Automatic Fault Detection using Cause and Effect Rules for In-vehicle Networks

Alexander Kordes, Sebastian Wurm, Hawzhin Hozhabrpour and Roland Wismüller
Operating Systems & Distributed Systems, University of Siegen, Hölderlinstraße 3, Siegen, NRW, Germany

Keywords: Cause and Effect Rules, Machine Learning, In-vehicle Sensor Network, Fault Detection.

Abstract: In-vehicle networks (IVNs) connect Electronic Control Units (ECUs) for automotive applications. Most of the communication on the IVNs directly affect the comfort or even the safety of the driver. Therefore, it is necessary to monitor these systems in order to find the cause and effect of a fault. Current developments use plausibility checks in automotive ECUs to enhance safety and security. Within the LEICAR project in cooperation with INVERS GmbH we focus on all sensors signals recorded directly from CAN bus IVNs for this positional paper. Even without the knowledge of the sensors semantics it is possible to extract cause and effect rules for all recorded sensor signal relationships of the vehicle, map them in a graph and extract certain situations. The proposed solution detects direct and slowly evolving changes even if they propagate across several involved sensor values. For the automatic fault containment we extract features from the cause and effect rules to train a machine learning model in order to make predictions on new data. Besides that it is possible to implement optimized error checking procedures for the involved ECUs.

1 INTRODUCTION

Robust fault detection is a stringent requirement for the evaluation of safety-critical applications on in-vehicle networks (IVNs). Unfortunately, the search space in recorded IVN sensor data increases with the number of sensors mounted to the vehicle. In terms of autonomous driving the number of involved sensors and actuators increases with every new vehicle model. Sensors and actuators are connected to electronic control units (ECUs) which transmit the sensor data over the internal network. Therefore, the automotive engineers of vehicle manufacturers spend a lot of time with manual fault detection for vehicle prototypes, if the faults are not detected by the on-board diagnostics (OBD), which only performs plausibility checks on the sensor data. However, faults effecting the causal relationship between the sensor signals in the IVN, which are not detected by the OBD, may occur during the machine life of a vehicle. These faults can propagate through the IVN and need knowledge about the whole system, to find the cause of them. It should be mentioned that the detection of faults, that occur but just affect the system slowly over a long period of time, like mechanical wear, is very time-consuming for vehicle repair shops.

This is why we focus in this paper on a method for automatic situation detection, in order to reduce the search space for the automotive engineers. It is a part of the LEICAR project[1]. The method presented in this paper is adapted from (Hira and Deshpande, 2016) and adjusted by us in order to manage IVN sensor data instead of socio-economic indicators, like in the original method. We calculate causal relationships between sensors, to obtain features for an automatic fault detection with machine learning algorithms. This is verified with simulation and real IVN data in which we first detect situations and later perform automatic fault detection. In the future it could be possible to integrate better fault detection mechanisms in the vehicle using the outcome of the proposed method.

This conceptional paper is organized as follows: We describe the relevant parts of the method of modeling cause and effect rules for socio-economic indicators in Section 2. In Section 3, we state the problems consisting of applying cause and effect rules to IVN sensor data and extending the rules for automatic fault containment. The related work is shown in Section 4. In Section 5, we present the proposed method based on the creation of extended cause and effect

[1]Partially funded by German Federal Ministry of Education and Research (BMBF).

Kordes, A., Wurm, S., Hozhabrpour, H. and Wismüller, R.
Automatic Fault Detection using Cause and Effect Rules for In-vehicle Networks.
DOI: 10.5220/0006792605370544
In *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018)*, pages 537-544
ISBN: 978-989-758-293-6

rules and their visualization. We also discuss detectable faults and the limits of the proposed method. The effectiveness of the method is reviewed in Section 6 using simulated and real data of a vehicles braking situations. We conclude this paper in Section 7.

2 EXTRACTION OF CAUSE AND EFFECT RULES

For a better understanding, we will briefly discuss the most important aspects of the calculation of binary cause and effect rules described in (Hira and Deshpande, 2016). The approach is used to model the depicted relationships in Figure 1 between various stock market prices. The arrows in Figure 1 indicate a relationship in the sense of a cause and effect rule between the two time series P_i and P_j. The arrow points from the cause to the effect and if the cause occurs, the effect takes place after a time lag $\ell > 0$. The time series of the stock market prices in (Hira and Deshpande, 2016) have one sample per year.

