Convolutional Neural Net and Bearing Fault Analysis

Dean Lee, Vincent Siu, Rick Cruz, and Charles Yetman
SPAWAR Systems Center Pacific
San Diego, CA, USA
Email: {dean.lee1,vincent.siu,rick.cruz,charles.yetman}@navy.mil

***Abstract*—There has been immense success on the application of Convolutional Neural Nets (CNN) to image and acoustic data analysis. In this paper, rather than preprocessing vibration signals to denoise or extract features, we investigate the usage of CNNs on raw signals; in particular, we test the accuracy of CNNs as classifiers on bearing fault data, by varying the configurations of the CNN from one-layer up to a deep three-layer model. We inspect the convolution filters learned by the CNN, and show that the filters detect unique features of every classification category. We also study the effectiveness of the various CNN models when the input signals are corrupted with noise.**

***Index Terms*—Classification algorithms; multi-layer neural network; signal analysis; fault diagnosis.**

I. Introduction

Anomaly detection from vibration signals has traditionally used sophisticated methods derived from signals processing, usually relating to power spectrum analysis or wavelet-based methods. While these tried-and-true techniques have become an important tool set for the prognostics and health management community, they require extensive expertise, expensive laboratory test rigs, and experiments to properly design robust fault detection algorithms. Moreover, these techniques are sensitive to changes in physical characteristics of the vibrating component, e.g. a change in the manufacture of the component may dramatically alter its vibration signatures. The laboratory tests must be re-run with the new components and the detection algorithms are then re-calibrated to account for the changes. This is a major disadvantage for physics-based models.

Advances in machine learning have enabled many data-driven techniques to overcome the drawbacks of traditional signals analysis techniques. In particular, neural-net-based methods have caught the attention of many communities in recent years due to their automated process of learning latent concepts and their ability to transform the learned models into powerful and highly accurate classification algorithms. The Convolutional Neural Network (CNN) is one such neural network architecture that has shown immense possibilities in image processing and audio processing [1]–[3].

Most of the recent applications of neural networks in vibration analyses have focused on denoising autoencoders, which is a type of neural network that is built with layers of overcomplete and undercomplete hidden neurons to extract the latent features from noisy data [4]–[6]. In this paper, we explore the use of CNN to classify different types of signals by using open rolling bearing vibration data sets [7,8]. In particular, we show that the convolution filters learned by the CNN enable the model to achieve high classification accuracy. We also experiment with a deep CNN architecture study the robustness of deep models in the presence of noisy signals.

II. Related Work

The most basic technique to detect bearing vibration is to establish a baseline using the Root Mean Square (RMS) or Crest Factor of the vibration levels of the bearing housing; a fault is detected when the vibration exceeds some threshold. Clearly this technique is overly simplistic and does not distinguish among different classes of faults.

More often, Fourier-analytic based methods are the tools of choice when it comes to bearing fault diagnosis. For example, time waveform analysis and frequency spectral analysis rely on extensive use of the Fourier transform to shuttle the signals between the time and frequency domains to detect faults; in particular, harmonic analysis is a cornerstone of these type of methods [9].

High frequency detection is yet another method that's used to detect faults in rotating machinery. The idea is that cracking or abrasive wear of the rotating element generates high stress waves, which can then be used as signatures to detect faults [10].

On the other hand, enveloping [11] is a sophisticated technique that was developed to uncover low frequency fault signals, which relies on determining the unique pass frequencies at which various faults can happen. Most enveloping methods seek to determine the optimal window to discern particular faults.

Wavelet-based methods are also popular. While they address many of the short-comings of Fourier-analytic methods, wavelet methods rely on finding the proper bases to be effective [12]. Still, there were some successes using wavelet-based methods combined with machine learning techniques [13].

All of the above methods require experts who are well-versed in signals analyses as well as interpretation of physical signals emanating from the machinery. Moreover, these methods are highly dependent on the physical characteristics of the machinery; for example, if the manufacturing processes of the rotating parts were to change, the fault detection algorithm would need to be re-tuned to account for the new vibration signatures. Recent advances in data-driven classification techniques have removed the need for such retooling. Instead, the algorithms can learn from the data and adapt to changing conditions.

