

Predicting Air Compressor Failures with Echo State Networks

Yuantao Fan¹, Sławomir Nowaczyk², Thorsteinn Rögnvaldsson³ and Eric Aislan Antonelo⁴

^{1,2,3} *Center for Applied Intelligent Systems Research (CAISR), Halmstad University, 30118 Halmstad, Sweden*

yuantao.fan@hh.se

slawomir.nowaczyk@hh.se

thorsteinn.rognvaldsson@hh.se

⁴ *Department of Automation and Systems, Federal University of Santa Catarina, 88040-900 Florianópolis, Brazil*

eric.antonelo@gmail.com

ABSTRACT

Modern vehicles have increasing amounts of data streaming continuously on-board their controller area networks. These data are primarily used for controlling the vehicle and for feedback to the driver, but they can also be exploited to detect faults and predict failures. The traditional diagnostics paradigm, which relies heavily on human expert knowledge, scales poorly with the increasing amounts of data generated by highly digitised systems. The next generation of equipment monitoring and maintenance prediction solutions will therefore require a different approach, where systems can build up knowledge (semi-)autonomously and learn over the lifetime of the equipment.

A key feature in such systems is the ability to capture and encode characteristics of signals, or groups of signals, on-board vehicles using different models. Methods that do this robustly and reliably can be used to describe and compare the operation of the vehicle to previous time periods or to other similar vehicles. In this paper two models for doing this, for a single signal, are presented and compared on a case of on-road failures caused by air compressor faults in city buses. One approach is based on histograms and the other is based on echo state networks. It is shown that both methods are sensitive to the expected changes in the signal's characteristics and work well on simulated data. However, the histogram model, despite being simpler, handles the deviations in real data better than the echo state network.

1. INTRODUCTION

A fleet of commercial heavy-duty vehicles is a very interesting application arena for prognostics and health management. Limited computational and communication capabilities

combined with large influence of environment and usage make predictive maintenance challenging. The availability of many (almost but never quite) identical vehicles allows for approaches based on the “wisdom of the crowd” idea. For example, other vehicles can be queried regarding what constitutes a normal behaviour or of possible origins of deviations. This information can be utilised to prevent unplanned stops, which are very problematic and costly for commercial heavy-duty vehicles, as they can mean failed deliveries, towing and additional waiting time at workshops.

The state-of-the-art approach for designing on-board diagnostic functions heavily relies on domain experts, who usually need to define possible faults, component-specific models for nominal operation, and the relevant external conditions that need to be taken into account. This paradigm has been very successful and is applied on many components, especially the critical ones that have large impact on safety or continuous operation (e.g. engine, braking system, gearbox, etc.). On the other hand, today's vehicles are equipped with large number of sensors and computing units. With increasing number of signals streaming over the controller area network, it is possible to monitor more and more components and subsystems. Some of them do not warrant the equivalent engineering effort in designing diagnostic functions, however, a generic method that is capable of discovering new knowledge and can fully utilise the data collected from electronic sensor system on-board, for different types of signals, can provide great benefits for the automotive industry.

Such an approach towards improving vehicle uptime was proposed by (Bytner, Rögnvaldsson, & Svensson, 2011). It allows for monitoring of on-board sensor streams using dedicated hardware, analysing them and transmitting compressed representations to a computing center. Those representations are then compared across the fleet to find deviations. The method, called Consensus Self-Organising Models (COSMO),

Yuantao Fan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

is based on the idea of “wisdom of the crowd” and deviation detection. It assumes that the majority of the vehicles are “healthy” and an individual that deviates from the group should be labelled as potentially “faulty”. Deviations are matched against the vehicle service records, which contain the operations performed in the workshop, so that a knowledge base can be built up with associations between deviations (and their disappearances) and repairs.

In this paper we study the air compressor system for city buses. The air compressor primarily provides air to the brakes, doors, gearbox and the suspension. If the air compressor fails, the bus cannot drive. Still, developing on-board diagnostics for compressed air system is not a high priority, since it is a component that, overall, fails very seldom, with failure rate of less than 5% across all types of vehicles. However, in our study cohort of 19 buses, there were nine air compressor failures that caused stops on the road, in many cases including towing, and all the buses in this fleet had its compressor replaced in workshop. Apparently, the lifetime of the compressor varies greatly between individual vehicles, based on usage patterns and external conditions, which makes predictive maintenance highly relevant.

There are several tests that can be performed in the workshop to test the health of the air compressor. There are also patents (Fogelstrom, 2006 (filed 2004), 2007 (filed 2005)) that suggest methods for detecting compressed air problems on-board heavy duty vehicles. Both the workshop tests and the on-board patents are based on the characteristics of the *Wet Tank Air Pressure* signal. This pressure increases when the compressor pushes air into the tank and decreases when the tank releases the air to drive other components.

The COSMO approach builds on capturing the signal’s characteristics with a model; a model that can then be compared to models built at earlier times or from other vehicles. This method requires that the model is rich enough to capture the essential characteristics of the signal (e.g. its dynamics) and that it is possible to define a distance metric between the models. In previous studies we have used histograms and shown that it is possible to successfully detect half the compressor failures that occurred on the road, with sufficient lead time to schedule workshop visits.

Histograms are memory efficient, robust against noise, natural to parametrise directly based on design-time knowledge, and easy to compute on-board. However, they are simply a density estimator, and do not capture dynamics of the signal in any way. One way around that limitation is to use histograms of signal changes, from one time step to another, as introduced in (Fan, Nowaczyk, & Rögnvaldsson, 2015). In this paper we explore the use of recurrent neural networks (RNNs) for modelling the signal, using a particular type that exhibit fast training without local optima. Recurrent neural networks of this type are called Echo State Networks (ESNs)

(Jaeger & Haas, 2004).

The main contribution of this work is the comparison of using histograms and ESNs in detecting deviations. Their respective performance is evaluated on synthetic data, to get an understanding regarding the specific strong and weak points of the models, as well as on a real data collected over a long-term study of a bus fleet in real operation. In the latter case the exact condition of the equipment is unknown, and the only reference data available are the decisions of whether to replace or repair a particular piece of equipment made by workshop personnel. This information is only partially trustworthy, and it also heavily depends on the maintenance strategy employed by the workshop and the fleet operator.

2. RELATED WORK

The need for autonomous and generic methods for equipment prognostics and health monitoring (PHM) was evident several years ago. It was, for example, expressed already in 1998 at a NIST (National Institute of Standards) workshop on condition-based maintenance (Bartholomew, 1998). However, in spite of this, there have been surprisingly few efforts directed towards autonomous methods for PHM.