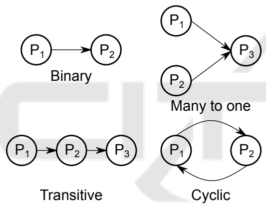

Figure 1: Causal relationships.

For the extraction of cause and effect relationships it is necessary to calculate the rate of change (ROC) per time series first. The ROC is an extension of the momentum, which is a simple technical analysis where indicators show the difference between, e.g., today's closing price and the close n days before. The ROC is expressed as a ratio between a change in one time series relative to a corresponding change in another. Graphically, the ROC is represented by the slope of a line.

To identify relationships between two time series P_i and P_j, first the ROC $\gamma_{i,k}$ of every time series P_i in the $k-th$ year is calculated. δ defines the minimum ROC used to consider a significant change. In (1) each time series value is categorized as a positive ROC (U), a negative ROC (D) or no ROC (Q). The type of change $R_{i,k}$ is defined as follows:

$$R_{i,k} = \begin{cases} U & \text{if } \gamma_{i,k} \geq \delta \\ D & \text{if } \gamma_{i,k} \leq -\delta \\ Q & \text{if } -\delta \leq \gamma_{i,k} \leq \delta \end{cases} \tag{1}$$

The result from equation (1) is used to calculate the strongest relationship between two time series with a time lag ℓ. The direct relationship $D_{i,j,k,\ell}$ between two time series P_i and P_j, regarding to a time lag ℓ, is calculated according to:

$$D_{i,j,k,\ell} = \begin{cases} 1 & \text{if}(R_{i,k} = U \wedge R_{j,k+\ell} = U) \vee \\ & (R_{i,k} = D \wedge R_{j,k+\ell} = D) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In case of the direct relationship, if the rate of change of P_i matches with the rate of change of P_j after a time period ℓ, the support count of the direct relationship is defined as:

$$S_D(P_i, P_j, \ell) = \sum_{k=1}^{n-\ell} D_{i,j,k,\ell} \tag{3}$$

The support percent of the direct relationship between P_i and P_j is defined as:

$$\alpha_D(P_i, P_j, \ell) = \frac{S_D(P_i, P_j, \ell)}{n - \ell} \tag{4}$$

Thus, the temporal direct relationship between two time series P_i and P_j for a time lag ℓ is defined in (5), with α_1 defined as the threshold for all causal relationships.

$$P_i \xrightarrow{\ell} P_j \text{ if } \alpha_D \geq \alpha_1 \tag{5}$$

The inverse relationship I is defined as:

$$I_{i,j,k,\ell} = \begin{cases} 1 & \text{if}(R_{i,k} = U \wedge R_{j,k+\ell} = D) \vee \\ & (R_{i,k} = D \wedge R_{j,k+\ell} = U) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The support count S_I is defined as:

$$S_I(P_i, P_j, \ell) = \sum_{k=1}^{n-\ell} I_{i,j,k,\ell} \tag{7}$$

Thus, the support percent of indirect relationship α_I is defined as:

$$\alpha_I(P_i, P_j, \ell) = \frac{S_I(P_i, P_j, \ell)}{n - \ell} \tag{8}$$

The strength of the relationship Θ_R is defined as:

$$\begin{aligned} \Theta_R(P_i, P_j) &= \alpha \cdot log(n), \text{ where} \\ \alpha &= \alpha_D(P_i, P_j, \ell) \text{ or } \alpha_I(P_i, P_j, \ell) \end{aligned} \tag{9}$$

The count of the number of pairs when no rate of change in P_i is associated with positive or negative rate of change in P_j after a time period ℓ is called neutral-change $C_E(P_i, P_j, \ell)$. The count for the inverse relationship is called change-neutral $C_F(P_i, P_j, \ell)$ and the count for a neutral ROC is called neutral $C_N(P_i, P_j, \ell)$. The definitions should be taken from (Hira and Deshpande, 2016).

