units and relations, and also coming from both advanced "top level"-sensors as well as simple "single measurement"-sensors. Hence regression analysis was used, a general statistical method for finding relationships between variables.

3.1.1 Regression Analysis

A list of variables, $X_1, \ldots, X_{n'}$, that have a high chance of being related to the sensor reading of interest, Y, is used as candidates in the regression analysis. Stepwise regression analysis is then used together with cross-validation on a representative data set to find a regression model for the sensor, $\hat{Y} = f(X_1, \ldots, X_n)$, that only includes the statistically significant predictor variables, $X_1, \ldots, X_n$.

Finding predictor variables

For the signals considered in this thesis the sensor reading, Y_i, is part of a time series. Hence, including the previous reading, Y_{i-1}, the discrete derivative, $Y_{i-1} - Y_{i-2}$, and the discrete second derivative, $Y_{i-1} - 2Y_{i-2} + Y_{i-3}$, in the set of predictor candidates is a given choice. If the sensor reading is related to a physical measure, finding other sensor readings that is expected to be related to Y, can be done as well. However, finding the most suitable sets of candidate prediction variables, $X_1, \ldots, X_{n'}$ is by no means easy and will depend on not only the particular sensor reading, Y, but also the configuration of both the particular sensor and the other available sensors [21]. With good understanding and insights in the particular sensor this can hopefully be done in a few tries by a group of experts. Finding suitable predictor variables will be a manual, sensor-dependent step that can not be fully automated when using regression analysis as a predictor model.

Removing outliers from training set

When the candidate predictor variables are decided a representative data set is constructed. As the goal of the predictor is to accurately predict normal sensor reading, the data set should only include such readings. As the data is not labelled the training set will initially include anomalies that should be removed [19]. This is done by creating a regression model using stepwise regression on the entire set and removing points that has a Cook's distance larger than $D_{max} = \frac{4}{n}$, where n is the number of data points in the data set. See Section 4.1 for more details. The regression model was created and Cook's distance calculated using the built-in function `stepwiselm()` in MATLAB.

Using stepwise regression with cross-validation

The trimmed data set is now divided into two parts, the training - and the validation set. Two thirds of the points, selected at random, is in the training set and the remaining one third is in the validation set. The regression model trained using the training set is initially empty and terms are added one by one using AIC (described in Section 4.3), until there are no more statistically significant terms to add. At each step the adjusted R^2-value for the new regression model is calculated for both the training and the validation sets. The regression formula is taken from the regression model that corresponds to the highest adjusted R^2-value for the validation set. This is done to avoid over fitting to the training set and find the regression model that most accurately predicts new sensor readings [22].

Finding the final regression model

A final step is performed to further eliminate the influence of possible remaining anomalies in the data set. That is to perform robust regression on the entire trimmed data set using the regression formula found using cross-validation. Robust regression is robust with regards to outliers in the data set and will result in the regression model that most accurately predict the normal sensor behaviour [23]. Examining the diagnostics of the regression analysis will reveal, statistically, how well the model fits the data and how much of the variance explained by the model. These diagnostic measures should be used to evaluate the choice of candidate predictor variables and can be used directly to determine whether the regression model should be used as the predictor or not [19].

3.1.2 Other predictors

To evaluate the performance of the regression model with respect to anomaly detection, a subset of predictive models from Section 2.3 were also implemented. The models based on assumptions, i.e the naïve predictor, the constant velocity - and constant acceleration model where all implemented along with using the moving average as predictor. The Kalman filter, Dynamic Bayesian Networks and Artificial Neural Network based predictors are other possible predictors that where not implemented and compared to the other predictors used in this project.

3.2 Analysing the Prediction Error

Even for normal records the sensor reading will probably differ from the predicted sensor reading, resulting in a prediction error, $\epsilon_i = Y_i - \hat{Y}_i$, where ϵ_i will have a certain distribution, $\hat{P}$. $\hat{P}$ can be stationary but it can also be related to a number of measurable features, $\xi_1, \ldots, \xi_m$. This can also be true for the predictive model, but in this thesis a single predictive model is used. A relationship between some set of features and $\hat{P}$ is probable for many sensor readings as sensors are usually optimised for certain working conditions, and will therefore exhibit different behaviour in different conditions.

Finding the relation between $\xi_1, \ldots, \xi_m$ and ϵ is no simple task and can be approached in different ways. Regression analysis could be used, but describing the relationship might not be possible using smooth functions. Especially as ϵ may also depend on unmeasured factors that may have some relation to the features $\xi_1, \ldots, \xi_m$ as well, this can result in very complex relationships between $\xi_1, \ldots, \xi_m$ and ϵ. Another approach would be to try to divide ϵ into a few groups with similar appearance and try to map these groups to different regions in ξ-space, so called sensor profiles. As we want a method that is as general as possible the latter approach was used, as it has higher potential to be effective on a larger number of sensor readings. The problem now lies in determining the method for clustering ϵ, finding good similarity measures, finding the most suitable number of groups, finding a method for mapping areas in ξ-space to the different groups and finally estimating the probability distribution of ϵ in each group. One could also create and use different regression models for each group, which might be easier if the original sensor signal, Y, is used to create the sensor profiles instead of the prediction error, ϵ.

To conclude, there are many method and parameter choices available, and when the best set of choices depends on the sensor, among other things, scientifically determining the optimal choices for any general sensor is far beyond the scope of this thesis. Finding a good candidate that works well for at least some set of sensors is considered a great result.

3.2.1 Finding Sensor Profiles

Sensor Profiles are used to determine the normal behaviour of the sensor under different conditions. The overall goal of the method is to divide the feature space (ξ-space) into regions, sensor profiles, where the distributions of the prediction error between sensor profiles are as different as possible. Each sensor profile should, at the same time, contain as many points as possible. It can be seen as a method for clustering distributions of a stochastic variable using dependent variables. As normal clustering or statistical techniques were proven to be unable to find these regions a novel method was used to perform the sensor profiling. The procedure is described in detail in Chapter 5 and the result of the method is a decision tree that is used to classify data points into the according sensor profile.

Estimating probability distribution of ϵ

The method used in this thesis for estimating the error distribution, $\hat{P}(\epsilon)$, is kernel density estimation, with the Gaussian kernel function. This method can be used on any data set, coming from either a specific group found by clustering and classification or a complete data set. The resulting probability density function (PDF) is parameter free and fully data-driven, however when created the bandwidth has to be set. The bandwidth used when creating the PDFs for the investigated sensor readings was decided using the estimated resolution of the particular sensor. For instance, a thermometer that measures the outside temperature may have the estimated resolution, and therefore the bandwidth, equal to 0.1°C. The chosen bandwidth should result in a PDF that captures the behaviour without being over-fitted to the particular data set. The method used for this was the built-in MATLAB-function `fitdist(...,'pdf','Kernel','BandWidth',h)`.

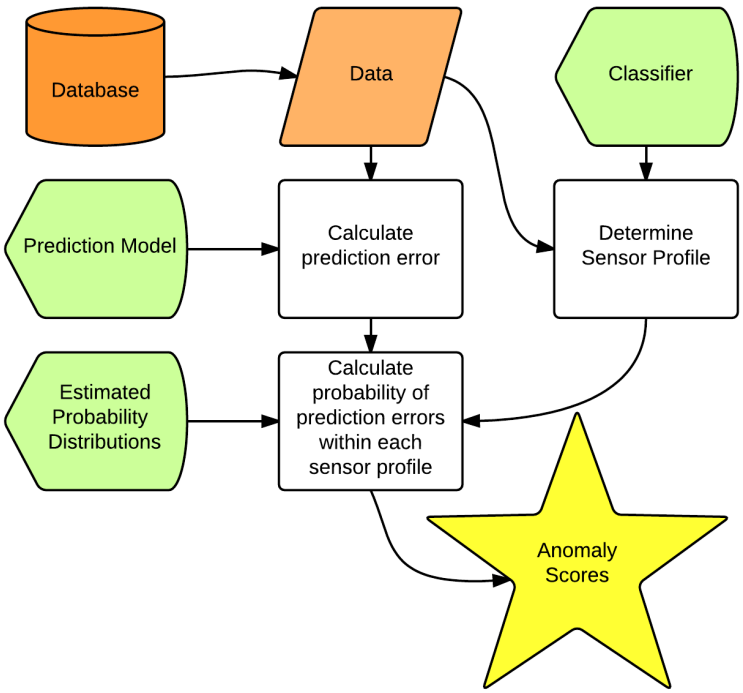


Figure 3.2: *A flow chart showing how the components are used to calculate the anomaly score using sensor profiles.*

3.3 Finding Anomalies

Both the prediction error, ϵ, and the estimated probability of the prediction error, $\hat{P}(\epsilon)$, either with or without first determining sensor profiles, can be used as anomaly scores, a_i. Computing the anomaly scores using a prediction model, sensor profile classifier and estimated probability distributions discussed above is visualised in Figure 3.2. Classifying data points as either anomalous or normal is done by comparing an anomaly score to a decision threshold, α. If we define anomaly scores so that a higher value corresponds to a larger probability of being an anomaly, the class of data point i is anomalous if $a_i > \alpha$ or normal if $a_i \leq \alpha$. The optimal decision threshold, α, depends on the specific sensor, which anomalies you are interested in and how many false positive values that can be accepted.

As we expect a large prediction error, ϵ, when anomalies occur, $|\epsilon|$ can be used directly as the anomaly measure. We can also make use of the sensor profiles, as $\hat{P}(\epsilon)$ is the estimated probability of the prediction error ϵ, $1 - \hat{P}(\epsilon)$, can be used as the anomaly measure.

3.4 Comparing Methods

A labelled evaluation set is needed to compare different combination of methods, parameter choices and anomaly measures and to find the optimal value of α. This is created manually using a simple GUI, where intervals are marked as either anomalous or normal. If the anomaly detection method marks at least one point in one of these areas as anomalous the entire area is flagged as anomalous. Otherwise it is flagged as normal. The result from the anomaly detector is compared against the labels from the GUI in order to compute the true positive rate and the true negative rate.

A ROC-curve is created by varying the anomaly threshold, α, between $-\infty$ and ∞. By manually comparing the appearance of the ROC-curves, created using the different anomaly detection methods, the most suitable detection method can be determined [20].

4 Regression Analysis

This chapter supplies the reader with a basic understanding of regression analysis and typical statistical diagnostics used to evaluate the resulting predictive model. Regression analysis is a powerful statistical tool with many applications, but very limited understanding of regression analysis is needed to understand this thesis. To the interested reader, who may want to gain a deeper understanding, it is recommended to look at one of the many books written within this subject, for example [7].

Introduction

In statistics regression analysis is used for modeling the relationship among variables. Regression analysis includes many techniques that are commonly used for prediction of time series with or without anomaly detection as the end goal [7]. Linear regression models are used to establish the relation between a response variable, Y and a number of predictor variables, $X_1, ..., X_n$. The case when $n = 1$ is called simple linear regression, opposed to multiple linear regression when $n > 1$. Both are different from multivariate linear regression, where Y consists of several variables.

4.1 Multiple Linear Regression

Using several predictor variables to predict a single variable is called multiple linear regression analysis. The goal is to find a function that is used to predict the response variable, Y, using the predictor variables, $X_1, ..., X_n$. A general example is given below:

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 \theta_1(X_1) + \beta_3 \theta_2(X_1, X_2) + ... + \beta_m \theta_p(X_n), \tag{4.1}$$

where $\beta_i, i = 1, \ldots, m$ are constants found through the regression analysis and θ_i could be any function of one or more variables. The function found through regression analysis is not necessarily linear in the predictor variables, but linear in the coefficients β_i. Regression analysis found in statistical tools, packages and specialised softwares, do not only calculate the values of β_i that minimises the distance between Y and $\hat{Y}$ but also calculates diagnostic measures that can be used for determining the significance of each prediction term and for the entire model. This gives the user the possibility to determine the regression model that is most suited for predicting future values and also makes it possible to easiliy compare models using statistically proven methods. Often these tools also let you perform stepwise regression which is a method where terms are added to or removed from the model based on their significance and some pre-set criteria. This gives the user the possibility to find a statistically significant regression model more or less automatically [7].

4.2 Regression Diagnostics

Finding and understanding measures to gauge the significance and the fit of the regression model is essential both for finding the most suitable model and when evaluating the final result. The measures and tests used in this thesis are listed and briefly explained below.

Coefficient of Determination

The coefficient of determination, R^2, is a measure that indicates how well a statistical model fits the data [17]. This is one of the measures used for analysing and comparing regression models and is defined as the ratio of the explained variance in the data, which is defined as follows. With n data instances, y_i, $i \in [1, n]$, mean value as in Equation (4.2) and modeled data, $\hat{y}_i$, we denote the total sum of squares as in Equation (4.3), the explained sum of squares as in Equation (4.4) and the residual sum of squares as in Equation (4.5). The definition of the coefficient of determination is seen in equation (4.6).

