Real-Time Motor Fault Detection by 1D Convolutional Neural Networks

Abstract— **Early detection of the motor faults is essential and Artificial Neural Networks (ANNs) are widely used for this purpose. The typical systems usually encapsulate two distinct blocks: feature extraction and classification. Such fixed and hand-crafted features may be a sub-optimal choice and require a significant computational cost that will prevent their usage for real-time applications. In this paper we propose a fast and accurate motor condition monitoring and early fault detection system using 1D Convolutional Neural Networks (CNNs) that has an inherent adaptive design to fuse the feature extraction and classification phases of the motor fault detection into a single learning body. The proposed approach is directly applicable to the raw data (signal) and thus eliminates the need for a separate feature extraction algorithm resulting in more efficient systems in terms of both speed and hardware. Experimental results obtained using real motor data demonstrate the effectiveness of the proposed method for real-time motor condition monitoring.**

***Index Terms*— Convolutional Neural Networks; Motor Current Signature Analysis**

I. INTRODUCTION

Motor fault detection and diagnosis methods can be divided into three major categories: model-based, signal-based, and knowledge-based. Model-based methods use mathematical models describing the normal operating conditions of the induction motors [1]. In model-based methods fault diagnosis algorithms are developed to monitor the consistency between the measured outputs of the practical systems and the model-predicted outputs [2]. The main advantage of a model-based method is that the fault diagnosis is very straightforward if the model parameter has a one-to-one mapping with the physical coefficients [3]. The signal-based methods usually employ one of four main classes of signal processing techniques [4]: time-domain analysis [5],[6], frequency-domain analysis [7],[8], enhanced frequency analysis [9],[10], and time–frequency analysis techniques [11],[13]. The signal-based systems do not require an explicit or complete model of the system but their performance may degrade when working in an unknown or unbalanced condition. It is a well-known fact that as the complexity of advanced signal processing tools used increases, fault detection capability is increased together with the computational cost [34]. The knowledge-based systems may be divided into two groups: qualitative methods on the basis of symbolic intelligence and quantitative methods on the basis of machine learning intelligence [3]. The qualitative methods include fault trees, diagraphs, and expert systems whereas quantative methods have both unsupervised learning systems such as K-means, C-means, nearest neighbor, principal component analysis (PCA), and self organizing maps (SOM), and supervised learning systems such as artificial neural networks (ANN), fuzzy logic (FL), support vector machines (SVM), partial least squares (PLS), and hybrid systems. The hybrid systems may be more suitable for complex fault detection problems where the features are extracted from statistical linear projection methods such as PCA and PLS, or signal processing methods such as fast Fourier transform (FFT) and wavelet transform (WT). The performance of knowledge-based methods relies on training data and quality of selected features heavily.

In several studies [14]-[26] different features are proposed. The selected features are presented to classifiers as inputs. Diagnosis of electric stator faults in induction machines using an ANN based approach is proposed in [16]. Machine fault conditions were predicted with less than 2.4% error using only 13 training data patterns and 9 validation data patterns. In [17], Li *et al* presents a neural-network based motor bearing fault diagnosis system using time and frequency based features and achieves average detection rates between 88.75% and 96.25% for different number of hidden neurons. In [20], a neural-network-based fault prediction scheme without using any machine parameter or speed information is presented. Speed is estimated from measured terminal voltage and current. With minimal tuning of the neural network, induction machines of different power ratings can be accommodated, and 93% or more detection performance is achieved. In [21], two types of neural detectors, feedforward multi-layer perceptron (MLP) and self-organized Kohonen's network, were employed to classify healthy or damaged bearings with 85% accuracy. Tung *et al* [22] proposed a CART-ANFIS based classifier to perform fault diagnosis of induction motors and for six different fault classes with 180 training and 90 test samples, obtained total classification accuracy of 91.11% and 76.67% for vibration and current signals, respectively. In another work [23], by using a self-organizing map, cluster information from frequency-domain features is extracted, and fault mode prediction with an error rate of 1.48% is achieved using a 2-dimensional multi-class SVM.

Although mostly satisfactory levels of anomaly detection accuracies were reported, most of these prior studies had to utilize different features and/or classifiers for various types of motor data. This basically shows how crucial the choice of the right features to characterize the specific signals used. Therefore, it is obvious that such features that are either manually selected or hand-crafted may not optimally characterize any motor current signal and thus cannot accomplish a generic solution that can be used for any motor data. In other words which feature extraction is the optimal choice for a particular signal (motor current data) still remains unanswered up to date. Furthermore, feature extraction usually turns out to be a computationally costly operation which eventually may hinder the usage of such methods in real-time monitoring applications. In this study we aim to address these drawbacks and limitations using Convolutional Neural Networks (CNNs).

CNNs are feed-forward and constrained 2D neural networks that has both alternating convolutional and subsampling layers. Convolutional layers basically model the cells in the human visual cortex [27]. The final layers after the convolutional layers are fully connected and thus resemble MLPs. CNNs aim to mimic the mammalian visual system which can accurately recognize certain patterns and structures such as objects in a visual scenery. CNNs have recently become the de-facto standard for "deep learning" tasks such as object recognition in large image achieves and achieved the state-of-the-art performances [28]-[30] with a significant performance gap. In our earlier work, the adaptive CNNs have successfully been used over 1D electrocardiogram (ECG) signals, in particular for the purpose of ECG classification and anomaly detection [31] and exhibit a superior performance in terms of both accuracy and speed. The main reason behind this is that during the training phase the convolution layers of the CNNs basically are optimized to extract highly discriminative features using a large set of 1D filter kernels. The latter layers basically mimic a MLP which performs the classification (learning) task. As a result, when trained properly for a particular signal collection (dataset), they can optimize both feature extraction and classification tasks according to the problem at hand. Usually the optimization technique is a gradient-descent method with random initialization, the so-called back-propagation (BP) method that iteratively searches for the optimal set of network parameters (filter coefficients, MLP weights and biases).

