A Study of Convolutional Neural Networks Learning Mechanisms for Machine Health Monitoring Applications

Mohammed Alabsi[1], Ahmad Saeed Mohammad [2], Ahmed Sammoud[3]

[1] *School of Engineering, The College of New Jersey, Ewing, NJ 08618*
alabsim@tcnj.edu,

[2] *College of Engineering, Mustansiriyah University, Baghdad, Iraq.*
ahmad.saeed@uomustansiriyah.edu.iq,

[3] *School of Engineering, Pennsylvania State University, The Behrend College, Erie, PA 16510*
ams1885@psu.edu

Abstract

In recent years, Deep Learning (DL) and Internet of Things (IoT) technologies have been used and deployed jointly to solve a wide range of modern technical challenges in different areas. With the continuous advancement of IoT connectivity solutions, the range of applications that can benefit from such an increase is limitless. One area that can benefit significantly from the combined strength of DL and IoT technologies is Machine Health Monitoring (MHM) Systems. MHM utilizes different analytical approaches and tools to determine the state and health of different components in running machinery leading to end-to-end prognosis. One crucial fact is that features learned by Deep Neural Networks (DNN) are part of a large black box, and there are valuable underlying physical meanings embedded within the features. Hence, there is an exciting research area which explores underlying mechanisms and interpret physical meanings within DNN. In this paper, learning mechanisms are evaluated using different models: stacked autoencoders (SAE), and convolutional neural networks (CNN). Results indicate that the autoencoder networks failed to regenerate discriminative representation of input signals without pre-processing. On the other side, the outputs of both the convolutional and activation layers in CNN showed clear distinction between different classes which led to substantial improvement in classification accuracy.

1. Introduction

The introduction of Industry 4.0 has revolutionized traditional manufacturing paradigm leading to smart manufacturing and digital transformation. In smart manufacturing, physical devices can communicate wirelessly creating Internet of Things (IoT) network. As a result, unprecedented volumes of data are continuously generated leading to a pressing need for an efficient analytical tools to transform the data into insightful information. Prior to the introduction of Industry 4.0, a form of IoT already existed in both discrete manufacturing and process industry plants. ANSI/ISA-95 standard reference model (Fig. 1) shows that the process industry achieves a nearly fully automated production process. However, existing ANSI/ISA-95 based infrastructure lacks the interconnection between lower and high level functionality. Hence, advanced data analytics techniques have been utilized extensively to empower autonomous interconnection between different layers and introduce data where it is needed, when it is needed and in the form it is needed.

Level 4	Business planning management
Level 3	Manufacturing operations management
Level 2	Monitoring, DCS, PLC control
Level 1	Sensors, actuators, robots
Level 0	Physical production process

Figure 1. ANSI/ ISA 95 reference model

Considering Machine Health Monitoring (MHM) as the main case study in this paper, digital transformation has influenced MHM techniques drastically. Machine Health monitoring (MHM) is critical for all manufacturing industries because of its potential in cost reduction, and improving reliability and safety. Traditional MHM systems relied on setting control limits on different sensory data i.e. vibration and temperature readings, and once the measured variables cross those limits the machine is sent for maintenance causing unplanned shutdowns and production delays. It is true that outlier sensory data prevents machine complete damage; however, there is nothing much can be done post failure and usually production interruption can not be avoided. Data driven approaches were utilized to develop prognosis capabilities and predict machine failures well before occurrence (Zhao et al., 2019). Data driven techniques combines handcrafted features engineering with shallow machine learning algorithms to predict the remaining useful life for machines. Logistic regression and support vector machines were amongst wide range of techniques utilized for data mining (Muralidharan & Sugumaran, 2012; Widodo & Yang, 2007); however, manual features extraction requires significant time and expertise and usually performed by data scientists. In manufacturing environment, hiring data scientists for MHM sounds not practical and requires lot of resources. Deep learning techniques has the potential to automate MHM pipeline by building deep networks that extract abstract representation from input data. Current deep learning research emphasis targets the development of end-to-end scheme which automatically learns feature from raw data and predict machines future failures and remaining useful life. One important fact is that features learned by Deep Neural Networks (DNN) are part of a large black box and there are valuable underlying physical meanings embedded within the features (L. Zhang et al., 2019). Hence, there is an interesting research area to explore underlying mechanisms and interpret physical meanings within DNN.