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{4.2}$$

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{4.3}$$

$$SS_{reg} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \tag{4.4}$$

$$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.5}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{4.6}$$

One issue that occurs when looking at R^2 is that it will always increase with an increase in the number of predictor variables as long as their coefficients are non-zero. Thus using R^2 as the only diagnostic measure when preforming regression analysis may result in an overly complicated model [19].

Adjusted R^2

To avoid an overly complicated model the adjusted R^2 can be used instead. The adjusted R^2 penalises the number of predictor variables, p, and will only increase when an added variable improves the model more than would be expected by chance [17]. It is defined by

$$\begin{aligned} R^2_{adj} &= 1 - \frac{SS_{res}/df_e}{SS_{tot}/df_t} \\ df_e &= n - p - 1 \\ df_t &= n - 1 \end{aligned} \tag{4.7}$$

where n is the sample size.
The adjusted R^2 is useful when deciding what predictor variables to include in the model.

F-statistic

To assess the statistical significance of the regression model an F-test is usually performed. This compares the model $\hat{Y} = f(X_1, .., X_n)$ to the constant model $\hat{Y} = c$. The F-statistic is defined as:

$$F = \frac{SS_{res}/(p-1)}{SS_{reg}/(n-p)} \tag{4.8}$$

The probability of this value is found for the F-distribution, $f(F; d_1, d_2)$, with parameters $d_1 = p - 1$ and $d_2 = n - p$ and compared to the selected significance level (usually 0.05) in order to determine the significance of the regression model [17].

T-statistic

To determine which predictor variables to include in the regression model, the t-statistic is used to test the significance of a single coefficient, β_i. If the estimated value of β_i is b_i, which has a standard error of $SE(b_i)$, the t-statistic is

$$t = \frac{b_i}{SE(b_i)} \tag{4.9}$$

The null hypothesis ($\beta_i = 0$) is rejected at significance level, a, if $P(t) < a$ [17]. When performing stepwise regression a coefficient typically has to pass the t-test at significance level $a = 0.05$ in order to be included in the model and an already included coefficient has to fail the t-test at significance level $a = 0.10$ in order to be excluded [19].

Leverage

Leverage is a measure used to identify observations that are outliers with regards to the predictor variables. These points may affect the regression model more than closely clustered points. In general, the farther the point is from the center of the predictor variables, the more leverage it has. The leverage score of a data point i is defined as:

$$h_{ii} = \frac{d\hat{Y}_i}{dY_i} \tag{4.10}$$

Points with high leverage indicate that the region of predictor variables they are in is sparse and more observations with similar predictor variable values can help to create a more suitable regression model [17].

Cook's Distance

Cook's distance can be used in the same way as leverage but also measures the effect of deleting the observation. The effect that deleting an observation has on the regression model may come from a high leverage score and/or large residual, $r_i = Y_i - \hat{Y}_i$. The Cook's distance is defined as:

$$D_i = \frac{r_i^2}{p \cdot MSE}\left(\frac{h_{ii}}{(1-h_{ii})^2}\right), \tag{4.11}$$

where p is the number of coefficients included in the model, MSE is the mean squared error of the regression model [17]. Deleting highly influential observations that has Cook's distance above a certain threshold, $D_i > D_{max}$, is a way to stablilise the resulting regression model. Selecting the threshold, D_{max}, can be done in several ways: simply $D_{max} = 1$, $D_{max} = \frac{4}{n}$, where n is the number of observations, or $D_{max} = 3 \times$the mean value of D_i [19]. In this thesis $D_{max} = \frac{4}{n}$ is used as it resulted in more accurate regression models for several sensor signals.

Other measures and tests

Other measures and tests may be of equal or greater importance when regression analysis is used outside the application of anomaly detection in time series. For example the Durbin-Watson test is used to find auto-correlation within the residuals, which, if present, can result in an under-estimated variance. This can ultimately in some cases render the F-statistic useless. However in auto-regressive models, where one predicts Y_i using Y_{i-1}, one is expected to have auto-correlation in the residuals and its existence does not affect the predictive power of the model itself [19].

4.3 Akaike Information Criterion

The Akaike information criterion (AIC) is a measure of the relative quality of any statistical model on a data set [24]. As AIC can be used to estimate the relative quality of each model it can be used for model selection [25]. AIC is based on information theory and defined as:

$$\text{AIC} = 2k - 2\log(L), \tag{4.12}$$

where k is the number of estimated parameters in the model and L is the maximised value of the likelihood function for the model. The likelihood function is a function of the parameters in a statistical model. The likelihood of a set of parameters, θ, given outcomes, x, is the probability of outcomes x given the parameters θ:

$$L(\theta|x) = P(x|\theta). \tag{4.13}$$

The maximum of L is found by finding the statistically most likely parameter values, θ^* [17]. As AIC is a general measure used for comparing statistical models it can be used for finding the most suitable regression model through stepwise regression, but its use is not limited to regression analysis. AIC can also be used, for example, for finding the best matching Gaussian mixture model or finding the best parameters for exponential smoothing.

Summary

The summarised workflow and uses of the different statistics is visualised in Figure 4.1.

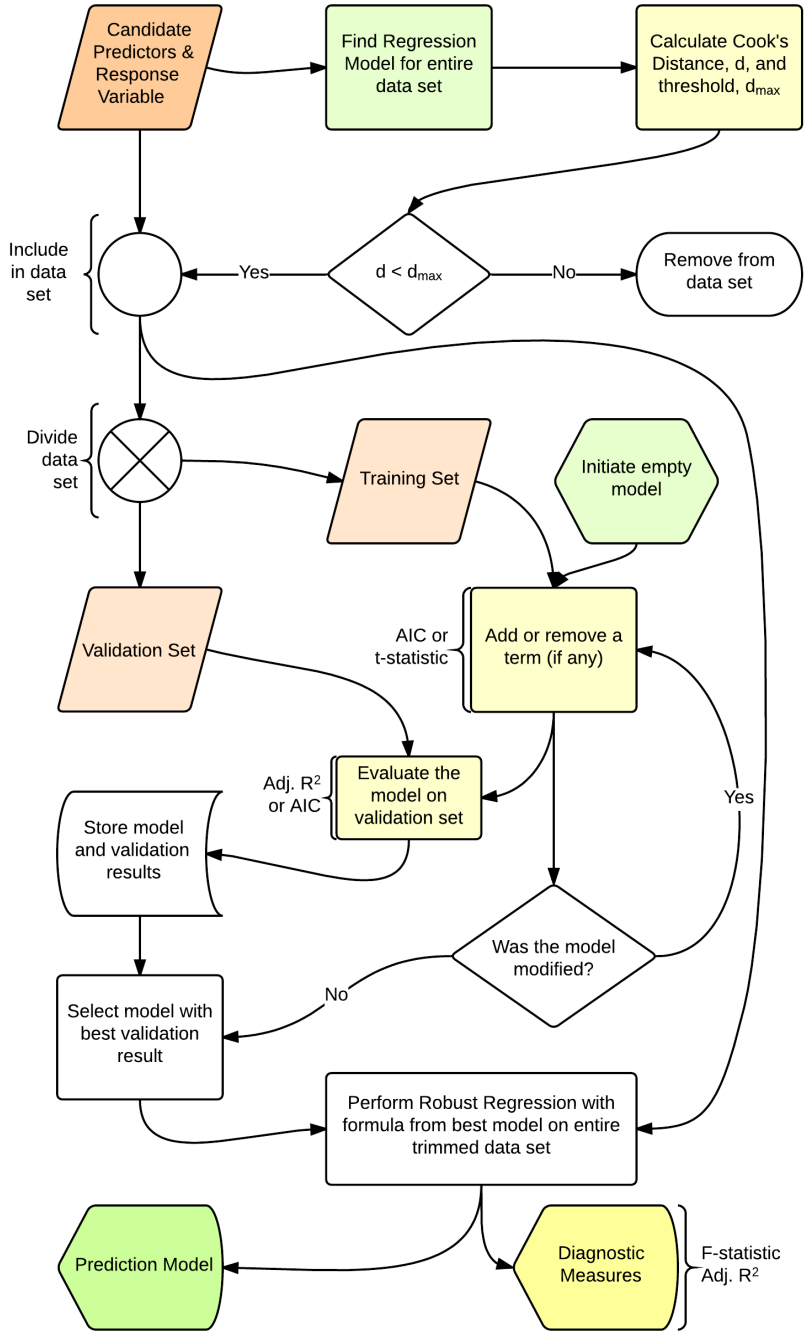


Figure 4.1: *A flow chart of the general method for finding the most suitable prediction model through robust regression. The diagnostic measures is used to evaluate the final model. The optimal formula of the model is found through stepwise regression and cross-validation on a data set without outliers. The outliers are found through Cook's distance and are removed from the original data set.*

5 Sensor Profiling

In this chapter sensor profiles and sensor profiling will be introduced and the motives behind it, together with the suggested methods for finding the profiles, will be discussed. As this is a novel method introduced in this thesis, the chapter includes detailed descriptions of the algorithms used. Understanding the main results and the overall method for detecting anomalies does not require a deeper understanding of sensor profiles or the methods used to find them. Knowledge of the motives behind sensor profiling and understanding the resulting classifier should suffice.

5.1 Motives behind Sensor Profiles

Intuitively, a sensor may behave differently under different circumstances. For example, consider a simple thermometer. Depending on the intended use it may be optimised for certain temperature ranges. A thermometer intended to be used for measuring the outdoor temperature may show a very small error when the temperature is between e.g. $-15°\text{C}$ and $+30°\text{C}$ and fairly constant over time. When the temperature is outside this range or changing very rapidly the thermometer may exhibit a larger error. If there is large difference between the errors in the different regions it can probably be found using regression analysis. However, small differences or changes in the appearance of the distribution of the measuring errors may not be so simply found. These changes may however be of interest when assessing the sensor or when finding anomalies.

The thermometer example will now be used to show how sensor profiles can be used to characterise the performance of a sensor. There are two features, the measured temperature, T, and measuring error, ϵ. The temperature is used as an input feature and the measuring error is used as the response feature. Having two features makes it possible to visualise the data in a 2D scatter plot. In real thermometer data the error might be correlated to other features as well, for example, the average wind speed, rainfall, temperature change rate, air pressure, humidity or incoming sunlight. A scatter plot of 100 000 records together with histograms of the features are shown in Figure 5.1. The temperature was drawn from a rough estimate of the distribution of temperatures in Gothenburg. The measuring error is drawn from one of two distributions depending on whether the temperature is within the range, $R = [-15°\text{C}, +30°\text{C}]$, or not. The two different distributions are visualised in Figure 5.2. Notice that about 99% of the temperatures drawn from the distribution of temperatures fall within the range, R, and the errors in those records will come from the narrow distribution.

The two distributions of the measuring error both have a mean value of zero, which should be the case for any good sensor, but their shapes are quite different. This, together with the distribution of temperatures and the large number of data points, makes it difficult to see a direct relation between temperature and error in the scatter plot in Figure 5.1. It also makes it difficult to extract any information using common statistical or machine learning tools, e.g. regression analysis or clustering. In this case these methods will not help us understanding the sensor and some other method must be used to determine the normal behaviour. What is suggested in this thesis is that a set of input features (temperature in this example) is used to divide the points into regions where the distribution of the response feature (measuring error) are as different as possible. In this thesis the different regions found are called *Sensor Profiles*. Finding such regions can hopefully be used to capture the normal behaviour of the response feature and will help analysts to gain insights about the data. The main challenge now lies in comparing and quantifying different divisions that results in different sensor profiles. This can then be used to find a *sensor profiler* which will actually divide the data into sensor profiles.

5.2 Comparing sets of Sensor Profiles

The main goal of sensor profiling is to find regions where the distributions of the response feature are as different as possible. The distributions in each region is estimated using a histogram and selecting the appropriate range and bin size is the first step.

Evaluating the difference, or distance, between two histograms is a field in itself [26]. There are many ways to compute the distance between two histograms. In this thesis Earth Mover's Distance (EMD) or Chi-squared Distance (χ^2) is used. χ^2 uses the difference in bin counts at each bin to compute the distance (bin-to-bin comparison). EMD uses the distance between the bins as well as the difference in bin counts when computing the distance (cross-bin comparison) [27]. Both distance measures requires normalised histograms that both cover the same region, D, and has the same bin locations.