In this paper, we propose a fast, generic and highly accurate motor anomaly detection and condition monitoring system using an adaptive 1D Convolutional Neural Network. With a proper adaptation over the traditional CNNs, the proposed approach can directly classify input signal samples acquired from the motor current, therefore, resulting in an efficient system in terms of speed that allows a real-time application. As mentioned earlier, due to the CNNs ability to learn to extract the optimal features, with the proper training, the proposed system can achieve an elegant classification and fault detection accuracy. The overview of the proposed system is illustrated in Figure 1.

In this study we further aim to demonstrate that simple CNN configurations can easily achieve an elegant detection performance rather than the complex ones commonly used for deep learning. In this way using compact 1D CNNs one can easily perform few hundreds of back-propagation (BP) iterations for efficient training after which a real-time monitoring and continuous anomaly detection can be accomplished since a compact CNN only performs few hundreds of 1D convolutions to generate the output decision vector. This makes them an ideal tool to be used in an accurate, real-time, and cost-effective motor fault detection and early fault alert system. In summary, the contributions of the paper are the following:

- We propose a novel approach for motor fault detection using 1D CNNs that can merge feature extraction and classification tasks into a single machine learner. To our knowledge, this is the pioneer work applied for this purpose.
- By directly learning the best possible features from motor's training data, the proposed generic classifier can adapt to possible variability of motor current signatures and it is applicable to different types of electrical machine failures.
- The proposed method does not require any form of transformation, feature extraction, and post-processing. It can directly work over the raw data, i.e., the motor current signal, to detect the anomalies.
- As a result, while achieving an elegant classification performance, the computational

complexity of the proposed method is significantly lower than any prior work and thus enables the real-time detection capability.

The rest of the paper is organized as follows: A brief introduction to motor faults is provided in Section II. Section III outlines the motor fault diagnosis dataset and the down-sampling process performed over the raw data. The proposed 1D CNNs along with the formulations of the back-propagation training are presented in Section IV. In Section V, the experimental results obtained using the real motor data are presented and performance of the proposed approach is evaluated using the standard performance metrics. Finally, Section VI concludes the paper and suggests topics for the future research.

II. Motor Faults

The main sources of failure for induction machines include both mechanical types caused by bearings faults and electrical types caused by insulation or winding faults. Bearing faults are by far the highest single cause of all motor failures. They are the most difficult to detect but the least expensive to fix when detected early enough and replaced [11]. Consequently, this study focuses on the detection of bearing faults in the earliest possible way.

Bearing faults are mechanical defects and they cause vibration at fault related frequencies. The fault related frequencies can be determined if both bearing geometry and shaft speed are available. Typical ball bearing geometry is depicted in Figure 2.

The equations used for calculating both characteristic vibration frequencies and current frequencies are given as follows [33]:

Outer race defect frequency, f_{OD}, the ball passing frequency on the outer race, is given by

$$f_{OD} = \frac{n}{2} f_{rm} (1 - \frac{BD}{PD} \cos\phi) \quad (1)$$

where f_{rm} is the rotor speed in revolutions per second, n is the number of balls, and the angle φ is the contact angle which is zero for ball bearings.

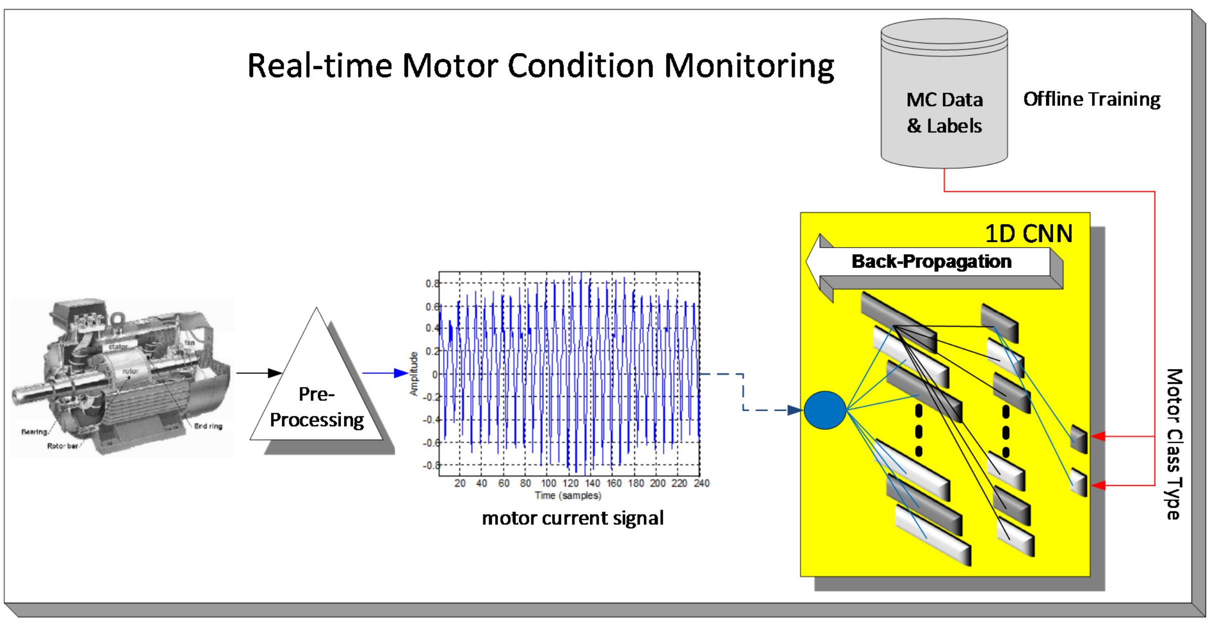

Figure 1: Overview of the proposed approach with training (offline) and real-time monitoring and fault detection phases.

