A robust method for detection and classification of permanent magnet synchronous motor faults: Deep autoencoders and data fusion approach

Jagath Sri Lal Senanayaka, Van Khang Huynh, Kjell G. Robbersmyr
Department of Engineering Sciences, University of Agder, 4879 Grimstad, Norway
E-mail: jagaths@uia.no

Abstract. Permanent magnet synchronous motors become popular in wind turbines and industrial applications. In critical machines, it is necessary to use robust condition monitoring and fault diagnosis algorithms to prevent faults or shutdowns. The data-driven approach with machine learning algorithms is widely used in industrial and research communities as this method does not require a mathematical model of the system, which is difficult to obtain in practical cases. Most of the successful machine learning methods are based on supervised learning approach, requiring labelled training data. The supervised learning approach cannot use the unlabelled data, while only a few labelled data is in place in the industry. This work uses a deep autoencoder based unsupervised learning method to identify the features of the fault classification algorithm in a self-supervised way, which overcome the shortage of labelled data. The proposed algorithm uses the benefits of available unlabelled data, but it needs only a few labelled data. The fault classification algorithm is based on artificial neural network SoftMax layer and Bayes classifier. The robustness of the algorithm is improved by fusing the current and vibration information. Experimental results are used to validate the robustness of proposed algorithms under noise conditions, and the results show that the algorithm could classify faults robustly.

1. Introduction
Condition monitoring of electrical machines is one of the most important steps in predictive maintenance. Condition monitoring of electrical machines has gained an increasing interest in research as the electric machines have been widely used to replace combustion engine or hydraulic-based machine in industrial applications such as powertrains and vehicles. In wind turbine applications, electrical generators and motors are the most common components. Among electric motors, permanent magnet synchronous motors (PMSMs) become more popular due to their merits of high-power and -torque density, efficiency, and reliability. Vibration and current signals are commonly used to monitor the health conditions of electrical machines. Bearing defects and stator short circuits are the most common faults in electrical motors [1]. The vibration signal is widely used for detecting mechanical faults, e.g. bearing or gearbox faults while current signature analysis is normally applied for detecting faults in electrical machines, e.g. stator or broken-bar faults. However, a bearing defect causes a variation in the air gap of the motor, and this variation produces additional harmonics in the current spectrum, which can be detected using current signature analysis [2], [5]. Furthermore, a stator winding fault produces an unbalance in the mechanical system, which can be detected by vibration signals.

Although many data-driven and machine learning algorithms are intensively applied in condition monitoring for rotating machines, their robustness and generalisability are doubtful. Most algorithms use only one single source of information, e.g. vibration or current signal, in which noise is present in
the data, resulting in a significant reduction of the accuracy. To address such a problem, integrating data fusion techniques into existing classification algorithms can improve the generalisation and robustness of algorithms [3]. This work presents a data fusion algorithm, which allows using both vibration and current information of the motor. If the noise on one data source causes a false indication of fault classification, another data source will justify the information for better fault classification.

Processing a big data is a challenge in condition monitoring practices. Most of the data is normally unlabelled. Thus the machine conditions are difficult to predict. One common solution in research and industry is based on supervised machine learning algorithms for condition monitoring, but it requires all labelled training data, which are hard to be obtained. For example, Support vector machine algorithm is used for bearing fault detection, which requires a large amount of labelled data [6]. Unsupervised machine learning methods can be used to solve this labelled data requirement. Most common unsupervised methods are clustering techniques, in which data is subdivided into clusters based on the found similarity among the data. Principle component analysis (PCA) and independent component analysis (ICA) are commonly used for reducing the dimensionality of features. Deep learning algorithms become popular among conventional machine learning methods because learning features themselves are among advantages of the deep algorithms [4]. In previous decades, deep learning methods were not much popular due to high computational requirements. The development of low-cost, powerful graphics processing units and central processing units removes the barrier of using deep learning methods. In this paper, sparse autoencoders (SAE) are used for automatic feature identification of PMSM faults with many unlabelled data. Few labelled data are used in the supervised SoftMax classification algorithm.