One such technique that has shown great promise is through the use of deep learning, in particular the usage of autoencoders [4]–[6]. Deep autoencoders are able to pick up latent signals by alternating undercomplete and overcomplete hidden layers. However, autoencoders are relatively simple constructs that do not account for the complexity of vibration analysis.

The Convolutional Neural Network, however, can learn optimal convolutional filters that minimizes an error criterion, and therefore is the ideal candidate for automated bearing fault detection. Similarly in automatic speech recognition (ASR), a CNN architecture using raw accoustic signals has already been shown competitive to standard spectral approaches [3]. In this paper, we use the CNN and also apply a deep-learning approach to bearing faults detection on single- and dual-channel raw vibration signal data sets.

III. Convolutional Neural Network

The Convolutional Neural Network is an architecture made up of three distinct layers: an input layer, a convolutional layer, and a pooling layer. The input layer is made up of neurons that hold data values; optionally, preprocessing of the data values may be done prior to the input.

The parameter κ specifies the number of *feature maps* in the convolutional layer. In the case of one dimensional input, such as vibration signals, κ is exactly the number of filters of which the CNN will learn. Let f and g be two functions, then the discrete convolution of f and g is given by

$$(f * g)[n] \equiv \sum_{m=-\infty}^{\infty} f[n-m]g[m].$$

In the CNN, f is the input layer, and g is one of the κ filters which the CNN will optimize to an objective function during the learning process.

The pooling layer downsamples the output from the convolutional layer. Most common strategies are min-, max-, and average-pooling [14].

A *deep* CNN is an architecture where multiple units of CNN are stacked on top of each other, such that the output from the pooling layer of the CNN below becomes the input for the current CNN. Deep learning has been shown to successfully tackle many problems [2,15,16].

However, deep architectures are notoriously difficult to train [17]. Not only does the network take a lot longer to train than other learning techniques, but often times the neural networks have many more hyper parameters for which to optimize and are very resource intensive. Thus, it often takes many trial and error before a suitable model is found. Part of the contributions of this paper is to determine if a deep CNN architecture is suitable for vibration analysis.

Note that it is common practice, especially with vibration signals analysis, to precondition the data to aid the learning process. Techniques such as Fourier analysis, Wavelet analysis, or dimensionality reduction are often applied to extract features or denoise the data before it is fed into the learning algorithm. In this paper, we also show that the CNN with minimal data conditioning can achieve high classification accuracy and that no sophisticated preprocessing is required.

IV. Experiments and Analysis

Methodology

The proposed architecture shown in Figure 1 consists of two main components: filtering layer(s) followed by a classification layer. The input is standardized to zero mean and unit variance, which will improve training performance [18]. The hidden neurons at each layer are ReLU units, which were empirically shown to be robust [19].

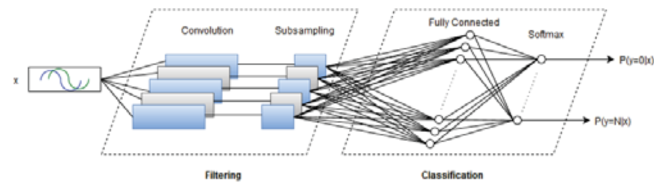


Fig. 1. Implementation of CNN classifier

The filter layers consists of adjacent convolutional and subsampling layers stacked 1 to N times. The convolutional layer convolves the input with a learnable kernel or filter and puts it through a nonlinear ReLU activation function to form an output map. Multiple filters results in the output map having a depth equal to the number of filters. These output maps will show the activation spots when the filters see specific features in the input. The subsampling layer reduces the spatial size of the output map, while the depth dimension remains the same. This is to reduce the amount of parameters in the network and minimize overfitting. Due to the reduced parameters, the computation time will also be reduced.

The output is then fed into two additional layers to perform classification. The first is a fully connected layer composed of sigmoid units that has dropouts enabled to minimize overfitting. The top layer has a softmax function, which will output a conditional probability for each class. A gradient descent optimizer is used to minimize the cross-entropy error when training the system.

