Proceedings of the 7th National Symposium on Rotor Dynamics
NSRD 2024
December 18 - 20, 2024, SRM Institute of Science and Technology, Tamil Nadu, India

NSRD2024-024

BAYESIAN OPTIMIZED CONVOLUTIONAL NEURAL NETWORKS FOR CRACK DETECTION IN SHAFTS.

Harmandeep Singh[a]
Undergraduate Student
Dept. of Mechanical Engineering
Thapar Institute of Engineering and Technology,
Punjab, India
Email: hsingh1_be21@thapar.edu

Sudhar Rajagopalan[b]
Research Scholar
Dept. of Mechanical Engineering
Thapar Institute of Engineering and Technology,
Punjab, India
Email: talktosudhar@gmail.com

Ashish Purohit[c]
Associate Professor
Dept. of Mechanical Engineering
Thapar Institute of Engineering and Technology,
Punjab, India
Email: ashish.purohit@thapar.edu

ABSTRACT

Purpose: *Cracks can lead to catastrophic failures if left unchecked since they propagate over time due to various stresses, condition monitoring can optimize the life of the machinery and avoid unplanned downtime as well as run-till-failure maintenance to minimize costs. The paper aims to study the implementation of Bayes Hyperparameter Optimization in Convolutional Neural Networks for Crack Detection in Shafts via time domain signals acquired from an experimental setup.*

Methods*: Bayes Optimization, which employs Gaussian processes, is used to optimize 'black-box' functions, which are defined as the performance metric of the model for a given combination of hyperparameters in this paper. Hyperparameter tuning is employed to achieve an optimized model for classifying the severity of the crack from the time domain signal without using any feature extraction methods. Hyperparameters such as filter size, kernel size, and pool size for a single convolutional layer, along with dropout are optimized from a given search space in minimum number of iterations, greatly reducing the time and effort required for finding the ideal hyperparameters as compared to traditional manual search. Comparisons have been drawn with Artificial Neural Networks, Support Vector Machines, and K-Nearest Neighbour, with the features being energies of frequency bands extracted using the Wavelet Packet Transform.*

The data is acquired at steady state conditions at 4 depths of transverse cracks, which are 0%, 20%, 30%, and 40% of the shaft's diameter, data was collected for each condition at two rotational speeds which are 1500 rev/min and 2000 rev/min over a period of 70 seconds. To prove the model's effectiveness, Gaussian Noise was added to the data, even with noisy data, the model performed effectively

Keywords**:** ***Rotor Crack, Bayesian Optimization, Wavelet Packet Transform, Deep Learning, Machine Learning, Vibration Analysis, Predictive Maintenance, Convolution.***

1 INTRODUCTION

Rapidly varying bending and torsional stresses as well as thermal stresses lead to fatigue cracks in shafts which are heavily used in industrial compressors, gas turbines, pumps, etc. Failure in timely detection may lead to catastrophic failure and cause injuries to the nearby workforce, avoidable downtime, and huge costs in damaged equipment when complete fracture occurs. Classifying the severity of the crack may extend the life of the machinery and help in planning the maintenance before complete failure occurs.