With the calculated relationships, the temporal odds ratio TOR is defined. The direct odds ratio OR_D

and the temporal indirect odds ratio OR_I between two time series P_i and P_j for a time lag ℓ are defined as:

$$OR_D(P_i,P_j,\ell) = \frac{S_D(P_i,P_j,\ell) \cdot C_N(P_i,P_j,\ell)}{C_E(P_i,P_j,\ell) \cdot C_F(P_i,P_j,\ell)} \quad (10)$$

$$OR_I(P_i,P_j,\ell) = \frac{S_I(P_i,P_j,\ell)}{C_E(P_i,P_j,\ell) \cdot C_F(P_i,P_j,\ell)} \quad (11)$$

If the result of a neutral relationship $C_N(P_i,P_j,\ell)$, $C_E(P_i,P_j,\ell)$ or $C_F(P_i,P_j,\ell)$ between two time series P_i and P_j is zero, it is considered as one to avoid an infinite temporal odds ratio TOR.

According to (Hira and Deshpande, 2016) a binary rule set (BRS) exists between P_i and P_j if there is a temporal association rule $P_i \xrightarrow{\ell} P_j$ and $OR_D \geq \beta$ or $OR_I \geq \beta$, where β is the threshold for the temporal odds ratio TOR. The BRS is defined as a tupel $BRS\langle P_i,P_j,y,\ell,\alpha,TOR\rangle$, with the names of the time series P_i and P_j, the trend y, direct D or inverse I, the time lag ℓ, the support percent of the relationship α and the temporal odds ratio TOR.

The rules not explained here, are described in the article (Hira and Deshpande, 2016) in more detail.

3 PROBLEM STATEMENT

Fault detection in IVNs sensor data currently does not consider causal relationships to automatically find defined situations of recorded vehicle data. A situation recognition can reduce the search area if, for example, an error occurs only in a certain driving situation, such as a braking situation. The causal relationships between sensor values can, on the one hand, indicate errors in situations where they change or no longer exist, but on the other hand they can also be used to set up dependency graphs between sensors. This makes fault detection much easier for the expert.

In this positional paper we examine IVN sensor data which are recorded on the CAN bus of a vehicle or simulated for test purposes. A recording of a trip is called a trace. We understand the individual recorded sensors data as a single time series. The entire recording is therefore taken as a set of time series of the form $T = \{T_i,T_j|i,j = 1,...,n\}$. When calculating causal rules, always two time series of set T, T_i and T_j are used for the method with $T_i \neq T_j$.

During fault analysis in IVNs, currently there are not enough causal rules to properly map the relationships between all recorded sensor signals. The cause and effect rules can be used to characterize various situations, to later identify them automatically in IVNs, where faults have occurred. This limits the search space for faults and can be used for visualization and verification using, e.g., a requirement specification. Extended cause and effect rules can also be used as a feature for machine learning procedures in automatic fault detection and fault prediction methods. In particular, changes that occur especially in the case of temporally changing relationships between components can be detected and predicted very effectively using cause and effect relationships, as is the example of mechanical wear in braking situation.

The recorded sensor signals from the IVN can either be noisy and unmodified raw output from sensors or pre-processed and fused signals. The IVNs sensor signals could not have the same sampling rate and therefore, cannot be compared directly at any time, which is not the case in the example of stock market prices (one sample per year). In order to calculate cause and effect relationships from sensor signals, different pre-processing steps must be carried out first, to reduce noise and change their sampling rate to an identical one by interpolation.

The next step is calculating the cause and effect relationships for given situations. The strongest relationships between the sensors describe this particular situation. To determine the correct mode of operation, the rules have to be visualized and proven by expert knowledge.

For the automatic detection of faults in certain situations, especially slightly changing relations over a long period of time, the cause and effect rules have to be extended in order to detect the situations of interest and provide good features for training and subsequent verification of, e.g., a machine learning model.

4 RELATED WORK

Vehicles can be affected by a wide variety of faults. In order to avoid personal injury it is of great interest to detect critical faults as early as possible. However, according to (Rosich et al., 2012) simple plausibility checks are carried out in many error diagnosis systems, like the self-diagnostic and reporting system (OBD) of a vehicle. These systems just check sensor values against predefined threshold values. There are also improved methods of verification like (Korte et al., 2012; Kordes et al., 2014; Deb et al., 2013). Above all, not only individual sensors can be checked, but also the entire sensor network of a machine can be examined. However, in (Rosich et al., 2012) it is pointed out that the complexity of methods should not become so high that they are no longer practically applicable. Thus, it is necessary to use techniques to model all causal relationships between all involved time series and have exact knowledge of it, which requires expert knowledge. In addition, this knowledge must be used to create competing models, from which the best is selected through system identification.