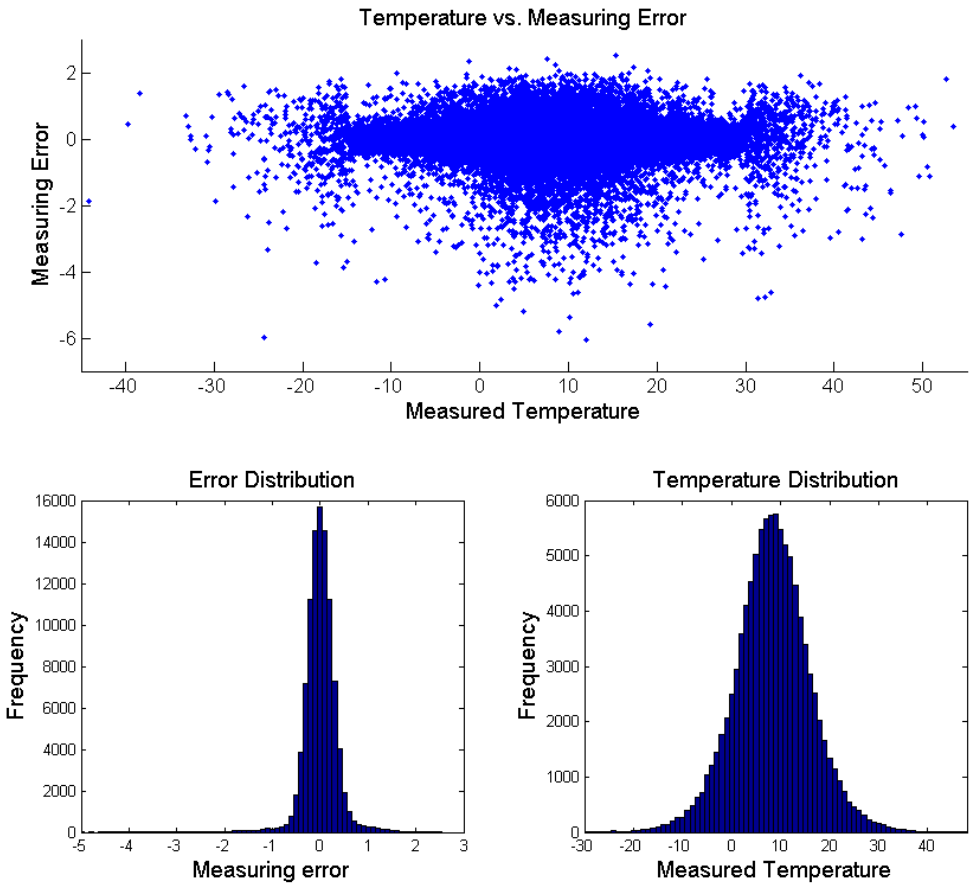


Figure 5.1: *In the top the measured temperature and measuring error from 100 000 simulated data instances are shown along with the distribution of the measuring error and the distribution of the temperature.*

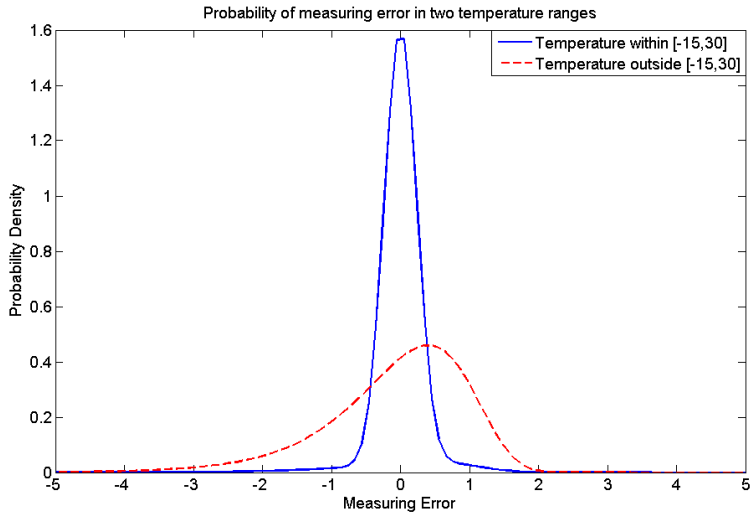


Figure 5.2: *The two different distributions the measuring errors are taken from. As 99% of the points have a temperature between $-15°C$ and $+30°C$ the total distribution of measuring errors is very similar to the solid-lined blue distribution, even though the appearance of the two distributions are quite different.*

Computing the EMD between two one-dimensional histograms, i and j, with equidistant bins is a simple special case and equal to:

$$EMD(p^i, p^j) = \sum_{k=1}^{n} \left| \sum_{l=1}^{k} p_l^i - \sum_{l=1}^{k} p_l^j \right|,$$

where p is a vector containing the normalised frequency for each bin over region D and has n elements. In general, two histograms can be seen as two ways of piling up probability over region D. If so, the EMD is the minimum cost of turning one pile into the other. Cost, in this case, is the amount of probability multiplied with the distance it is moved. This makes the EMD symmetric and the distance between two equal histograms is zero. In any case, the EMD will depend on the choice of the region D and the number of elements in p^i. I.e. the EMD between two sets of data heavily depends on the bin locations and the total number of bins.

EMD is a cross-bin comparison since distance between bins is taken into account. Chi-squared distance is a bin-to-bin comparison that sums up the relative difference between the estimated densities in each bin. It is defined as:

$$\chi^2(p^i, p^j) = \sum_k \begin{cases} \frac{(p_k^i - p_k^j)^2}{2(p_k^i + p^j k)} & \text{if } p_k^i > 0 \text{ or } p_k^j > 0 \\ 0 & \text{else,} \end{cases}$$

where k is the index in p that contains the estimated probability for bin k.

Only using a distance measure (EMD or χ^2) to evaluate a set of sensor profiles will give rise to unwanted results as a histogram containing few records may have a large distance to another histogram even if all the records comes from the same distribution. This occurs as the estimated distribution is inaccurate when based on few data instances. What we really want is to find ways to divide the data so that as many records as possible are divided into as different probability densities as possible. With this in mind, the profiling score, S, of a certain set of sensor profiles is defined as

$$S = \sum_i \sum_j w_i w_j d_{i,j} \quad , \tag{5.1}$$

where w_i is the fraction of records that are in sensor profile i, with an estimated probability of p^i and $d_{i,j}$ is the distance (EMD or χ^2 in this thesis) between p^i and p^j. The estimated probability is the normalised histogram in this case but can be estimated in other ways. The choice of distance measures has not been evaluated in this thesis and it is quite possible that another measure is more suitable for evaluating sensor profiles. The profiling score, S, can be computed for any set of sensor profiles containing any number of different sensor profiles. Using this score makes it possible to compare different sets of sensor profiles to each other which is crucial when trying to find the best way to divide the records.

One issue that arises when comparing sets of sensor profiles using the profiling score in Equation (5.1) is that a higher number of different sensor profiles always allows for higher profiling score. This originates from the fact that the estimated probabilities becomes slightly different leading to an increased distance between them. Consider a large set of points drawn from the same distribution. Using this as a single sensor profile gives us a profiling score $S = 0$, as $d_{i,i} = 0$. In this set there exists a small subset that, if put into a separate sensor profile, will increase the profiling score. This is true as long as not all the original points are placed in the same bin when estimating the probability distribution. It may only be a single record in the second sensor profile and still the profiling score will increase. This issue can be addressed in several ways, for example setting a lower limit on w_i or introduce a penalty for the number of different sensor profiles. In this thesis it is solved by finding different sets of sensor profiles using either EMD or χ^2 and an upper limit on the number of sensor profiles allowed. These different sets are then compared against each other manually to find the most suitable set. The best choice of the number of sensor profiles may depend on the distance measure used.

5.3 Finding the most suitable Sensor Profiler

Now that sets of sensor profiles can be compared against each other the final step is to find the optimal divider i.e. sensor profiler. Two different algorithms are implemented that both aim to produce a sensor profiler. The resulting sensor profiler shall optimise the profiling score for the resulting sensor profiles for a given dataset while fulfilling the constraint on the number of profiles.

The sensor profiler can be any type of classifier and is used to determine the sensor profile of each data instance. As one of the goals is to gain insights in the normal sensor behaviour a classifier that can be understood easily by a human is preferred over a "black-box"-classifier.

A decision tree was chosen as a classifier as it is a fast and simple classifier that can be analysed by a human. Each tree is also easily expanded by adding additional branches, which means that a straight-forward greedy algorithm can be used to find a solution.

Finding the best possible decision tree is however probably impossible for real data, especially if it includes a large number of features. This as the decision tree can be of any size and include any combinations of features and values making it possible to put almost any point in any group in an infinite amount of different ways. A stochastic algorithm may be more suited for finding a decision tree in this case as the mapping from decision trees to S is rather complex. The complex mapping makes it impossible to determine gradients and even if it were possible S will consist of a large amount of local optima.

Heuristic algorithms may fail to reach the global optimum, but may result in a locally optimal solution that can be considered satisfactory. Stochastic algorithms can find better solutions than other algorithms and even has the potential of finding the global optimum. The main drawback is that stochastic algorithms usually require more computational time than heuristic algorithms to find better solutions [28].

Two different algorithms are developed to find the sensor profiler, one heuristic (greedy) algorithm and one stochastic (genetic) algorithm. These will be described and evaluated as follows.

5.3.1 Finding a Decision Tree through a Greedy Algorithm

A greedy algorithm finds a solution to a problem by iteratively making locally optimal decisions [29]. A greedy algorithm may not find the global optimum but it is a fast algorithm that will, for many problems at least, find a locally optimal solution. To demonstrate a greedy algorithm, consider the change-making problem: *how can a given amount of money be made with the least number of coins of given denominations?*

A greedy algorithm will pick the largest denomination that is not greater than the remaining amount and withdraw it from the amount until the remaining amount is equal to zero. For certain coin systems, like the one used in Sweden, this will produce the optimal result. For example, the solution for 17 SEK = 10 + 5 + 1 + 1 will be found by this greedy algorithm. For other coin systems, for example if the coin denominations were 1, 3 and 4 and the amount of money was 6, the greedy algorithm would produce 6 = 4 + 1 + 1, whilst the optimal solution is 6 = 3 + 3.

Greedy algorithms are, however, commonly used for training decision trees where one has labelled data [30]. The common approach used when training decision trees on labelled data is to optimise the splitting criterion at each node and use some form of pruning criterion to avoid creating a too large and complex tree. This inspired the algorithm used in this thesis, but as we have unlabelled data and the goal is to maximise the profiling score, S, some modifications are made.

The greedy algorithm created for finding a decision tree that maximises S consists of two main steps. The first step is to find the locally optimal division of a group of records. This division turns one group into two sub-groups. The second step is to combine sub-groups, this is done either to maximise S or to ensure that the upper limit of the number of groups is not violated. The first step is the greedy step, where S is computed for a number of splitting criteria and the best one is selected.

Let us return to the example with the thermometer, visualised in Figure 5.1. If we run the greedy algorithm for two iterations, with a maximum number of groups equal to two and χ^2 as the distance measure, the steps can be exemplified and the intermediate results can be evaluated. The results at each step is visualised in Figure 5.3 and the procedure is described as follows:

1. The first step is to divide the records into two groups. One group will have $T \leq \kappa$ and the other will have $T > \kappa$. The locally optimal choice of κ is found by computing S for several values of κ and choose the threshold that maximises S. If there are more than one input feature they are all evaluated with different thresholds in order to find the splitting criterion that maximises S.

2. When a splitting criterion is found there are two subgroups, combining these will not lead to any improvement of S.

3. This step will divide each sub-group into two sub-groups. This is done in the same greedy way as described in step 1. This will result in four sub-groups. Numbering the groups is done as follows: sub-group 1 is divided into sub-groups 1 & 2 and sub-group 2 is divided into sub-groups 3 & 4.

4. As the maximum number of groups is two, these four sub-groups must be combined into two subgroups (1 & 2). This is done exhaustively i.e. all the viable combinations are tested and the combination that yields the highest value of S is chosen. The combinations and resulting value of S are:

Group 1	**Group 2**	S
1	2, 3, 4	0.0008
1, 2	3, 4	0.0033
1, 3	2, 4	0.0058
1, 4	2, 3	0.0040
1, 2, 3	4	0.0033
1, 2, 4	3	0.0066
1, 3, 4	2	0.0026

The final decision tree and the resulting histograms of the groups are shown in Figure 5.4. This tree divides the records so that the records with $T \in [-14.9°\text{C}, +30.4°\text{C}]$ is in one group and the other group consists of the records with T outside this range. The profiling score for the decision tree found using the greedy algorithm is $S = 0.0066$. The two sensor profiles are very similar to the original two distinct groups. In Figure 5.4 the two distributions used to create the data are shown together with the histograms of the two sensor profiles. This sensor profiler will classify 0.057% of new records into the incorrect sensor profile.