The COSMO method is based on searching for good representations, comparing these representations across systems (or time), and doing novelty detection. However, in this paper we only study the effect different representations have on deviation detection performance. There is a huge amount of literature on novelty detection, and an emerging body of literature on learning representations, typically in the context of deep neural networks. We do not review these here but focus on work related to the automotive domain, and on using ESNs as representations.

In their early work, (Filev & Tseng, 2006; Filev, Chinnam, Tseng, & Baruah, 2010) have presented an approach for deviation detection for PHM, where Gaussian mixture model fuzzy clusters are used as generic models to represent the signals. Their work is close to Angelov’s, who has made many important contributions to autonomous knowledge creation (Angelov, 2013). The work by (Alippi, Roveri, & Trovò, 2012, 2014) have used linear models to capture the relationships between signals, and lagged values of signals. Vachkov (2006) used self-organised neural gas models to capture non-linear relationships between signals for diagnostic purposes. None of these, however, have used a distance metric between models, which allows the deviation detection to be done in model space.

In their recent work, (Chen, Tiño, Rodan, & Yao, 2014) uses the model space (based on building models of the data, defining a metric to compare those models, and detecting deviations based on the differences between model parameters), with restricted ESNs as models, to do deviation detection

with single class support vector machines. They refer to this as cognitive fault diagnosis and apply it to simulator models of the Barcelona water distribution network (Quevedo et al., 2014) and of the Tennessee Eastman Kodak process (Chen, Tiño, & Yao, 2014). Their conclusion was that the model space approach was superior to other approaches on these problems.

The COSMO approach, which also uses the model space for the deviation detection, was suggested by (Byttner, Rögnvaldsson, & Svensson, 2007), using linear models of relationships between signals. It has since been further developed in a number of studies: on a heavy-duty truck with injected faults (Byttner, Rögnvaldsson, Svensson, Bitar, & Chominsky, 2009), on a city bus driven on a test track (Byttner et al., 2011), and on a fleet of city buses in regular operation (Byttner, Nowaczyk, Prytz, & Rögnvaldsson, 2013).

3. DATA

3.1. The vehicle fleet

The on-board empirical data used in this paper were collected over four years, from June 2011 to September 2015, on a commercial bus fleet with 19 buses. The buses operate in a municipality on the west coast of Sweden. The buses run on average 100,000 km per year and vehicle. Approximately 100 signals, e.g., different temperatures, pressures, wheel speeds, engine speed, GPS, etc., are logged at 1 Hz frequency by an on-board embedded device.

In addition to the on-board data, different kind of off-board data is also available, such as Vehicle Service Records (VSR) or vehicle configuration. The VSR contains repair information, in particular, the date and the mileage, operations performed and workshop personnel's comments. The off-board data are available for the full operational time for the buses, which goes further back than 2011. In the fleet are four buses manufactured in 2009, one in 2008, and the remaining fourteen in 2007. The buses are all of the same model.

In this work we focus on the air compressor, for reasons already mentioned in the introduction. There was only one air compressor failure in total during the buses' first five years of operation, and this was caused by a human mistake during a different repair. However, compressors started to fail frequently in the following four years and have caused several occurrences of bus breakdown on the road. By now all vehicles in this fleet have its compressor replaced.

The air compressor is a water cooled Wabco twin cylinder compressor, 636 cm³, which can produce a flow rate of 3000 litres per minute. The only air pressure signal we can measure on-board during operation of the vehicle is the *Wet Tank Air Pressure*. This is the air pressure in the wet tank, which is the first tank downstream of the compressor. The characteristics of this signal are described in the next section.

In this study we focus on compressor replacement operations. Replacing the compressor occurs in two situations: first, when the compressor fails completely, so that the vehicle is inoperable and towed to the workshop for repair; second, when the workshop personnel determines that the compressor is not functioning satisfactorily and decides to replace it. The latter is based on diagnostic tests done in the workshop, e.g., the time needed to reach to maximum pressure, as well as on subjective assessment of the technicians. On the other hand, the former case is an example of "run to failure."

One particular challenge we face is the lack of ground truth. The VSR comments do not specify the results of the compressor tests, only that they were done. Furthermore, there is no information about the actual condition of a replaced compressor, neither from the workshop nor from the manufacturer. It is therefore impossible to tell exactly why the compressor broke (when it broke down on the road), or how worn the compressor was (when the maintenance personnel decided to replace it). There is a significant difference between a verified failure and a workshop decision, which is why we introduce two fault categories of repairs: *Compressor Failure* and *Compressor Replacement*. In this paper we only focus on the first case, i.e. compressor on-road failures, when the vehicle was towed to the workshop.

Another challenge is that there are several other faults that can affect the wet tank air pressure. These are not treated in this study but they include air leaks, congested air pipes, and malfunctioning regulators. This makes the analysis more complicated, since it is often impossible to know if there was only one fault, or what the root cause for the fault was. For example, an untreated congested air pipe can strain the compressor and lead to compressor wear out. Similarly, a leak in the air system can affect the pressure signal in a similar way a worn compressor would, leading to false fault symptoms.

The lack of ground truth about the compressor's health status and the complexity of possibly multiple faults in the empirical data are challenges but they reflect the reality if one desires to build up knowledge from normal operation data. It is therefore necessary to see how well an approach does under such circumstances.

3.2. Wet Tank Air Pressure

Figure 1 shows the *Wet Tank Air Pressure* (WTAP) signal of one vehicle during normal operation. The pressure signal consists of a charging period (marked with red points) and a discharging period (marked with blue points). The pressure is expected to lie between cut-in and cut-out limits.

3.3. Synthetic Data

Conceptually, the WTAP consists of a charging period and a discharging period. Synthetic data set was constructed and

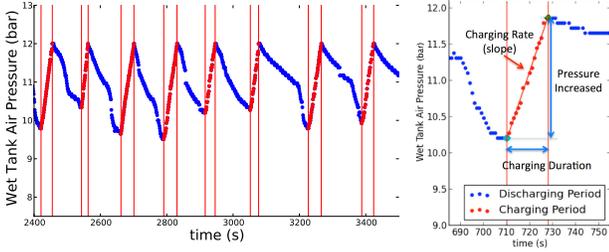


Figure 1. The WTAP signal. Red points correspond to charging periods and blue points correspond to discharging periods. The right panel shows features that can be extracted from a charging cycle.

designed to simulate a WTAP signal based on two main characteristics, namely slopes and spread. Each period has its own features, i.e., slope k as well as maximum and minimum pressure values V_{max} and V_{min} . The synthetic data were created by drawing values for these from the following distributions:

$$k_c \sim N(\mu_{up}, \sigma_k) \quad k_d \sim N(\mu_{down}, \sigma_k) \quad (1)$$

$$V_{min} \sim N(\mu_{min}, \sigma_V) \quad V_{max} \sim N(\mu_{max}, \sigma_V) \quad (2)$$

where $N(\mu, \sigma)$ denotes a normal distribution with mean μ and standard deviation σ . Data samples for a single charging (or discharging) period were generated by uniformly interpolating points between V_{max} and V_{min} based on change rate k . Several charging and discharging periods were then concatenated into a synthetic time series.