DNNs have proven feature extraction capabilities in fault diagnosis and classification applications (Mohammad & Al-Ani, 2018). Also, it is so important to optimize the DNN structure to obtain a compact and powerful DNN scheme (Mohammad, Rattani, & Derakhshani, 2018). It is usually challenging to explain underlying mechanisms that lead to such outstanding performance because of the difficulty to track sensory data flow through large number of deep layers in presence of nonlinear operators. Heydarzadeh et. al. stated that nonlinear sigmoid mapping complicates the understanding of deep layers of an autoencoder DNN (Heydarzadeh et al., 2019). On the other hand, it was stated that the first hidden layer behaves similar to FIR filter; hence visualization task of the first hidden layer was quite simple. Zhang et. al. attempted to visualize features learned by CNN first layer by plotting filter kernels along with their frequency transformation and the results showed few sinusoidal decompositions (W. Zhang, Peng, Li, Chen, & Zhang, 2017). Another research article also investigated the understating of CNN deep layers in addition the first layer (Jia, Lei, Lu, & Xing, 2018). The results also indicated that the kernels in the first convolutional layer behave like a set of band-pass filters while the kernels in the second convolutional layer represents a set of more complex filters. It was noted that previous literature emphasized on understanding well tuned DNNs which achieves more than 90% classification accuracy. In this paper, learning mechanisms from the Case Western Reserve University (CWRU) bearing dataset are evaluated considering stacked autoencoders (SAE), and convolutional neural networks (CNN). For those networks, classification accuracy varies drastically and the learning mechanisms are compared in view of classification performance.

2. DATASET DESCRIPTION

CWRU bearing dataset was used in this paper (Smith & Randall, 2015). Total number of 12 failure classes were considered as shown in Table 1 and the collected data was sampled at 12 kHz rate. The dataset included different component failures and failures of the same class but with different severity level. For training and feature extraction purpose, a matrix is created combining samples from all datasets. Considering 800 datapoints (rows), the shortest dataset yields 150 observations (columns) and this number is considered for each class regardless of the length of the original dataset. The dataset for each is partitioned between training and testing sets: 100 samples for training and 50 for testing (for each class) which yields 1200 observations for training and 600 observations for testing (for all classes).

Class label	Operating condition	Fault diameter [in]	Orientation	Dataset length	Defect frequency [Hz]
1	Normal	0	-	243200	-
2	Ball fault	0.007	-	122400	132
3		0.014	-	121600	
4		0.021	-	121600	
5	Inner race fault	0.007	-	120800	161.8
6		0.021	-	121600	
7		0.028	-	120800	
8	Outer race fault	0.007	Vertical @ 3	121600	107.7
9		0.007	Center @ 6	121600	
10		0.014	Center @ 6	121600	
11		0.021	Vertical @ 3	121600	
12		0.021	Center @ 6	122400	

Table 1. Bearing dataset description

As shown in table 1, each class represents a separate operating condition. Hence, each time series contains data points corresponding to a unique operating condition and there is no overlap between operating conditions during different periods of the same time series. For this reason, data was proportioned between training and testing for each class such that initial 100 observations are for training and the latter 50 for testing.

3. PROPOSED METHOD

Figure 2 shows the overall of the suggested workflow which consists of three main stages namely, data preparation, 2D convolutional neural network and deep auto-encoder neural network.

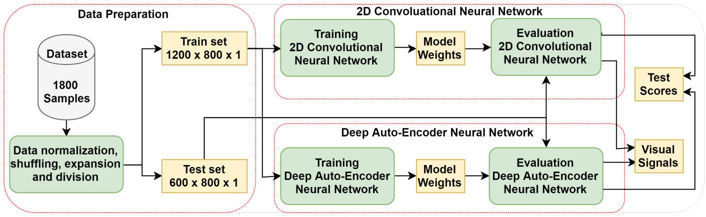
Data Preparation
Dataset
1800 Samples
Data normalization, shuffling, expansion and division
Train set 1200 x 800 x 1
Test set 600 x 800 x 1
2D Convoluational Neural Network
Training 2D Convolutional Neural Network
Model Weights
Evaluation 2D Convolutional Neural Network
Deep Auto-Encoder Neural Network
Training Deep Auto-Encoder Neural Network
Model Weights
Evaluation Deep Auto-Encoder Neural Network
Test Scores
Visual Signals

Figure 2. Visualization workflow

3.1. Data Preparation

Initially, each observation data was normalized using min-max normalization as shown in Equation (1). Then, the data was shuffled, expanded (for 2D CNN only) and divided into two subsets training and testing with 66.6% and 33.3%, respectively.