We take two publicly available bearing vibration data sets [7,8], which are single- and dual-channel data sets, respectively, and study the effects of tuning various CNN parameters. There are literature that study various methods on extracting bearing fault information from these public data sets [6,11] and self generated data sets [20], but to the best of our knowledge there does not exist a CNN-based method yet for bearing fault analysis that operates on raw vibration signals. Additionally, our experiments test the robustness of CNN when the vibration signals come from multiple channels. We then extend the base CNN configurations to a deep architecture and study the efficacy of deep-CNN on these data sets.

Also, it was found in [21,22] that the injection of corrupted values into the training set can add robustness to the network.

In our experiments, we extend this idea further by randomly corrupting the training data and the test data with Gaussian noise. There are three objectives: 1) when training data is corrupted with noise, the accuracy of the final model should be higher than the model trained without noise corruption [21]; 2) when the test data is corrupted with noise, the robustness of the model trained with uncorrupted data is tested; and 3) when both the training and test data are corrupted with noise, the resiliency of the model against noisy data is tested.

A. Data Sets

The MFPT data set [7] is made up of three sets of bearing vibration data: 1) a baseline set, sampled at 97656 Hz for 6 seconds in each file; 2) an outer race faults set, sampled at 48828 Hz for 3 seconds in each file; and 3) an inner race faults set, sampled at 48828 Hz for 3 seconds in each file. The data points come from a single-channel radial accelerometer. The baseline set was downsampled to 48828 Hz to match the other fault sets. There are additional data files included in the MFPT data set, but they are not used in the experiments.

The Case Western data set [8] is made up of ball bearing data collected from motor bearings that are either normal or faulty. The testbed consisted of a 2 hp motor, torque transducer/encoder, and dynamometer. Accelerometers sampling at 12 kHz were attached to the drive end (DE) and fan end (FE) of the motor housing to measure the vibration. Vibration data was recorded with various engine loads (0 to 3 hp) running at 1700 RPM, motor shaft bearings with faults ranging in depth (none, 0.007, 0.014, 0.021 inches), and fault orientation (inner race, rolling element, and outer race). The outer race fault is centered relative to the load zone positioned at 6:00. The baseline test had 240k data points, while each fault test had 120k data points from each DE and FE sensor.

B. MFPT Data

The data set is divided into three categories: baseline condition, outer race fault conditions, and inner race fault conditions. The data is normalized to 0 mean and unit variance. The goal of the classifier is to correctly categorize the input data to the three conditions.

A single layer CNN was trained with variable number of filters, from 1 to 50 filters. See Figure 2(a) for partial results. We find that the number of filters required to learn the MFPT signals is small; at 5 filters the model achieved roughly 98.2% accuracy. A single layer CNN with up to 50 filters was tested (not shown) and we confirmed that the accuracy does not improve much beyond the 5-filter model. The 5-filter CNN model was then extended to 2 and 3 layers, but we found that the deep CNN model did not improve upon the accuracy of the single layer model.

The learned filters are taken from the single layer CNN and applied to the raw input signals. See Figures 3 and 4 for a comparison between the frequency spectrum of the raw input data and the results of applying convolution using the learned filters from the CNN to the raw input data. The learned filters seemed to pick up distinctive features from each data set which help the classifier categorize the data with high accuracy.

		ν					
		0.05	0.1	0.15	0.2	0.25	0.3
η	0.05	0.9801	0.9777	0.9695	0.9694	0.9584	0.9558
	0.1	0.9725	0.9723	0.9731	0.9664	0.9643	0.9575
	0.15	0.9504	0.9619	0.9658	0.9656	0.9596	0.9514
	0.2	0.9024	0.9407	0.9532	0.9557	0.9508	0.9489
	0.25	0.8379	0.9058	0.9296	0.9397	0.944	0.9418
	0.3	0.7635	0.8486	0.893	0.9025	0.925	0.929