To validate individual sensor values, the principal component analysis is used by (Kerschen et al., 2005) and artificial neural networks are used by (Xu et al., 1999; Klimkowski, 2016). In the mentioned procedures, only individual sensors are checked for faultlessness. This means that no relationships (causality) between the sensors are used to draw conclusions about the entire system state or related sub-sections.

According to the current state of research, there are several methods for including causality. The approach of (Tucker and Liu, 2004) describes how dynamic Bayesian networks, trained from existing sensor data, are used to model the dependencies between sensors as a dependency graph. This method is used to visualize changes in the dependency structure and thus identify faults. In the approach of (Alippi et al., 2014) a dependency graph is created to find relations between sensors, based on the Granger causality concept. The proposed statistical framework is then combined with a hidden Markov model to find and isolate errors in the sensor network.

The proposed method of (Hira and Deshpande, 2016) for creating cause and effect rules is compared to both, the Granger causality and the Bayesian networks. They conclude that their method is faster and can determine not only binary but also transitive, cyclic and many to one relations. Since we want to model all possible causal relationships between sensor signals, we will expand the described method and apply it to IVNs.

5 PROPOSED SOLUTION

In this section we describe our analysis and implementation of an offline situation and fault detection method for IVNs, based on the approach of (Hira and Deshpande, 2016). It is extended by a pre-processing step to handle IVN sensor signals and is corrected in some places. In the current state of research, we only use the calculation of the BRS[2].

On the one hand, the calculation of cause and effect rules is used to create graphs with causal relationships of the sensors in various situations and on the other hand, for the reduction of the search space and further feature extraction for error detection. The extracted features are used for training, test and validation of a decision tree algorithm to automatically detect faults.

In the following sections we describe the adjustments to the original method of (Hira and Deshpande, 2016) and pre-processing steps for the use of IVN sensor signals with it. Afterwards we describe our method of situation learning in order to detect them automatically in recorded traces. Finally, we discuss certain fault detection methods for different faults.

5.1 Adjusting Cause and Effect Rules for IVN Data

Generally, we follow the implementation of (Hira and Deshpande, 2016). Instead of the ROC we use the slope $m_{i,k}$, to calculate the relationships between two time series T_i and T_j. δ is redefined to the minimum slope to consider a significant change and instead of stings we use numerical values[3] to calculate the $R_{i,k}$, redefined as:

$$R_{i,k} = \begin{cases} 1 & \text{if } m_{i,k} \geq \delta \\ -1 & \text{if } m_{i,k} \leq -\delta \\ 0 & \text{if } -\delta \leq m_{i,k} \leq \delta \end{cases} \tag{12}$$

Furthermore, the definition of the temporal inverse odds ratio OR_I defined in 11 has been corrected by us, using C_N as a multiplier in the numerator, because the odds ratio is the same calculation as the cross product ratio defined in (Mosteller, 1968). The new definition of the temporal inverse odds ratio OR_I is:

$$OR_I(P_i,P_j,\ell) = \frac{S_I(P_i,P_j,\ell)\cdot C_N(P_i,P_j,\ell)}{C_E(P_i,P_j,\ell)\cdot C_F(P_i,P_j,\ell)} \tag{13}$$

Finally, we redefined the BRS by deleting the ROC and adding the mean of the slope μ and the standard deviation of the slope σ to $BRS\langle T_i,T_j,y,\ell,\alpha,TOR,\mu,\sigma\rangle$. These values are also used as features for an automatic fault detection.

Further adjustments will be done in the future work, if we add other causal relationship types to the binary relationship, used in the proposed solution.

5.2 Pre-processing IVN Sensor Data

The starting point of our analysis is comparing the features of time series from the stock market prices, which are used as input for the cause and effect rule modeling of (Hira and Deshpande, 2016) and time series from recorded test drives in IVNs in different situations like parking, braking, normal drive behavior

[2]The future work consists among other things of modeling all causal relationships presented in (Hira and Deshpande, 2016).