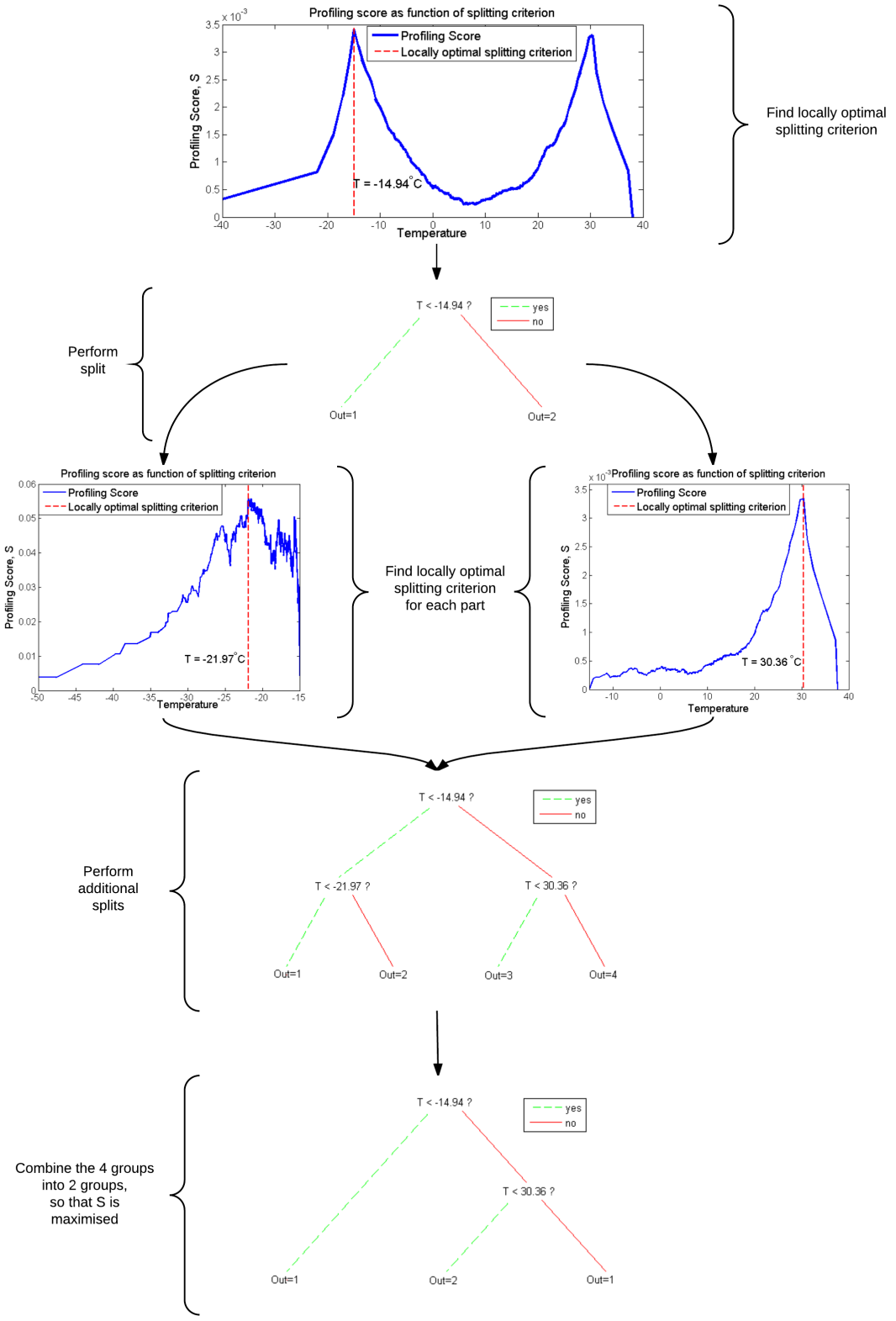
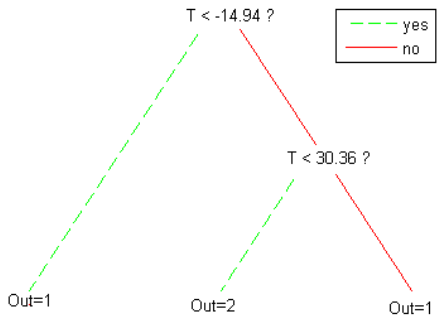
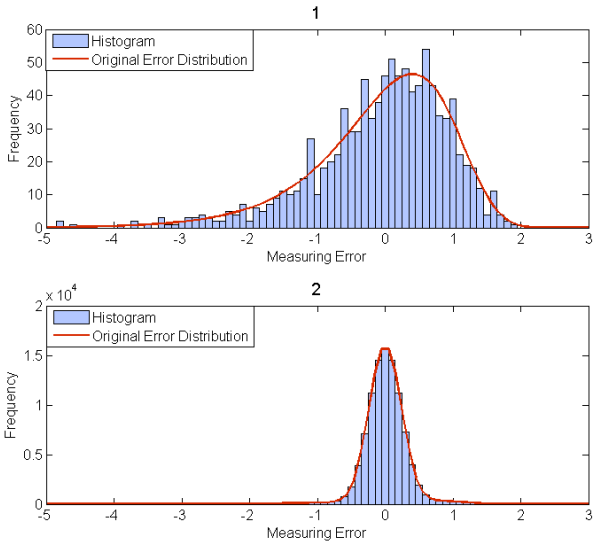


Figure 5.3: *The locally optimal splitting decisions in each iteration of the greedy algorithm together with the resulting decision trees. Note that the final step is an exhaustive search, that will result in the optimal combination given a constraint on the number of sensor profiles.*



(a) *The sensor profiler found using the greedy algorithm.*



(b) *The resulting histograms for each sensor profile.*

Figure 5.4: *The chosen sensor profilers and the resulting histograms. The shape of the original distributions match the histograms rather accurately. Using the known distribution of the temperature (figure 5.2) 0.057% of data points will be classified into the incorrect sensor profile using this sensor profiler.*

5.3.2 Evolving a Decision Tree

Another algorithm that can be used to find a decision tree is based on genetic programming. Genetic programming, where the desired outcome is a program, in our case a decision tree, is a special case of genetic algorithms. A genetic algorithm is an evolution-inspired optimisation technique that includes the driving forces behind evolution. Those driving forces of evolution are:

- Selection: The individuals with the highest fitness are more likely to reproduce. In this case fitness is defined to be the profiling score, S.
- Crossover: Two selected individuals are mixed in some way to produce two new individuals. The common way of performing crossover in tree structures is to switch one branch in the first individual with a branch from the second individual. This usually makes the crossover operator effective and results in individuals that are different from their "parents".
- Mutation: This adds variety to the population by changing information in some nodes in some individuals. In this case mutation will change either the value or the variable of the splitting criterion or the assigned group of a node.

The work flow of a general genetic algorithm is shown in Figure 5.5. This visualises the main parts of the algorithm used for finding a decision tree. Computing the fitness for a decision tree is done by dividing the data using the tree and computing the profiling score, S. Computing S for a set of decision trees can be computationally heavy for a large set of records and large decision trees. There are several different ways of performing the selection based on the computed fitness [28]. In this thesis *tournament selection* was used. Tournament selection is performed by repeatedly randomly selecting a group of individuals for a tournament from which a single individual is selected. The tournament size, s_T, determines how many individuals are selected for each tournament. These individuals are ordered by their fitness value and the participant with the highest fitness is selected to be the winner with a probability, p_T. If the participant with the highest fitness does not win the tournament the participant with the second highest fitness is declared the winner with probability p_T. This continues until a winner is declared or the individual with the lowest fitness is the only candidate left, in which case, that individual is declared the winner. Let us order participants in one tournament so that participant 1 has the highest fitness and participant s_t has the lowest. Now the probability that participant i

will be selected, p_i, is given as:

$$p_i = \begin{cases} (1-p_T)^{i-1} p_T & \text{if } i < s_T \\ (1-p_T)^{s_T-1} & \text{if } i \equiv s_T. \end{cases} \tag{5.2}$$

Tournament selection does not consider the actual difference in fitness between individuals, only the order. Using a small value for p_T and / or s_T will result in more random selection, which generally leads to wider and more stochastic search. In any case, one should always ensure that $p_T \geq 0.5$ so that individuals with higher fitness never has a lower probability of being selected than individuals with lower fitness.

Crossover combines two individuals by switching a branch in the first individual with a branch in the second individual. The branches are chosen randomly and can be of different sizes. This results in two new individuals that can be very different from the original individuals.

There are three different types of mutation implemented in this algorithm, standard mutation, creep mutation and prune mutation. Standard mutation changes either the variable, the value or the assigned group to a random value within the valid range. Creep mutation can only affect the value of the splitting criterion and will only perform a slight change in either direction. The size of the creep mutation depends on the density of points in the region. A high density will result in a smaller change than a low density, which will allow for a larger change. This is implemented by modifying the fraction of points fulfilling the criterion and not the criterion itself. This also makes it possible to use the same creep mutation size for several different variables. Prune mutation will cut branches. Performing prune mutation will decrease the size of the decision trees. Mutation ,creep mutation and prune mutation occurs with certain probabilities or rates. Selecting these rates is of course crucial. Unfortunately the theory behind genetic algorithms is limited and there exists only rules of thumb to select these rates. Using a varying mutation rate, for example, may will lead to more diversity in the population which may yield better results. A varying mutation rate is only used for the standard mutation, while creep and prune mutation rates are kept constant. The algorithm used for finding the decision trees uses a periodic mutation rate with three parameters: the minimum mutation rate, p^-_{mut}, the maximum mutation rate, p^+_{mut}, and a period, P, measured in generations.

One challenge when using genetic algorithms is that it usually requires assigning a large set of parameter values. This is true in our case as well, the name of the parameters, a short description and the value used is given as follows:

Name	**Description**	**Approximate value**
Min. Mutation Rate	Determines the min. probability of mutation occurring at each node	$^1/_{30}$
Max. Mutation Rate	Determines the max. probability of mutation occurring at each node	$^1/_{10}$
Mutation Period	Determines the period of the varying mutation rate	60
Creep Mutation Rate	Determines the probability of performing creep mutation on the value of the splitting criterion	$^1/_{10}$
Creep Mutation Size	Determines the largest fraction of points affected by the mutation	0.01
Prune Mutation Rate	Determines the probability of a node becoming a terminal node	$^1/_{200}$
Minimum Prune Depth	Determines the minimum length of a branch before prune mutation can occur	3
Crossover Probability	The probability of performing crossover on the selected individuals	0.6
Tournament Probability	p_T in equation (5.2)	0.75
Tournament Size	s_T in equation (5.2)	6
Initial Tree Sizes	Sets the range of the number of nodes in each tree in the initial population	$[3, 12]$
Population Size	The number of individuals in the population	120
Max. Generations	The maximum number of generations	200

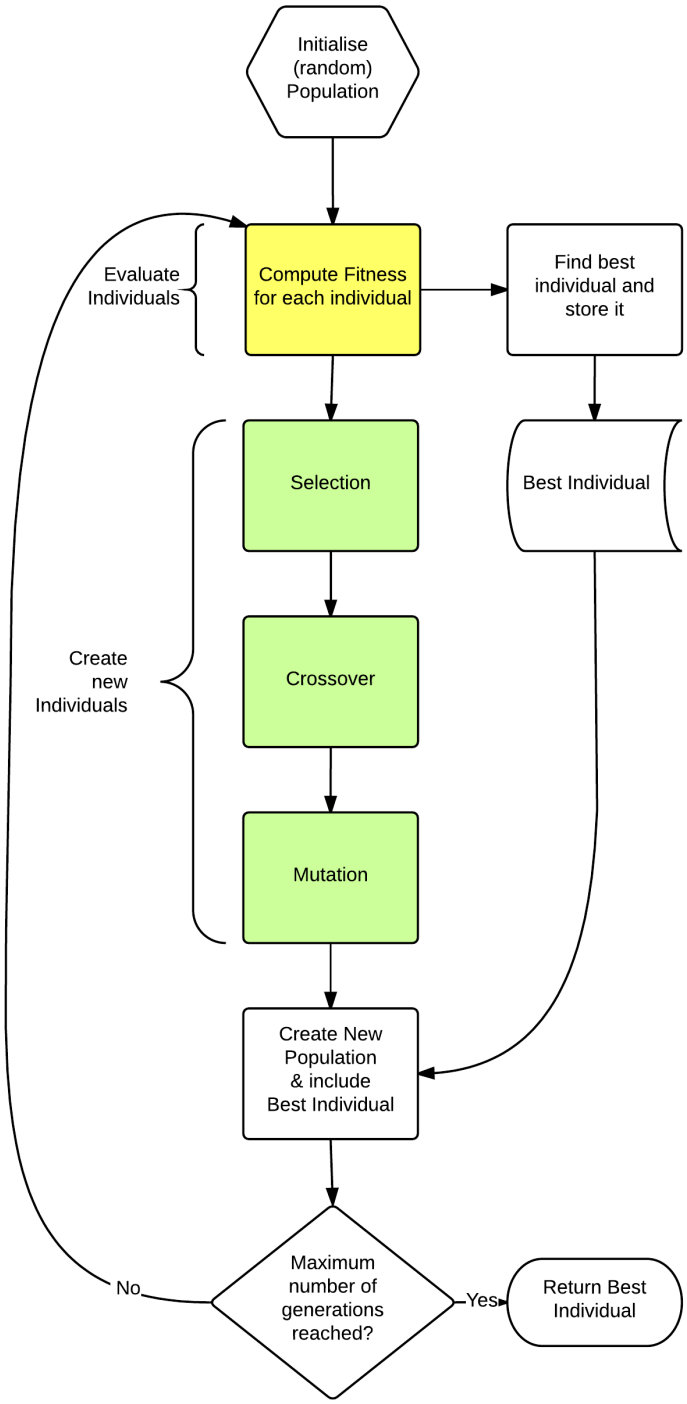
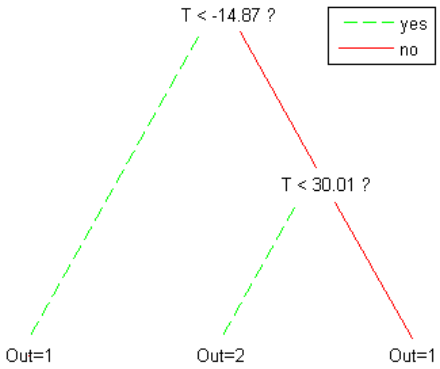
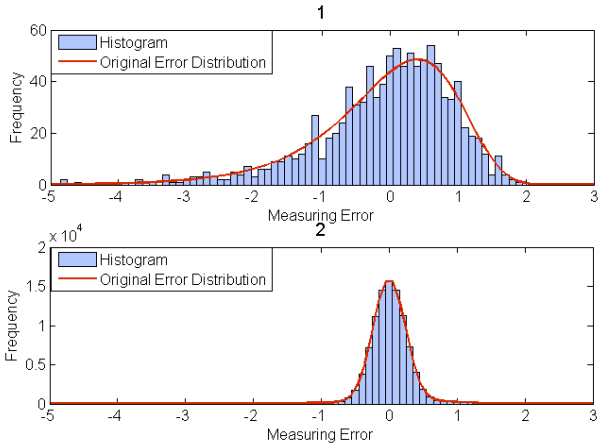