TABLE I
MFPT DATA CLASSIFICATION ACCURACY, 1 LAYER CNN, CORRUPTED TRAINING AND TEST SET

		ν					
		0.05	0.1	0.15	0.2	0.25	0.3
η	0.05	0.9818	0.9826	0.9796	0.9795	0.9656	0.96
	0.1	0.9715	0.9769	0.9688	0.9666	0.9731	0.9579
	0.15	0.9532	0.9695	0.9719	0.9616	0.9707	0.9625
	0.2	0.9117	0.9473	0.9602	0.9634	0.9611	0.9581
	0.25	0.8525	0.901	0.9371	0.9498	0.9483	0.9444
	0.3	0.7859	0.8321	0.8919	0.928	0.9304	0.9321

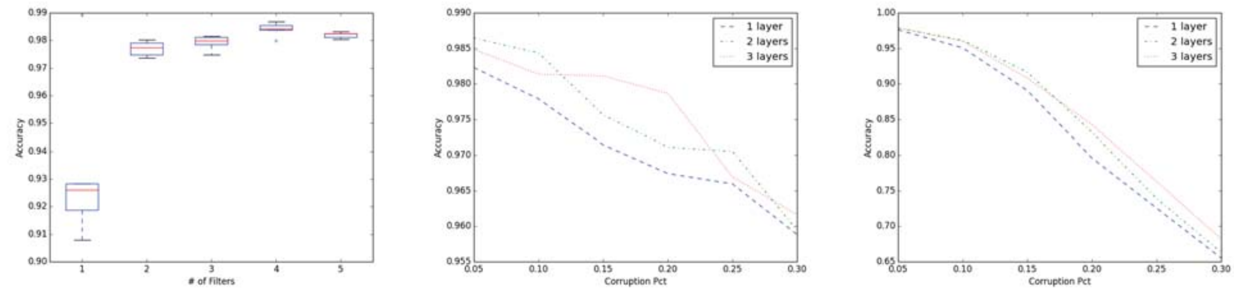
TABLE II
MFPT DATA CLASSIFICATION ACCURACY, 2 LAYER CNN, CORRUPTED TRAINING AND TEST SET

Noise Analysis: We examine the impact of deep architectures when noise is present in the vibration signals. For this analysis, the input signal is randomly corrupted with Gaussian noise. The corruption can happen at 1) the training set, 2) the test set, and 3) both the training and test sets. The corruption parameter ν is the proportion of the training set that is randomly corrupted, and ν ranges from 0.05 to 0.3. The corruption parameter η is the proportion of the test set that is randomly corrupted, and η ranges from 0.05 to 0.3.

Figure 2(b) shows the results of deep CNN configurations against various values of ν. Somewhat surprisingly, the performance of 3-layer CNN is relatively unstable, although its performance at $\nu = 0.2$ is still around 98%, handily beating the other deep-CNN models. Overall, it can be seen that the addition of random perturbation to the training set increases the robustness of the models at all configurations and slightly improves upon the 5-filter single layer CNN classification accuracy.

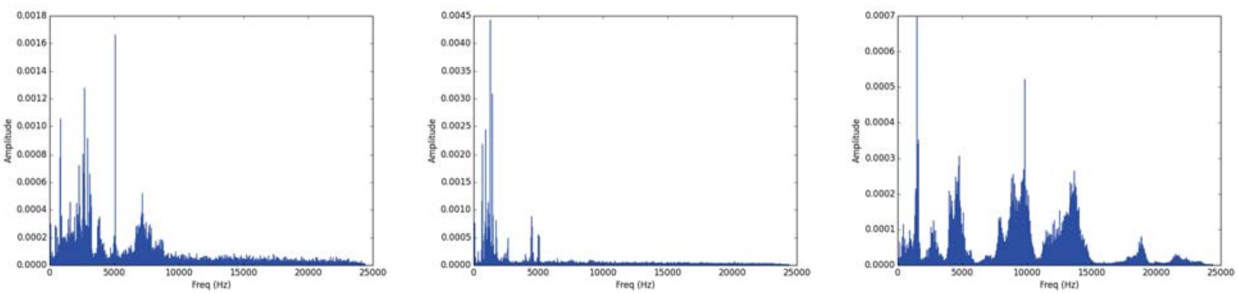
Figure 2(c) shows the results of varied number of layers of CNN that is trained with uncorrupted data, but is tested with corrupted data. It shows that when the input is corrupted up to $\eta = 0.1$, the deep CNN still has relatively high accuracy. However, beyond $\eta = 0.1$ the model begins to lose fidelity quickly. This analysis shows that while a deeper architecture proves to be more resistant against corrupted signals, it is only effective up to a certain point. Furthermore, the deep architectures show the same downward trend as the 1-layer CNN.