[3]The use of numerical values is convenient for computation for large, multidimensional arrays.

over short and longer time periods etc. The main differences are: The time series of IVNs do not all have the same sampling rate, they can have different resolutions starting from one bit[4] up to several bytes, they can have positive and negative values, they have a lot more data points and they can be affected by noise. Therefore, we implemented different pre-processing steps.

First, we extract the sensor signals from the IVNs recorded trace file as described in (Kordes et al., 2018). The signals are re-sampled to construct new data points within the range of the discrete set of known data points of every sensor signal. To reduce the noise of some signals, we use a low pass filter. Figure 2 a) shows the original signal and b) shows the smoothed signal[5]. After the pre-processing we

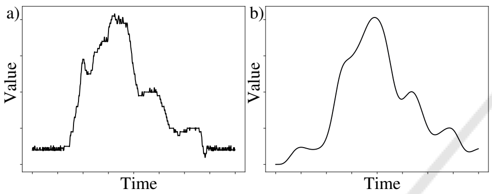

Figure 2: Low pass filter.

get a set of time series $T = \{T_1, T_2, ..., T_n\}$. The pre-processing has to be done just for IVN sensor data.

5.3 Situation Learning

A situation is described by the strongest relationships between all involved sensor values during the period of time, the situation occurs. In the current state of research, we only use the BRSs as describing relationships. In case of IVN sensor data, we assume that the pre-processing step has already been performed.

To automatically learn and detect certain situations, we follow the pipeline of situation learning which is segmented in the four computing steps, time series extraction, time lag calculation, BRS calculation and the optional step of graph visualization. In the first step of the pipeline, it is necessary to create a simulation or record all involved sensors in a situation of interest (or more than one) and extract the time series as set $T = \{T_0, T_1, ...T_n\}$, as depicted in Figure 3.

After the extraction step, we compute causal rules of all subsets of set T, with two differing elements $\{T_i, T_j\}$, to determine the optimal time lag[6] ℓ of the

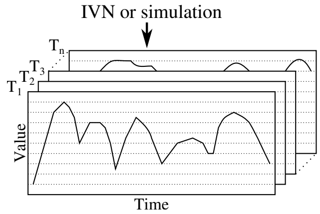

Figure 3: Time series extraction.

subsets. This second step of the pipeline is depicted in Figure 4. We use a predefined $\alpha_1 = -1$ and $\beta = -1$ in order to ensure a BRS is calculated per compared time series.

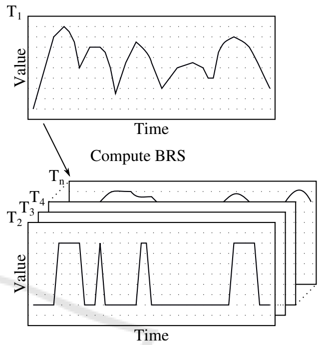

Figure 4: Time lag calculation.

In the third step of the pipeline, we first segment the time series T_i into small sub windows of equal length. The sub windows overlap each other $(sw_0, ..., sw_n)$, like depicted in Figure 5. The time series T_j is also segmented in sub windows starting with an offset of time lag ℓ, as calculated in the second step of the pipeline. We now calculate the strongest BRS with the longest sub sequence for all different sets of T_i and T_j.

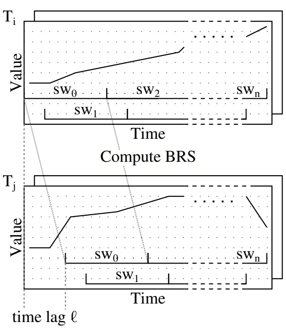

Figure 5: BRS calculation.

Therefore, we first compute a causal rule for $sw_k(T_i)$ and $sw_k(T_j)$. If $k = 0$ and we get a BRS ($\alpha \geq \alpha_1$ and $TOR \geq \beta$), we increment k and go on with the next sub window. Else if $k > 0$ and we get a BRS for $sw_k(T_i)$ and $sw_k(T_j)$, we merge the actual sub

[4]Time series, which are only represented by binary values, have been ignored for simplification in the proposed method.

