Automated fault detection using deep belief networks for the quality inspection of electromotors

Beiträge

Jianwen Sun*, Reto Wyss, Alexander Steinecker, and Philipp Glocker

Automated fault detection using deep belief networks for the quality inspection of electromotors

Automatische Fehlerdetektion mittels Deep Belief Netzwerken zur Qualitätskontrolle von Elektromotoren

Abstract: Vibration inspection of electro-mechanical components and systems is an important tool for automated reliable online as well as post-process production quality assurance. Considering that bad electromotor samples are very rare in the production line, we propose a novel automated fault detection method named "*Tilear*", based on Deep Belief Networks (DBNs) training only with good electromotor samples. *Tilear* consctructs an auto-encoder with DBNs, aiming to reconstruct the inputs as closely as possible. *Tilear* is structured in two parts: training and decision-making. During training, *Tilear* is trained only with informative features extracted from preprocessed vibration signals of good electromotors, which enables the trained *Tilear* only to know how to reconstruct good electromotor vibration signal features. In the decision-making part, comparing the recorded signal from test electromotor and the *Tilear* reconstructed signal, allows to measure how well a recording from a test electromotor matches the *Tilear* model learned from good electromotors. A reliable decision can be made.

Keywords: Electromotor, fault detection, deep belief networks, vibration signals, non-desctructive testing, online quality inspection.

Zusammenfassung: Die Analyse von Vibrationssignalen zur Fehelerdetektion elektromechanischer Komponenten und Systeme stellt ein wichtiges Werkzeug in zuverlässiger und automatischer Qualitätssicherung des Produktionsprozesses dar. Davon ausgehend, dass fehlerhafte Elektromotoren nur einen geringen Anteil einer Charge ausmachen, schlagen wir einen neuen Inspektionsansatz namens „*Tilear*" vor. Dieser Ansatz basiert auf einem Deep Belief Netzwerk (DBN), welches mit unterschiedlichen Signalmustern guter Elektromotoren trainiert wurde. *Tilear* generiert einen Auto-Encoder mittels DBNs mit dem Ziel, die Eingangssignale so genau wie möglich zu rekonstruieren. *Tilear* besteht aus zwei Teilen: (i) Training und (ii) Entscheidung. In der Trainingsphase wird *Tilear* nur mit Vibrationssignalen guter Motoren angelernt. Auf diese Weise kann *Tilear* ausschliesslich Signalmuster rekonstruieren, die sich einem guten Motor zuordnen lassen. In der nachfolgenden Entscheidungsphase wird ein aktuelles Vibrationsmuster mit der entsprechenden Rekonstruktion von *Tilear* verglichen. Auf diese Weise wird die Abweichung vom idealen, vorab gelernten Motorsignal quantifiziert und kann für eine Entscheidung in der Qualitätskontrolle verwendet werden.

Schlüsselwörter: Elektromotor, Fehlerdetektion, Deep Belief Netzwerke, Vibrationssignal, zerstörungsfreie Prüfung, Echtzeit Qualitätskontrolle.

*Corresponding Author: Jianwen Sun, CSEM Alpnach/Institute of Neuroinformatics, University/ETH Zürich, Switzerland, e-mail: jianwen.sun@csem.ch
Reto Wyss: ViDi Systems SA, Villaz-St-Pierre, Switzerland
Alexander Steinecker, Philipp Glocker: CSEM Alpnach, Alpnach Dorf, Switzerland

1 Introduction

Electromotors play an important role in nowadays industrial applications. Quality inspection of electromotors is essential for manufacturers to assure their products leaving the factory timely with expected quality. It is likewise critical for users to perform early failure detection to avoid possible malfunctions. Automated quality inspection has always been a popular research topic for both scientists and engineers.

Different techniques have been proposed for electromotor fault detection. These techniques can be classified as signal analysis based methods (SAMs), motor dynamic model based methods (MMs), and knowledge based methods (KMs) [1]. SAMs directly analyze measured signals, such as the vibration signal, without a need of accurate motor modeling. However, the dependence of output signals on input signals is ignored, and it is difficult to reproduce analysis results [1]. In contrast, MMs, including e.g. parameter estimation methods [2, 3] and state estimation methods [4, 5], take both input and output signals into account. But an accurate model for each specific motor type is needed.

Recently, methods based on artificial intelligence algorithms, a subdomain of the KMs [1], have been widely studied. Most of these readily available techniques are on the basis of discriminative learning model. A certain amount of fault samples are required to perform the fault type classification [6–8]. However, in practical applications, for a well-designed product model, it is extremely difficult to get fault samples in abundance for discriminative learning purpose. What makes matters worse is that even a single type of defect typically has many different sensory manifestations. Alternatively, some previous studies [9–11] treated the fault detection problem as an anomaly detection problem. The core of anomaly detection is to recognize the inputs that differ from those under normal conditions. Thus, it is possible to perform fault detections without the need of collecting a large amount of failure data.