Figure 5.5: *A flow chart visualising a general genetic algorithm. Computing the fitness of each individual is commonly the most computationally heavy step. This also holds true in this case. The driving forces behind evolution (selection, crossover & mutation) can be implemented in different ways depending on the problem and the structure of the individuals. Keeping the best individual through generations is called elitism, and will often improve the performance of the genetic algorithm. Other termination conditions than setting a maximum number of generations can be used. For example running until the fitness of the best individual has not improved over n generations or terminating when the result of the best solution reaches a certain threshold.*



(a) *A typical sensor profiler found using the evolutionary algorithm.*



(b) *The resulting histograms for each sensor profile.*

Figure 5.6: *The chosen sensor profilers and the resulting histograms. The shape of the original distributions match the histograms accurately. This sensor profiler will classify 0.015% of data points into the incorrect sensor profile.*

Let us return once again to the example with the thermometer visualised in Figure 5.1. Selecting the upper limit of groups to two and running the algorithm, using the parameter values above, typically results in similar decision trees between runs. A representative decision tree found using genetic programming is shown in figure 5.6. This tree divides the records so that the records with $T \in [-14.9°\text{C}, +30°\text{C}]$ is in one group and the other group consists of the records with T outside this range. The profiling score for the decision tree found using the evolutionary algorithm is $S = 0.0069$. The worst resulting profiling score found by running the evolutionary algorithm five times on this problem with these parameters was $S = 0.0068$. This can be compared to $S = 0.0066$ which was the result from the greedy algorithm. The typical sensor profiler displayed in figure 5.6 will classify 0.015% of new records into the incorrect sensor profile; the corresponding result for the greedy algorithm was 0.057%. Hence, even in this rather simple case the evolutionary algorithm finds a better sensor profiler than the greedy algorithm. The main drawback of the evolutionary algorithm is that the running time needed is at least 100 times the running time for the greedy algorithm in this example and for more complex cases the difference might be even larger.

5.4 Estimating Probability Density Function within each Sensor Profile

As the shape of the distributions within each sensor profile is unknown and may also differ between profiles a non-parametric density estimation technique should be used. This as fitting a parametric probability density function, *pdf* or *density*, with a specified shape to general data may result in a density dissimilar from the distribution of the data. The most basic form of non-parametric density estimation is a normalised histogram, used for calculating the distance between profiles earlier. A more advanced method that is commonly used for estimating the density and will result in a smooth estimation is *Kernel Density Estimation*.

Kernel density estimation creates a non-parametric probability density estimate based on the data. This is useful when data comes from an unknown distribution. The estimated distribution is similar to a histogram of the data, but has been smoothed using a kernel function. The estimated probability density function is defined as:

$$\widehat{pdf}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

where n is the number of data points, $K(\cdot)$ is the kernel smoothing function and h is the bandwidth. The kernel smoothing function can be any non-negative function that has mean equal to zero and integrates to one. Different kernel functions are commonly used and will result in different estimated distributions. Common

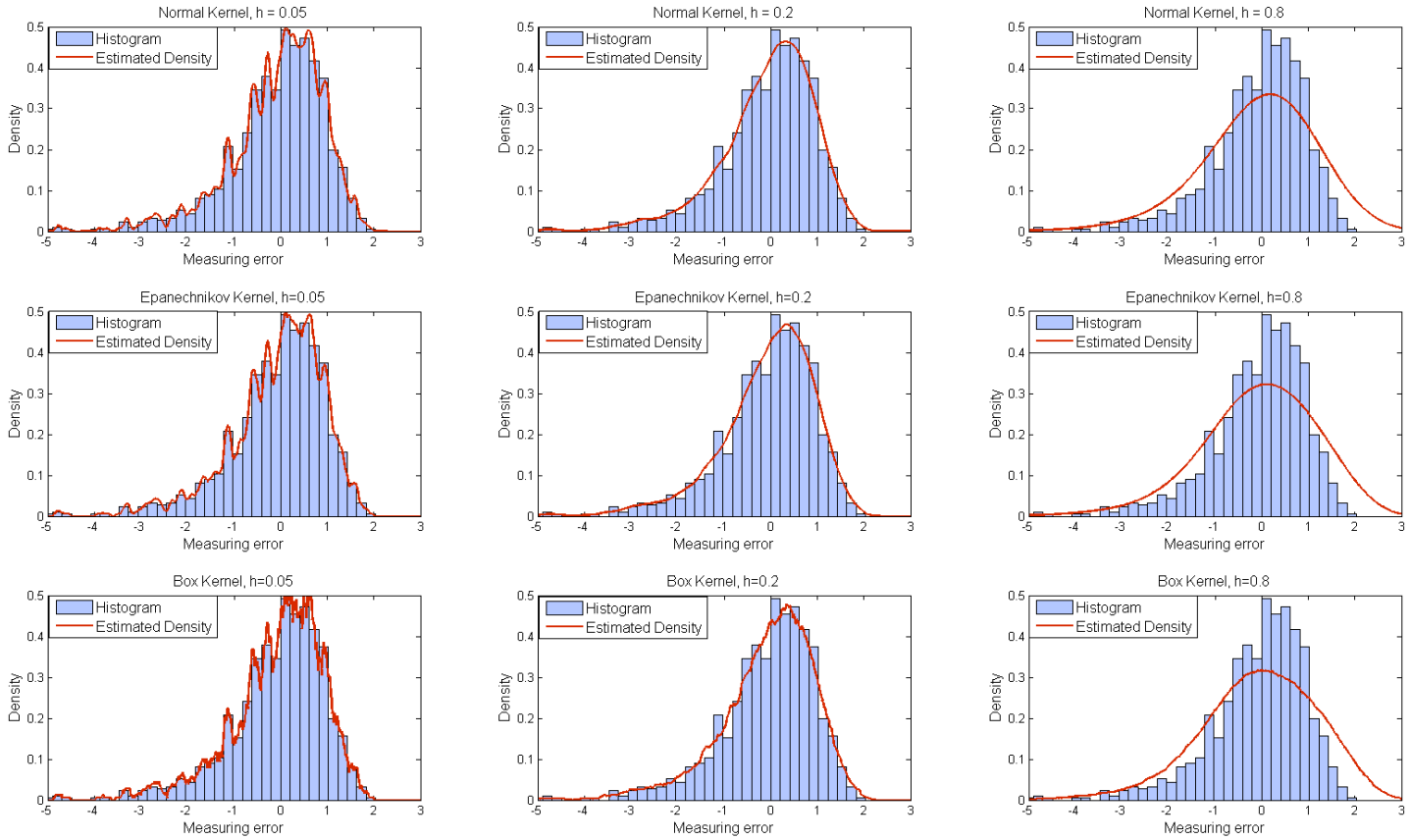


Figure 5.7: *The choice of bandwidth, h, and the choice of kernel smoothing function affects the estimated density. The data comes from the dotted red distribution in figure 5.2 which corresponds to the measuring error in the extreme temperature region. Each row corresponds to a different kernel function and each column a different bandwidth. For this data set the best choice of bandwidth is $h \approx 0.2$.*

examples of a kernel smoothing function are the normal, Epanechnikov or box kernels [31]. However specifying the correct bandwidth is more important with regards to the estimated distribution as seen in Figure 5.7.

Selecting bandwidth, h

The optimal bandwidth is the one that creates an estimated distribution that is most similar to the original distribution which the data comes from. However, since kernel density estimation is used when the original distribution is unknown the optimal bandwidth will remain unknown as well. The bandwidth used is determined based on a rule of thumb, manual trial-and-error or assumptions about the data. If a too small bandwidth is selected the estimate produces a very rough curve, capturing the smallest features of the specific data. Selecting a too large bandwidth produces a wide, overly smoothed, density that may destroy important features of the underlying distribution. MATLAB uses the bandwidth that is optimal if the original distribution is the normal distribution. Another rule of thumb for normal distribution is to use $h = (\frac{4\hat{\sigma}^5}{3n})^{\frac{1}{5}}$, where $\hat{\sigma}$ is the estimated standard deviation of the data [31]. In the application of anomaly detection in sensor readings one can examine the scale of the sensor reading and select a bandwidth equal to the estimated resolution of the signal, this will result in a distribution that at least is not unnecessarily rough [31]. In the example with the thermometer an apt bandwidth choice would be $\approx 0.1°$C, if this can be considered the smallest scale that the outside temperature is measured by the thermometer.

5.5 Conclusions

In general, both the greedy and the evolutionary algorithm produced sensor profiles very close to the original grouping in the data. This was however a constructed example and in real data the changes in distributions will probably not be a sharp edge, but more likely a smooth transition. If this is the case records near the splitting criteria in the decision tree may end up in the wrong profile no matter what the splitting criterion is. This can occur due to hysteresis, if for example, transitions from profile 1 to profile 2 requires different conditions compared to transitions from profile 2 to profile 1. It can also be a result of records near the borders has some probability of belonging to either profile. In either case, records located close to splitting criteria may require special attention depending on the application.

Uses for Sensor Profiles

Finding regions where a certain variable has different distributions can have many applications. It can be used for sensor validation; to create more accurate descriptions of sensor performance across working conditions. It can also be used to create data driven sensor models. Beyond the distribution of measuring errors the auto-correlation within and between different sensor regions may also be needed for creating models. The goal of this thesis is to find anomalies and sensor profiles can be used to separate normal sensor errors from anomalous errors. A record with a certain error might be normal if placed in a one sensor profile, but anomalous if placed in another. Accurately placing records in the correct sensor profile along with accurately estimating the distribution of normal errors becomes important when finding anomalies through the use of sensor profiles. This sensor profiling approach is not in any way limited to sensor data and there could exist applications in other fields where one want to map a set of conditions to different distributions of a specific variable.

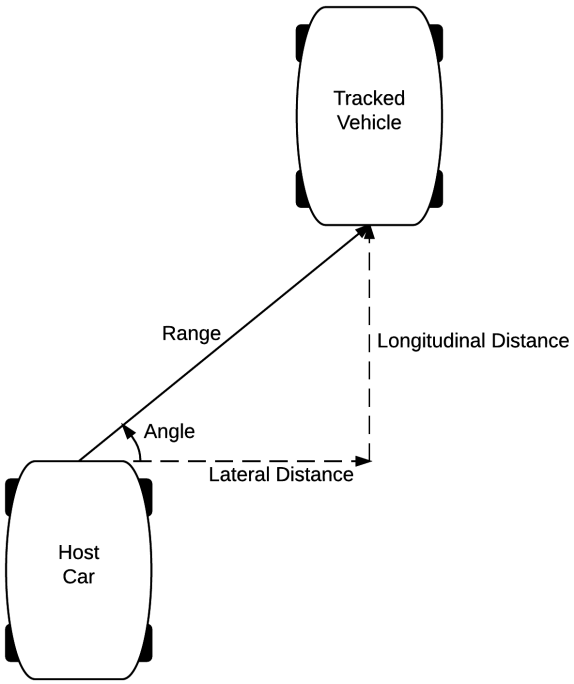


Figure 6.1: *The relationship between some sensor signals given by a sensor used by Volvo for active safety. The longitudinal distance has been used to evaluate the anomaly detection method.*

6 Results

The performance of the general anomaly detection method is evaluated using only a single sensor signal. This chapter contains the results from the regression analysis, sensor profiling and the final anomaly detection results for this single sensor signal. Evaluating the performance of the anomaly detection method across several different sensor signals to form a general conclusion is beyond the scope of this thesis.