$$Pnorm_i = \frac{P_i - \min(P)}{\max(P) - \min(P)} \tag{1}$$

3.2. Stacked Autoencoders Neural Network

Autoencoder (AE) is an unsupervised neural network which learns feature using three layers architecture: input, hidden layer (composed from encoder and decoder) and the output layers. The encoder transforms the input $\mathbf{x}$ into hidden representation h using nonlinear mapping:

$$h = f(\mathbf{Wx}\text{+}\mathbf{b}) \tag{2}$$

where f is a nonlinear activation function. By doing so, autoencoder learns abstract representation from the input. The next step is to decode the hidden representation using decoders. A decoder maps the hidden representation back to the original data:

$$z = f(\hat{\mathbf{W}}\mathbf{x} + \hat{\mathbf{b}}) \tag{3}$$

AEs are trained in an unsupervised manner to minimize the reconstruction error between z and x by adjusting the model parameters: $\boldsymbol{\theta} = [\mathbf{W}, \hat{\mathbf{W}}, \mathbf{b}, \hat{\mathbf{b}}]$ Autoencoders can be stacked to learn higher-level representation forming a deep network. Stacking of layers is accomplished by feeding the output of layer n to the input of layer n+1 and the training is done for each layer separately. This step is usually referred as pre-training of a DNN. Pretraining of DNN initializes the model weights and then the supervised training is performed to fine tune the model. Regression layer / softmax layer is usually added at the end of AE model to map the last AE output to the targets. The network (AEs+softmax layer) is then trained in a supervised manner utilizing labeled training data to minimize the model prediction error.

Figure 3 shows the suggested structure of deep auto-encoder which consists of seven layers. The first and last layers are the input and output layers of size $\mathbf{800} \times \mathbf{1}$ (matches the input size). The other five layers (layer 2 through 6) are the fully connected layers (Multi-layer perceptron (MLP)). The MLP layers have sizes of $\mathbf{128}$, $\mathbf{64}$, $\mathbf{32}$, $\mathbf{64}$ and $\mathbf{128}$ which were designed after a large scale evaluation. Besides, adadelta optimizer was used to train this network. This optimizer used a dynamical adaptive algorithm to achieve a stochastic gradient descent. Also, this learning method shows a robustness against the noisy gradient information(Zeiler, 2012).

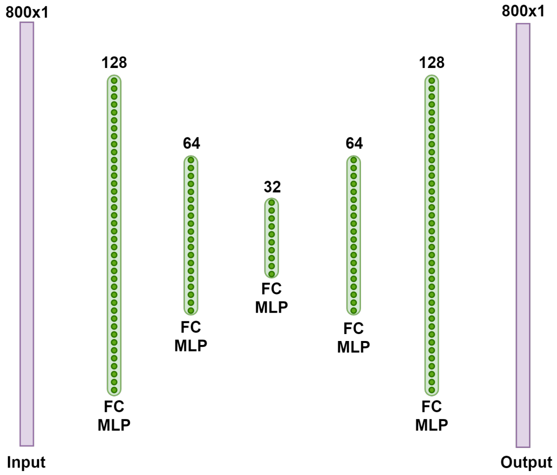

Figure 3. Stack auto-encoder structure

3.3. Convolutional Neural Networks

CNN has more complicated structure than a Multi-Layer Perceptron (MLP). It has convolutioal layers, pooling layers, and full connected layers. The process of CNN forward propagation follows below steps (Li, Hu, Li, & Zheng, 2020)

$$\begin{aligned}
\boldsymbol{z_1} &= \boldsymbol{ReLU(x * \omega_1 + b_1)} \\
\boldsymbol{conv_1} &= \boldsymbol{pool(z_1)} \\
&\vdots \\
\boldsymbol{z_i} &= \boldsymbol{ReLU(conv_{i-1} * \omega_i + b_i)} \\
\boldsymbol{conv_i} &= \boldsymbol{pool(z_i)} \\
\boldsymbol{h_1} &= \boldsymbol{ReLU(conv_i * \omega_{conv_1} + b_{conv_1})} \\
\boldsymbol{y} &= \boldsymbol{softmax(\omega_{conv_2} * h_1 + b_{conv_2})}
\end{aligned} \tag{4}$$

Where $\boldsymbol{x}, \boldsymbol{y}$ are the input and output vectors, respectively. $\boldsymbol{ReLU}$ is the rectified linear unit activation function.$\boldsymbol{z_i}$ is the $\boldsymbol{i^{th}}$ convolutional layer matrix. $*$ is the convolutional operation. $\boldsymbol{pool}$ refers to the pooling operation. $\boldsymbol{\omega}$ and $\boldsymbol{b_i}$ are the weight matrix and bias vectors. $\boldsymbol{conv_i}$ is the convolutional layer after applying the pooling operation. $\boldsymbol{h_i}$ is a hidden layer. In each training iteration, the back propagation algorithm minimizes the loss function and updates both $\boldsymbol{\omega}$ and $\boldsymbol{b_i}$.

Figure 4 shows the proposed structure for the CNN. The first layer is the input layer which receives the input signal that contains 800 data points. The first convolutional layer is composed of 2D convolutional layer (layer 2) and ReLU activation layer (layer 3) and it applies 64 filters on each input signal. The max pooling layer (layer 4) basically squeezes the data from previous layers by calculating the largest value with a window size of 50. The second and third convolutional layers run 16 filters and are composed of layers 5,6,7 and 8.