Being a very popular research topic in machine learning society recently, Deep Belief Network's (DBN) generative nature enables itself a strong feature learning ability. It has shown promise in many tasks, such as hand written digit recognition [12] and speech recognition [13]. Industrial applications based on DBN have also appeared, e.g. CSEM's quality inspection application of complex surfaces [14].

The objective of this work is to develop a new automated fault detection system for electromotor quality inspection. Treating the fault detection as an anomaly detection problem, this system is based on a Deep Belief Network (DBN) auto-encoder. It learns the sensory signals only from good samples, and makes decisions for test samples with the trained model.

The content of this paper is organized in the following way: In Section 2, theoretical basis and the architecture of the proposed system are introduced. In Section 3, the vibration signal dataset used in experiments is described. In Section 4, the experiment to prove the feasibility of proposed system is presented and discussed. In addition, the performance of the proposed method is compared with a state-of-the-art method, Support Vector Machine (SVM). Finally, conclusions and future directions of this work are given in Section 5.

2 Automated fault detection using DBNs

Anomaly detection (also called "novelty detection" or "outlier detection") has been a mature and active field within diverse research areas [15]. There are many other proposed anomaly detection techniques [16], such as classification based anomaly detection techniques or statistical anomaly detection techniques. DBN is selected because of its strong abilities to model high-dimensional data and to reconstruct the input signal as closely as possible [17, 18].

2.1 Deep belief networks

DBN is a probabilistic generative model, which employs a hierarchical structure constructed by stacking Restricted Boltzmann Machines (RBMs) [17, 19]. As shown in Figure 1, RBM contains two layers of neurons: a binary visible layer and a binary hidden layer. Ws are the symmetric weights between visible units v and hidden units h. c and b are biases to v and h respectively. Each unit is fully connected to units in the other layer, but there is no connection between units in the same layer.

An energy function is used in RBM to model the joint configuration between visible units v and hidden units h. The energy of all possible joint configurations (v,h) in a RBM is given by

$$E(v, h) = -h^T W v - b^T h - c^T v \qquad (1)$$

The probability of every possible joint configuration (v,h) in this energy-based model is

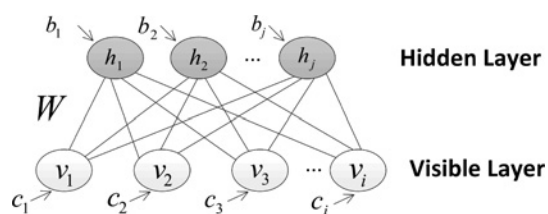$$P(v, h) = \frac{1}{z} e^- E(v, h) \qquad (2)$$

Figure 1: Schematic representation of RBM.

where $z = \sum_{v,h} \exp(-E(v,h))$ is a normalization constant, also referred as the partition function. The conditional distributions of v_i and h_j in a logistic RBM can be expressed as Equations (3) and (4).

$$P\left(h_j = 1|\boldsymbol{v}\right) = \mathrm{Sigm}\left(-b_j - \sum_i v_i W_{ij}\right) \quad (3)$$

$$P\left(v_i = 1|\boldsymbol{h}\right) = \mathrm{Sigm}\left(-c_i - \sum_j h_j W_{ij}\right) \quad (4)$$

The training rule of a logistic RBM is to maximize the probability of the observed data. Ideally, this maximization can be accomplished by gradient descent along the derivative of the log probability of the data taken with respect to the weights $\partial \log P(\boldsymbol{v})/\partial W_{ij}$ [17]. However, because the gradient of the log likelihood of $\boldsymbol{v}$ is intractable, contrastive divergence [20] after n iterations of Gibbs sampling is usually used to approximate it:

$$\frac{\partial \log P(\boldsymbol{v})}{\partial W_{ij}} = \langle v_i h_i \rangle^0 - \langle v_i h_i \rangle^n, \quad (5)$$

where $\langle \cdot \rangle^n$ denotes the average value of the contrastive divergence after n iterations of Gibbs sampling. With $n \to \infty$, it approaches the maximum likelihood learning.

To construct a DBN, a number of RBMs are stacked on top of each other. The hidden layers of lower level RBMs are the visible layers of the adjacent higher level RBMs, as shown in Figure 2. A greedy layer-wise training algorithm is applied to train the DBN, which is actually training the RBMs individually under the contrastive divergence rule. Trained in this way, the DBN can perform a fast inference
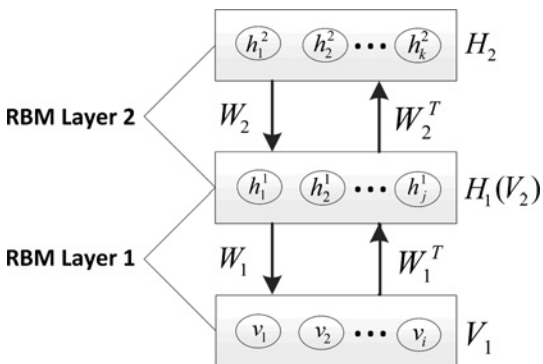
Figure 2: Schematic representation of DBN with 2 RBM layers where v_i represents the visible units of the lowest level RBM, h_j^1 represents the hidden units of the lowest level RBM, and h_k^2 represents the hidden units of the second level RBM. The hidden layer H_1 of the *RBM layer*1 is the visible layer V_2 of the *RBM layer*2. The number of units and layers are just examples.