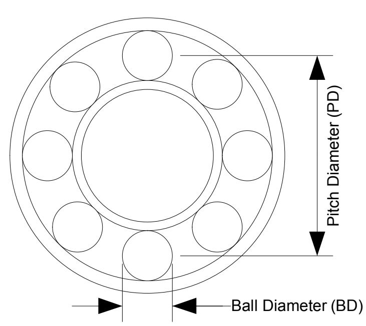

Figure 2: Ball bearing geometry

Inner race defect frequency f_{ID}, the ball passing frequency on the inner race, is expressed as

$$f_{ID} = \frac{n}{2} f_{rm} (1 + \frac{BD}{PD} \cos\phi) \quad (2)$$

Cage defect frequency f_{CD}, caused by irregularity in the rolling element train, is given by

$$f_{CD} = \frac{1}{2} f_{rm} (1 - \frac{BD}{PD} \cos\phi) \quad (3)$$

Ball defective frequency f_{BD}, the ball spin frequency, is given by

$$f_{BD} = \frac{PD}{2BD} f_{rm} (1 - (\frac{BD}{PD})^2 \cos^2 \phi) \qquad (4)$$

The bearing dimension data (*n, PD, BD*) can be easily obtained from the manufacturer in most cases. The mechanical vibration due to the bearing defect results in air gap eccentricity. Oscillations in air gap width in turn cause variations in flux density. The variations in flux density affect the machine inductances producing stator current vibration harmonics [7]. The characteristic current frequencies, f_{CF}, due to bearing characteristic vibration frequencies can be expressed as,

$$f_{CF} = \left| f_e \pm m f_v \right| \qquad (5)$$

where f_e is the line frequency, m is an integer and f_v is characteristic vibration frequency obtained from Eqs. 1-4.

III. MOTOR FAULT DATA PREPARATION

Motor fault related frequency components usually show up in close neighborhood of fundamental frequency in motor current spectrum. Their magnitudes are very small compared to the magnitude of power system fundamental frequency. Therefore, the presence of electrical noise and dominant power system fundamental component in the current frequency spectrum complicate the motor fault detection process. Usually notch filters are used for pre-processing of motor current data to suppress power system fundamental frequency in the current spectrum.

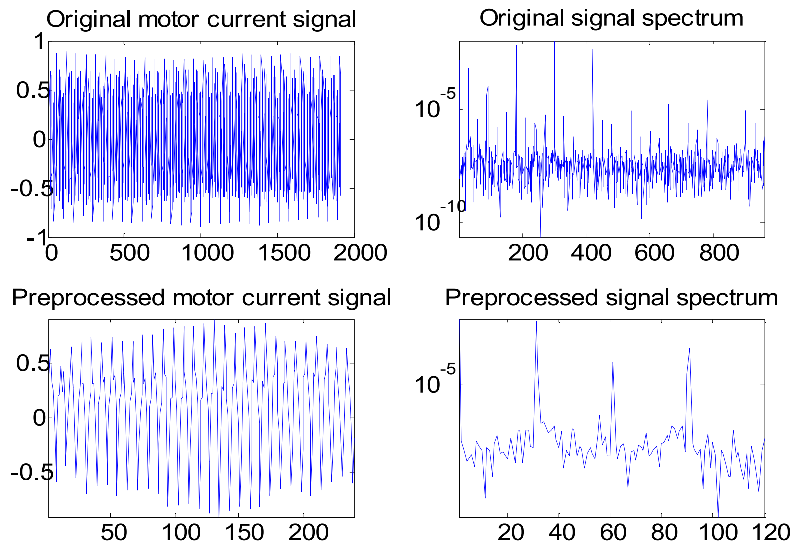

Figure 3: Sample healthy motor current signal and its amplitude spectrum before and after preprocessing

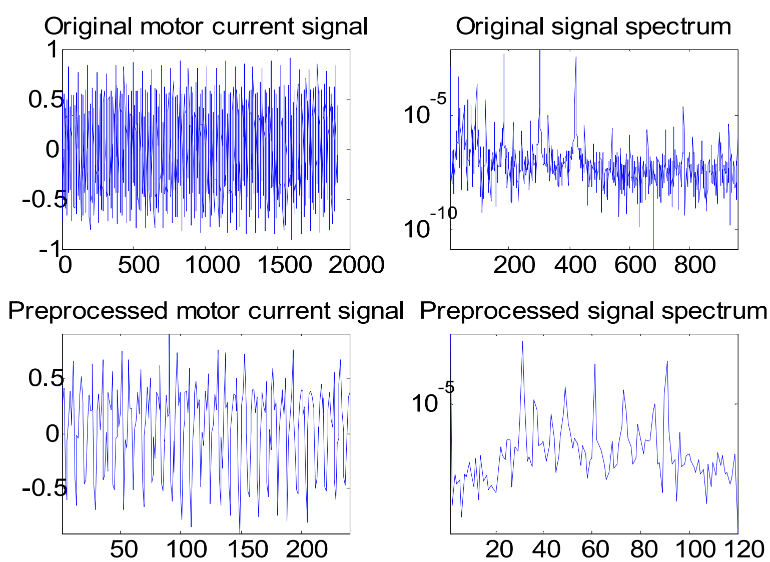

Figure 4: Sample faulty motor current signal and its amplitude spectrum before and after preprocessing.

The test system consists of a three-phase, one hp, 200 V, four-pole, 1750 r/min induction motor (US Motors Frame 143T) and a SquareD CM4000 industrial circuit monitor to capture current data. The shaft end ball bearing is a 6205-2Z-J/C3 (9 balls) and the opposite end ball bearing is a 6203-2Z-J/C3 (8 balls).

In data collection, baseline data is taken for the motor under monitoring using a healthy set of bearings. Then, the cage of shaft end bearing is dented to simulate a cage defect, and line current is sampled under same loading condition to collect data from a motor with a faulty bearing. Motor current is captured at 128 point per cycle for a minute in each trial. The current data is then filtered by a second order notch filter to suppress the fundamental frequency for preprocessing.