Different synthetic *Wet Tank Air Pressure* signal datasets were generated based on different sets of means and standard deviations, representing *healthy* and *faulty* cases. For example, the faulty cases would have a lower average charging rate than the healthy cases, or lower maximum, or higher minimum. An example is shown in Fig. 2, where the faulty case has a lower charging slope than the healthy case.

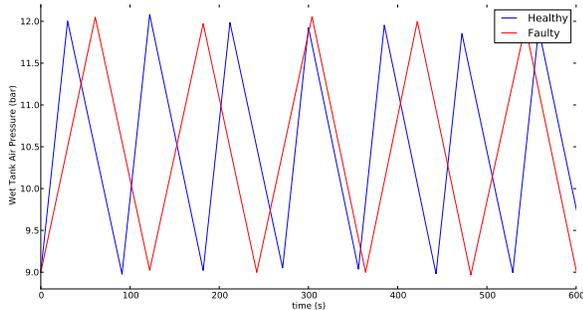


Figure 2. Synthetic data sample with lower charging rate for the faulty case than for the healthy case.

4. METHOD

4.1. COSMO

The Consensus Self-Organised Models (COSMO) method is based on three steps: capture the signal characteristics with the selected model; compute the distances between models;

and find deviations with a “wisdom of the crowds” approach.

In this paper we employ two quite different types of models, histograms and ESNs, to represent the signal. In a series of experiments we compare how well can they capture the differences in signal characteristics between healthy and faulty compressors, i.e., how well do they perform in predictive maintenance setting.

For the empirical case, models are based on daily data. The data are sampled at 1 Hz and at least 3600 samples (one hour) are required to build a model, or else the day is scrapped. The typical case is that about ten hours of data are available each day for the model construction. The daily models for the fleet are then compared over a week, i.e. in the best case there are 133 (19×7) models that can be compared to each other. The pairwise distances between the models are computed, resulting in a symmetric distance matrix \mathbf{D} .

The deviation detection is done with a method described by (Rögnavaldsson, Norrman, Byttner, & Järpe, 2015). First the row in \mathbf{D} with minimum row sum is chosen as *the most central model* (denoted by c). The z-score for a model m is then the number of models that are further away from the most central model c than m is:

$$z(m) = \frac{|\{i = 1, \dots, N : d_{i,c} > d_{m,c}\}|}{N} \quad (3)$$

where $|\cdot|$ denotes cardinality of the set. The null hypothesis is that all samples are drawn from the same distribution, in which case the z-scores should be uniformly distributed between zero and one. This hypothesis is tested by comparing the average z-score over a certain period (we use 30 days) with the value expected from a uniform distribution. We compute the p-value for the mean using a one-sided test, since we are interested in samples that lie at the edge of the distribution, i.e. when the z-score is small. It is important to keep track of models so that models relating to one vehicle are kept separate from the fleet when that vehicle is tested.

The negative logarithm of the one-sided p-value is used as the deviation level:

$$\text{deviation level}(\bar{z}) = -\log_{10} \left[\Phi \left(\frac{\bar{z} - 0.5}{\sigma_n} \right) \right], \quad (4)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function, \bar{z} is the average of the z-scores, $\sigma_n = (12n)^{-1/2}$, and n are the number of valid days during the 30 days (i.e. the number of samples that were used when computing \bar{z}).

4.2. Evaluation with reference data

Ideally, deviations should be detected early enough so that they are actionable, i.e., it is possible to schedule workshop visit to fix the problem before it causes damage to the system. For simplicity, we assume that this period of interest is constant, and refer to it as the *prediction horizon*. In previous

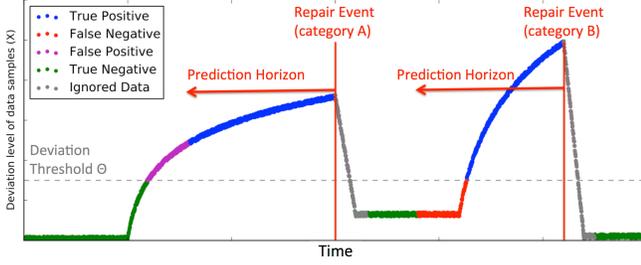


Figure 3. Labelling of deviations in relation to repair actions.

work (Fan et al., 2015) we have described several types of faults related to air systems, grouped them into different categories, and described a way to evaluate the outcome when there are many conflicting faults. The setup is illustrated in Fig. 3. The expected *healthy* observations are “shared” between all faults, since they correspond to times when a vehicle is believed to be operating without any problem. *Faulty* observations, on the other hand, depend on the particular repair action that was performed.

Based on the negative logarithm of the p-value, described in Section 4.1, we can calculate elements in the confusion matrix (true positives, false positives, true negatives and false negatives) for any decision threshold θ . By varying θ we can map out the receiver operating characteristic (ROC) curve, which is the relationship between the true positive rate and the false positive rate. The area under the ROC curve (AUC) is used as the primary quality measure in the final evaluation. The confidence interval of the AUC value for the bus fleet data is computed using leave-one-out (i.e. the individual buses are left out one by one).

4.3. Models

4.3.1. Histograms

A histogram is an estimate of the probability distribution of the signal. Creating a histogram is straightforward, it is robust against outliers, easy to compute and easy to implement in on-board hardware. The parameters required to construct a histogram are the range of variable and number of bins that divide the whole range into a series of intervals. In practice, both of them can be set based on the property of sensor and signal as specified when producing the vehicle.

4.3.2. Distance between histograms

We use normalized histograms, i.e. histograms where the bin values sum to one. The distance between histograms is measured with the Hellinger distance:

$$h(\mathbf{r}, \mathbf{s}) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^M (\sqrt{r_i} - \sqrt{s_i})^2} \quad (5)$$

where \mathbf{r} and \mathbf{s} are normalized histograms with M bins. We have tested many different distances between histograms and concluded that the Hellinger distance works very well.

We use both histograms of the signal $r(t)$ and of the signal changes $\Delta r(t) = r(t) - r(t-1)$, in order to capture some of the signal dynamics. The information from these can be fused in different ways. One way is to create two-dimensional histograms.