2. Proposed algorithm
The main objective of the proposed algorithm is to improve the robustness of fault classification and reduce the requirement of many labelled training data. Figure 1 shows a block diagram of the proposed fault diagnosis algorithm.

![Figure 1. The block diagram of the proposed fault diagnosis algorithm](image-url)
The algorithm can be produced in three steps. In the first step, both current and vibration signals are collected, and then continuous wavelet transformation (CWT) is used to generate time-frequency maps. The CWT is selected due to its ability to capture non-stationary frequency components effectively. The signals associated with three machine health statuses are then used to train the SAEs in a self-supervised way without any labels. In the second step, the learned features have been applied to the SoftMax layer for fault classification. Since the SAE has already learned to distinguish different fault classes, less amount of labelled data is sufficient to make a supervised classifier. For each current and vibration information, individual SAE and SoftMax layers have been trained, and the decision level fusion is applied to each individual classifier results, enhancing the robustness of final classification results. Finally, trained fault classifier can be employed to predict and classify faults from new data.

2.1. Autoencoder feature learning

An autoencoder (AE) is a neural network which tries to replicate its input at its output through unsupervised learning [7-8]. An AE is architecturally like a feedforward multilayer perceptron (MPL), but its objective is to replicate its input at its output. Various types of AEs are available in the literature, and each method has different ways to extract features or compress information. Denoising AEs are designed to use partially corrupted inputs data and capture the original data removing noise [7]. Sparse AE is designed to learn the features and structures within input data. In the training process, sparsity is imposed to the hidden units to extract the features [8]. A block diagram of an AE with a single hidden layer is given in Figure 2. The simplest AE can be built using three layers. A deep AE can be built in various ways. Several AEs can be trained stepwise, and later all trained AE can be stacked to make one deep AE. Another option is using the Restricted Boltzmann machine (RBM) to make a deep belief network (DBN). In this study, three SAEs are trained and stacked to build a deep network where the training process is given in Figure 3.

![Figure 2. An AE with one hidden layer](image1)

Figure 2. An AE with one hidden layer

![Figure 3. Training steps of a stacked AE](image2)

Figure 3. Training steps of a stacked AE

An AE with a single hidden layer is given in Figure 2, and it tries to learn a function to produce its output same as its input.

\[ \hat{x} = f_{w,b}(x) = x \]

(1)

where \( x \) is the input vector and \( x \in [0,1]^n \). \( W = \{W_1,W_2\} \) denotes the weights and \( b = \{b_1,b_2\} \) represents the bias. First the input vector \( x \) is mapped into a hidden representation by using a parameter \( \theta_1 = \{W_1,b_1\} \), which can be represent by:

\[ h = g_{\theta_1}(x) = \sigma(W_1 x + b_1) \]

(2)
where \( h \in [0, 1]^n \), \( W_1 \in \mathbb{R}^{n \times n'} \), \( b_1 \in \mathbb{R}^{n'+1} \) and \( \sigma(\cdot) \) represents the activation function, which can be selected as sigmoid, Rectified linear units (ReLU) etc. Then the hidden representation \( h \) is mapped back to reconstructed vector \( \hat{x} \in [0, 1]^n \) by similar mapping with \( \theta_2 = \{W_2, b_2\} \)

\[
\hat{x} = g_{\theta_2}(x) = \sigma(W_2 x + b_2)
\]

where \( W_2 \in \mathbb{R}^{n \times n'} \) and \( b_2 \in \mathbb{R}^{n+1} \). The average reconstruction error can be considered as a cost function to optimise the model parameters \( \theta = \{\theta_1, \theta_2\} \);

\[
J_E(W, b) = \frac{1}{m} \sum_{r=1}^{m} \frac{1}{2} \|\hat{x}^{(r)} - x^{(r)}\|^2
\]

where the number of training samples is given by \( m \). If the dimension in the hidden layer is less than the input \( (n' < n) \), then the AE \( h \) learns a compressed representation of the input, which is like the PCA dimension reduction method. The basic AEs discussed so far is capable of dimension reduction only. The above AE can be improved to extract the hidden structure from the input. A SAE has dimension reduction functionality and it can extract the important structure of the input data in which the cost function should be modified as follows.