The sensor signal used to evaluate the method comes from the RaCam-system used in certain modern Volvo cars and is the longitudinal distance from the host car (where the RaCam is mounted) to another vehicle. A description of the sensor reading used and the relation to other distance measures is shown in Figure 6.1. This signal was used because it is a physical measure (position) where you expect to find a linear relationship to other physical measures (e.g. position at previous time step, velocity, acceleration). It should also be relatively simple for a human to detect anomalies in a sensor signal containing physical measures. Last but not least, being able to find anomalies in this signal is of interest to Volvo Car Corporation.

6.1 Predictive Model

To create a predictive model a number of possible predictors are used to predict a response variable, as described in chapter 4. The response variable is the longitudinal position, x_i, at time step i. The initial set of predictors used are given below, together with a short description. The predictors marked with * where not included in the final model as the method deemed them not to be statistically significant.

Index	Name or Value	Description
X_1	x_{i-1}	Long. position
X_2	$x_{i-1} - x_{i-2}$	Numeric velocity of long. position
X_3	$\dot{x}_{i-1}$	Long. velocity
X_4^*	$\ddot{x}_{i-1}$	Long. acceleration
X_5	y_{i-1}	Lateral position
X_6	$\dot{y}_{i-1}$	Lateral velocity
X_7^*	$\ddot{y}_{i-1}$	Lateral acceleration
X_8	ν_{i-1}	Speed of the host car
X_9	γ_{i-1}	Yaw rate of the host car
X_{10}^*	α_{i-1}	Acceleration of the host car
X_{11}	$x_{i-1} + x_{i-3} - 2x_{i-2}$	Numeric acceleration of long. position

The final model consists of 22 terms in 8 predictors and includes cross-terms, X_2X_3 for instance. As the method can determine the significance of the predictors a large set candidate predictors can be included. It is not required for predictors to have a physical relationship; the amplitude of the sound emitted by the car stereo may be included in the model if it has a statistically significant relationship to the response variable. This enables the statistical model to capture sensor behaviour that would be missed by a traditional physically based model.

Most of the variance ($\approx$99 %) is explained using only the previous position (X_1) and the longitudinal velocity (X_2). The adjusted R^2-value for this model on the total data set with the original outliers removed is $1 - 4 \cdot 10^{-6}$. The adjusted R^2-value for new data, still containing outliers, is 0.9997. This leaves only 0.03 % of the variance that is not explained by the predictive model, which suggests that the predictive model is indeed very accurate. The accuracy may however not be constant. There could be conditions where the model struggles to predict the sensor reading. For example, one can imagine that tracking a fast-moving object causes a different prediction error compared to tracking an object at a constant distance from the host car. Sensor profiles are used to find and describe such differences.

6.2 Sensor Profiles

The existence of sensor profiles is hypothetical at this stage. A natural first step is to visualise the data and manually inspect it in search for any signs of structure in the prediction error. The prediction errors from 180 000 randomly selected records are shown in Figure 6.2. In this data one can see some structure. There are two small bumps in the histogram centred around -0.5 m and 0.5 m and one can also see areas in the time plot where the variance of the prediction error seems to be constant. Deciding the initial set of features, ξ, that may affect the prediction error, ϵ_i, at time step i, was done by consulting engineers with experience of the sensor data and by guessing. As the method (described in Chapter 5) makes use of the most useful features one may include insignificant features as well. Since the running time depends on the number of features included it is however recommended to only use features with a reasonable chance of being significant. The set of ξ is as follows:

Index	Name or Value	Description
ξ_1	x_i	Long. position
ξ_2	$\dot{x}_i$	Long. velocity
ξ_3	$\lvert\phi_i\rvert$	Absolute angle to target
ξ_4	$\lvert\gamma_i\rvert$	Absolute yaw rate of the host car
ξ_5	ν_i	Speed of the host car
ξ_6	α_i	Acceleration of the host car
ξ_7	τ_i	Tracking time of the target

The tracking time of the target measures for how long the object has been tracked by the RaCam and is measured in seconds. To make sure that there actually are regions with differences in prediction error present in the data, one can plot the prediction error against the initial set of features and inspect the data manually. The two most interesting plots, prediction error versus longitudinal velocity and prediction error versus tracking time, are shown in Figure 6.3. As seen in Figure 6.3a the behaviour of the prediction error for negative longitudinal velocities differs from the behaviour where the longitudinal velocity is positive. In

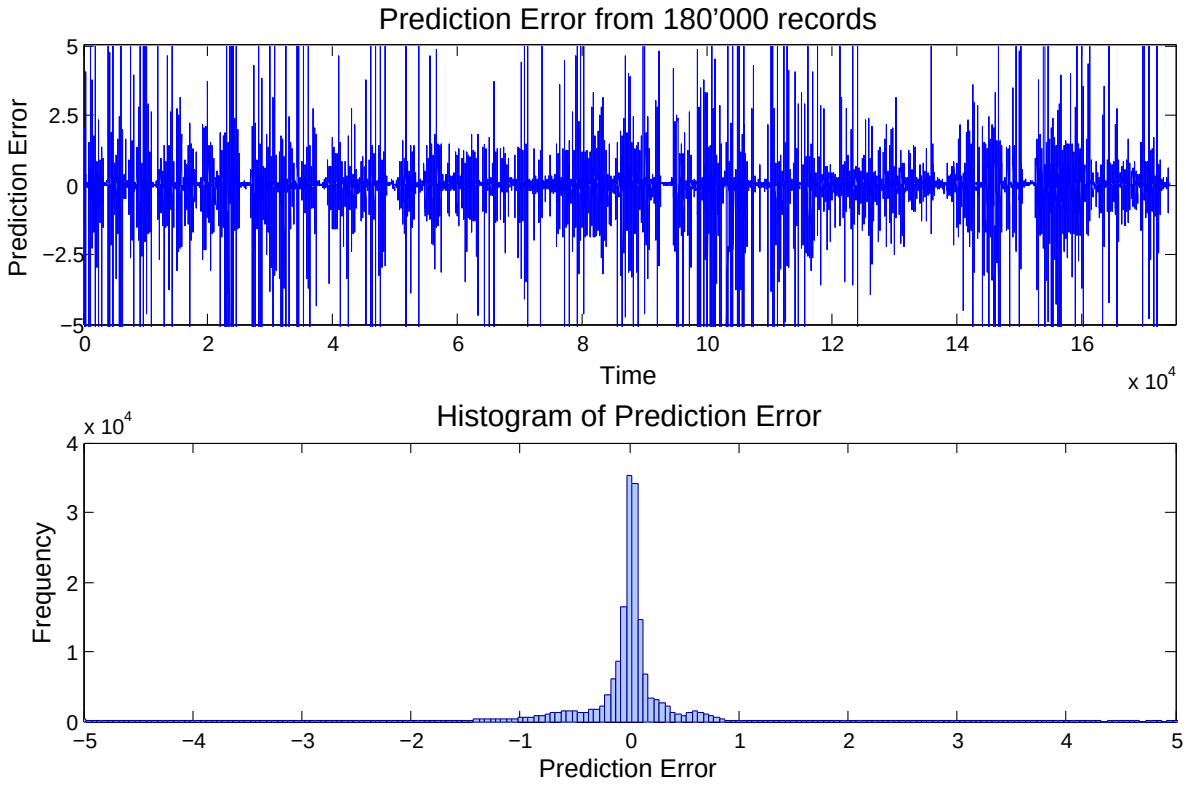


Figure 6.2: *The prediction error as a time series and as a histogram. As seen in the histogram, the majority of the data has a very low prediction error.*

Figure 6.3b one can also distinguish an almost exponential decay in the variance of the prediction error as the tracking time increases.

Determining the number of groups was done using trial and error. Both methods and both distance measures described in Chapter 5 was used. The maximum number of sensor profiles allowed ranged from 2 to 7. The goal was to find a sensor profiler that results in unique histograms in each sensor profile. The chosen sensor profiler was found using the evolutionary algorithm, Earth Mover's Distance and a limit on the number of sensor profiles equal to 3. Additional sensor profiles all exhibited histograms very similar to one of the histograms present in one of the existing sensor profiles. The chosen sensor profiler accompanied by the resulting histograms is shown in Figure 6.4. Of the 3 sensor profiles 2 of them are obtainable through different sets of conditions. As seen in Figure 6.4b the distributions of the errors in the different sensor profiles are all quite different from each other. The different profiles corresponds to the structure visible in Figure 6.3.

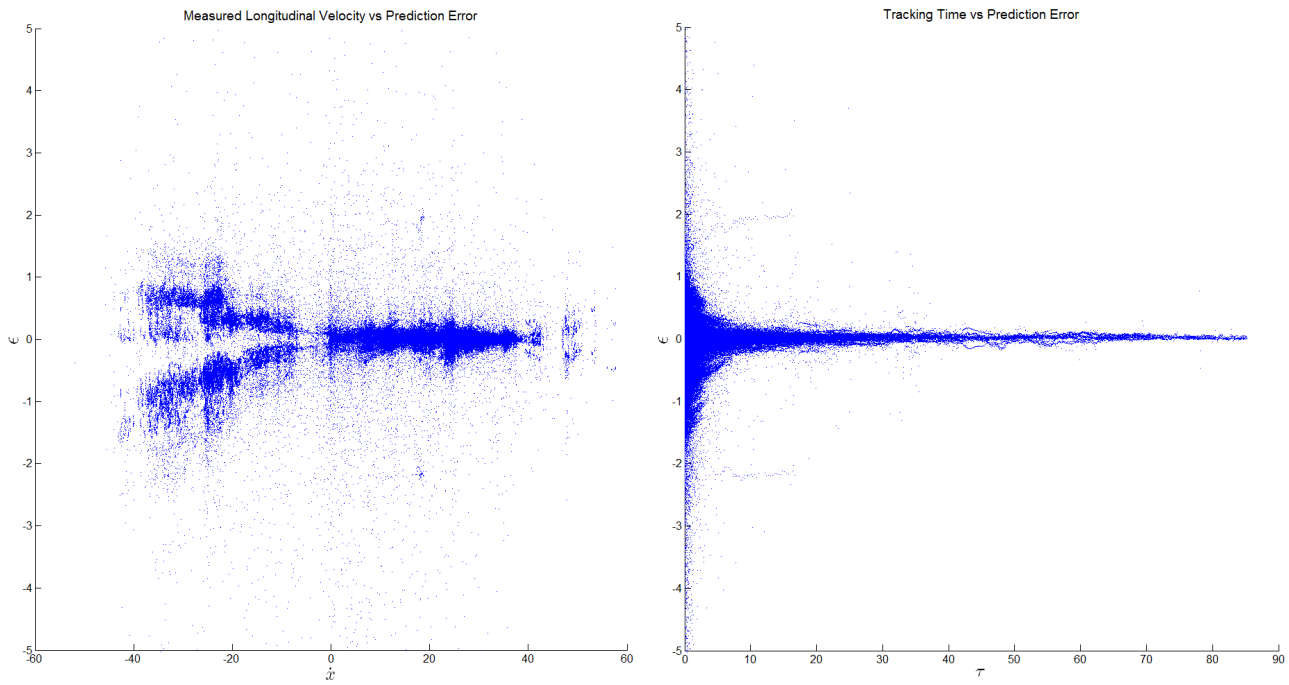
6.3 Defining Normal Behaviour

Using the sensor profiler shown in Figure 6.4a one can establish the normal behaviour of the prediction error within each sensor profile. Comparing new data to the estimated normal behaviour is the basis of the anomaly detection technique. To ensure that the differences in the data set used to find the sensor profiler are representative for new data, a new data set was created and the data points classified using the sensor profiler. The resulting histograms and the estimated density for each sensor profile is shown in Figure 6.5.