Another way is to treat the signal and signal change as independent and compute a total distance with e.g. a p-norm:

$$h_{tot}(\mathbf{r}, \mathbf{s}) = [h(\mathbf{r}, \mathbf{s})^p + h(\Delta \mathbf{r}, \Delta \mathbf{s})^p]^{1/p} \quad (6)$$

Here \mathbf{r} (and \mathbf{s}) refers to the normalized histogram of the signal, and $\Delta \mathbf{r}$ (and $\Delta \mathbf{s}$) refers to the normalised histogram of the signal changes. Using a p-norm requires that the two distances $h(\mathbf{r}, \mathbf{s})$ and $h(\Delta \mathbf{r}, \Delta \mathbf{s})$ are of similar scale, or else the larger of them will dominate the combined distance.

A third alternative to combine the two is to calculate the deviation level (see equation 4) for the signal (d_s) and signal change (d_Δ) separately, and then combine those deviation levels instead (again, using e.g. p-norm):

$$\|d\|_p = [d_s^p + d_\Delta^p]^{1/p} \quad (7)$$

In particular, by setting $p \rightarrow \infty$, we obtain:

$$\|d\|_\infty = \max(d_s, d_\Delta), \quad (8)$$

which corresponds to a “worst case” approach, i.e., to always use the more severe deviation level. This makes sense for example if the two models react to different types of faults.

4.3.3. Echo state networks

We use Echo State Networks (ESNs) for the reservoir computing models. The general structure of a reservoir computing model is shown in Figure 4. The internal reservoir units, $\mathbf{x}(t)$, are leaky integrator neurons with a discrete state update equation

$$\mathbf{x}(t+1) = (1 - \alpha)\mathbf{x}(t) + \alpha f[\mathbf{W}^{xx}\mathbf{x}(t) + \mathbf{W}^{ux}\mathbf{u}(t)] \quad (9)$$

where $\mathbf{u}(t)$ denotes the input vector at time t ; α is the leaking rate; $f[\cdot] = \tanh[\cdot]$ is the hyperbolic tangent function; \mathbf{W}^{xx} is the connection matrix of the recurrent reservoir; and \mathbf{W}^{ux} is the weight matrix connecting the input to the reservoir.

The input’s influence into the reservoir dynamics is controlled by a parameter v that rescales the input connection matrix to $v\mathbf{W}^{ux}$. This scaling impacts the reservoir dynamics and requires some tuning. Similarly, the reservoir weights are rescaled such that the resulting system is stable but still exhibits rich dynamics. This is achieved by making its spectral radius $\rho(\mathbf{W}^{xx}) < 1$. Usually, this rescaling influences the

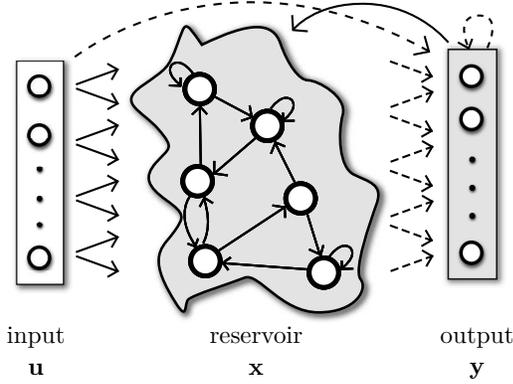


Figure 4. Reservoir Computing (RC) network. The reservoir is a non-linear dynamical system usually composed of recurrent sigmoid units. Solid lines represent fixed, randomly generated connections, while dashed lines represent trainable or adaptive weights.

performance and, as such, is tuned to give the best results on a validation set. Neither v nor $\rho(\mathbf{W}^{\mathbf{x}\mathbf{x}})$ need fine tuning.

The leaking rate $\alpha \in [0, 1]$ changes the dynamics of the reservoir, eventually matching better the dynamics of the input signal or of the task. Low leaking rates yield reservoirs with longer memory.

The output is computed as:

$$\mathbf{y}(t+1) = g[\mathbf{W}^{\mathbf{x}\mathbf{y}} \mathbf{x}(t+1)], \quad (10)$$

where $g[\cdot]$ denotes the identity function (we introduce it here for convenience later) and $\mathbf{W}^{\mathbf{x}\mathbf{y}}$ is the only trainable weight matrix. This weight matrix can also have direct input to output connections, or feedback connections between the output units, as indicated in Fig. 4, but we did not use this. The weight matrix is optimised using ridge regression:

$$\mathbf{W}^{\mathbf{x}\mathbf{y}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \hat{\mathbf{Y}}. \quad (11)$$

The matrix \mathbf{X} is of size $T \times (R+1)$; it is a row-wise concatenation of reservoir states $\mathbf{x}(1), \dots, \mathbf{x}(T)$, where there are R reservoir units (the extra 1 is for the bias). $\hat{\mathbf{Y}}$ is composed of the row-wise collection of the corresponding desired outputs $\hat{\mathbf{y}}(t)$. T is the total number of training samples and the initial state is $\mathbf{x}(0) = \mathbf{0}$. The regularisation parameter λ is set by cross-validation (it is not tuned for each model).

After generation of \mathbf{X} , a process called *warm-up drop* is used to get rid of undesired initial transients in the reservoir. This is done by dropping some of the first samples of \mathbf{X} and using the resulting matrix in equation (11). Training ESNs is explained in detail by Jaeger (2010) and we therefore describe it only briefly here.

In this work, the ESN is driven by a one-dimensional signal $u(t)$, where the desired output $y(t)$ is $u(t + \Delta)$. Thus, we train an ESN to perform a Δ -steps ahead prediction. For

each given period containing sensor data, such as a day, and for each bus, an ESN is trained with the signal $u(t)$ available in the respective period. The overall process is thus unsupervised since only sensor data $u(t)$ are used during training.

Chen et al. (2014) use a constrained version of ESN, introduced by Rodan and Tiño (2012). This ESN model is a cycle reservoir with jumps (CRJ) where all cycle connections have the same value and the jump connections have the same value, i.e. there are only two weight values between units in the reservoir. This CRJ model is a lot less complex than the full ESN and has been shown to outperform the latter on several time series tasks (Rodan & Tiño, 2012).

We tested the CRJ model too on our task but with limited exploration of hyperparameters (i.e. λ , jump sizes, weight scales, etc.).