\[
J_{SAE}(W, b) = J_E(W, b) + \beta |\lambda|E(p|\beta) + \lambda \sum_{l=1}^{s1} \sum_{i=1}^{s1+1} (w_{li}^{(l)})^2
\]

where \( J_{KL}(p|\beta) \) is the Kullback–Leibler (KL) divergence function, which helps to add the sparsity constraint to the AE. \( \frac{\lambda}{2} \sum_{l=1}^{s1} \sum_{i=1}^{s1+1} (w_{li}^{(l)})^2 \) is the weight decay term to prevent overfitting. The \( \beta \) control the sparsity penalty and \( \lambda \) control the penalty for the weight decay. More details about AEs can be found in [7-8].

2.2. The softmax regression

In the previous section, the features have been learned in an unsupervised way. The next step is utilizing the learned features to classify faults. Since there are three classes in the classification task, a multi-class classifier is required. The SoftMax regression is a supervised multi-class classifier [9] and used as the final layer in most of the neural network architectures, such as multilayer perceptrons (MLP), convolutional neural networks (CNN). In this study, a SoftMax layer is added to the end of learned features to make the final classifier.

Consider a training set \( \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\} \) of \( m \) labelled examples where \( x^{(m)} \in \mathbb{R}^n \) and \( y^{(i)} \in \{1, 2, \ldots, K\} \). There are \( K \) types of labelled outputs (or fault classes). Now the conditional probabilities \( P(y = k|x) \) for each value of \( k = 1, \ldots, K \) can be written as:

\[
h_\theta(x) = \begin{bmatrix}
P(y = 1|x; \theta) \\
P(y = 2|x; \theta) \\
\vdots \\
P(y = K|x; \theta)
\end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)^T} x)} \begin{bmatrix}
\exp(\theta^{(1)^T} x) \\
\exp(\theta^{(2)^T} x) \\
\vdots \\
\exp(\theta^{(K)^T} x)
\end{bmatrix}
\]

where \( \theta \) represents the all the parameters in the model.
The cost function for a given model parameter $\theta$ can be given as:

$$J(\theta) = - \left[ \sum_{i=1}^{m} \sum_{k=1}^{K} 1\{y^{(i)} = k\} \log \frac{\exp(\theta(k)^T x^{(i)}(k))}{\sum_{j=1}^{K} \exp(\theta(j)^T x^{(i)})} \right]$$

(7)

where $1\{\cdot\}$ is the indicator function and if $1\{true\} = 1$ and $1\{false\} = 0$. The cost function $J(\theta)$ should be minimized to get optimum $\theta$. An iterative optimization algorithm can be used to solve equation (7).

2.3. Decision level data fusion

The vibration signal based classifier is useful for classifying bearing faults, and the current signal based classifier is more suitable for classifying stator winding faults. A probabilistic classifier is used to combine the results of these individual classifiers. In this study, a Naïve Bayes model is used to fuse the classification results of each individual classifiers since it is simple to build and effective for decision fusion applications. The Naïve Bayes combiner is a supervised learning algorithm. In this decision fusion algorithm, the training data and the confusion matrix results of each classifier are combined to create a new confusion matrix. The training steps of The Naïve Bayes classifiers are [10];

**Step 1:** Get an array $E_{(M,q)}$, which contains the outputs of the q classifiers for M entities in the training set. The true health class labels can be collected from the training set and included in $Z_{(M,1)}$ array.

**Step 2:** Calculate $M_1, M_2, ..., M_c$ values, which represent the number of entities in each health class within $Z_{(M,1)}$. The $c$ represents the number of health classes.

**Step 3:** For each classifier $D_i$, $i = 1, 2, ..., q$, calculate a bespoke $c \times c$ confusion matrix $C_i$.

$$C_i(h_1, h_2) = \frac{K(h_1, h_2) + \frac{1}{c}}{M_{h_1} + 1}$$

(8)

where $K(h_1, h_2)$ is the number of entities in training set with the true class label $h_1$, labelled by the classifier $D_i$ in class $h_2$.