Finally, the effects of varying η and ν are tested against the deep CNN architecture. See Tables I, II, and III for results. As expected, deeper architectures with higher values of ν (up to a certain point) seems to handle signal noise better.



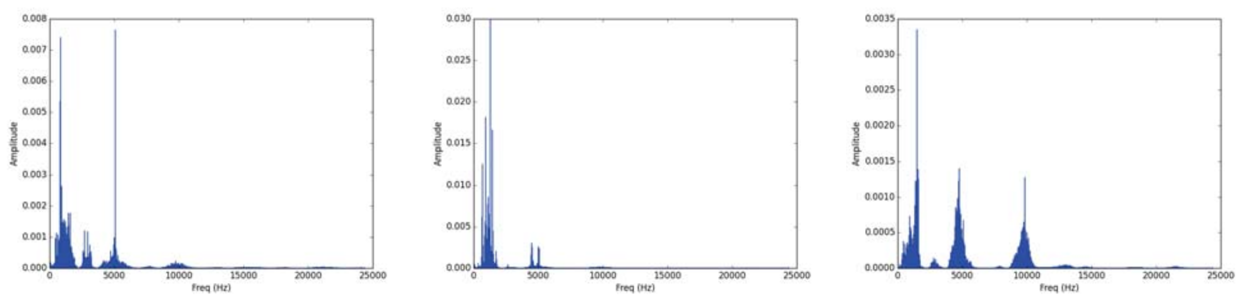
(a) 1-Layer CNN with variable number of filters (b) Deep CNN with Corrupted Training Set (c) Deep CNN with Corrupted Test Set

Fig. 2. MFPT Data, CNN Performance



(a) Raw Baseline (b) Raw Outer Fault (c) Raw Inner Fault

Fig. 3. MFPT, Raw Input Data



(a) Smoothed Baseline (b) Smoothed Outer Fault (c) Smoothed Inner Fault

Fig. 4. MFPT, Smoothed Data

		ν					
		0.05	0.1	0.15	0.2	0.25	0.3
η	0.05	0.9832	0.9781	0.9773	0.9716	0.9712	0.9522
	0.1	0.9716	0.9769	0.9783	0.9757	0.9676	0.9617
	0.15	0.9496	0.9697	0.9713	0.974	0.9692	0.9603
	0.2	0.9074	0.9487	0.9659	0.9678	0.9615	0.9584
	0.25	0.8782	0.9119	0.938	0.9516	0.9535	0.9545
	0.3	0.7635	0.8575	0.8955	0.9273	0.9318	0.9409

TABLE III
MFPT DATA CLASSIFICATION ACCURACY, 3 LAYER CNN, CORRUPTED TRAINING AND TEST SET

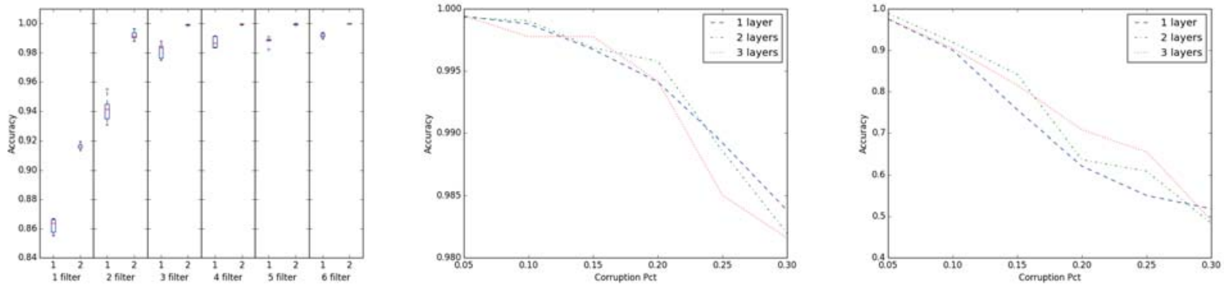
C. Case Western Data

The data set is divided into four categories: baseline condition, outer race fault conditions, rolling element (ball) fault conditions, and inner race fault conditions. The data is standardized to 0 mean and unit variance. The goal of the classifier is to correctly match the input data to the four conditions.