[5]Noise detection is done manually.

[6]To simplify matters, it is assumed that the requested situation corresponds to the total length of the recorded trace.

window sw_k with the pre sub window sw_{k-1} and calculate a new BRS. We increment k and proceed with the next sub window. The termination criteria for the actual calculation are $\alpha < \alpha_1$, $TOR < \beta$ or the last sub window of T_j is reached. We get the strongest BRS with the longest sub sequence for the sets T_i and T_j. We perform this for all sets, until we have calculated all BRSs[7].

The fourth step of the pipeline is the optional visualization as a relationship graph. An example of a relationship graph is depicted in Figure 6.

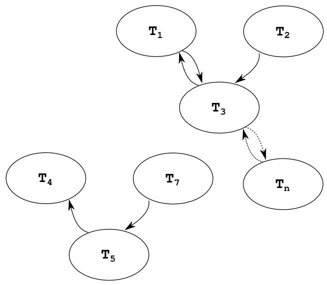

Figure 6: Example of relationship graph.

To detect the learned situation in a recorded driving behavior, we use the calculated causal relationships with the defined time lag ℓ and sub windows like in step three of the pipeline of situation learning. The difference is, that we compare each found BRS with the BRSs of the learning phase. If it matches a BRS of the learning phase, we try to find the other corresponding BRSs with the involved T_i. If we get the whole set of BRSs, we found the exact situation, if it differs in small subset, this subset has to be reviewed for fault detection. Depending on the recording, we start over with the next sub window, to find other situations of the same type.

The proposed solution is still work in progress. In the current state of research we set values manually, like the minimum slope δ, α_1, β and the sub window size used in the pipeline. The next step is to improve the calculation steps of the pipeline, to make the method more robust and more precise. Therefore we can label the BRSs found within the positive situations recorded and use them as feature for a machine learning algorithm (i.e., a support vector machine), to identify the situations in new recordings, with no knowledge about the content.

5.4 Fault Detection

The situation recognition is carried out as preprocessing so that only the situations itself can be analyzed for the fault detection.

Faults that affect the causal relationships can be detected by simply comparing the causal relationship graphs of positive and negative cases. In the case a BRS is missing between two time series, it can be the case if an ECU goes into a fault state and transmits only a fixed value or a short circut has occurred. The causal relationship graph can be used by automotive engineers to identify all affected relationships and sensors, if a fault propagates through the IVN.

To detect an error that manifests itself in a temporal offset, like a mechanical wear, we can use the aforementioned BRSs with the new values of the μ and σ as features for a supervised machine learning algorithm. Therefore it is necessary to record positive and negative examples in order to train the algorithm and find faults or slightly differing values over time, in new recordings. With predictive modeling, we could predict in the future.

There are also algorithms that detect unusual behavior (Shirahama et al., 2016). The proposed method could be used as a pre-processing step in order to reduce the search space for such algorithms in the future work.

6 EXPERIMENTAL RESULTS

In Section 5, we present the proposed situation detection method based on the creation of adjusted cause and effect rules and their visualization. We also discussed detectable faults and the current limits of the proposed method. The effectiveness of the method is reviewed in this section using simulated and real data of a vehicles braking situations.

6.1 Simulated Braking Situations

To evaluate the proposed method, an idealized deceleration is simulated. The required formulas are taken from (Mitschke and Wallentowitz, 2014). It consists of three different time series: velocity v, deceleration a and distance traveled s. The simulation can be initialized with the threshold time and the friction coefficient. The range for positive or negative samples can be taken from Table 1 which is a collection from (Burckhardt, 1991; Gomeringer et al., 2014). The situation detection described in Section 5 is not used because the whole simulation models only the situation to be detected.

In the first step the algorithm for computing BRSs is used to determine the relationships between the three time series. The resulting graph is shown in Figure 7.

[7]The method will be adjusted in order to make it more robust.

Table 1: Parameter of braking situation.

	t_s $[s]$		a_{max} $[m/s^2]$		
case	min	max	min	max	V_A $[m/s]$
pos.	0.14	0.18	$0.76g$	$0.8g$	13.88
neg.	0.19	0.30	$0.75g$	$0.66g$	13.88

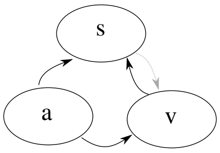

Figure 7: Relationship graph for simulated brake situation.