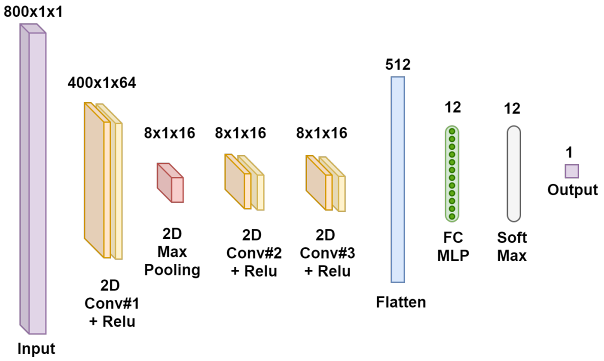

Figure 4. CNN structure

4. RESULTS AND DISCUSSION

4.1. Auto-encoder evaluation results

Figure 5 shows an AUC of 0.7574 and EER of 0.3003 (30.03) were obtained. AE performance was evaluated in the literature considering different model structures and it was shown that data preprocessing is required to boost the AE classification accuracy (Alabsi, Liao, & Nabulsi, 2020; Hou, Wen, & Dong, 2017; Haidong, Hongkai, Xingqiu, & Shuaipeng, 2018). However, the scope of this paper is not focused on improving classification accuracy but rather understanding and evaluation of learning mechanisms. Hence, the classifier is evaluated without further data preprocessing and it will be considered for future investigation.

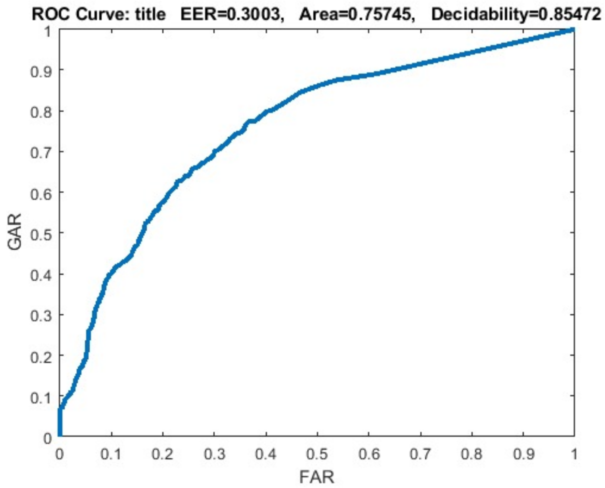

Figure 5. Receiver Operating Curve of autoencoder classifier

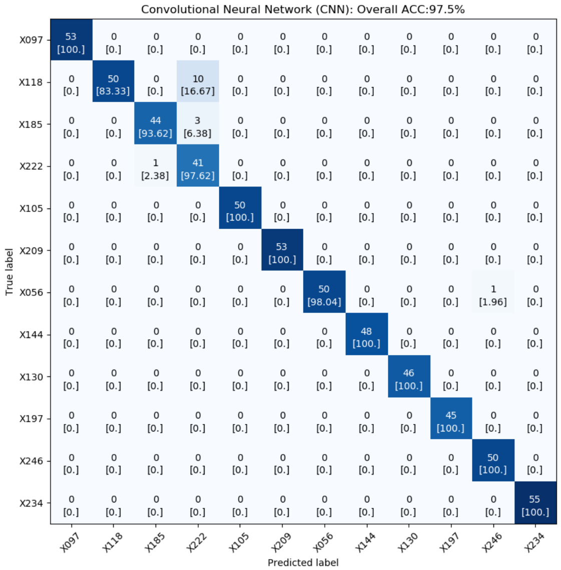

Figure 6. CNN classifier confusion matrix

4.2. CNN evaluation results

Figures 6 and 7 present the CNN-based classifier performance using confusion matrix and ROC curves for each class. The overall accuracy is 97.5% and almost all classes obtained 100% except for classes 2,3 and 4. Those three classes correspond to same fault (ball fault) with different severity, it is clear from Table 1 that the severity level is very close which led to similarities in the fault signatures between those classes.