and extract high level representations, or features, of the input data. Thorough descriptions of DBNs' mathematical and technical details are available elsewhere [17, 19].
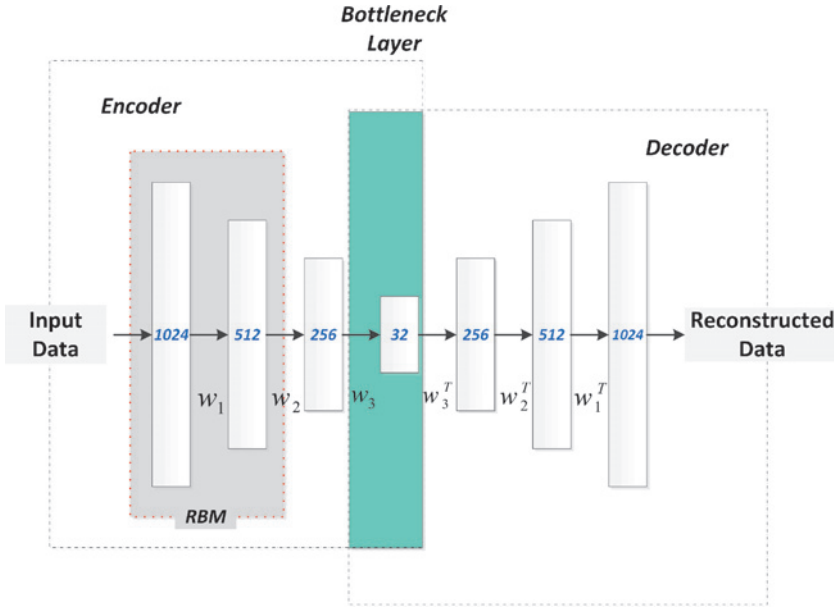
2.2 DBN based auto-encoder

An autoencoder is to learn the representations for the input and then to reconstruct the input based on the learned representations as closely as possible. By unfolding the stacked n RBMs, an auto-encoder composed by $(2n - 1)$ RBMs is constructed. This $(2n - 1)$ directed auto-encoder can be fine-tuned with backpropagation [18]. As shown in Figure 3, the first n RBMs act as an encoder. High-level features of the input data are extracted by this encoder and stored at the hidden layer of the top RBM. The last n RBMs, including the top RBM of the encoder, form a decoder. This decoder reconstructs the input data with the extracted high-level features stored in the top RBM of the encoder. The number of units in the hidden layer of the n_{th} RBM is an important parameter of this auto-encoder, since it limits the number of high-level features used for reconstruction. Thus the hidden layer of the n_{th} RBM is called the bottleneck layer, shown as the blue marked layer in Figure 3.

Generally speaking, training a DBN based auto-encoder is to learn the weights and biases among the layers, such that the input samples can be reconstructed as closely as possible [13].

2.3 Tilear

Taking advantage of DBN auto-encoder's capability to reconstruct the input data as closely as possible, we propose a novel anomaly detection model, named *Tilear*, used for electromotor quality inspection.

Tilear has two functions: "Teacher" for the training phase, "Tester" for the decision making phase. During training phase, only input data from good samples will be learned by the auto-encoder. Small anomalies in the "good" data are tolerable variances. The scarcity of anomalies prevents the DBN from learning and reconstructing those. This property results in an additional reconstruction error for the data containing anomalies. Therefore, the higher the reconstruction error, the more anomalies the data sample contains. An anomaly detector thus can be made by using the reconstruction error with a threshold. In *Tilear*, the reconstruction error S_i, also named score, is the Root Mean Squared Error (RMSE) between the input data I_i and corresponding reconstructed

Figure 3: Architecture of DBN based auto-encoder. Numbers in the blocks represent the number of units in each layer. Unit number in the bottom layer represents the sampling points from the input data. Unit number of the rest layers in the encoder represents the number of extracted high-order features for their respective input data. The number of units and layers are only examples. It is not required to have the same numbers in the experiments, or to be 2^n.

data R_i, averaged over n dimensions of the data, as expressed in Equation (6).

$$S_i = \sqrt{\frac{\sum_{j=1}^{n} \left(R_{ij} - I_{ij} \right)^2}{n}} \qquad (6)$$

The reconstruction error threshold S_{th} demarcating the anomaly boundary is another model parameter. This is determined by searching the reconstruction error space of a validation dataset containing labeled good samples and defective samples with anomalies. The algorithm of "Teacher" can be generalized as Algorithm 1.