After the training process, the trained bespoke $c \times c$ confusion matrix $C_i$, can be used to fuse new results of individual classifiers. The calculation steps are as follows:

**Step 1:** For each new entity, find the class labels $s_1, s_2, ..., s_L$ assigned by the $L$ base classifiers.

**Step 2:** For each class $\omega_k$, $k = 1, ..., c$ find the probability $P(k)$ of each health class.

Set $P(k) = \frac{M_k}{M}$

(9)

Calculate $P(k) = P(k)C_i(k, s_i)$ for $i = 1, ..., L$

(10)

**Step 3:** Assign label $k^*$ to the entity, where

$k^* = \arg \max_{k=1}^{c} P(k)$

(11)

**Step 4:** Return the final label of the new entity.
3. Experimental setup and results

3.1. The experimental setup

Experimental results are used to test the proposed algorithm. Figure 4 shows the experimental equipment used to collect the data. There are two 400V, 375 rpm, 8 pole pairs PMSMs which are directly coupled. One motor is used as the test motor, and another one is used as the load motor. The load motor is connected to a resistor bank.

![Figure 4. The experimental setup](image)

Three different types of health status are tested namely, healthy, bearing outer-race fault and 10% inter-turn short circuit fault. The faulty components are shown in Figure 5. The data has been collected at 150, 250 and 350 rpm constant speeds. After the experiments, 600 samples images have been produced (200 images from each speed test) using CWT for each health class, and total 1800 images are included in the collected dataset.

3.2. The stacked SAEs

The collected data set is divided into 3 sections where 60% of data (dataset-1) is considered as unlabelled. 20% of data is labelled (dataset-2) and used for supervised training the SoftMax layer and fine-tuning. The remaining 20% data (dataset-3) is used to test the performance of the proposed deep neural network based classifier. As shown in Figure 6, the sparse AEs are trained using several steps. Each input image consists of 48180 data points.

![Figure 6. Unsupervised training steps of each individual SAEs. (a) SAE-1 (b) SAE-2 (c) SAE-3](image)
The first SAE is trained using the unlabelled dataset-1, and the input is reduced to 1000 features. Then in the second step, derived 1000 features are used as the input for the second SAE and 500 features have been extracted. Next, the derived 500 features reduced to 100 features using the third SAE. Sparsity is added to the training process so that the algorithm learns the most important features from the unlabelled data.

After the unsupervised training, the next step is the supervised training. The derived 100 features have been utilised in the SoftMax layer for initial supervised training with dataset 2. Finally, three autoencoders and SoftMax layer are stacked together to make the complete neural network. The final network with stacked SAEs and SoftMax layer is given in Figure 7, which is again trained using the labelled training data (data set 2) for fine-tuning. This trained neural network can be used for classifying faults with new test data (data set 3).

![Figure 7. The Final Neural network with stacked SAEs and SoftMax layer](image)

3.3. The reconstruction performances of SAEs
As discussed earlier, the main objective of an AE is to replicate its input at its output. The average reconstruction performances of self-supervised SAEs are summarised in Table 1. A small reconstruction error doesn’t guarantee an exact image reconstruction, but a useful indicator to measure reconstruction ability of an autoencoder.

| SAE ID | Reconstruction error for vibration signal | Percentage (%) | Reconstruction error for current signal | Percentage (%) |
|--------|------------------------------------------|----------------|----------------------------------------|----------------|
| SAE-1  | 4.77×10^{-4}                            | 0.1            | 2.60×10^{-3}                           | 0.66           |
| SAE-2  | 1.25×10^{-4}                            | 0.02           | 5.13×10^{-4}                           | 0.08           |
| SAE-3  | 1.90×10^{-7}                            | 0.01           | 2.40×10^{-3}                           | 0.80           |