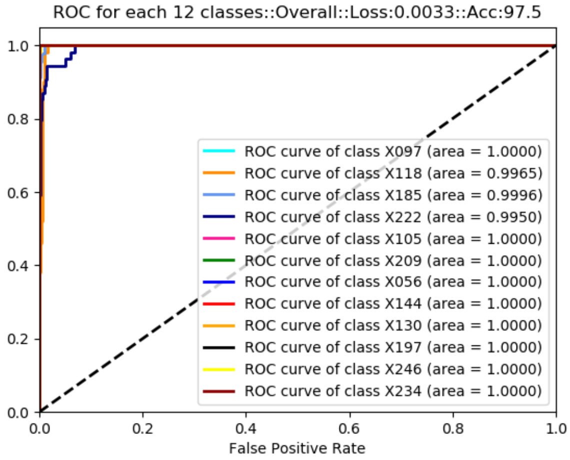

Figure 7. ROC curves for each class

4.3. Autoencoder visualization results

Neural Network (NN) models are usually treated as black box. One way to understand the feature extraction process is through quantitative analysis of weight matrices (kernels in case of CNNs) (Jia et al., 2018; Wang et al., 2018). However, the novelty of this paper is to study activations and emphasize the importance of studying the reconstructed signal and relate that to classification accuracy. Figure 8 shows the output signals for the first two layers of the proposed auto-encoder neural network. The similarity between classes is so high and this is due to the fact that MLP-01 and MLP-02 outputs is showing the activation. Hence, if the signal is corrupted with noise or if the SNR is low the network fails to learn key features from different classes.

4.4. CNN visualization results

Classification accuracy is studied in view of layers 2 and 3 visualization. Figure 9 shows one sample from each class (left column), the average signal visualization after applying 64 filters in the second layer (middle column), and the third layer activation (right column). It is clear that signals that belong to classes 5,6,8,9 and 12 show clear fault impulses while it is hard to infer interpretation from other signals. The second layer applies 64 filters and the average signal is presented in the middle, it is clear that an impulsive patterns are noticed in classes 2,3,4 and 7 which were not noticed prior to applying filtering in the second layer. After applying the ReLU activation, it can be seen that only impulses are present in all signals with minimal noise and those signatures represent key features extracted from layers 2 and 3. However, activations from classes 2,3 and 4 do not show clear impulses compared to other classes and closer look to deeper layer is needed.

Figures 10 and 11 show visualization of the CNN 8 layers considering classes 2 and 12, respectively. As mentioned before, layers 2 and 3 apply filters and calculate activations however, the max pooling layer with a window size of 50 reduces the input dimension from 400 data point into 8. It is clear that the size reduction in class 12 retained the impulses characteristics after max pooling and all subsequent layers. However, the pattern looks more complicated for class 2 and the max pooling did not retain major characteristics. Hence, the reduced classification accuracy for class 2 may be influenced by the max pooling window size. Decreasing the window size gradually would preserve the content of the original signal and lead to better classification accuracy.

5. CONCLUSION

This paper presented learning mechanisms of different NN models. The results indicated that the autoencoder networks failed to regenerate discriminative representation of input signals without pre-processing. Hence, the poor reconstruction performance by the autoencoder network resulted in degraded classification accuracy (Li, Zhang, & Ding, 2019). On the other side, the outputs of both the convolutional and activation layers in CNN showed clear distinction between different classes which led to substantial improvement in classification accuracy. Care should be taken while choosing the max pooling layer dimensions since it was shown that significant reduction might result in loosing key features in different classes. The approach described herein provides a direct approach to investigate NN classification accuracy using MLPs, convolutional and activation layers outputs. For future work, a cascaded neural network will be implemented for multiple deep auto-encoder and convolutional neural networks to achieve a robust scheme for unlabeled data with powerful classification and visualization performance.