The raw input current signal is down-sampled by a factor of 8 by performing a decimation preceded by an anti-aliasing filtering. The decimated signal is then normalized properly to be the input of the 1D CNN classifier. The decimation allows the usage of a simpler CNN configuration, which in turn improves both training and detection speeds. Finally, the training and test sets are normalized to have zero mean and unity standard deviation to remove the effect of dc offset and amplitude biases, and then linearly scaled into [-1, 1] interval before being presented to the CNN classifier. Sample healthy and faulty motor current signals and their amplitude spectrum before and after preprocessing are shown in Figures 3 and 4.

IV. THE PROPOSED SYSTEM WITH 1D CNNS

A. *Overview of CNNs*

CNNs are biologically inspired feed-forward ANNs that present a simple model for the mammalian visual cortex. They are now widely used and become the *de-facto* standard in many image and video recognition systems. Figure 5 illustrates a 2D CNN model with an input layer accepting 28x28 pixel images. Each convolution layer after the input layer alternates with the sub-sampling layers which decimate propagated 2D maps from the neurons of previous layer. Unlike hand-crafted and fixed parameters of the 2D filter kernels, in CNNs they are trained (optimized) by the back-propagation (BP) algorithm. However, the kernel size and the sub-sampling factor that are set to 5 and 2 for illustration in Figure 5, are the two major parameters of the CNN. The input layer is only a passive layer which accepts an input image and assigns its (R,G,B) color channels as the feature maps of its three neurons. With forward propagation over sufficient number of sub-sampling layers, they are decimated to a scalar (1-D) at the output of the last sub-sampling layer. The following layers are identical to the layers of a MLP, fully-connected and feed-forward networks that has the output layer estimating the decision (classification) vector.

In order to accomplish decimation until a scalar at the output CNN layer, the entire CNN configuration (number of convolutional, sub-sampling and MLP layers) has to be arranged according to the input image dimensions. Usually it is the other way around, i.e., the input image dimension is adapted according to the CNN configuration. To address this drawback we performed certain modifications on the CNN topology and further formulated the back-propagation training of a 1D CNN that works over 1D (time) signals.

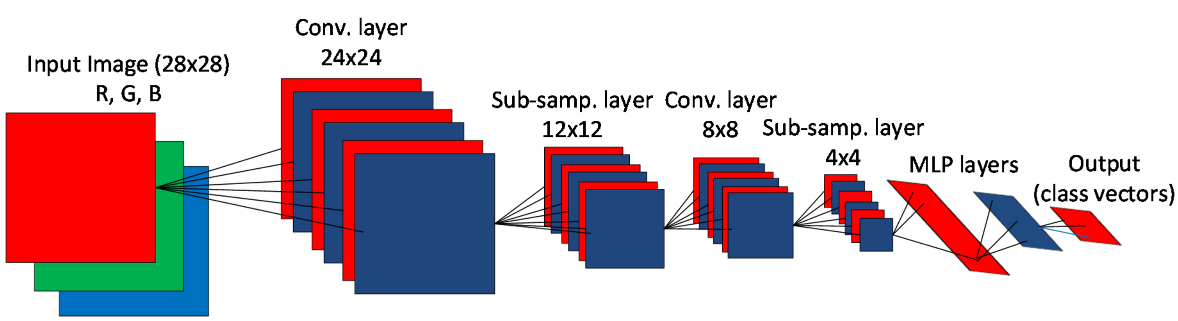

Figure 5: Overview of a sample conventional CNN.

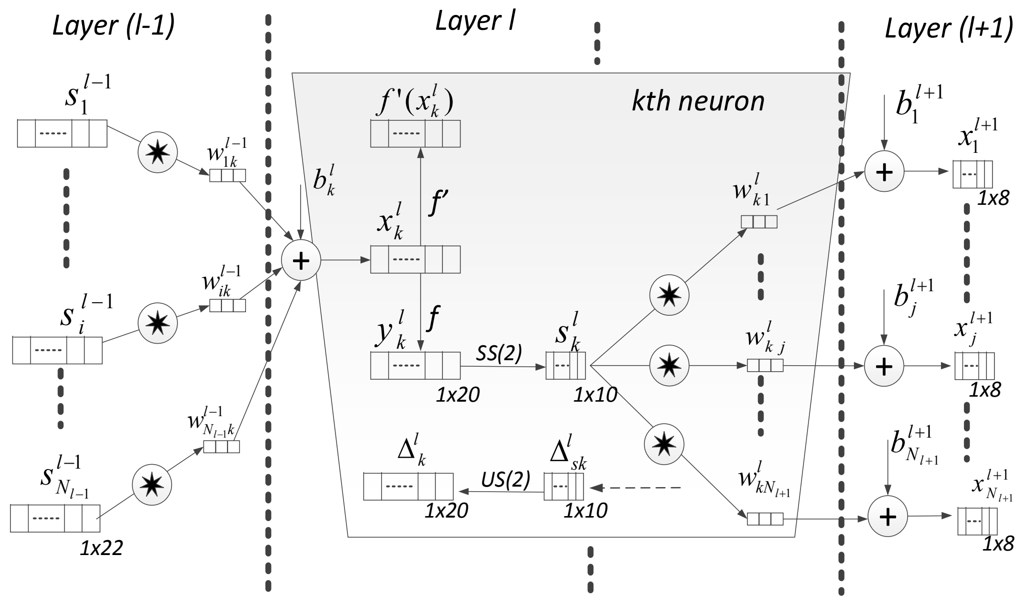

Figure 6: The convolution layers of the proposed adaptive 1D CNN configuration.