4.3.4. Distances between ESNs

Inspired by the work of Chen et al. (2014), we use the L_2 norm to measure the distance between two trained ESNs (or CRJs). Two versions are tested: one based on Monte Carlo sampling in the state space and one based on an analytical approximation. Denoting the two ESNs by g_1 and g_2 , the L_2 distance is defined as

$$L_2(g_1, g_2) = \left[\int \|g_1(\mathbf{x}) - g_2(\mathbf{x})\|^2 \Omega(\mathbf{x}) d\mathbf{x} \right]^{1/2} \quad (12)$$

where the density $\Omega(\mathbf{x})$ is over the states \mathbf{x} . The Monte Carlo based distance is computed by sampling $\|g_1(\mathbf{x}) - g_2(\mathbf{x})\|^2$ over the state space. The analytical distance is computed by assuming a uniform $\Omega(\mathbf{x})$ over the state space, which yields (Chen, Tiño, Rodan, & Yao, 2014)

$$L_2(g_1, g_2) \propto \left[\sum_j w_{0,j}^2 + \frac{1}{3} \sum_{i,j} w_{i,j}^2 \right]^{1/2}. \quad (13)$$

Here $w_{0,j}$ denotes the difference between bias weights in the matrix $\mathbf{W}^{\mathbf{x}\mathbf{y}}$ for the two models g_1 and g_2 , and $w_{i,j}$ denotes the difference in weights from reservoir units in the matrix $\mathbf{W}^{\mathbf{x}\mathbf{y}}$ for the two models.

The Monte Carlo sampling is done as follows. The reservoir states $\mathbf{x}(t)$ using two different input sequences result in different sequences of states in the reservoir space. This means that g_1 and g_2 are trained using different \mathbf{X} matrices. We denote them \mathbf{X}_1 and \mathbf{X}_2 , respectively. Random states are sampled from them, half from \mathbf{X}_1 and half from \mathbf{X}_2 , and then used in the computation of $L_2(g_1, g_2)$. The order of the states is ignored.

5. RESULTS

5.1. Training the ESNs on WTAP

For the experiment with both datasets, the following setup was used for constructing the ESN models. The modeling task for the reservoir was five steps ahead prediction ($n_{ahead} = 5$). The number of reservoir units was set to a small value ($n_r = 25$) so that it should not take long to train each model. Using the Monte Carlo distance between ESNs required generating and storing the matrix of reservoir states besides training the models, whereas the analytical distance only required the ESN model weights.

After some experimentation (grid search and empirical tuning), the following values were selected: input scaling $v = 0.16$, spectral radius $\rho(\mathbf{W}^{xx}) = 0.1$, and leak rate $\alpha = 0.5$. The regularization parameter was set to $\lambda = 0.0001$. The non-trainable reservoir weight matrix \mathbf{W}^{xx} values were set to random gaussian numbers with zero mean and unit variance, except for bias weights, which were drawn from the uniform $[-1, 1]$ distribution. The non-trainable input to reservoir weight matrix \mathbf{W}^{ux} values were set to random numbers from the set $\{-1, 1\}$.

The CRJ models had a reservoir with 25 units (i.e. the same as for the ESN models). The jump size was $l = 5$, with one-way cycle connection strength (weight) $r_c = 0.92$ and bidirectional jump connection strength (weight) $r_j = 0.17$.

An example output sequence from a trained ESN with five step prediction is shown in Fig. 5. All ESN models were trained to produce similarly accurate time series with respect to the real WTAP. The CRJ models were also trained to do the same but did not achieve quite the same precision.

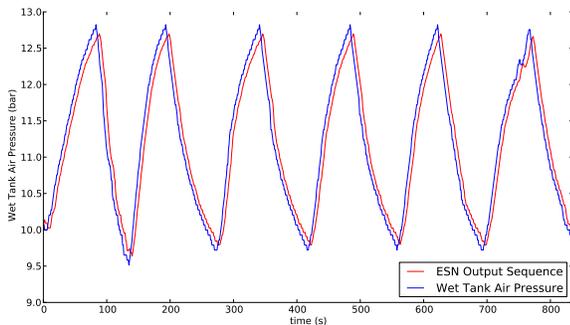


Figure 5. An example output sequence from a trained ESN with five step ahead prediction

5.2. Synthetic Data Generation

The experiments with synthetic data were designed to showcase the differences between the histogram and ESN models. Conceptually, we introduced two different distributions representing the two groups of vehicles: the healthy group and the faulty group. With the synthetic data we are able to control the type of the introduced “fault,” as well as its seriousness,

which allows us to visualise the differences in behaviour of the two models.

The signal we generated was a simplification of the actual wet tank air pressure signal: a sawtooth (Fig. 2) with alternating upwards and downwards slopes, as described in Section 3.3. The two “faults” that were simulated were a lower charging slope (corresponding to a weak air compressor) and a shift in the minimum and maximum values (corresponding to a broken regulator).

For the first fault dataset (weak compressor) was the upwards (charging) slope, see equation (1), higher for the healthy group than for the faulty group. The mean charging slope for the healthy group was kept fixed at $\mu_{up} = 0.1$, whereas the mean charging slope for the faulty group was varied $\mu_{up} \in [0.95, 1] \times 0.1$. The downwards slope, $\mu_{down} = 0.1$ was the same for both groups. The standard deviations, for both upwards and downwards slopes, was kept fixed at $\sigma_k = 0.001$.

For the second fault dataset (broken regulator) was the minimum and the maximum values, see equation (2), shifted downwards for the faulty group. The mean maximum value for the healthy group was $\mu_{max} = 12$, and the mean minimum value was $\mu_{min} = 9$. The corresponding values for the faulty group were $\mu_{max} \in [0.98, 1] \times 12$ and $\mu_{min} \in [0.98, 1] \times 9$. Both the minimum and the maximum were varied at the same time so that the mean spread (distance between maximum and minimum) remained fixed.

5.3. Synthetic Data Results

Figure 6 shows how distances between models change as the difference between the healthy and the faulty data is increased. The upper two subplots show the distances between different histogram models as the charging slope (left) or the min-max values (right) change. As expected, the signal histogram is insensitive to changes in the slope (weak compressor fault), while both the signal change histogram and the combined 2D histogram are very sensitive to this. In contrast, the signal change histogram is insensitive to changes in the min-max values (broken regulator fault), while both the signal histogram and the 2D histogram are sensitive to them.

The notation “All” and “CP” denote, respectively, whether all the data or just the data from the charging period were used to populate the histograms — the latter was motivated by existing diagnostic knowledge, which suggest that the charging periods are more informative with regards to condition of air compressor. As expected, the models based only on the upwards slope of the signal generally outperform the models based on all the data.