Table 1: Generalized algorithm of *Tilear's* Teacher function.

Algorithm 1 Teacher of *Tilear*

1: Construct a DBN with specific parameters.
2: Train the constructed DBN with training dataset which only contains input data from good samples.
3: Unroll the trained DBN to construct a DBN based auto-encoder.
4: Use backpropagation to fine tune the DBN based auto-encoder.
5: Learn the statistics of the scores of the validation dataset. Determine the threshold score $S_{threshold}$.

With the selected S_{th}, "Tester" can judge the health status of test sample T_i by computing its reconstruction score S_i, as explained in Algorithm 2.

Table 2: Generalized algorithm of *Tilear's* Tester function.

Algorithm 2 Tester of *Tilear*

1: Reconstruct the test signal with the well trained DBN based auto-encoder.
2: Calculate the reconstruction score of the test signal.
3: **if** $S_{test} > S_{threshold}$ **then**
4:   $T_i \rightarrow$ bad
5: **else**
6:   $T_i \rightarrow$ good
7: **end if**

---

3 Datasets

The vibration dataset used in this paper was collected from two different electromotor models with gearbox, namely RE16 and GB20, available at Maxon Motor AG. The data acquisition system consists of one PCB accelerometer to acquire the vibration signal, one NI9234 analog digital converter (ADC), a computer with an i5-2400s CPU and 4 G RAM, and a data acquisition software developed with the NIDAQmx C API.
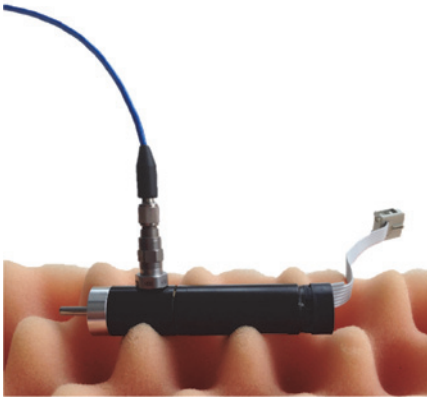
Figure 4: Position of the accelerometer during recording.

Table 3: Recording conditions and the composition of the samples for two electromotor models.

Electromotor Model		RE16	GB20
	Total	170	36
	Good	120	21
No. of Samples	1S	10	4
	2x1S	10	2
	2S	10	4
	2x2S	10	2
	1&2S	10	3
Recording Condition		Changing voltage	Fixed RPM = 10 K
Signal Time Length		12s	5s

According to Maxon engineers' suggestions, electromotors were placed on a piece of foam without any load. The accelerometer was placed at the 12 o'clock position at the center of the gearbox, as shown in Figure 4. Vibration signals were recorded via the NI9234 ADC with sampling rate 51.2 kHz for different time lengths. Each defective sample had one fault of five different types:

- 1 missing gear on the 1st stage, *1S*
- 2 missing gears on the 1st stage, *2x1S*
- 1 missing gear on the 2nd stage, *2S*
- 2 missing gears on the 2nd stage, *2x2S*
- 1 missing gear on the 1st stage and 1 missing gear the on 2nd stage, *1&2S*

There were 170 samples in total for model RE16 consisting of 120 good samples, and 36 samples in total for model GB20 consisting of 21 good samples. Each sample was recorded twice under the same recording condition. In total, 340 data samples and 72 data samples were collected for RE16 and GB20 respectively.

Next, we split the dataset into 3 parts: training, validation and test. The training dataset consisted of 50% of the vibration signals only of good electromotors. The validation dataset consisted of the training dataset, additional 10% of good ones and 50% of bad ones. The validation dataset will be used to determine the reconstruction error threshold for fault detection (as shown in Figure 7). The
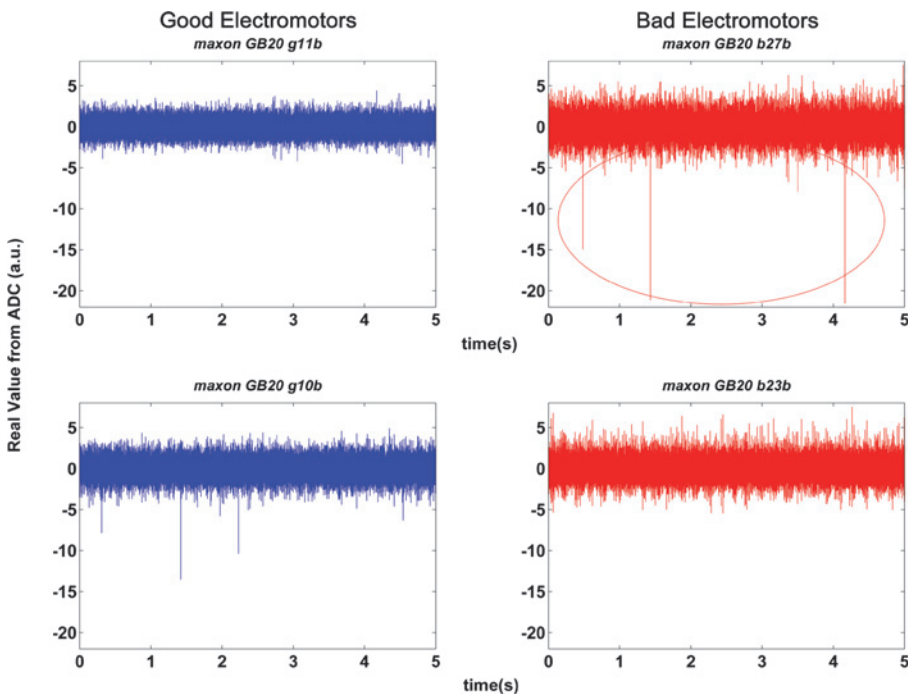