The SAE-1 takes 48180 data points at the input and encodes them to 1000 data points and then again decodes 1000 data points back to 4810. The average reconstruction error for test data set is very low for both vibration signal and the current signals. In this process, the SAE-1 has self-learned the most important 1000 features of the original spectrogram images, and it can regenerate images using derived 1000 features with a very small error. The SAE-2 uses the 1000 features generated by SAE-1 and encodes it to 500 data points and decodes them back to 1000 data points. According to the Table 1, a very low average reconstruction error can be observed, and the SAE-2 works well. The objective of the SAE-3 is to compress the 500 data points to 100 points and decode them back to 500 data points. According to the test data set given in Table 1, a very small reconstruction error can be observed for SAE-3.

3.4. Comparison of the proposed method with CNN
To test the effectiveness of the proposed method, the performances are compared with a convolutional neural network (CNN) results. The convolutional neural networks are one of the best methods used for pattern recognition and image classification applications [11]. However, the main limitation of CNN is that it requires many labelled training data to get better performance from a CNN. But in condition monitoring and fault diagnosis applications, it may difficult to find a large amount of labelled training data.
Table 2. The performance of proposed Stacked AE and SoftMax layer based neural network

| Test case | Training data | Test Accuracy (%) | Training data | Test Accuracy (%) |
|-----------|---------------|-------------------|---------------|-------------------|
| 1         | CNN with large data Dataset-1: 1800×0.6 samples | 98.1% | 90.6% | Dataset-2: 1800×0.2 samples | 89.2% | 57.5% |
| 2         | CNN with few data Dataset-2: 1800×0.2 samples | 89.2% | 57.5% |
| 3         | Proposed Stacked AE and SoftMax layer based neural network Dataset-1: 1800×0.6 samples | 99.2% | 68.9% | Dataset 2: 1800×0.2 samples | |

Table 2 summarises the performances of test cases. In this comparison, the CNN is trained with dataset-1 and dataset-2 for two test cases. In the test case 1, the CNN is trained with dataset-1 in a supervised way. The performance of each individual classifiers is 98.1% for the current signal and 90.6% for the vibration signal. In case-2, the dataset-2 is used to train the CNN, where the performance of the individual classifiers has been reduced because of less number of training data. The proposed method is tested in test-3 where 3 AEs have been trained in an unsupervised way with many unlabeled data (dataset-1) and with less number of labelled training data in dataset-2. The performance of the proposed method is better than test case-2, where CNN is trained using less number of training data. These results demonstrate that the proposed method can give better results than CNN when many unlabelled training data and less number of labelled training data is available.

3.5. Decision level fusion and robustness test

In this study, the Naïve Bayes combiner is used to fuse the result of each individual classifiers. The main objective of this decision level fusion step is to improve the overall accuracy of the algorithm by combining the classification results of each individual classification outputs. Figure 8 shows the performance of the current signal based classifier where the overall classification accuracy is very high as 99.2%. As given in Figure 9, the performance of vibration based classifier is moderate as it is about 68.9%.

![Figure 8](image1.png)

**Figure 8:** Confusion matrix for current signal based classifier  

![Figure 9](image2.png)

**Figure 9:** Confusion matrix for vibration signal based classifier

In the given confusion matrixes, the class-1 belongs to health status, and the class-2 represents the bearing fault case. The class-3 represents the stator winding fault case. The main reason for moderate overall performance in the vibration signal based classifier is that it is not good for detecting stator
winding faults where the accuracy is 45%. Also, it has a confusion to differentiate healthy status with stator winding fault.

White Gaussian noise is added to the input vibration and current signals for testing the performance of each individual classifier with noisy data. The signal to noise ratio is set to 14 dB. The performance of current signal based classifier with the noisy current signal is shown in Figure 10. The classification accuracy of this classifier has been reduced from 99.2% to 74.7%. However, as shown in Figure 11, the classification accuracy of vibration signal based classifier has not been reduced much from its original performance as it is reduced from 68.9% to 65.6%.