The two lower subplots in Fig. 6 show the distances between different ESN models as the charging slope (left) or the min-max values (right) change. It is clear that both the ESN and the CRJ models, as well as both the Monte Carlo and the

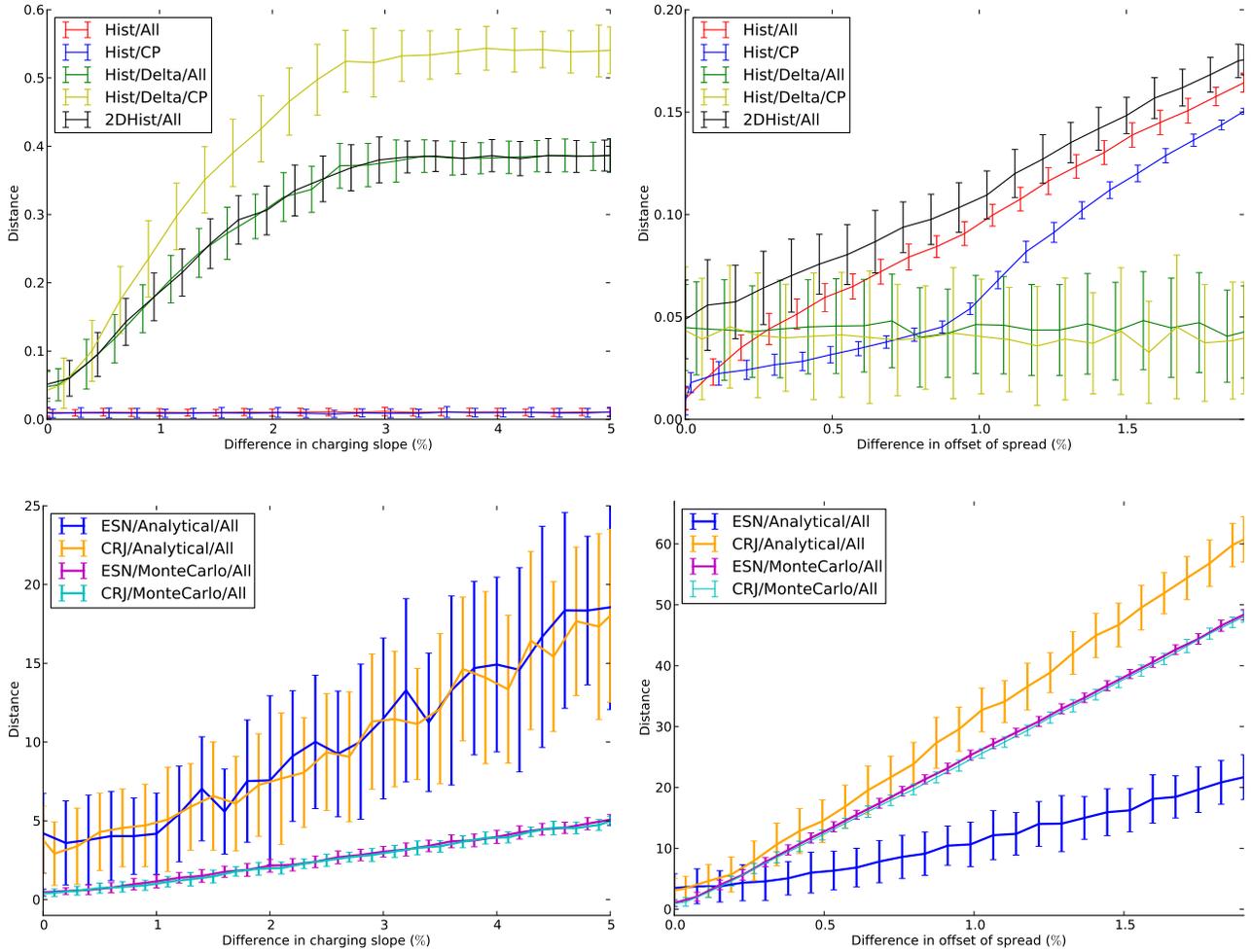


Figure 6. Distances between healthy and faulty samples as a function of the difference in charging slope (left side) or min-max offset (right side). The Hellinger distance is used for histograms (upper plots). The L_2 distance is used for ESN and CRJ models (lower plots). The results with Monte Carlo L_2 distance have been scaled up by a factor of 10 to make them more visible.

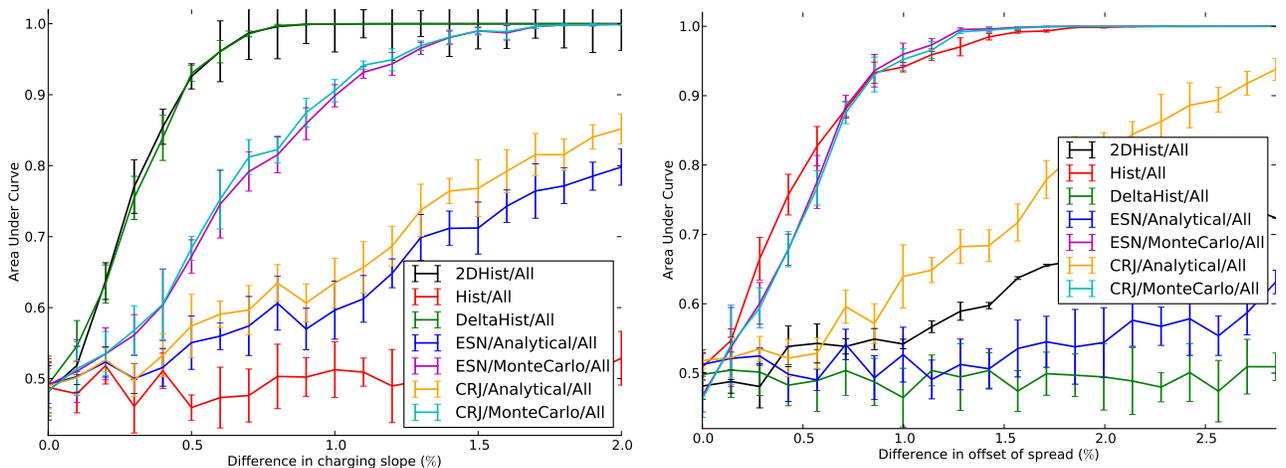


Figure 7. AUC values when distance in model space from the healthy group are used to categorise the models.

analytical distance, are sensitive to both induced fault types. For the left subplot (changing slope), there does not seem to be any significant difference between ESN and CRJ models — however, CRJ with analytical distance outperforms ESN when min-max values are changing. The comparison between Monte Carlo or analytical distances (see Section 4.3.4) does not yield clear results. The analytical distance is generally larger than the Monte Carlo distance, but it also has a much larger variation (the Monte Carlo distances are rescaled in the plots to make them more visible).

The full value of the method, however, is in how well can it classify the datasets into healthy and faulty ones (or, how well it can classify the models into representations of healthy and faulty cases). This is illustrated in Fig. 7, which shows the AUC values based on classification that can be achieved based on the distances shown above.