Figure 5: Vibration signal examples in GB20. The left column shows two signals of good electromotors, and the right column of bad ones. The horizontal axis of these figures is the time scale, and the vertical axis represents the signal amplitude with same scale.

test dataset consisted of the rest 40 % of good ones and 50 % of bad ones.

Figure 5 displays selected examples of vibration signals from either good or bad electromotors of model GB20. Although anomalous vibration signals of bad electromotors showed relatively high amplitudes (as shown in the red marked area in the top right panel in Figure 5), it is observed that some bad electromotors' vibration signals look close to that from good electromotors (as shown in the bottom right panel in Figure 5). Thus a simple amplitude threshold is not reliable to distinguish these two classes.

4 Experimental results

The proposed DBN based automated fault detection approach for electromotor quality inspection is first demonstrated. The performance of this proposed method is compared with SVM classification technique.

4.1 Feasibility of Tilear for electromotor quality inspection

Cepstrum analysis is widely used for machinery fault detection and diagnosis especially for gearbox analysis [21]. Cepstrogram, consisting of cepstrums of consecutive windowed vibration signal clips, was used as the signal feature for *Tilear*'s training and testing.

To verify the feasibility of using a DBN auto-encoder for fault detection, experiments were conducted with GB20 and RE16 datasets respectively. Cepstrograms of vibration

(a) Good electromotor sample g2b from GB20

(b) Defective electromotor sample b4b from GB20

Figure 6: Cepstrograms of electromotor samples from GB20. The left column displays the original cepstrograms, and the right one displays the reconstructed ones. The horizontal axis is the time scale, and the vertical is the quefrency.

signals were the feature put into *Tilear*. *Tilear* was first trained with the training set. Reconstruction error threshold was selected with the validation set. *Tilear*'s detection ability was verified with the test set.

Figure 6 shows the input and output of the DBN auto-encoder, which are the original and reconstructed cepstrograms of either good (g2b) or bad (b4b) electromotor samples from GB20. The green-yellow colored block in reconstructed cepstrograms represents the area where a reconstruction error occurs. The stronger the color is, the higher the reconstruction error is. With high enough error, the block can be turned into red. It is observed that the reconstructed cepstrograms look close to the original ones with the first look. This firstly proves that *Tilear* can reconstruct the input as closely as possible, which is one cornerstone for our proposed method. On top of that, it is also observed that the reconstructed cepstrogram of b4b has more and larger yellow marked areas than that of g2b. This means that vibration signal containing more anomalies was reconstructed with a higher error, which is another cornerstone for our proposed method.

Figure 7 shows the reconstruction error cumulative distributions of the validation and test sets of GB20. The cumulative distribution of good samples is marked with green, while that of bad samples with red, which is flipped vertically to help examine the overlap between the two distributions. The less overlapped they are, the better performance *Tilear* has. No overlap means the detector can always make the right decision. The threshold used to validate unlabeled samples is shown as a red dashed line along with its actual value.

