Gearbox Degradation Prediction Through Deep CNN and Bayesian Optimization

Kai Shen[1], Haoyu Wang[2], Yuwei Liao[3], Dev Kakde[4], Arin Chaudhuri[5] and Zohreh Asgharzadeh[6]

[1,2,3,4,5,6] *IoT division, SAS Institute, USA, 27513*
kai.shen@sas.com

1. Data Preprocessing

1.1. Downsampling

The original dataset, recorded at a high sampling rate of 20,480 Hz, presents a considerable volume of data. To facilitate efficient processing and analysis, we implement a data reduction strategy, which involves downsampling the signal to a more manageable frequency of 2,048 Hz.

1.2. Short-Time Parametric Power Spectral Density (ST-PPSD)

Upon downsampling, we employ the Short-Time Parametric Power Spectral Density (ST-PPSD) technique. This method divides the vibration data into short, overlapping segments with a 50% overlap. For each segment, ST-PPSD applies a parametric model, specifically the Yule-Walker autoregression model, to estimate the underlying signal characteristics.

In this approach, the assumption is that the signal is generated by a known underlying model. Consequently, parameters of this model are estimated for each short segment, allowing for a more precise characterization of the signal's frequency spectral curve.

The frequency spectral curves obtained from ST-PPSD are subsequently concatenated column by column, forming a 2D spectral heatmap. This heatmap provides insights into how the vibration data evolves in the frequency domain over time. For the purpose of feature extraction, we calculate the average of these spectral curves over a defined time period.

There are two primary reasons for averaging the spectral curves:

- Through observation, we have noted that the pattern within the two-dimensional heatmap remains relatively consistent over time. Averaging the spectral curves enables us to capture the signal's stable features while simplifying the subsequent analysis.
- Averaging also has the potential to mitigate the impact of noise in the spectral density. By smoothing out variations introduced by noise, we obtain a more stable and robust representation of the underlying signal. This, in turn, enhances the accuracy and reliability of our analysis.

Following the averaging of ST-PPSD results, we are left with 2,049 features (representing magnitude in the frequency domain) for each of the x, y, and z coordinates.

1.3. Autoencoder for further dimension reduction

With the 2,049 features obtained from the previous step, we further use autoencoder to achieve additional dimension reduction in our feature set for gear degradation prediction.

The autoencoder architecture consists of two main components: an encoder and a decoder.

- Encoder: The encoder takes the original 2,049 features as input and compresses them into a lower-dimensional latent feature vector.
- Decoder: The decoder then takes the compressed latent feature vector and attempts to reconstruct the original 2,049 features from it.

Once the autoencoder is trained, we extract the latent feature vector from the encoder. This vector represents the most salient information from the original 2,049 features while reducing dimensionality to 512 features. While reducing dimensionality, the autoencoder focuses on the most informative aspects of the data, which can also help suppress some of the noise presented in the original 2,049 features obtained from ST-PPSD.

2. Modeling with 1D CNN network

In this section, we delve into our 1D Convolutional Neural Network (1D CNN) model. Our focus lies on the intricacies of our network structure.

Our 1D CNN model (as shown in figure 1) features several convolution blocks, each consisting of a convolutional layer, max-pooling layers for downsampling, batch normalization

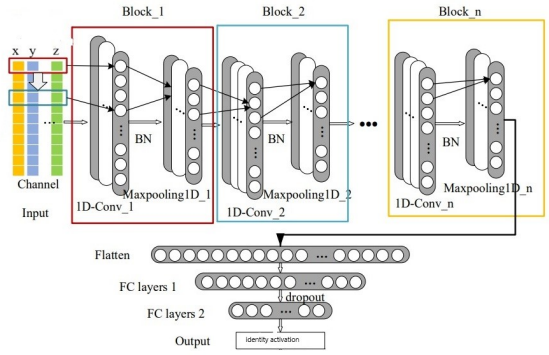

Figure 1. 1d CNN structure

for regularization, and activation functions for introducing non-linearity. The output of the activation function serves as input for the subsequent convolution block.

After the convolutional layers, we flatten the output vectors and introduce fully connected layers. To enhance the model's comprehension of the data, we augment it with operational condition features, specifically speed and torque, which are concatenated with the fully connected layers. Dropout layers are placed between these fully connected layers to mitigate overfitting.

For the output layer, we initially employed the softmax activation function to generate probabilities for different classes. However, we observed challenges when dealing with unseen classes. To address this issue, we used identity activation function and adopted Mean Squared Error (MSE) loss, essentially treating the problem as regression. This modification enables the model to extrapolate to unseen classes to some extent.

To generate class probabilities, we devised a method based on the distance between the rounded-up and rounded-down integers of the raw continuous output. This approach assigns probabilities to the two classes nearest to the raw output, setting others to 0.

3. HYPERPARAMETER TUNING WITH BAYESIAN OPTIMIZATION (BO)

As demonstrated in the previous section, the architecture of a 1D CNN can become quite intricate, involving several crucial hyperparameters. The success of utilizing a 1D CNN for gear degradation level prediction depends on identifying the appropriate network architecture and hyperparameters. To tackle this challenge, we turn to hyperparameter optimization techniques, with a specific focus on Bayesian Optimization (Li, Chen, Huang, & Qu, 2022).

Hyperparameter optimization typically involves methods like grid search and random search, which systematically explore the hyperparameter space. However, BO offers several distinct advantages over these traditional methods:

- Better Results: BO often yields superior hyperparameters when compared to grid or random search. It utilizes probabilistic models to pinpoint promising regions in the hyperparameter space, resulting in improved optimization outcomes.
- Resource Saving: BO's efficient exploration of the space conserves computational resources. This is particularly beneficial in scenarios where objective function evaluations are computationally expensive or time-consuming.

In our application, we leverage BO to explore a total of eight hyperparameters. These hyperparameters encompass elements related to the network structure and training, such as the number of convolution blocks, the number of filters, kernel sizes of the filters, activation functions, and dropout ratios in the fully connected layer. The parameter names, their respective search ranges, and the selected value by BO are outlined in the table 1. The search range for each hyperparameter is determined by considering commonly used values and reasonable bounds. For surrogate model and acquisition function, we used Gaussian process and Expected Improvement. These are commonly used choices within the BO framework.

Hyperparameter Name	Range (select value)
No. of Conv_Block	1 to 5 (4)
No. of filters	8, 16, 32 (16)
Kernel Size	3 to 15 (3)
Padding type	Same, Valid (same)
Activation Function	ReLU, Tanh, Leaky ReLU (ReLu)
No. of FC layers	2 to 5 (3)
No. of nodes in FC layers	64 to 512 (step 32) (64)
Dropout ratio	0, 0.1, 0.2, 0.3, 0.4, 0.5 (0.1)

Table 1. Hyperparameter Configuration and Selection by Bayesian Optimization (BO).

4. CONCLUSION

Our approach combines ST-PPSD and Autoencoder for efficient data preprocessing and feature extraction. Then we used an optimized 1D CNN model to solve the complex gear degradation level classification problem. Bayesian Optimization is employed for hyperparameter tuning, by which the model's performance is further enhanced.

REFERENCES

Li, J., Chen, R., Huang, X., & Qu, Y. (2022). Development of deep residual neural networks for gear pitting fault diagnosis using bayesian optimization. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–15.