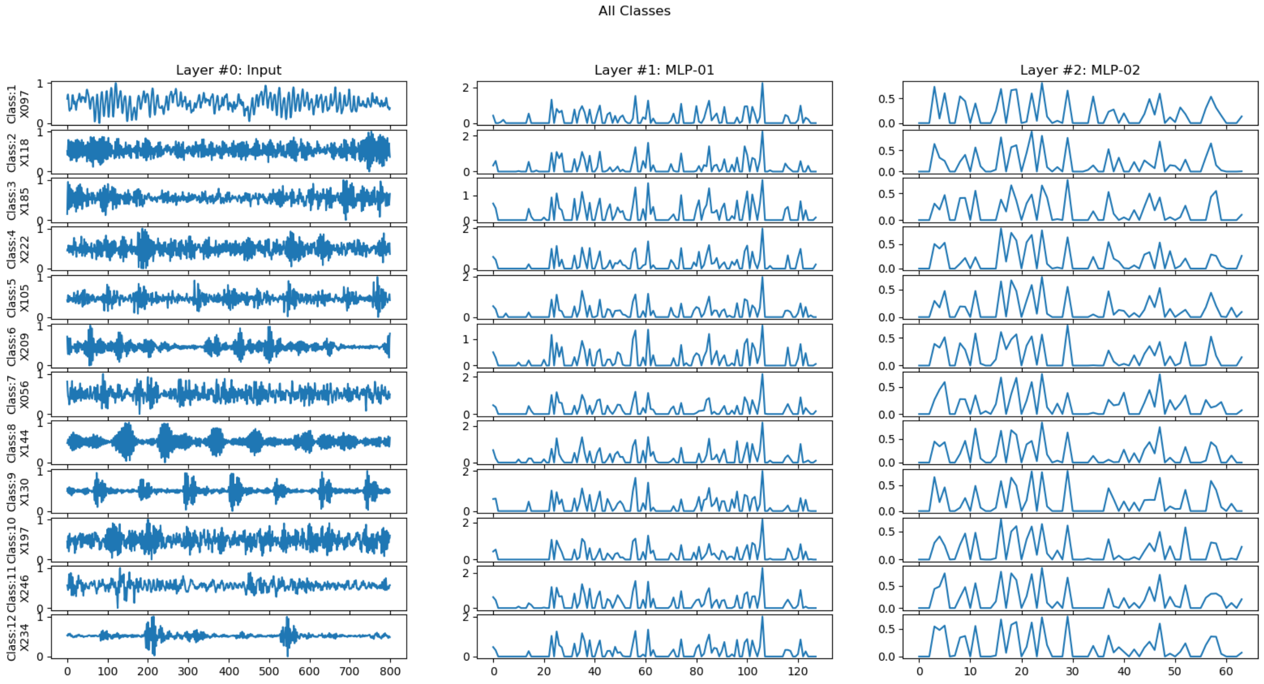

Figure 8. Auto-Encoder visualization for first and second layer

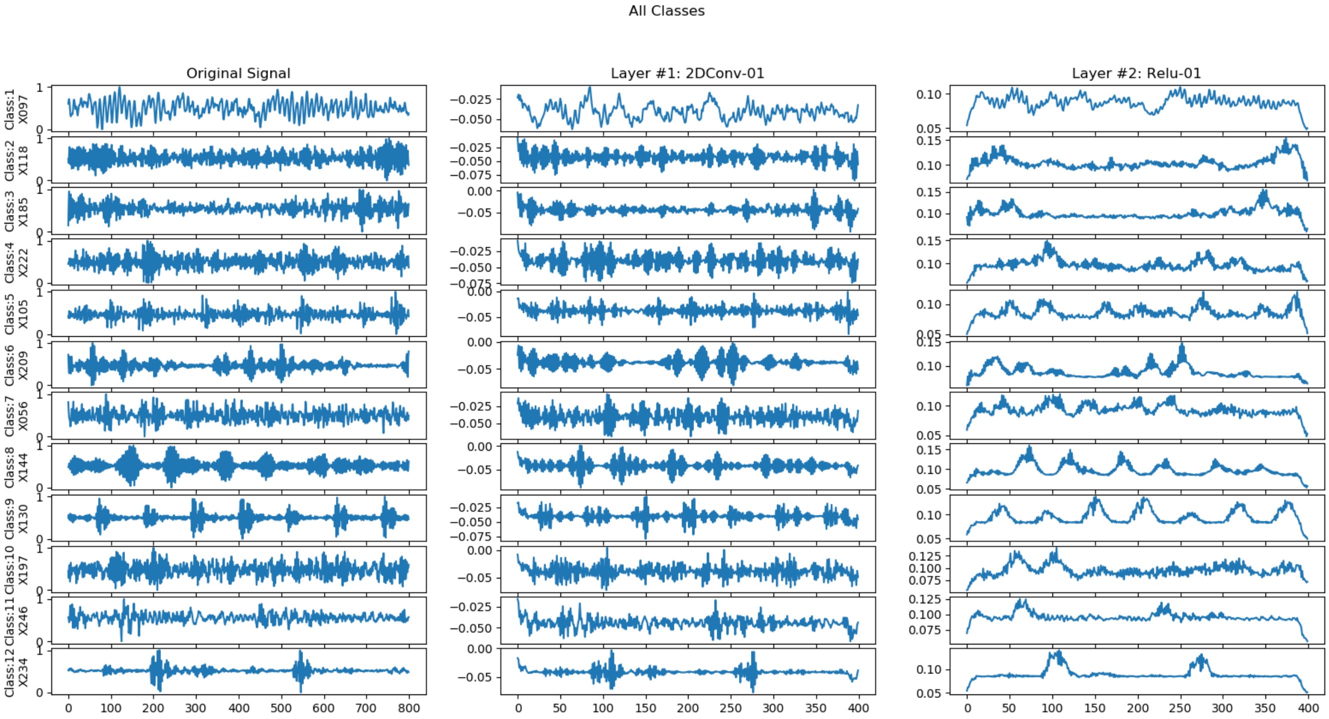

Figure 9. Convolutional Neural Network visualization for first and second layer

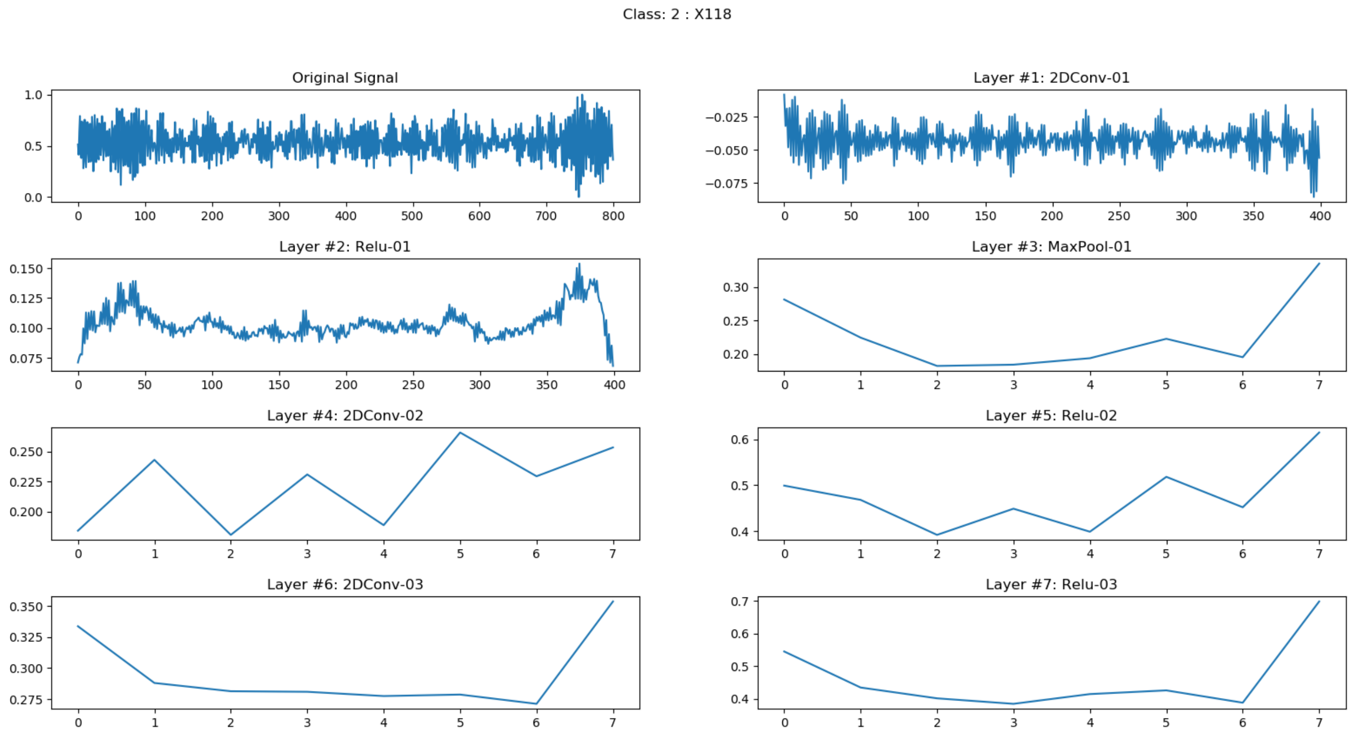

Figure 10. CNN layers visualization for class 2

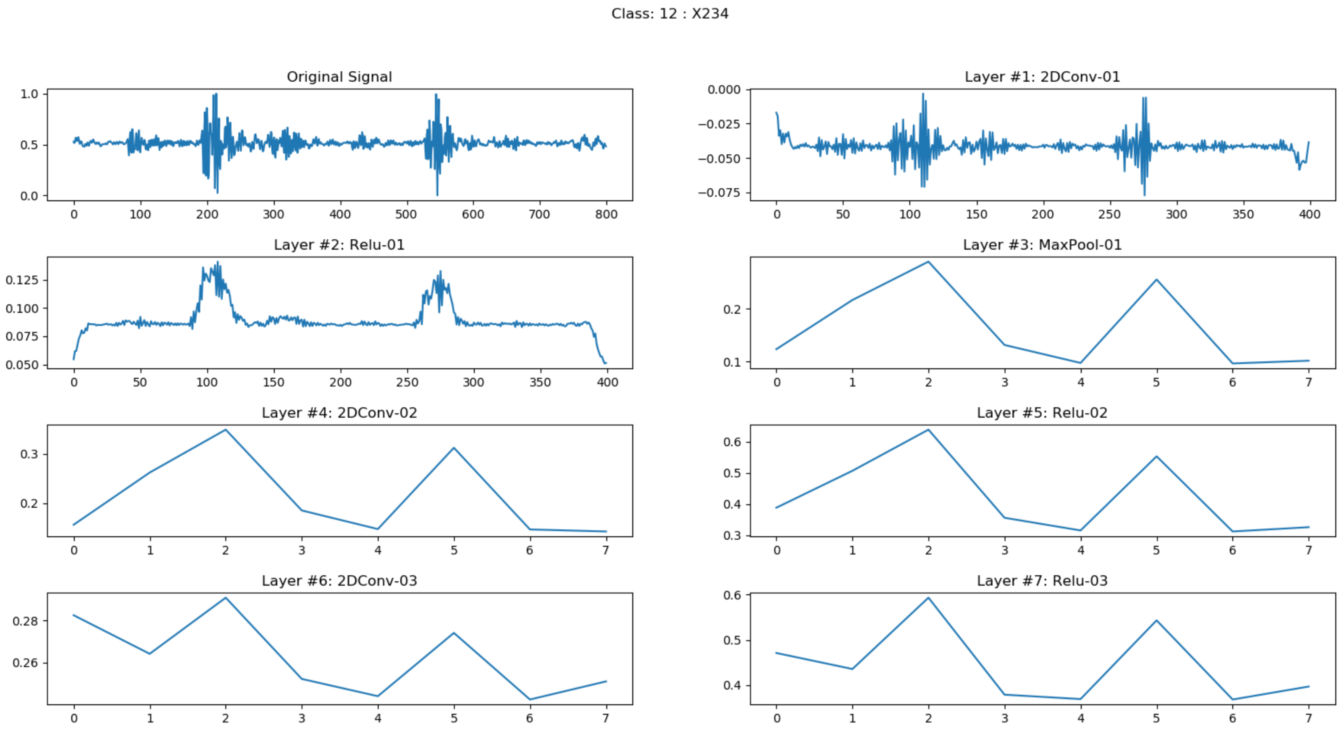

Figure 11. CNN layers visualization for class 12

REFERENCES

Alabsi, M., Liao, Y., & Nabulsi, A.-A. (2020). Bearing fault diagnosis using deep learning techniques coupled with handcrafted feature extraction: A comparative study. *Journal of Vibration and Control*. doi: 10.1177/1077546320929141

Haidong, S., Hongkai, J., Xingqiu, L., & Shuaipeng, W. (2018). Intelligent fault diagnosis of rolling bearing using deep wavelet auto-encoder with extreme learning machine. *Knowledge-Based Systems*, *140*, 1–14.

Heydarzadeh, M., Mohammadi, A., Hedayati Kia, S., Nourani, M., Henao, H., & Capolino, G.-A. (2019). Feature learning using deep neural networks for fault diagnosis in electromechanical systems. In *Asme 2019 dynamic systems and control conference.*