It is observed that there is an overlap between good and bad samples for both validation and test sets, but the overlap doesn't cover the whole reconstruction error range. The appearance of overlap can be explained as follows: Ideally, the bad samples should have higher reconstruction error than good samples. However, since there are some bad electromotors having "good-looking" vibration signals, as shown in Figure 5 (bottom-right), it is possible that enough anomalies were present in the training set. DBN then learned not only to represent common patterns of the normal signals, but also to represent these presented anomalies reasonably well. This reduces the reconstruction errors of "good-looking" samples. What's more, if the good sample contains too many anomalies, like the "defective-looking" sample in Figure 5 (bottom-left), *Tilear* cannot reconstruct it very well, resulting in a larger reconstruction error than other good ones. These deviations of the reconstruction errors make the overlap between the two classes. A good point of this overlap is that it doesn't cover the whole reconstruction error range, which sug-
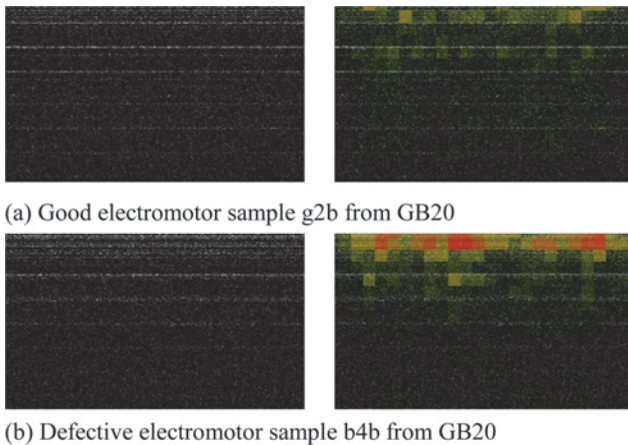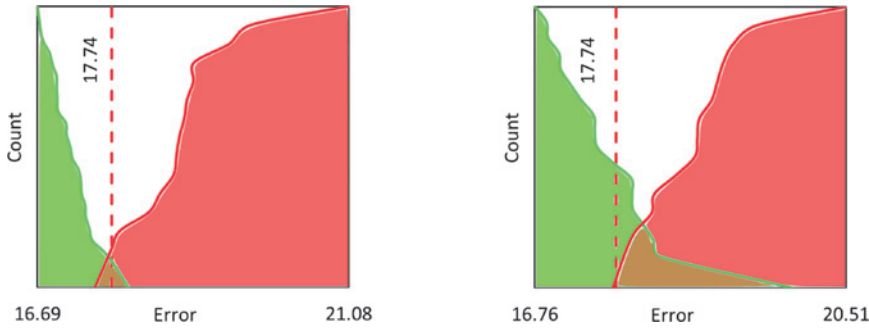
Figure 7: Reconstruction error cumulative distribution of the validation (left) and test (right) sets of GB20.
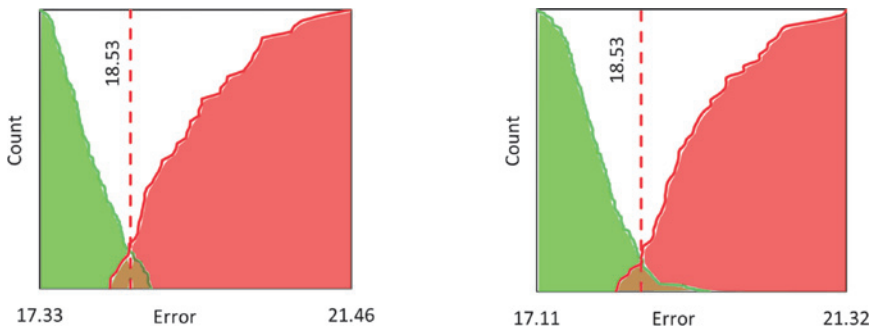
Figure 8: Reconstruction error cumulative distribution of the validation (left) and test (right) sets of RE16.

gests that it is still possible to fully sort out bad samples with the penalization of a decreased true positive rate. From this point, we can conclude that the fault detection for electromotor quality inspection is feasible.

Tilear with same parameters was applied to dataset RE16, which has much more samples of both classes. Experiment results for RE16 are shown in Figure 8. As for GB20, overlap can be found in the cumulative distribution for both sets. Likewise, the overlap doesn't cover the whole reconstruction error range. The feasibility of using DBN auto-encoder for fault detection is verified again.

To sum up, *Tilear* can reconstruct the input as closely as possible. Since there is indeed a reconstruction error difference between good and bad samples, the feasibility of fault detection using DBN is verified. How to reduce the overlap and how to find the best boundary for discrimination are topics to be studied. In addition to that, influences of different parameters on *Tilear*'s performance will be investigated.

4.2 Comparison between Tilear and SVM

Our DBN based anomaly detector *Tilear* was compared to the SVM, a commonly used technique. SVM is to map the

Table 4: Comparison between *Tilear* and SVM.

Model	GB20		RE16	
Method	*Tilear*	SVM	*Tilear*	SVM
Validation	0.986	0.979	0.962	0.938
Test	0.867	0.912	0.951	0.944
Average	0.927	0.946	0.960	0.941

input data, which is not linearly separable, into a high-dimensional space, where a hyperplane can separate the mapped data. Thorough descriptions about SVM and SVM classifier are available elsewhere [14, 22, 23]. In this experiment, LIBSVM [23] was used to construct the SVM.