Using the predictive model, the sensor profiler in Figure 6.4a and the estimated densities in Figure 6.5 allows us to assign any new record to a sensor profile and compute the probability density for the error to occur within the sensor profile. The probability density value of the prediction error is used as a 'normality measure', n_i of the record with index i. The anomaly score of the record, a_i, is defined as

$$a_i = 1 - n_i. \tag{6.1}$$

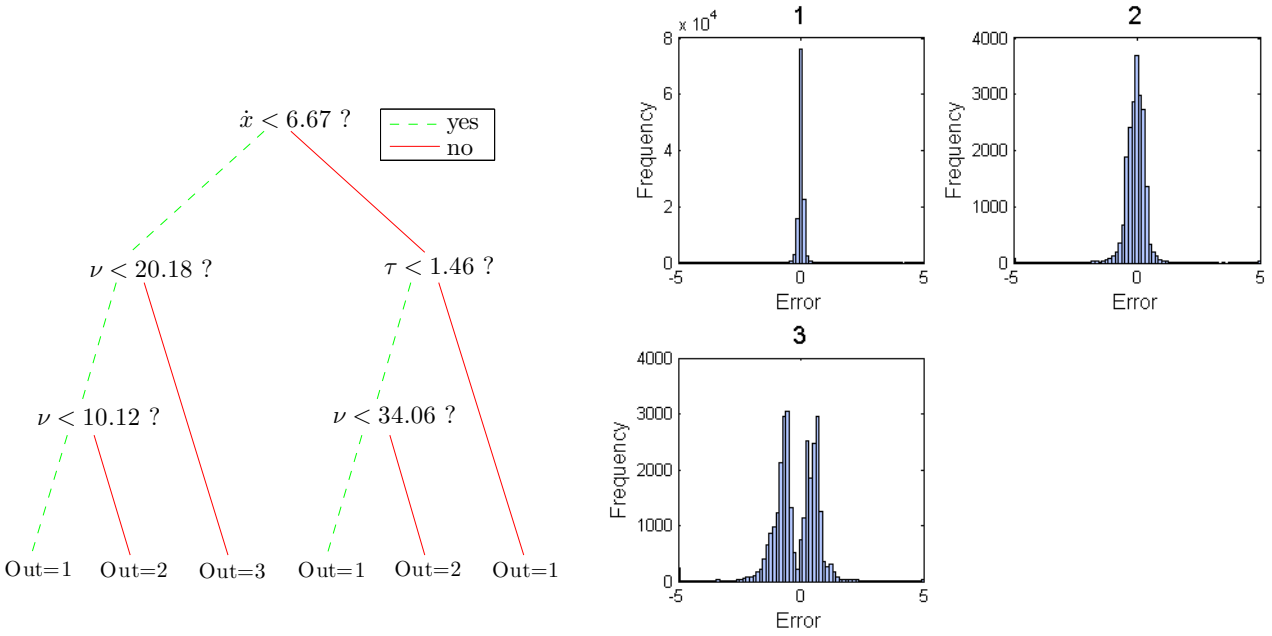
Anomalies now correspond to records with $a_i >$ threshold. The remaining records are considered to be normal. Selecting a suitable threshold is done simultaneously as evaluating the performance of the anomaly detection method. To perform the evaluation and threshold selection a labelled data set is needed.



(a) *Prediction error versus longitudinal velocity*

(b) *Prediction error versus tracking time*

Figure 6.3: *There are some relation between the prediction error and both longitudinal velocity and tracking time. This figure is used to justify the use of sensor profiles.*



(a) *The chosen sensor profiler*

(b) *The resulting histograms for each sensor profile*

Figure 6.4: *The chosen sensor profilers and the resulting histograms. This sensor profiler was found through genetic programming and as you can see the appearance of the three histograms are qualitatively different and corresponds to the structure seen in Figure 6.3a.*

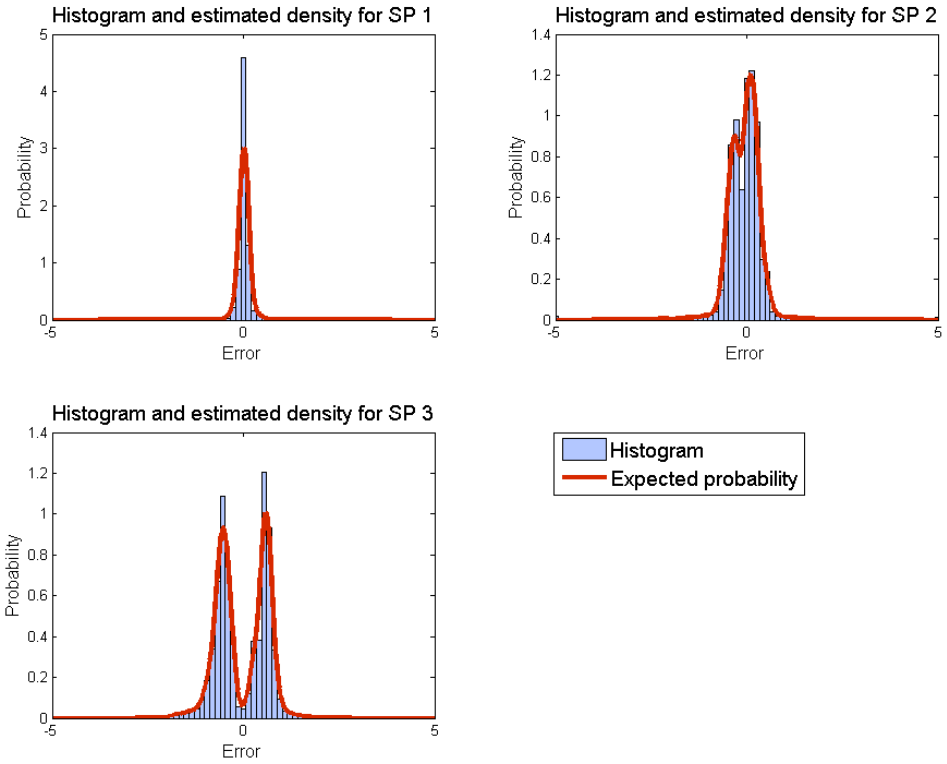


Figure 6.5: *The histograms and estimated densities for each sensor profile found through the use of the sensor profiler in Figure 6.4a. The histogram for sensor profile 2 differs slightly from the corresponding histogram in Figure 6.4b. The resulting histograms are however considered to be similar enough to ensure that the estimated density is representative for the entire data set.*

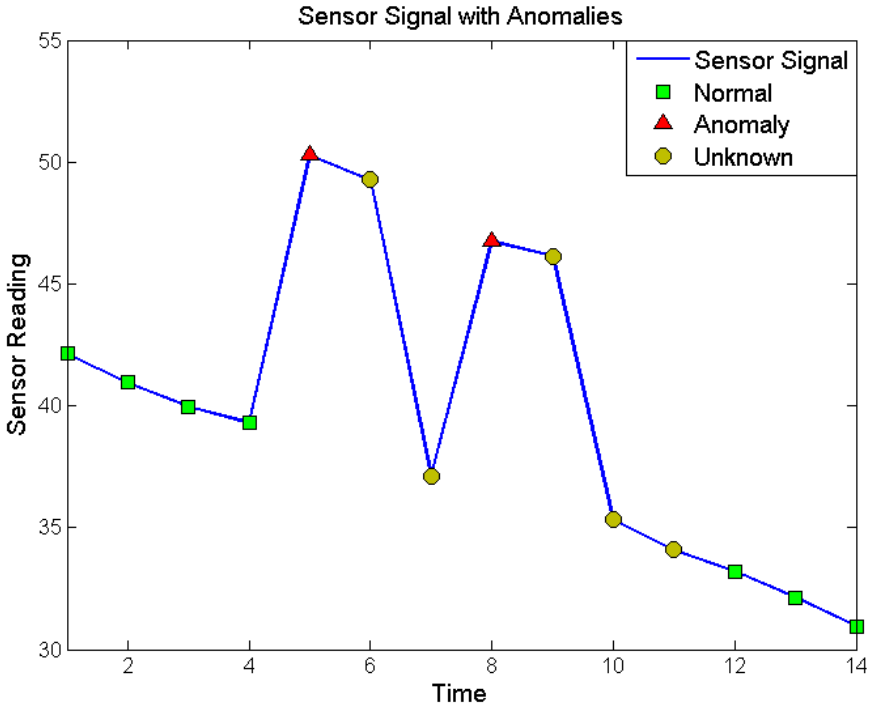


Figure 6.6: *A constructed example of a sensor signal containing anomalies. The records marked 'Unknown' is difficult to label as either normal or anomalous. Establishing the normal sensor behaviour at those time steps is difficult since the sensor exhibits abnormal behaviour.*

6.4 Creating a Dataset for Evaluation

A labelled data set with known anomalies and normal records is needed to evaluate the anomaly detection method and to select the threshold. The only way to accurately label a data set is to do it manually. Labelling each individual record accurately was considered unreliable, even when performed manually. Even if a human can accurately distinguish between most anomalies and normal readings, there can exist readings that are 'quite anomalous', but maybe not anomalies. An example of this is shown in figure 6.6. Anomalies are in this thesis defined as unexpected measurements. In regions where anomalies occur the expected measurements can be difficult to assess. It becomes subjective and dependent on what records are included in the assessment. Evaluating the performance of the anomaly detection method on subjectively (and possibly false) labelled records may lead to inaccurate conclusions or choice of threshold.

An attempt to eliminate the risk of false labelling is to manually label regions instead of individual records. It was found to be an easier task to manually label an interval where the sensor readings exhibit normal or anomalous behaviour. The data set is divided into regions where a region that contains at least one anomaly is labelled anomalous and regions that contain only normal records are labelled normal. Labelling the regions in Figure 6.6 will result in two normal regions (Time $\in[1, 4]$ & Time $\in[12, 14]$) and one anomalous region (Time $\in[5, 11]$). A region that consists of possible non-normal records but no record that can be considered an anomaly is excluded from the data set to avoid false labels. An example of such a region is when there too few data points to manually determine the normal behaviour.

Evaluating the performance of the anomaly detection using regions instead of individual records should not compromise the accuracy of the evaluation. The future use of the anomaly detection method is to automatically find sensor malfunctions. If the anomaly detection method marks a number of records located close together that region will require further analysis. Missing a possible anomaly in that region, or falsely marking a possibly normal record, will not affect the performance of the anomaly detector. Marking a single record as an anomaly in a region containing only normal records will however have a negative effect on the performance.

The Dataset

The dataset consists of 46 anomalous regions and 448 normal regions. 79 regions where excluded from the original data due to difficulties labelling the regions. In general the normal regions are larger than the anomalous regions. The data used to create the dataset comes from the same database as the data used for training the anomaly detection methods but was not used during any training.

6.5 Evaluating the Performance of the Anomaly Detection

All variants of the anomaly detection method in this thesis assigns an anomaly score to each record. To convert it to regions an anomaly score must be assigned to each region. This score is equal to the maximum anomaly score found on the records which make up the region. The anomaly detection method is evaluated on the data set by computing the true positive rate and the false positive rate for each of a set of thresholds to create a ROC-curve for each of the anomaly detection methods.

The method resulting in the anomaly score in Equation (6.1) is compared against two simpler anomaly detection methods. The first consists of assigning an anomaly score equal to:

$$a_i = |\epsilon_i|, \tag{6.2}$$

where ϵ is the prediction error computed using the predictive model from section 6.1.

The other simple anomaly detection method, computes the moving average and defines the anomaly score to be the difference between the sensor reading and the moving average:

$$a_i = \left| x_i - \frac{1}{5} \sum_{k=i-2}^{i+2} x_k \right|. \tag{6.3}$$

The resulting ROC-curves is seen in Figure 6.7. The performance of the three anomaly detection methods are rather similar, but using sensor profiling improves the result slightly. For a certain threshold this anomaly detection method accurately classifies 94% of both the normal and the anomalous regions.

ROC-curves were created using other sensor profilers as well. The other profilers were found using other parameters and had different numbers of sensor profiles. The performance of these were all very similar to the performance of the method in Figure 6.7. Thus they were excluded from the report. A possible reason that sensor profiling does not drastically improve the anomaly detection result is that almost all anomalies lie outside the region where there is a significant difference between the different sensor profiles. Sensor profiling will only improve the result when records are anomalous if within a certain sensor profile, but normal in another profile. For this sensor signal almost all anomalies are considered anomalous no matter what sensor profile they belong to.

The anomaly detection based on moving average has comparable performance to the other, more complicated, methods. This is probably due the nature of the sensor signal and the anomalies. Moving average will be a rather accurate predictor if the velocity is constant. Even if the moving average has a worse overall performance, most anomalies are much larger than the typical error. This enables the moving average anomaly detection method to be just as good as the best method for finding the most obvious anomalies. The predictive model is also affected by errors in the predictor signals, whilst the moving average is immune to such errors and will only be affected by errors in the response variable.