A single layer CNN was trained with a variable number of filters (1 to 10 filters) and data channels (single- or dual-). See Figure 5(a) for a partial graph of the results. We found that the amount of filters to achieve 99% accuracy was small, 6 filters for single-channel and 3 filters for dual-channel. The single-channel exhibited a higher amount of variance and required 10 filters to match the accuracy of the dual-channel. When the single layer 3-filter dual-channel model was extended to 2 or 3 layers, there were negligible improvements.

The frequency spectrum of input data from each category convolved with the learned filters in a single layer CNN

ISBN: 1-60132-431-6, CSREA Press ©



(a) 1-Layer CNN with variable number of filters (b) Deep CNN with Corrupted Training Set (c) Deep CNN with Corrupted Test Set

Fig. 5. Case Western Data, CNN Performance

can be seen in Figures 6 and 7. Only one filtered result with the highest frequency spectrum amplitude is displayed for the baseline and each fault category. It can be seen that a large amount of high frequency energy is produced due to ball bearing defects, resulting in the typical bearing fault frequencies (BPFO, BPFI, BSF) being modulated by a high frequency carrier signal [23]. The figures show that the CNN has learned filters to extract a frequency spectrum profile of the various bearing conditions regardless of high frequency behavior of the signal. This differs from current popular methods of high frequency and envelope vibration analysis, where carrier demodulation is usually required prior to successful diagnosis using the fault frequencies [24].

Noise Analysis: We perform a CNN noise analysis using the same methodology described in Section IV-B.

Figure 5(b) shows the results of deep CNN configurations against various values of ν. The 2-layer CNN is slightly more resilient to corruption of the training data for some small level of ν, but at $\nu > 0.2$ the single layer CNN performs slightly better. The 3-layer CNN is unstable when varying ν. Regardless, the accuracy among all configurations remain relatively high, above 98% even for $\nu = 0.3$.

Figure 5(c) shows the results of varied number of layers of CNN that is trained with uncorrupted data, but is tested with corrupted data. It shows that when the input is corrupted up to $\eta = 0.10$, the deep CNN still has relatively high accuracy. At higher η, the deep CNN has a higher overall accuracy than the single layer CNN but the accuracy still similarly drops off steeply.

Finally, the effects of varying η and ν are tested against the deep CNN architecture. See Tables IV, V, and VI for results. Similar to the results found in IV-B, deeper architectures with ν below some threshold handle signal noise better.

		ν					
		0.05	0.1	0.15	0.2	0.25	0.3
η	0.05	0.9965	0.9960	0.9949	0.9903	0.9842	0.9702
	0.1	0.9903	0.9934	0.9902	0.9876	0.9831	0.9740
	0.15	0.9633	0.9851	0.9879	0.9797	0.9813	0.9689
	0.2	0.9074	0.9592	0.9737	0.9783	0.9778	0.9651
	0.25	0.7903	0.9079	0.9457	0.9640	0.9610	0.9626
	0.3	0.6969	0.8413	0.9007	0.9316	0.9465	0.9458

TABLE IV
CASE WESTERN DATA CLASSIFICATION ACCURACY, 1 LAYER CNN, CORRUPTED TRAINING AND TEST SET

		ν					
		0.05	0.1	0.15	0.2	0.25	0.3
η	0.05	0.9975	0.9965	0.9524	0.9907	0.9790	0.9691
	0.1	0.9881	0.9945	0.9803	0.9920	0.9813	0.9761
	0.15	0.9609	0.9865	0.9879	0.9907	0.9816	0.9679
	0.2	0.9052	0.9576	0.9721	0.9796	0.9739	0.9445
	0.25	0.8580	0.9121	0.9493	0.9626	0.9651	0.9649
	0.3	0.7276	0.8424	0.8898	0.9378	0.9521	0.9427