Again, the histograms are insensitive to weak compressor faults, and the change histograms are insensitive to broken regulator faults. 2D histograms, as well as all RNN models, are able to detect both faults. The two best model for detecting weak compressors are, virtually indistinguishable, the change histogram and the 2D histogram. Somewhat weaker are the RNNs using Monte Carlo distance.

The best model for detecting broken regulators is the signal histogram, very closely followed by CRJ and ESN using Monte Carlo distance. On the other hand, 2D histogram and ESN with analytical distance perform quite poorly, with CRJ using analytical distance ending up in the middle.

Figure 8 demonstrates the performance depending on whether all the data or just the charging period data are used to create the models when detecting difference in charging slope. Using only the charging period data generally achieves slightly better performance than using all the data, which is to be expected.

5.4. Compressor Classification on Bus Fleet data

Figure 9 shows the z-scores, see equation (3), and the corresponding deviation level, see equation (4), for one of the buses in the fleet. The values were computed with a signal change histogram as model and Hellinger distance as metric in the model space. The red vertical line corresponds to a *Compressor Failure*, i.e. a compressor breakdown on the road. In this case, the method flagged a deviation from the fleet several months before the breakdown. After the repair, the z-scores are approximately uniformly distributed between 0 and 1, as expected under normal behaviour.

The deviation levels for all buses in the fleet were computed, using different models and different model distance measures. Following the description in Section 4.2, the ROC curves were computed for the case of *Compressor Failure*, with a prediction horizon of 60 days (i.e. we want to see a deviation

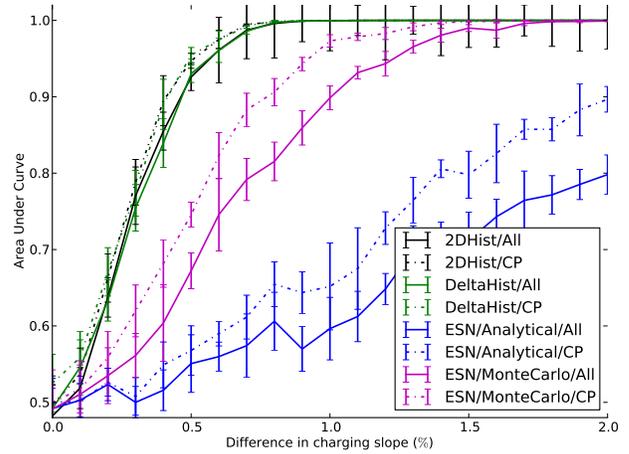


Figure 8. AUC values when the charging slope is varied, for different models. Here it is shown how the results are affected by using all the data (“All”) or just the data during the charging period (“CP”) when building the models.

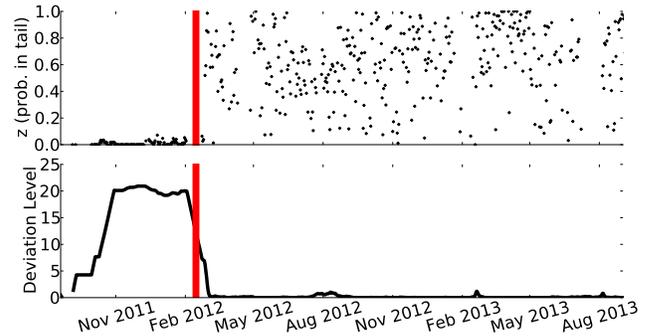


Figure 9. The z-scores and deviation level for one bus when the WTAP signal is represented with a change histogram. The red vertical line corresponds to an occurrence of a *Compressor Failure*

60 days ahead of the breakdown, but not earlier than that).

Figure 10 shows the ROC curves when the COSMO method was used with different types of histograms (signal histogram, signal change histogram, and combined 2D histograms) as models of WTAP. The best performance was achieved with a signal change histogram, or when the change histogram and the signal histogram were combined by taking the maximum of their deviation levels, i.e. equation (8). The signal histogram by itself performed poorly (not better than random guessing). Combining the signal and signal change histograms with a p-norm, i.e. equation (6) with $p = 2$ and $p \rightarrow \infty$, was virtually the same (slightly better) than using signal histogram alone (not shown).

Figure 11 shows the ROC curves when the COSMO method is used with ESN and CRJ as models of WTAP, with L_2 distance (estimated with the Monte Carlo method or the analytical approximation). They all perform very similar to the distance combinations of the signal and signal change his-

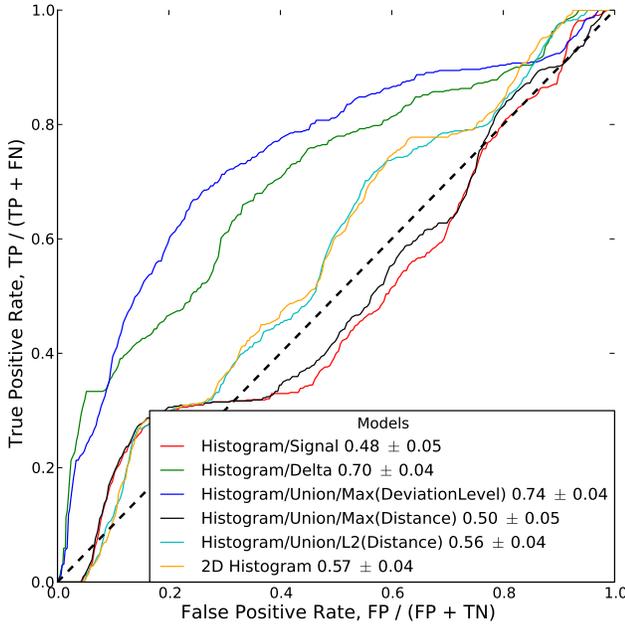


Figure 10. ROC curves and AUC values, with 95 % confidence intervals, when detecting *Compressor Failure* with histogram models.

grams (i.e. not when the deviation levels are combined).

6. DISCUSSION AND CONCLUSION

In this paper we compare ESNs and histograms as models for deviation detection using COSMO method. We have performed experiments both on synthetic data, as well as on a data set coming from longitudinal study on 19 city buses.

The results showed that histograms are more sensitive to changes in slope and equally sensitive to changes in offset of spread, compared to reservoir computing models. However, to detect different types of deviations, histograms need to be calculated over appropriate feature, corresponding to the interesting characteristics of the signal. For example, histogram of the signal is not able to capture differences in slope, and histogram of the signal changes is not able to capture differences in the offset. If another fault characteristic would become relevant (e.g. some form of periodicity disturbance), it would be necessary to design yet another variant of the histogram. In contrast, RNNs detected both types of faults, by being able to capture differences in both slope and signal offset, without providing any additional explicit knowledge of regarding input features. It is plausible that most typical faults can be detected using such models.