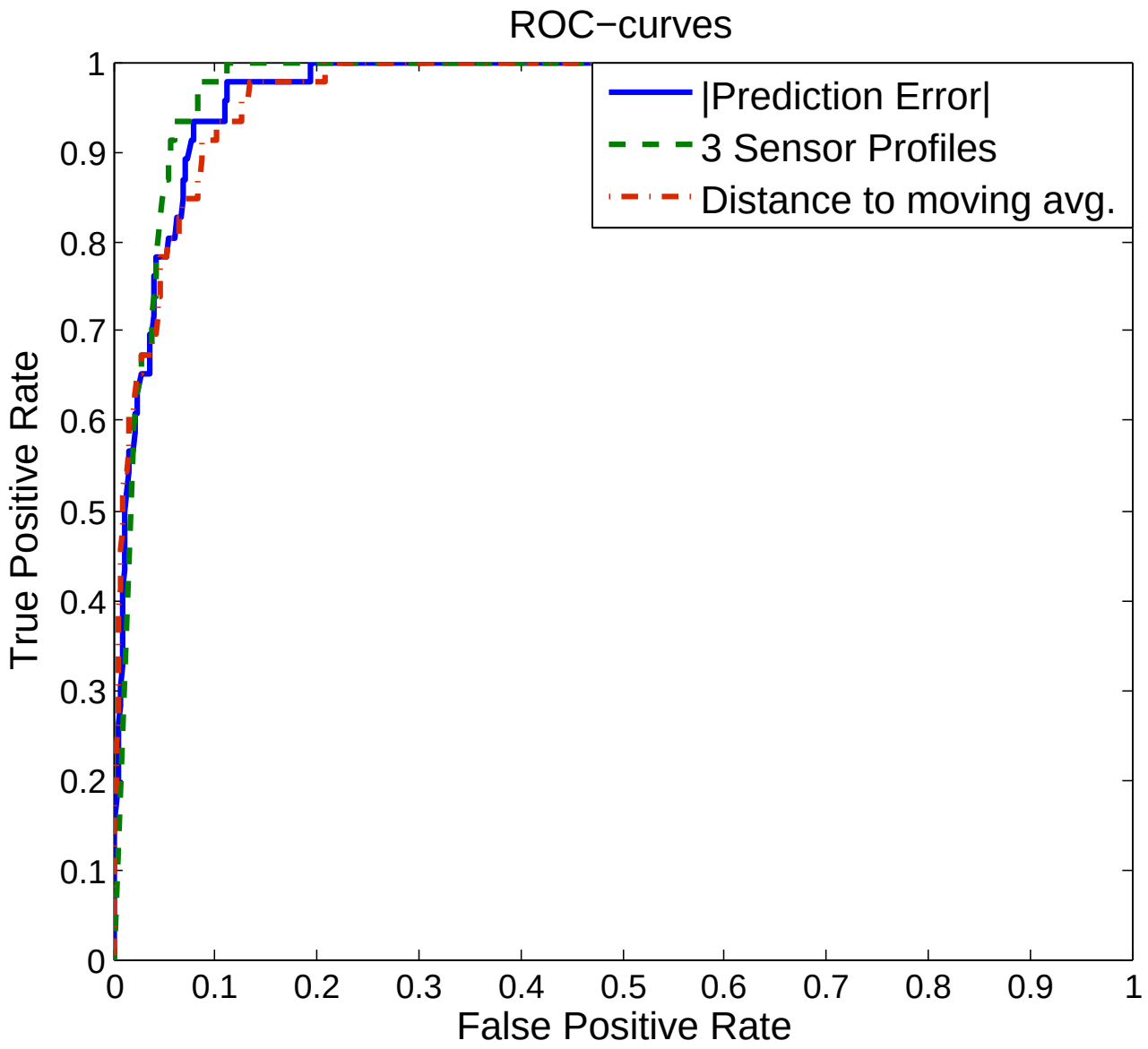


Figure 6.7: *The ROC-curves from 3 different anomaly detection methods created by varying the threshold of the anomaly score. The optimal result is found in the top left corner and the most suitable threshold depends on the application. The performance of the anomaly detection methods are comparable but the use of sensor profiles yields a slightly better result.*

7 Discussion

In this chapter the general anomaly detection method and the results are discussed. Other uses for the methods included in this thesis will also be presented. An attempt to describe the importance of suitable predictors and how the behaviour of the sensor that is to be analysed affects the performance will also be made.

7.1 Predictive Model

The performance of the predictive model stands or falls with the statistical significance of the initial set of predictors. The method for finding this model in this thesis will result in a good sensor model when given access to significant external predictors. Typical external predictors are sensor signals which measure the same entity or system. Advanced sensor or sets of sensors measuring the same entity may exhibit partial redundancy. This can be used to validate a sensor reading through prediction. Without access to external predictors the model may only explain the most basic behaviour of the sensor and will not account for dynamic changes in the monitored entity. A sensor measuring the distance to objects surrounding the sensor may be effected by the velocity of the sensor itself, the structure of the objects or the acceleration of the objects. Including external predictors in a model can enable the model to foresee changes in a sensor signal that can not be predicted using only the history of that signal. In extreme cases can a sudden and seemingly unpredictable change in a signal be predicted by a model using external predictors; a large change in position and velocity is predicted by observing a large difference in acceleration. If such a change is considered (by human analysts) to be an anomalous event it may pass the anomaly detection method undetected. This as such an event do not fit the definition of an anomaly being unpredictable. Analysing the predictive model to not only ensure the future performance of the anomaly detection, but also to understand the behaviour of the sensor is therefore important.

7.2 Sensor Profiles

Sensor profiles finds differences in the performance of the predictive model. These differences will originate from differences in the behaviour of either the sensor signal that is to be analysed or one or several predictors. The usage of sensor profiles can be extended beyond anomaly detection to sensor verification. Instead of the prediction error the sensor can be compared to a reference sensor that is known to be very accurate. If this error is used one can find sensor profiles where the true performance of the sensor differs. Automatically finding regions where the sensor performance differs is useful when deciding uses of the sensor and also when evaluating results based on senor readings. If an active safety function has a certain success rate using ideal sensor readings, the perceived performance of the function in an actual car will also depend on the accuracy of the sensor signal it relies upon. The method for finding sensor profiles is general and can be used on any dataset where the behaviour of a stochastic variable is of interest. It can be seen as a method for clustering distributions and can be used when other statistic and machine learning tools are unable to extract any information from the data.

7.3 Most Suitable Anomaly Detection Method

The accuracy of the predictive model and the differences between the distributions in the sensor profiles determines whether sensor profiles will be useful for anomaly detection or not. Basically if a record is an anomaly in one sensor profile and normal in another, sensor profiles may be useful, given that the regions are separable. For the sensor signal used to evaluate the method in chapter 6 the use of sensor profiling improves the result of the anomaly detection slightly. Using the prediction error directly as an anomaly measure will yield almost identical results for this particular sensor signal. This as most anomalies are located outside the region where there is a significant difference between the sensor profiles. In signals where anomalies occur in regions where there is a large difference in density between senor profiles the use of sensor profiling may greatly improve the results. However, it will only be able to do so if it can accurately map regions, where the behaviour of the error differs, to the correct sensor profile. To achieve an improvement the access to features $(\xi_1, \ldots, \xi_m)$ that are related to the error is therefore crucial. Using sensor profiling when finding anomalies in other sensor signals may yield a much better result than simply using the prediction error, the choice of methods depends on the accuracy of the predictive model and the behaviour of the sensor signal.

8 Future Work

8.1 Evaluate the Anomaly Detection Method on General Data

The aim of this project was to produce a general anomaly detection method for sensor data. Examining how this method performs against other methods on several types of data, with different types and sizes of anomalies, would of course be very interesting. The possibe need for access to external predictors can be seen as a drawback, but the possibility of using correlated sensor signals may prove very useful in certain situations.

8.2 Evaluate True Sensor Performance with Sensor Profiles

The novel method introduced in this thesis used to determine sensor profiles might have several uses beyond anomaly detection. Establishing ranges of conditions where the true sensor performance differs would be useful. Having an accurate description of the sensor performance may, at least in the case regarding active safety in vehicles, lead to lives being saved.

8.3 Use Methods to Create Sensor Model

Predicting measurements and knowing the distribution of prediction error takes you close to creating an accurate virtual sensor model. Creating a model, using the statistical tools used in this thesis and real world data, could result in a rather simple and statistically proven model. This model would be data-driven and not rely on assumptions regarding the sensor or the traffic environment. Further analysis using other statistical methods is probably needed to create such a model, estimating the auto-correlation to predict sequential readings for instance. Also, instead of predicting sensor output based on predictors from the sensor itself true readings is probably needed. The predictive model only produces the expected output of the sensor, it does not reveal any information regarding the difference between sensor reading and the true reading. A sensor signal can behave normally and still show extremely inaccurate readings. Constant errors can not be seen in the prediction error. If one wishes to create a sensor model it will need to map a description of the measured entity to sensor outputs. To create an accurate sensor model a data set containing a true description of the measured entity and the measured sensor output is needed. An accurate sensor model could be used to virtually assess the performance of active safety functions which would hasten the development of such functions.

8.4 Real Time Implementation

The anomaly detection method proposed in this thesis can be used in real time. Using it to flag anomalies as they occur is of course also an interesting application. The main drawback of this anomaly detection method is that it can not determine the correctness of the anomalous record. It may very well be the case that the measurements leading up to the anomaly are incorrect and the anomalous measurement is accurate. A function that can determine that something has gone wrong is not very useful without the abilities to determine when the actual error occurred, why it has occurred or even what the correct result actually is.

9 Conclusion

This thesis presents a method and tool chain to automatically detect anomalies in any sensor data that can be represented as a time series. The performance of this method is still to be proven on a large set of general sensor data, but it shown to perform well on the one signal it was evaluated on. The method should have particularly good performance when used on sensor systems consisting of several sensor signals. This as it makes use of partial redundancy and the statistical relationship between different signals.

Sensor profiling, that was introduced in this thesis, may prove useful in other areas as it is a general way to describe the relationship between continuous variables and a continuous stochastic variable.

References

[1] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection", *Network, IEEE*, vol. 8, no. 3, pp. 26–41, 1994.

[2] R. J. Bolton and D. J. Hand, "Statistical fraud detection: a review", *Statistical science*, pp. 235–249, 2002.

[3] C. D. Spence, J. C. Pearson, and P. Sajda, *Method and apparatus for training a neural network to learn hierarchical representations of objects and to detect and classify objects with uncertain training data*, US Patent 6,018,728, 2000.

[4] T. Yairi, Y. Kato, and K. Hori, "Fault detection by mining association rules from house-keeping data", in *Proc. of International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Citeseer, vol. 3, 2001.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey", *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[6] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[7] T. W. Anderson, *The statistical analysis of time series*. John Wiley & Sons, 2011, vol. 19.

[8] A. Soule, K. Salamatian, and N. Taft, "Combining filtering and statistical methods for anomaly detection", in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, USENIX Association, 2005, pp. 31–31.

[9] D. J. Hill, B. S. Minsker, and E. Amir, "Real-time bayesian anomaly detection for environmental sensor data", in *Proceedings of the Congress-International Association for Hydraulic Research*, Citeseer, vol. 32, 2007, p. 503.

[10] K. Chakraborty, K. Mehrotra, C. K. Mohan, and S. Ranka, "Forecasting the behavior of multivariate time series using neural networks", *Neural networks*, vol. 5, no. 6, pp. 961–970, 1992.

[11] B. D. Ripley, *Spatial statistics*. John Wiley & Sons, 2005, vol. 575.

[12] F. J. Anscombe, "Rejection of outliers", *Technometrics*, vol. 2, no. 2, pp. 123–146, 1960.

[13] A. J. Fox, "Outliers in time series", *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 350–363, 1972.

[14] B. Pincombe, "Anomaly detection in time series of graphs using arma processes", *ASOR BULLETIN*, vol. 24, no. 4, p. 2, 2005.

[15] H Zare Moayedi and M. Masnadi-Shirazi, "Arima model for network traffic prediction and anomaly detection", in *Information Technology, 2008. ITSim 2008. International Symposium on*, IEEE, vol. 4, 2008, pp. 1–6.

[16] A. M. Bianco, M Garcia Ben, E. Martinez, and V. J. Yohai, "Outlier detection in regression models with arima errors using robust estimates", *Journal of Forecasting*, vol. 20, no. 8, pp. 565–579, 2001.

[17] B. S. Everitt, "The cambridge dictionary of statistics", *Cambridge: Cambridge*, 2002.

[18] F. V. Jensen, *An introduction to Bayesian networks*. UCL press London, 1996, vol. 210.

[19] M. W. Lipsey and D. B. Wilson, *Practical meta-analysis*. Sage publications Thousand Oaks, CA, 2001, vol. 49.

[20] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms", *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[21] I. Welch and A. Goyal, "A comprehensive look at the empirical performance of equity premium prediction", *Review of Financial Studies*, vol. 21, no. 4, pp. 1455–1508, 2008.

[22] R. Kohavi et al., "A study of cross-validation and bootstrap for accuracy estimation and model selection", in *Ijcai*, vol. 14, 1995, pp. 1137–1145.

[23] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John Wiley & Sons, 2005, vol. 589.

[24] H. Akaike, "A new look at the statistical model identification", *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, 1974.

[25] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, "Akaike information criterion statistics", *Dordrecht, The Netherlands: D. Reidel*, 1986.

[26] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions", *City*, vol. 1, no. 2, p. 1, 2007.

[27] O. Pele and M. Werman, "The quadratic-chi histogram distance family", in *Computer Vision–ECCV 2010*, Springer, 2010, pp. 749–762.
[28] M. Wahde, *Biologically inspired optimization methods: an introduction*. WIT press, 2008.
[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, et al., *Introduction to algorithms*. MIT press Cambridge, 2001, vol. 2.
[30] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology", 1990.
[31] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.