The BRS calculated between s and v is grayed out, because the distance can not be the cause of the velocity, thus expert knowledge is used to identify and discard it. If we generate a negative brake situation, the BRS sets are the same as in the positive case. Thus we take μ and σ into account to identify this.

In the experiment we generate 200 positive and 200 negative braking situations by randomly assigning equally distributed values from Table 1. For each braking situation the three time series are generated. We cut all of them at the minimum threshold time of $0.14s$, because else it is possible to distinguish between the positive and negative situations by just counting the data points of the time series.

Then for each braking situation the BRSs mentioned above are computed and labeled. For each type of BRS a feature matrix is compiled, consisting of the μ and σ as input for a machine learning model. In this example a decision tree algorithm is used. The results are the same for all BSRs and are cross validated (3-Folds). This is shown in Table 2.

Table 2: Metrics.

class	precision	recall	f1-score	support
neg	1.00	1.00	1.00	200
pos	1.00	1.00	1.00	200
avg/total	1.00	1.00	1.00	400

The results are questioned, because 100% accuracy is reached and could be the cause of overfitting. But if one examines the decision tree, it has only a height of one. This means only the feature μ is used to make the split between the cases. If we compute a probability density function (PDF) of μ in the positive and negative situations, we can separate very clearly between the situations, shown in Figure 8. Therefore, we can validate the result of the decision tree.

We conclude, in the simulated situation we get very good results. This has to be validated with real IVN data.

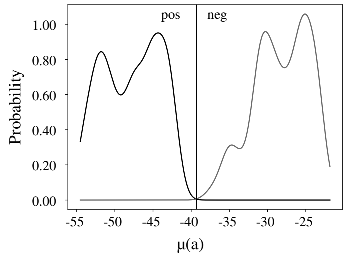

Figure 8: Probability density function of $\mu(a)$.

6.2 Real Braking Situations from IVN Sensor Data

The next step we evaluate the situation detection with real IVN data of a Ford Fiesta MK-II. We recorded CAN bus data for a test drive which contains several braking situations to automatically detect only the situations braking from $20km/h$ down to $15km/h$. In the recorded trace there are only positive examples of a brake without mechanical wear.

With expert knowledge we modeled a sub graph of involved time series during a brake situation. It contains the verified relationships of the deceleration sensor a, the velocity v and the brake pedal position b: $BRS_{b\to a}$, $BRS_{b\to v}$ and $BRS_{a\to v}$. An unfiltered excerpt of the time series is shown in Figure 9.

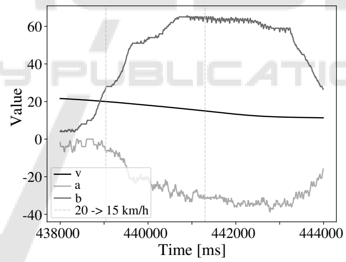

Figure 9: IVN brake situation (Ford Fiesta MK-II).

The pre-processing steps described in section 5 are applied on the time series to eliminate noise and re-sample them and the optimal time lag is determined by the proposed method. Next we applied the verified relationships to the method in order to find braking situations in a trace of several minutes. Through empirical tests a sub window size of $500ms$ equivalent to 50 data points and a overlap of 25 data points is used. The α-value is set to 90%. The result is shown in Figure 10, where the brake situations are marked between dashed lines. The proposed method found every brake situation in the trace[8].

We also modeled good and bad brake examples using the IVN sensor data, depicted in the left-hand

[8]Is verified by manual situation detection.

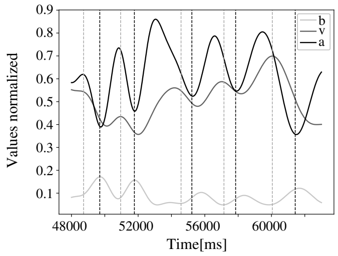

Figure 10: Found IVN brake situations.

side of Figure 11[9] and got the result the PDF of $\mu(v)$ on the right-hand side of Figure 11. The decision tree has, like in the simulation, a depth of one. So the PDFs were examined and result that even in the real IVN sensor data, there are features with optimal selectivity, like the $\mu(v)$.