Histogram based method turned out to be better on the real data, when predicting *Compressor Failures*. One potential explanation is that real data is noisy, contains missing values, as well as errors in sensor readings that break the structure of the signal and therefore make ESN unable to properly learn

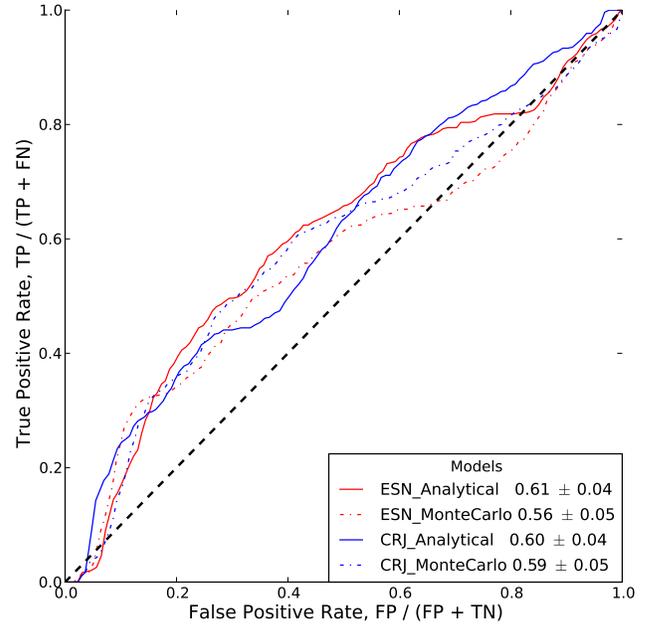


Figure 11. ROC curves and AUC values, with 95 % confidence intervals, when detecting *Compressor Failure* with ESN and CRJ models.

the dynamics. Moreover, equal time intervals between two consecutive sensor readings is assumed when training ESN and CRJ. However, for real *Wet Tank Air Pressure* signal data collected on-board, such accurate timing of sampling is not guaranteed. Therefore histogram, a model that is robust against both those types of noise, may be more successful in accurately capturing the essential characteristics of the signal.

Nevertheless, the ESN and CRJ results on the synthetic data were quite encouraging. There are still unexplored issues regarding the fitting of the ESN and CRJ models. For example, there is more hyper-parameter search to be performed, since we are not certain that the network structure we have selected is the best one. Various techniques for dealing with noise in the data can also be investigated. If the performance of RNN models can be improved, they can be very attractive general models for capturing the dynamics of signals on-board equipment networks and as building blocks in autonomous knowledge creation systems.

REFERENCES

- Alippi, C., Roveri, M., & Trovò, F. (2012). A learning from models cognitive fault diagnosis system. In *Artificial neural networks and machine learning – ICANN 2012* (Vol. 7553, pp. 305–313). Springer Berlin Heidelberg.
- Alippi, C., Roveri, M., & Trovò, F. (2014). A self-building and cluster-based cognitive fault diagnosis system for sensor networks. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6), 1021–1032.

- Angelov, P. (2013). *Autonomous learning systems – from data streams to knowledge in real-time*. John Wiley & Sons, Ltd.
- Bartholomew, R. (1998). *Condition-based maintenance*. www.atp.nist.gov/www/cbm/cbm_wpl.htm.
- Byttner, S., Nowaczyk, S., Prytz, R., & Rögnvaldsson, T. (2013). A field test with self-organized modeling for knowledge discovery in a fleet of city buses. In *2013 IEEE International Conference on Mechatronics and Automation (ICMA)* (p. 896-901).
- Byttner, S., Rögnvaldsson, T., & Svensson, M. (2007). *Modeling for vehicle fleet remote diagnostics* (Technical paper No. 2007-01-4154). Society of Automotive Engineers (SAE).
- Byttner, S., Rögnvaldsson, T., & Svensson, M. (2011). Consensus self-organized models for fault detection (COSMO). *Engineering Applications of Artificial Intelligence*, *24*, 833–839.
- Byttner, S., Rögnvaldsson, T., Svensson, M., Bitar, G., & Chominsky, W. (2009). Networked vehicles for automated fault detection. In *IEEE International Symposium on Circuits and Systems*.
- Chen, H., Tiño, P., Rodan, A., & Yao, X. (2014). Learning in the model space for cognitive fault diagnosis. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(1), 124–136.
- Chen, H., Tiño, P., & Yao, X. (2014). Cognitive fault diagnosis in Tennessee Eastman Process using learning in the model space. *Computers and Chemical Engineering*, *67*, 33–42.
- Fan, Y., Nowaczyk, S., & Rögnvaldsson, T. (2015). Evaluation of self-organized approach for predicting compressor faults in a city bus fleet. *Procedia Computer Science*, *53*, 447–456.
- Filev, D. P., Chinnam, R. B., Tseng, F., & Baruah, P. (2010). An industrial strength novelty detection framework for autonomous equipment monitoring and diagnostics. *IEEE Transactions on Industrial Informatics*, *6*, 767–779.
- Filev, D. P., & Tseng, F. (2006). Novelty detection based machine health prognostics. In *Proceedings of the 2006 international symposium on evolving fuzzy systems, september, 2006 (efs06)* (pp. 193–199). IEEE Press.
- Fogelstrom, K. A. (2006 (filed 2004)). *Air brake system characterization by self learning algorithm* (Nos. 7,089,815 B2).
- Fogelstrom, K. A. (2007 (filed 2005)). *Prognostic and diagnostic system for air brakes* (Nos. 7,216,552 B2).
- Jaeger, H. (2010). *The echo state approach to analysing and training recurrent neural networks – with an erratum note* (Tech. Rep.). Fraunhofer Institute for Autonomous Intelligent Systems.
- Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, *304*, 78–80.
- Quevedo, J., Chen, H., Cugueró, M. À., Tiño, P., Puig, V., García, D., ... Yao, X. (2014). Combining learning in model space fault diagnosis with data validation/reconstruction: Application to the Barcelona water network. *Engineering Applications of Artificial Intelligence*, *30*, 18–29.
- Rodan, A., & Tiño, P. (2012). Simple deterministically constructed cycle reservoirs with regular jumps. *Neural Computation*, *24*, 1822–1852.
- Rögnvaldsson, T., Norrman, H., Byttner, S., & Järpe, E. (2015). Estimating p-values for deviation detection. In *8th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2014), London, UK, September 8-12, 2014* (pp. 1–4). IEEE Computer Society.
- Vachkov, G. (2006, July). Intelligent data analysis for performance evaluation and fault diagnosis in complex systems. *IEEE International conference on fuzzy systems*, 6322–6329.