TABLE V
CASE WESTERN DATA CLASSIFICATION ACCURACY, 2 LAYER CNN, CORRUPTED TRAINING AND TEST SET

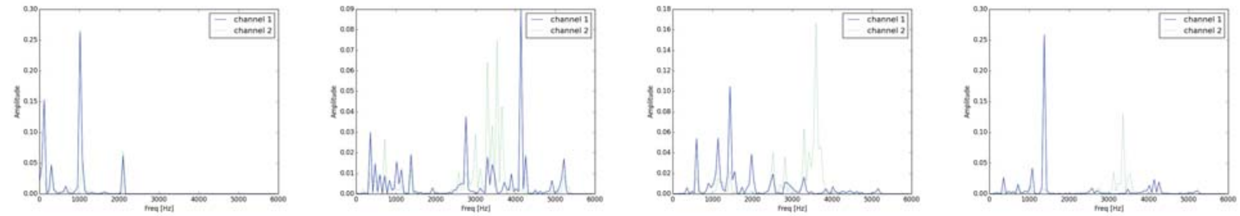
V. DISCUSSION

In the construction of the CNN models, we shunned the traditional signals processing methodologies and opted to directly input the normalized data into the CNN. For example, we found that as long as the input window size is large enough (e.g. more than 100 sample points), then the classifier accuracy doesn't improve even with larger window sizes. We also experimented with input windows with variable sizes of overlap, and found that the size of the overlap does not directly correlate to the accuracy of the final models. These findings imply that both the time taken and required computation resources to train CNN models can be minimized, because the size of the training data is reduced (since the size of window overlap can be zero), and that the input size can be restrained to lessen the memory footprint. The small number of filters required for a CNN to learn the data also reduces the training time. With our hardware setup (2.8 GHz Intel i7 QuadCore CPU, 16GB RAM), we are able to train most of the models under ten minutes. Thus, while neural network architectures in general are resource intensive, when applied to vibration analysis, it is a viable alternative to the manual signals processing techniques that may take days and weeks to complete.

VI. CONCLUSION

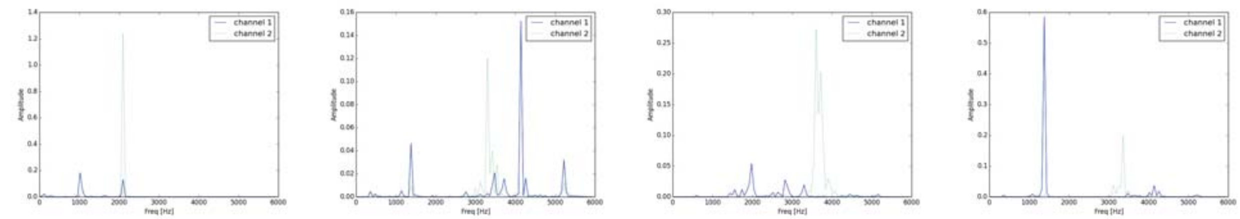
In this paper, we have demonstrated that usage of CNN type architectures shows promise in raw vibration signal ball bearing analysis. The results, 98%-99% depending on the data set used, show that preprocessing of the input data to

ISBN: 1-60132-431-6, CSREA Press ©



(a) Raw Baseline (b) Raw Outer Fault (c) Raw Inner Fault (d) Raw Ball

Fig. 6. Case Western, Raw Input Data



(a) Smoothed Baseline (b) Smoothed Outer Fault (c) Smoothed Inner Fault (d) Smoothed Ball Fault

Fig. 7. Case Western, Smoothed Data

		ν					
		0.05	0.1	0.15	0.2	0.25	0.3
η	0.05	0.9972	0.9953	0.9943	0.9914	0.9868	0.9762
	0.1	0.9862	0.9869	0.9935	0.9916	0.9775	0.9810
	0.15	0.9721	0.9861	0.9844	0.9866	0.9851	0.9566
	0.2	0.9241	0.9621	0.9624	0.9375	0.9781	0.9614
	0.25	0.8585	0.9104	0.9470	0.9583	0.9375	0.9563
	0.3	0.7787	0.8650	0.9063	0.9391	0.9475	0.9550

TABLE VI
CASE WESTERN DATA CLASSIFICATION ACCURACY, 3 LAYER CNN, CORRUPTED TRAINING AND TEST SET

denoise or extract features is not required to achieve a high classification accuracy. The CNN is shown to be able to learn spectral frequency profiles for each category of fault with a very sparse architecture of 1-layer CNN and low amount of filters. It was found that the use of dual-channel data to train the CNN reduces the amount of filters necessary to achieve a high accuracy. Lastly, the addition of corrupted training input data and deep CNN architectures can each provide varying levels of resilience when there exists a certain threshold of signal noise.