B. Adaptive 1D CNNs and Back-Propagation

As mentioned earlier we used an adaptive 1D CNN configuration in order to fuse feature extraction and learning (fault detection) phases of the raw motor current signals. The adaptive CNN topology will allow us to work with any input layer dimension. Furthermore, the proposed compact CNN have now the hidden neurons of the convolution layers that can perform both convolution and sub-sampling operations as shown in Figure 6. This is why we call the fusion of a convolution and a sub-sampling layer as the "CNN layer" to make the distinction but still call the remaining layers as the MLP layers. So, the 1D CNNs are composed of an input layer, hidden CNN and MLP layers and an output layer.

Further structural differences are visible between the traditional 2D and the proposed 1D CNNs. The main difference is the usage of 1D arrays instead of 2D matrices for both kernels and feature maps. Accordingly, the 2D matrix manipulations such as 2D convolution (*conv2D*) and lateral rotation (*rot180*) have now been replaced by their 1D counterparts, *conv1D* and *reverse*. Moreover, the parameters for kernel size and sub-sampling are now scalars, *K* and *ss* for 1D CNNs, respectively. However, the MLP layers are identical to 2D counterpart and therefore, has the same, traditional BP formulation.

In 1D CNNs, the 1D forward propagation (FP) from a previous convolution layer, *l*-1, to the input of a neuron in the current layer, *l*, can be expressed as,

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} conv1D(w_{ik}^{l-1}, s_i^{l-1}) \tag{6}$$

where x_k^l is the input, b_k^l is a scalar bias of the k^{th} neuron at layer *l*, and s_i^{l-1} is the output of the i^{th} neuron at layer *l*-1. w_{ik}^{l-1} is the kernel from the the i^{th} neuron at layer *l*-1 to the k^{th} neuron at layer *l*. The intermediate output of the neuron, y_k^l, can then be expressed from the input, x_k^l, as follows:

$$y_k^l = f(x_k^l) \text{ and } s_k^l = y_k^l \downarrow ss \tag{7}$$

where s_k^l is the output of the neuron and $\downarrow ss$ represents the down-sampling operation with the factor, *ss*.

The adaptive CNN configuration requires the automatic assignment of the sub-sampling factor of the output CNN layer (the last CNN layer). It is set to the size of its input array. For instance, in Figure 6 assume that the layer *l*+1 is the last CNN layer, then *ss* = 8 automatically since the input array size is 8. Such a design allows the usage of any number of CNN layers. This adaptation capability is possible in this CNN configuration because the output dimension of the last CNN layer can be automatically downsized to 1 (scalar) regardless from the native sub-sampling factor parameter that was set in advance for the CNN.

We shall now briefly formulate the back-propagation (BP) steps. The BP of the error starts from the output MLP layer. Let *l*=1 and *l*=*L* be the input and output layers, respectively. Also let N_L be the number of classes in the database. For an input vector *p*, and its corresponding target and output vectors, t_i^p and $[y_1^L, ..., y_{N_L}^L]$, respectively , the mean-squared error (MSE) in the output layer for the input *p*, E_p, can be expressed as follows:

$$E_p = MSE\left(t_i^p, \left[y_1^L, ..., y_{N_L}^L\right]\right) = \sum_{i=1}^{N_L}\left(y_i^L - t_i^p\right)^2 \quad (8)$$

The objective of the BP is to minimize the contributions of network parameters to this error. Therefore, we aim to compute the derivative of the MSE with respect to an individual weight (connected to that neuron, k) w_{ik}^{l-1}, and bias of the neuron k, b_k^l, so that we can perform gradient descent method to minimize their contributions and hence the overall error in an iterative manner. Specifically, the delta of the k^{th} neuron at layer l, Δ_k^l will be used to update the bias of that neuron and all weights of the neurons in the previous layer connected to that neuron, as,

$$\frac{\partial E}{\partial w_{ik}^{l-1}} = \Delta_k^l y_i^{l-1} \quad and \quad \frac{\partial E}{\partial b_k^l} = \Delta_k^l \quad (9)$$

So from the first MLP layer to the last CNN layer, the regular (scalar) BP is simply performed as,

$$\frac{\partial E}{\partial s_k^l} = \Delta s_k^l = \sum_{i=1}^{N_{l+1}} \frac{\partial E}{\partial x_i^{l+1}} \frac{\partial x_i^{l+1}}{\partial s_k^l} = \sum_{i=1}^{N_{l+1}} \Delta_i^{l+1} w_{ki}^l \quad (10)$$

Once the first BP is performed from the next layer, l+1, to the current layer, l, then we can further back-propagate it to the input delta, Δ_k^l. Let zero order up-sampled map be: $us_k^l = up(s_k^l)$, then one can write:

$$\Delta_k^l = \frac{\partial E}{\partial y_k^l}\frac{\partial y_k^l}{\partial x_k^l} = \frac{\partial E}{\partial us_k^l}\frac{\partial us_k^l}{\partial y_k^l} f'(x_k^l) = up(\Delta s_k^l)\,\beta\, f'(x_k^l) \quad (11)$$

where $\beta = (ss)^{-1}$ since each element of s_k^l was obtained by averaging ss number of elements of the intermediate output, y_k^l. The inter BP of the delta error ($\Delta s_k^l \xleftarrow{\sum} \Delta_i^{l+1}$) can be expressed as,

$$\Delta s_k^l = \sum_{i=1}^{N_{l+1}} conv1Dz\left(\Delta_i^{l+1}, rev\left(w_{ki}^l\right)\right) \quad (12)$$

where rev(.) reverses the array and $conv1Dz$(.,.) performs full convolution in 1D with K-1 zero padding. Finally, the weight and bias sensitivities can be expressed as,

$$\frac{\partial E}{\partial w_{ki}^l} = conv1D(s_k^l, \Delta_i^{l+1}) \quad \frac{\partial E}{\partial b_k^l} = \sum_n \Delta_k^l(n) \quad (13)$$

As a result, the iterative flow of the BP algorithm can be stated as follows:

1) Initialize all weights (usually randomly, U(-a, a))
2) For each BP iteration DO:
   a. For each item (or a group of items or all items) in the dataset, DO:
      i. **FP**: Forward propagate from the input layer to the output layer to find outputs of each neuron at each layer, $y_i^l, \forall i \in [1, N_l]\, and\, \forall l \in [1, L]$.
      ii. **BP**: Compute delta error at the output layer and back-propagate it to first hidden layer to compute the delta errors, $\Delta_k^l, \forall k \in [1, N_l]\, and\, \forall l \in [2, L-1]$
      iii. **PP**: Post-process to compute the weight and bias sensitivities.
      iv. **Update**: Update the weights and biases with the (accumulation of) sensitivities found in (c) scaled with the learning factor, ε:

$$w_{ik}^{l-1}(t+1) = w_{ik}^{l-1}(t) - \varepsilon \frac{\partial E}{\partial w_{ik}^{l-1}}$$
$$b_k^l(t+1) = b_k^l(t) - \varepsilon \frac{\partial E}{\partial b_k^l} \quad (14)$$

V. Experimental Results

In this section the experimental setup for the test and evaluation of the proposed motor condition monitoring approach is first presented. Then, the overall results obtained from the experiments using real motor data are presented in terms of the most common metrics found in the literature: classification accuracy (*Acc*), sensitivity (*Sen*), specificity (*Spe*), and positive predictivity (*Ppr*). While accuracy measures the overall system performance over the two classes of motor data, Healthy (H) and Faulty (F), the other metrics are specific to each class and they measure the recall rate of the classification algorithm to each class. The expressions of these standard performance metrics using the hit/miss counters, e.g., true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*), are as follows: *Accuracy* is the ratio of the number of correctly classified patterns to the total number of patterns classified, *Acc = (TP+TN)/(TP+TN+FP+FN)*; *Sensitivity* (Recall) is the rate of correctly classified *fault* events among all data, *Sen = TP/(TP+FN)*; *Specificity* is the rate of correctly classified normal (H) events among all H events, *Spe = TN/(TN+FP)*; and *Positive Predictivity* (Precision) is the rate of correctly classified F events in all detected F events, *Ppr = TP/(TP+FP)*. Finally, the computational complexity of the proposed method for both training (offline) and classification (online) will be discussed.

A. Experimental Setup

As described in Section III, motor current signals are represented as 240 time-domain samples after preprocessing at input of the proposed classifier. The 1D CNN-based motor fault detection system used in all experiments has a compact configuration with only three hidden convolution layers and 2 MLP layers. In this way we aim to accomplish an elegant computational efficiency required for training and particularly for real-time anomaly detection. Besides that, this will also demonstrate that deep and complex CNN configurations are not really needed to achieve the desired detection performance. The 1D CNN configuration used in all experiments has [60 40 40] neurons on the 3 hidden convolution layers and 20 neurons on the hidden MLP layer. The output (MLP) layer size is 2 which is the number of classes and there is a single input neuron which takes the input signal as the 240 (time-domain) samples of the decimated motor current data. The two parameters of the1D CNN, the kernel size, *K*, and the sub-sampling factor, *ss*, are set to 9 and 4, respectively. In this case, the sub-sampling factor for the last CNN layer is set to 4, which is automatically determined in the proposed adaptive CNN implementation.

For all experiments we assigned a two-fold stopping criteria for BP training: the minimum train classification error (CE) is 0.5% or the maximum number of BP iterations is 100. Whenever either criterion is met the BP training stops. The learning factor, ε, is initially set as 0.001 and the global adaptation is performed during each BP iteration: for the next iteration if the train MSE decreases in the current iteration ε is increased by 5%; otherwise, reduced by 30%. We repeated 10 individual BP runs for each data partition and we reported the average anomaly detection performances.

B. Detection Performance Evaluation

An extensive set of experiments are performed using real motor current data samples for a total of 260 healthy (H) and 260 faulty (F) cases. The dataset is obtained from a three-phase squirrel cage induction motor using an industrial circuit monitor for capturing motor current data. The proposed adaptive 1D CNN classifier is implemented by C++ using MS Visual Studio 2013 in 64bit. For training the 1D CNNs, 10-fold cross-validation technique is applied to improve generalization and avoid the over-fitting problem. Table I presents the confusion matrix of motor fault detection problem for all (10) test runs.

For comparison with major competing signal processing techniques for current-based bearing fault detection we implemented wavelet packet decomposition [11], [12] and Fast Fourier Transform (FFT) [7], [17] based feature extraction techniques with three commonly used classifiers from the literature: Multi-Layer Perceptron (MLP) [17], Radial Basis Function Networks (RBFN) [12], and Support Vector Machines (SVM) [23]. We explored various configurations for these classifiers and empirically selected the configurations that achieved the best performances, ([32 64 32 2] for MLP, [32 32 2] for RBFN, and SVM with the linear kernel). Classification results using the aforementioned common metrics are summarized in Table II. While accuracy measures the overall system performance over all classes, the other metrics are specific to each class and they measure the ability of the classification algorithm to distinguish certain events (i.e., faulty motor) from nonevents (i.e., healthy motor). In addition, the region of convergence (ROC) plots are presented in Figure 7 for better visualization of the performance of the proposed method.

Table I: Confusion matrix of the motor fault detection problem for the 10 test runs.

		Classification Result	
		H	**F**
Ground Truth	**H**	2522	78
	F	58	2542

Table II: Motor fault detection performances of the proposed method with six major algorithms.