Hou, P., Wen, C., & Dong, D. (2017). Rolling bearing fault diagnose based on stacked sparse auto encoder. In *2017 36th chinese control conference (ccc)* (pp. 7027–7032).

Jia, F., Lei, Y., Lu, N., & Xing, S. (2018). Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. *Mechanical Systems and Signal Processing*, *110*, 349–367.

Li, X., Hu, Y., Li, M., & Zheng, J. (2020). Fault diagnostics between different type of components: A transfer learning approach. *Applied Soft Computing*, *86*, 105950.

Li, X., Zhang, W., & Ding, Q. (2019). Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism. *Signal Processing*, *161*, 136–154.

Mohammad, A. S., & Al-Ani, J. A. (2018). Convolutional neural network for ethnicity classification using ocular region in mobile environment. In *2018 10th computer science and electronic engineering (ceec)* (pp. 293–298).

Mohammad, A. S., Rattani, A., & Derakhshani, R. (2018). Comparison of squeezed convolutional neural network models for eyeglasses detection in mobile environment. *Journal of Computing Sciences in Colleges*, *33*(5), 136–144.

Muralidharan, V., & Sugumaran, V. (2012). A comparative study of naïve bayes classifier and bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis. *Applied Soft Computing*, *12*(8), 2023–2029.

Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical Systems and Signal Processing*, *64*, 100–131.

Wang, J., Li, S., Han, B., An, Z., Xin, Y., Qian, W., & Wu, Q. (2018). Construction of a batch-normalized autoencoder network and its application in mechanical intelligent fault diagnosis. *Measurement Science and Technology*, *30*(1), 015106.

Widodo, A., & Yang, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, *21*(6), 2560–2574.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701.*

Zhang, L., Lin, J., Liu, B., Zhang, Z., Yan, X., & Wei, M. (2019). A review on deep learning applications in prognostics and health management. *IEEE Access*, *7*, 162415–162438.

Zhang, W., Peng, G., Li, C., Chen, Y., & Zhang, Z. (2017). A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, *17*(2), 425.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, *115*, 213–237.