REFERENCES

[1] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 609–616.

[2] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1096–1104.

[3] D. Palaz, M. Magimai.-Doss, and R. Collobert, "Analysis of cnn-based speech recognition system using raw speech as input," Idiap, Idiap-RR Idiap-RR-23-2015, 6 2015.

[4] W. Yan and L. Yu, "On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach," in *Annual Conference of The Prognostics and Health Management Society 2015*, vol. 6, 2015. [Online]. Available: http://www.phmsociety.org/node/1652/

[5] S. Tao, T. Zhang, J. Yang, and X. Wang, "Bearing fault diagnosis method based on stacked autoencoder and softmax regression," in *Control Conference (CCC), Hangzhou, China, July 28-30, 2015*. IEEE, July 2015, pp. 6331–6335.

[6] T. Junbo, L. Weining, A. Juneng, and W. Xueqian, "Fault diagnosis method study in roller bearing based on wavelet transform and stacked auto-encoder," in *Control and Decision Conference (CCDC), 2015 27th Chinese*. IEEE, 2015, pp. 4608–4613.

[7] http://www.mfpt.org/FaultData/FaultData.htm.

[8] http://csegroups.case.edu/bearingdatacenter/home.

[9] J. I. Taylor, *The Vibration Analysis Handbook*, 2nd ed. Vibration Consultants, 2003.

[10] S. J. Lacey, "An overview of bearing vibration analysis," 2007.

[11] E. Mendel, T. W. Rauber, F. M. Varejão, and R. J. Batista, "Rolling element bearing fault diagnosis in rotating machines of oil extraction rigs," in *17th European Signal Processing Conference, EUSIPCO 2009, Glasgow, Scotland, UK, August 24-28, 2009*, 2009, pp. 1602–1606.

[12] L. Tóth and T. Tóth, "On finding better wavelet basis for bearing fault detection," *Acta Polytechnica Hungarica*, vol. 10, no. 3, pp. 17–35, 2013.

[13] P. K. Kankar, S. Sharma, and S. P. Harsha, "Rolling element bearing fault diagnosis using wavelet transform," *Elsevier - Neurocomputing*, vol. 74, pp. 1638–1645, 2011.

[14] D. Scherer and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *In: Artificial Neural Networks (ICANN), 20th Int. Conf*, 2010.

[15] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription," in *Proceedings of the 29th International Conference on Machine Learning*. icml.cc / Omnipress, 2012, p. 244.

[16] A. rahman Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," 2011.

[17] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng, "Measuring invariances in deep networks," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 646–654. [Online]. Available: http://papers.nips.cc/paper/3790-measuring-invariances-in-deep-networks.pdf

[18] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Mueller, "Efficient backprop,"

in *In Neural NetworksTricks of the Trade, Lecture Notes in Computer Sciences 1524*. Springer, 1998, pp. 5–50.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, J. Frnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 807–814. [Online]. Available: http://www.icml2010.org/papers/432.pdf

[20] Z. Chen, C. Li, and R.-V. Sanchez, "Gearbox fault identification and classification with convolutional neural networks," *Shock and Vibration*, vol. 2015, 2015.

[21] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. antoine Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," 2010.

[22] Q. Xu, C. Zhang, and L. Zhang, "Denoising convolutional neural network," in *Information and Automation, 2015 IEEE International Conference on*. IEEE, 2015, pp. 1184–1187.

[23] H. Balderston, "Incipient failure detection: Incipient failure detection in ball bearings." DTIC Document, Tech. Rep., 1969.

[24] J. L. Frarey, "The history and application of the envelope detector," DTIC Document, Tech. Rep., 1996.