	Method	Fault detection			
		Acc	*Sen*	*Spe*	*Ppr*
1	Proposed 1D CNN	97.4	97.8	97.0	97.0
2	WP – MLP	97.9	97.0	98.8	98.9
3	WP - RBFN	99.8	100	99.7	99.7
4	WP - SVM	99.2	100	98.4	98.3
5	FFT – MLP	92.7	90.8	94.9	95.1
6	FFT - RBFN	92.5	90.8	94.4	94.6
7	FFT - SVM	84.2	85.0	83.3	82.9

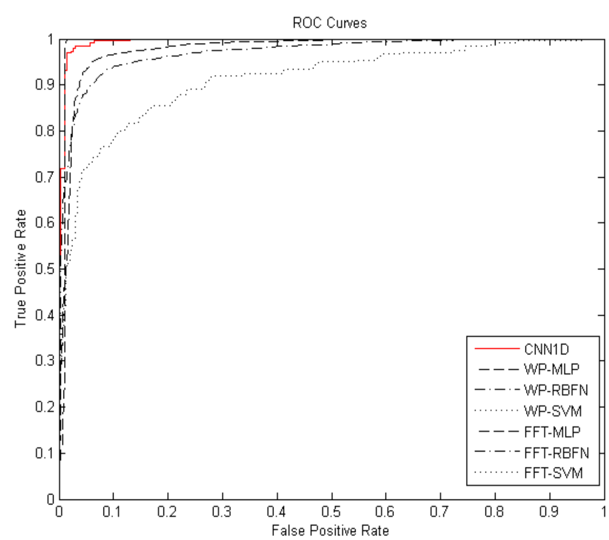

Figure 7: ROC plots of classifiers for comparison. The x- and y-axis represent the false positive rate and true positive rate, respectively.

From the results in Table I and Table II, it is fairly evident that the proposed method based on 1D CNN classifier can be effectively used for motor bearing fault diagnosis. In our implementation with Intel ® OpenMP API the training time of the proposed system was around 4.8 minutes. Note that the training will be performed only once per motor. Specifically, for the single-CPU implementation, the total time for a forward propagation of a single input current data to obtain the class vector is less than 1 *msec*. The average execution time of the proposed algorithm and that of six major algorithms are compared in Figure 8.

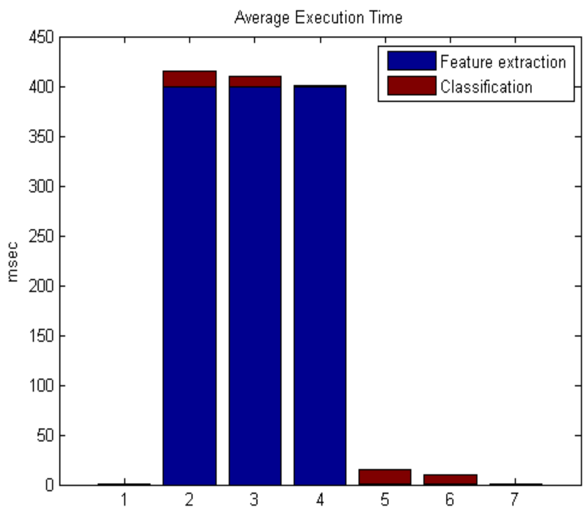

Figure 8: The average execution times (msec) of the proposed algorithm (1) and six major algorithms (2-7, in the same order as in Table II)

VI. Conclusions

In this work, we proposed a novel motor condition monitoring system with an adaptive implementation of 1D Convolutional Neural Networks (CNNs) that are able to fuse the two major blocks of a traditional fault detection approach into a single *learning* body: feature extraction and classification. The proposed system has the ability (to learn) to extract the optimal features with the proper training and thus it can be applied to any motor data. This not only achieves a high level of generalization but also voids the need for manual parameter tuning or hand-crafted feature extraction and furthermore, promises an optimized solution for the problem at hand.

The proposed system is tested with real motor current data and the experimental results demonstrate its potential and effectiveness as a real-time motor condition monitoring system. It can be easily modified to include the detection and classification of both mechanical and electrical faults with signatures on mechanical or electrical quantities (i.e. current). With the BP training the convolutional layers of proposed 1D CNN can learn to extract optimized features while the MLP layers perform the classification task. Experimental results demonstrated that an elegant fault detection accuracy (> 97%) can thus be achieved. Due to the simple structure of the 1D CNNs that requires *only* 1D convolutions (scalar multiplications and additions) any hardware implementation of the proposed system will be quite feasible and cheaper. It is therefore suitable for FPGA or ASIC implementations [32]. Such a hardware implementation and classification of more fault types for real-time monitoring will be the topic of our future work.

References

[1] A. Giantomassi, “Electric Motor Fault Detection and Diagnosis by Kernel Density Estimation and Kullback–Leibler Divergence Based on Stator Current Measurements,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, Mar. 2015.

[2] Z. Gao, C. Cecati and S. X. Ding, “A Survey of Fault Diagnosis and Fault-TolerantTechniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[3] X. Dai and Z. Gao, “From Model, Signal to Knowledge: A Data-Driven Perspective of Fault Detection and Diagnosis,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.

[4] F. Filippetti, A. Bellini, and G. A. Capolino, “Condition monitoring and diagnosis of rotor faults in induction machines: State of art and future perspectives,” in Proc. IEEE WEMDCD, Paris, pp. 196–209, Mar. 2013.

[5] W. Zhou, T. Habetler, and R. Harley, “Bearing fault detection via stator current noise cancellation and statistical control,” *IEEE Transactions on Industrial Electronics*, vol. 55, no.12, pp.4260-4269, Dec. 2008.

[6] C. Kral, T. G. Habetler, R.G. Harley, “Detection Of Mechanical Imbalances of Induction Machines without Spectral Analysis of Time Domain Signals” *IEEE Transactions on Industry Applications*, vol. 40, no. 4, pp. 1101–1106, 2004.

[7] R.R. Schoen, T.G. Habetler, F. Kamran, and R. G. Bartheld, “Motor bearing damage detection using stator current monitoring,” *IEEE Transactions on Industry Applications*, vol. 31, pp. 1274-1279, Nov./Dec. 1995.