![Confusion Matrix for current signal based classifier with a noisy signal](image1)

![Confusion Matrix for vibration signal based classifier with a noisy signal](image2)

Figure 10: Confusion matrix for current signal based classifier with a noisy signal  
Figure 11: Confusion matrix for vibration signal based classifier with a noisy signal

The performance of the decision fusion method is summarised in Table 3.

| Test case     | Current signal | Vibration signal | Decision fusion |
|---------------|----------------|------------------|-----------------|
| Without noise | 99.2%          | 68.9%            | 91.1%           |
| With noise    | 74.7%          | 65.6%            | 88.3%           |

For all test cases in Table 2, the current signal gives better results than vibration signal. A reason for this observation might be that the current signal is less noisy than vibration signal. As given in Table 3, when noise is added to the current signal, the classification accuracy is largely reduced while less reduction is observed for vibration signal. These results confirm that the noise level is a critical factor in determining the classification accuracies of individual classifiers.

In the noiseless cases, the overall classification accuracy of the decision fusion is 91.1 %, which is a high value, making the result highly acceptable. However, this value is less than the individual classification accuracy of a current signal of 99.2%, which is higher than the vibration signal based classifier. With the noisy signals, the classification accuracy of individual classifiers has been reduced. The individual classification accuracy is 74.4% for the current signal based classifier and 65.6% for vibration signal based classifier. However, the decision fusion algorithm can keep the classification accuracy at the high level of 88.3%. Therefore, the decision level fusion maintains the overall classification accuracy in the presence of noises, allowing to ensure the robustness of the proposed method.
4. Conclusion
In this paper, two major improvements have been proposed for supervised machine learning based fault diagnosis algorithms. The demand for a large amount of labelled data is reduced by using many unlabeled data in the self-learning sparse autoencoders for feature learning and using less labelled data for the classification algorithm. For traditional shallow learning clustering methods such as support vector clustering (SVC) require additional feature extraction steps and proposed method does not require additional feature learning steps. The robustness of fault diagnosis algorithm is improved using both the vibration and current signature of the permanent magnet synchronous motor and fusing the individual classification results with the Bayes combiner. Furthermore, the robustness of the proposed method is validated using the experimental data under the presence of noises. The experimental results confirm the effectiveness of the proposed method on robust automatic condition monitoring and fault diagnosis system for wind turbines applications.

References
[1] Allbrecht F et al 1986 Assessment of the reliability of motors in utility applications IEEE Trans. Energy Conv. 1 39-46
[2] Gritli Y et al 2017 Condition monitoring of mechanical faults in induction machines from electrical signatures: Review of different techniques IEEE 11th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (Tinos: Greece) pp 77-84
[3] Choi K et al 2009 Novel Classifier Fusion Approaches for Fault Diagnosis in Automotive Systems IEEE Trans. on Instrumentation and Measurement 58 602-611
[4] Zhao G et al 2016 Research advances in fault diagnosis and prognostics based on deep learning 2016 Prognostics and System Health Management Conference (PHM-Chengdu) pp 1-6
[5] Nandi S, Toliyat H A and Li X 2005 Condition monitoring and fault diagnosis of electrical motors-a review IEEE Trans. on Energy Conv. 20 719-729
[6] Senanayaka J S L et al 2017 Early detection and classification of bearing faults using support vector machine algorithm IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD) (Nottingham: UK) pp 250-255
[7] Vincent P et al. 2008 Extracting and composing robust features with denoising autoencoders Proc. 25th Int. Conf. Mach. Learn. (Helsinki: Finland) pp. 1096–1103
[8] Hosseini-Asl E et al. 2016 Deep Learning of Part-Based Representation of Data Using Sparse Autoencoders With Nonnegativity Constraints IEEE Trans. on Neural Networks and Learning Systems 27 2486-98
[9] Andrew N et al 2018 SoftMax Regression Stanford University Online: http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/ [Accessed February 2018]
[10] Kuncheva L 2014 Combining Pattern Classifiers: Methods and Algorithms (New Jersey: John Wiley & Sons)
[11] Tajbakhsh N et al 2016 Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning IEEE Trans. Med. Imag. 35 1299-1312