Due to the imbalanced class distribution in dataset RE16, Area under the Curve (AUC) obtained from the Receiver Operating Characteristics (ROC) curve plot is employed to be the detector performance evaluation metric [24]. With the false positive rate being x-axis and true positive rate being y-axis, ROC is a plot to illustrate how a binary classifier's performance changes with the variation of its discriminative threshold. It reflects the cost/benefit tradeoff in the detector decision making. AUC is used to quantitatively assess the classifier's performance. Generally speaking, a random classier would be

expected to have an AUC number of 0.5. The larger the AUC, the better the classifier is.

We used the cepstrograms of the train set as the input data for both *Tilear* and SVM. AUC is used to evaluate detectors' performances.

As shown in Table 4, AUC values of *Tilear* on the two motor types RE16 and GB20 are higher than those of SVM. This indicates that, compared with SVM, *Tilear* has comparable or even better performances on these two datasets. Furthermore, it is observed that AUC of RE16 test set is higher than that of GB20 test set, and that of validation set for both models are close. This may have resulted from the difference of the training sample number. Number of training samples for GB20 and RE16 is respectively 21 and 120. With more training samples, trained DBN can cover more diverse good samples, and has better generalization ability, which can improve the performance on test set. Nevertheless, a thorough experiment needs to be made regarding the influence of training sample number on *Tilear*'s performance.

Training and testing time of *Tilear* and SVM is not compared here, because *Tilear* was developed to use the Graphical Processing Unit (GPU) while LIBSVM not. A comparison of computational cost between these two methods on GPU platform will be investigated in the future. However, it is worth pointing out that the training time of *Tilear* for these two datasets was 29 minutes and 40 minutes respectively, with the help of NVIDIA GeForce GTX 560 Ti graphic card. It is observed that the training and test time of *Tilear* varies with the change of input data size. The training dataset size increased 13.7 times from 21 samples * 256 000 data points to 120 samples * 614 400 data points, while the training time only increased 37.9%. This increase of training time is acceptable.

As for the average query time, which represents the decision making time for one recorded signal, it was 0.73 second and 1.62 second for RE16 and GB20 respectively. It is fast enough for industrial applications.

5 Conclusions

We have developed a new automated fault detection method for electromotor quality inspection based on Deep Belief Network auto-encoder. This method, named *Tilear*, makes use of Deep Belief Network's strong capabilities of modeling high-dimensional data and reconstructing inputs to construct an anomaly detector. The feasibility of fault detection using *Tilear* is verified with our acquired vibration signal datasets. It is shown that *Tilear* has comparable performance with the state-of-art technique, Support Vector Machine, using the Area under the Curve as the performance metric. *Tilear*'s training time increases with the data size within an acceptable range; query time of *Tilear* is very fast regardless of the data size. The experiments in this paper show the possibility of online fast fault detection for electromotors. It is believed that DBN not only can be used for fault detection, but also has the potential in the fault classification area.

References

1. Liu, X., Zhang, H., Liu, J., and Yang, J., "Fault Detection and Diagnosis of Permanent-Magnet DC Motor Based on Parameter Estimation and Neural Network," IEEE Transactions on Industrial Electronics, vol. 47, no. 5, pp. 1021–1030, 2000.
2. Filbert, D., "Advanced fault diagnosis for the mass production of small-power electric motors." In Measurement Technology and Intelligent Instruments, pp. 100–110. International Society for Optics and Photonics, 1993.
3. Isermann, R., "Process fault detection and diagnosis methods," in Proc. IFAC Fault Detection, Supervision and Safety for Technical Processes, Espoo, Finland, pp. 597–612, 1994.
4. Watanabe, K., Sasak, M., and Himmelblan, D. M., "Determination of op- timal measuring sites for fault detection of nonlinear systems," Int. J. Syst. Sci., vol. 16, pp. 1345–1363, 1985.
5. Chan, C. W., Cheung, K. C., Zhang, H. Y., and Wang, Y., "Fault detec- tion of DC-motors using nonlinear observer based on recurrent B-spline neurofuzzy network," in Proc. 14th IFAC World Congr., vol. B, Beijing, China, pp. 511–516, 1999.
6. Yang, J., Zhang, Y., and Zhu, Y., "Intelligent fault diagnosis of rolling element bearing based on SVMs and fractal dimension," Mechanical Systems and Signal Processing, vol. 21, no. 5, pp. 2012–2024, Jul. 2007.
7. Wu, S.-D., Wu, P.-H., Wu, C.-W., Ding, J.-J., and Wang, C.-C., "Bearing Fault Diagnosis Based on Multiscale Permutation Entropy and Support Vector Machine," Entropy, vol. 14, no. 12, pp. 1343–1356, Jul. 2012.