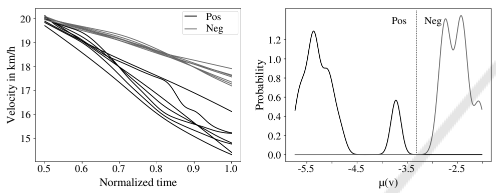

Figure 11: Modeled brake examples and resulting PDF.

7 CONCLUSIONS

The experimental results show that the adjusted method of (Hira and Deshpande, 2016) performs very well with a simulated situation and situations derived from real traces of IVN sensor data, shown in Section 6. It is able to learn defined situations and find them in unknown time series.

The automatic fault detection, in the case of mechanical wear, is verified with a positive result in a simulated situation as well as in real traces of IVN sensor data. Therefore we used positive examples from test rides and modeled negative examples.

According to the results, the method of modeling causal relationships between time series can be applied to sensor signals of IVNs very well. It has to be adjusted in the future work in order to make it more robust and decrease the cost in means of computation time, by applying new techniques like machine learning algorithms in intermediate steps of the method. All previously manually defined variables have to be decided automatically.

[9] Samples are divided into small sub windows in order to get more training data.

REFERENCES

Alippi, C., Roveri, M., and Trov, F. (2014). Learning causal dependencies to detect and diagnose faults in sensor networks. In *2014 IEEE Symposium on Intelligent Embedded Systems (IES)*, pages 34–41.

Burckhardt, M. (1991). *Fahrwerkstechnik: Bremsdynamik und Pkw-Bremsanlagen*. Vogel-Verlag.

Deb, A. K., Kordes, A., and Vermeulen, H. G. H. (2013). Flexray network runtime error detection and containment. US Patent 20140047282A1.

Gomeringer, R., Heinzler, M., Kilgus, R., Menges, V., Näher, F., Oesterle, S., Scholer, C., Stephan, A., and Wieneke, F. (2014). *Tabellenbuch Metall*. Europa-Fachbuchreihe für Metallberufe. Europa-Lehrmittel.

Hira, S. and Deshpande, P. S. (2016). Mining precise cause and effect rules in large time series data of socio-economic indicators. *SpringerPlus*, 5(1):1625.

Kerschen, G., Boe, P. D., Golinval, J.-C., and Worden, K. (2005). Sensor validation using principal component analysis. *Smart Materials and Structures*, 14(1):36.

Klimkowski, K. (2016). An artificial neural networks approach to stator current sensor faults detection for dtc-svm structure. *Power Electronics and Drives*, 1 (36)(1).

Kordes, A., Hozhabrpour, H., Wismüller, R., and Grzegorzek, M. (2018). Vehicle-independent interpretation of sensor signals without a-priori knowledge of their semantics. *In press, VDE GMM-Fachtagung AmE Automotive meets Electronics, Dortmund, Germany*.

Kordes, A., Vermeulen, B., Deb, A., and Wahl, M. (2014). Startup error detection and containment to improve the robustness of hybrid flexray networks. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 5. European Design and Automation Association.

Korte, M., Holzmann, F., Kaiser, G., Scheuch, V., and Roth, H. (2012). *Design of a Robust Plausibility Check for an Adaptive Vehicle Observer in an Electric Vehicle*, pages 109–119. Springer Berlin Heidelberg, Berlin, Heidelberg.

Mitschke, M. and Wallentowitz, H. (2014). *Dynamik der Kraftfahrzeuge*. Springer Vieweg.

Mosteller, F. (1968). Association and estimation in contingency tables. *Journal of the American Statistical Association*, 63(321):1–28.

Rosich, A., Frisk, E., Aslund, J., Sarrate, R., and Nejjari, F. (2012). Fault diagnosis based on causal computations. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 42(2):371–381.

Shirahama, K., Köping, L., and Grzegorzek, M. (2016). Codebook approach for sensor-based human activity recognition. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM.

Tucker, A. and Liu, X. (2004). A bayesian network approach to explaining time series with changing structure. *Intell. Data Anal.*, 8(5):469–480.

Xu, X., Hines, J. W., and Uhrig, R. E. (1999). Sensor validation and fault detection using neural networks.