[8] G.B. Kliman, W.J. Premerlani, B. Yazici, R.A. Koegl, and J. Mazereeuw, “Sensorless online motor diagnostics,” *IEEE Computer Applications in Power*, vol. 10, no.2, pp. 39–43, 1997.

[9] J. Pons-Llinares, J.A. Antonino-Daviu, M. Riera-Guasp, S. B. Lee, and T.J. Kang and C. Yang "Advanced induction motor rotor fault diagnosis via continuous and discrete time–frequency tools," *IEEE Transactions on Industrial Electronics*, vol. 62, no 3, pp. 1791 – 1802, Mar. 2015 .

[10] D. Z. Li, W. Wang, and F. Ismail, “An enhanced bispectrum technique with auxiliary frequency injection for induction motor health condition monitoring,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no 10, pp. 2279-87, Oct. 2015.

[11] L. Eren, M. J. Devaney, “Bearing Damage Detection via Wavelet Packet Decomposition of the Stator Current”, *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 2, pp. 431–436, 2004.

[12] L. Eren, A. Karahoca, and M. J. Devaney, "Neural network based motor bearing fault detection," IMTC-Instrumentation and

Measurement Technology Conference, pp. 1657-1660, May 2004.

[13] Z. Ye Z, B. Wu B, A. R. Sadeghian, “Current signature analysis of induction motor mechanical faults by wavelet packet decomposition,” *IEEE Transactions on Industrial Electronics*, vol. 50, no. 6, pp. 1217-28, 2003.

[14] J. Liu, W. Wang, F. Golnaraghi, and K. Liu, “Wavelet spectrum analysis for bearing fault diagnostics,” *Measurement Science and Technology*, vol. 19, no. 1, pp. 1-10, 2008.

[15] R. Yan, R. X. Gao, and X. Chen,”Wavelets for fault diagnosis of rotary machines: A review with applications,” *Signal Processing*, vol. 96 Part A, pp 1–15, 2014.

[16] R. Di Stefano, S. Meo, M. Scarano, "Induction motor fault diagnostic via artificial neural network," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE' 94)*, Santiago, 1994, pp. 220-225.

[17] B. Li, M-Y, Chow, Y. Tipsuwan, J.C. Hung, “Neural Network Based Motor Rolling Bearing Fault Diagnosis,” *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, Oct. 2000.

[18] G.F. Bin, J.J. Gao, X.J. Li, and B.S. Dhillon, “Early fault diagnosis of rotating machinery based on wavelet packets – empirical mode decomposition feature extraction and neural network,” *Mechanical Systems and Signal Processing*, vol 27, pp 696–711, 2012.

[19] X. Li, A. Zhang, X. Zhang, C. Li, and L. Zhang, “Rolling element bearing fault detection using support vector machine with improved ant colony optimization,” *Measurement*, vol. 46, no.8, pp 2726-2734, 2013.

[20] K. Kim, A. G. Parlos, and R. M. Bharadwaj, “Sensorless fault diagnosis of induction motors,” *IEEE Transactions on Industrial Electronics*, vol. 50, no. 5, pp. 1038–1051, Oct. 2003.

[21] C.T. Kowalski and T.O-Kowalska, “Neural network application for induction motor faults diagnosis,” *Mathematics and Computers in Simulation*, vol. 63, no. 3-5, pp.435-448, Nov. 2003.

[22] Tung, V. T., Yang, B.-S., Oh, M.-S., & Tan, A. C. C., “Fault diagnosis of induction motor based on decision trees and adaptive neurofuzzy inference,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 1840–1849, 2009.

[23] W.-Y. Chen, J.-X. Xu, S.K. Panda, "Application of artificial intelligence techniques to the study of machine signatures," in *Proceedings of the XX IEEE International Conference on Electrical Machines (ICEM' 2012)*, Marseille (France), 2012, pp. 2390-2396.

[24] M.S. Ballal, Z. J. Khan, H. M. Suryawanshi, R. L. Sonolikar, “Adaptive Neural Fuzzy Inference System for the Detection of Inter-Turn Insulation and Bearing Wear Faults in Induction Motor”, *IEEE Transactions on Industrial Electronics*, vol. 54, no. 1, pp. 250–258, 2007.

[25] F. Zidani, D. Diallo, M. E. H. Benbouzid, R. Nait-Said, “A Fuzzy Based Approach for the Diagnosis of Fault Modes in a Voltage-Fed PWM Inverter Induction Motor Drive”, *IEEE Transactions on Industrial Electronics*, vol. 55, no. 2, pp. 586–593, 2008.

[26] K. Kim and A. G. Parlos, “Induction motor fault diagnosis based on neuropredictors and wavelet signal processing,” *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 2, pp. 201–219, Jun. 2002.

[27] D. H. Wiesel and T. N. Hubel, “Receptive fields of single neurones in the cat’s striate cortex,” *Journal of Physiology*, vol. 148, pp. 574–591, 1959.

[28] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, “Deep big simple neural nets for handwritten digit recognition,” *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010.

[29] D. Scherer, A. Muller, and S. Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition,” in *Int. Conf. on Artificial Neural Networks*, 2010.

[30] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks", In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[31] S. Kiranyaz, T. Ince and M. Gabbouj, “Real-Time Patient-Specific ECG Classification by 1D Convolutional Neural Networks”, *IEEE Transactions on Biomedical Engineering*, doi: 10.1109/TBME.2015.2468589, Aug 2015.

[32] C. Farabet, C. Poulet, J. Han, and Y. LeCun, “CNP: An FPGA-based Processor for Convolutional Networks,” in *International Conference on Field Programmable Logic and Applications*. Prague: IEEE, September 2009.

[33] V. Wowk, Machinery Vibration, Measurement and Analysis, McGraw-Hill 1991.

[34] A. Bellini, F. Filippetti, C. Tassoni and G. A. Capolino, “Advances in Diagnostic Techniques for Induction Machines,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4109–4125, 2008.