8. Samanta, B., "Gear fault detection using artificial neural networks and support vector machines with genetic algorithms," Mechanical Systems and Signal Processing, vol. 18, no. 3, pp. 625–644, May 2004.
9. Pan, Y., Chen, J., and Guo, L., "Robust bearing performance degradation assessment method based on improved wavelet packet–support vector data description," Mechanical Systems and Signal Processing, vol. 23, no. 3, pp. 669–681, Apr. 2009.
10. Wulsin, D. F., Gupta, J. R., Mani, R., Blanco, J. A., and Litt, B., "Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement.," Journal of neural engineering, vol. 8, no. 3, p. 036015, Jun. 2011.
11. Fernández-Francos, D., Martínez-Rego, D., Fontenla-Romero, O., and Alonso-Betanzos, A., "Automatic bearing fault diagnosis based on one-class ν-SVM," Computers & Industrial Engineering, vol. 64, no. 1, pp. 357–365, Jan. 2013.
12. Hinton, G., "To recognize shapes, first learn to generate images," Progress in brain research, 165, 535–547, 2007.
13. Mohamed, A., and Dahl, G., "Deep belief networks for phone recognition." NIPS Workshop on Deep Learning for Speech Recognition and Related Applications, 2009.
14. Wyss, R., et al., "Bio-inspired Quality Inspection", CSEM Scientific and Technical Report 2009, page 102, 2010.
15. Hodge, V. and Austin, J., "A survey of outlier detection methodologies," Artificial Intelligence Review, no. 1969, pp. 85–126, 2004.
16. Chandola, V., Banerjee, A., and Kumar, V., "Anomaly detection," ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, Jul. 2009.
17. Hinton, G., Osindero, S., and Teh, Y., "A fast learning algorithm for deep belief nets," Neural computation, vol. 1554, pp. 1527–1554, 2006.
18. Hinton, G. and Salakhutdinov, R., "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. July, pp. 504–507, 2006.
19. Bengio, Y., "Learning Deep Architectures for AI," Foundations and Trends® in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009.
20. Hinton, G. E., "Training products of experts by minimizing contrastive divergence," Neural computation, vol. 14, no. 8, pp. 1771–800, Aug. 2002.
21. Dalpiaz, G., Rivola, A., and Rubini, R., "Gear fault monitoring: Comparison of vibration analysis techniques," Proc. of the 3rd Int. Conf. on Acoustical and Vibratory Surveillance Methods and Diagnostic Techniques, pp. 623–632, 1998.
22. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C., "Estimating the support of a high-dimensional distribution.," Neural computation, vol. 13, no. 7, pp. 1443–71, Jul. 2001.
23. Chang, C.-C. and Lin, C.-J., "Libsvm: A Library for Support Vector Machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp. 1–27, Apr. 2011.
24. Ferri, C., Flach, P., Orallo, J., and Lachice, N. (eds), ECAI' 2004 First Workshop on ROC Analysis in AI.ECAI, 2004.

Jianwen Sun
Microassembly & Robotics, Untere Gründlistrasse 1, CSEM Alpnach, 6005 Alpnach Dorf, Switzerland, Tel.: +41(0)41 672 7551, Fax: +41(0)41 672 7500
jianwen.sun@csem.ch

Jianwen Sun is a PhD student working at Microassembly & Robotics group, CSEM Alpnach. He is Also affiliated to Institute of Neuroinformatics, University/ETH Zürich, Switzerland. His main research area includes signal representation and processing, deep learning theory, and quality inpsection using machine learning techniques.

Dr. Reto Wyss
ViDi Systems SA, Z. I. du Vivier 22, 1690 Villaz-St-Pierre, Switzerland

Reto Wyss received his PhD degree from Institute of Neuroinformatics, University / ETH Zürich, Switzerland. His main research area includes invariant pattern recognition and navigation in autonomous agents. He is currently the CEO of ViDi Systems SA offering a full range of ground breaking vision tools and services.

Dr. Alexander Steinecker
Microassembly & Robotics, Untere Gründlistrasse 1, CSEM Alpnach, 6005 Alpnach Dorf, Switzerland

Alexander Steinecker is a business development manager at CSEM. He is a physicist and received the PhD degree (Dr. rer. nat.) in natural sciences from University of Bonn, Germany. For CSEM he is in charge of strategic positioning and industrializing CSEM's technology platforms in the microrobotics and packaging domain. From his former experience as R&D engineer he gained a profound expertise in delivery of advanced solutions in robotics and automation to industry.

Philipp Glocker
Microassembly & Robotics, Untere Gründlistrasse 1, CSEM Alpnach, 6005 Alpnach Dorf, Switzerland

Philipp Glocker is the section head of Microassembly & Robotics groupd consisting of 8 R&D Engineers, CSEM Alpnach. As an electrical engieer with an MBA degree, he received the Diploma degree in electrical engineering from ETH Zürich. He has a background in industrial R&D for automation and robotics, and has been dealing with all major production leaders like Siemens, Bosch, etc within his 25 years experience in the machinery industry. He has been working for CSEM Alpnach since 2007. He is also coordinating the national R&D initiatives.