[1] conducted an in depth analytical and experimental study of cracked Jeffcott rotors and compared their frequency spectrums while at 1/5, 1/3 and ½ critical speed (subcritical resonances). [2] investigated the Peak Position Component Method on Power Spectral Density of an experimental setup with crack and further used Artificial Neural Networks for diagnosis of the shaft. [3] found the use of the Daubechies wavelet to be effective in detecting cracks in shafts in transient conditions. [4] employed the Morlet wavelet to detect cracks in shafts by subjecting them to transient torsional vibrations and studying its lateral vibrations. [5] studied the use of Continous Wavelet Transform for crack detection in shafts at transient conditions and used Artificial Neural Networks to determine their depth and location. [6] employed Wavelet Packet Transform using Daubechies wavelet to determine the 3x energy of the signal and used probability of detection (POD) curves to determine the relation between the 3x energy and crack depth. [7] used ANNs to classify the depth of the crack using energies extracted from WPT. [8] conducted an experimental study and proposed the combined use of Discrete Wavelet Transform, Continuous Wavelet Transform, Wavelet Packet Transform and power spectral density to determine the presence of a crack under various operating conditions. [9] determined the crack depth ratio and eccentricity angle using an inverse method in conjunction with ANNs. [10] developed a physics informed ML approach and classified the fault type and its location via an FEM model. [11] determined the dimensionless crack depth and its location in an analytical study using Radial Basis Function (RBF) Neural Networks. [12] created a Finite Element Model of an experimental setup which was then used to generate labelled training data. The data is then used in a SVM Classifier with RBF Kernel and compared with the data from the experimental setup with an induced crack in a carbon fiber reinforced polymer rotating shaft via a transmittance function. [13] proposed convolutional neural networks and deep metric learning to identify the position of a crack in a dual disk hollow shaft rotor system. [14] employed CNNs and optimized the parameters of the model using genetic algorithm to diagnose cracks and shaft misalignments in noisy conditions, The CNNs were used for feature extraction after the signals were denoised through variational mode decomposition. [15] proposed Bayes Optimized CNNs with LSTM for fault diagnosis in hydro-turbines using acoustic vibration signals.

It is observed that there is a gap in literature where Bayesian Optimization is used in CNNs with dropout on data from proximity probes. The study aims to address the gap and draws comparison by studying the use of WPT energies as features in ML algorithms for crack detection in shafts.

2 METHODOLOGY

Section 2.1 discusses the implementation of Bayesian Optimization on Convolutional Neural Networks. Convolutional Neural Networks are proven to be effective at signal classification as studied by [16] in domains such as structural health monitoring and vibration monitoring. Their ability to extract features and detect anomalies in vibration data of bearings has been studied by researchers [17-19]. The accuracy of classification is however determined by the choice of hyperparameters, which vary as per use case and optimizing them has been studied in great detail by [20]. Bayesian Optimization is more efficient at optimizing the hyperparameters as compared to Random Search and Grid Search since it reaches the global maxima in less iterations as the next sampling point is chosen based on the prior information. The data obtained from a proximity probe on one axis was chosen for the study, a single layer of CNN was used and it’s hyperparameters were optimized for supervised classification of crack severity.

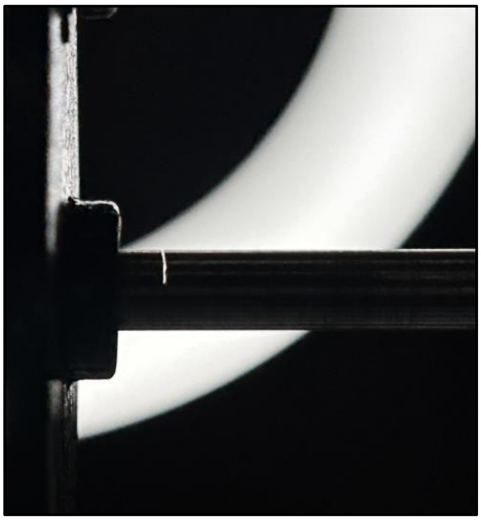

Fig. 1: Transverse slit seeded at the center of the shaft.

Section 2.2 discusses the use of WPT to extract energies from the vibration data, it has been observed that the 3x energy of the shaft has been a reliable parameter in determining the presence of crack [6]. For 25 Hz and 33 Hz, energies corresponding to 1× till 3× were chosen as features for the ML models. No hyperparameter optimization was performed on the ML algorithms and their values are stated in Section 4.2.

2.1 BAYESIAN OPTIMIZED CONVOLUTIONAL NEURAL NETWORKS

1D Convolutional Neural Networks have proven effectiveness in the domain of Structural Health Monitoring because of their ability to capture patterns in the signal as stated in literature. For a convolutional layer with kernel $\boldsymbol{j}$, where $\boldsymbol{j} = [j_1, j_2 \dots . j_w]$ of size w and discrete signal $\boldsymbol{x}$, which has been defined as $x_i = [x_i, x_{i+1} \dots x_q]$. q is the range of the signal considered for convolution and is shown in Equation 1, the resultant y is obtained on rolling the kernel across the signal as stated in equation. For the study, the stride has been set as 1 which corresponds to the kernel moving across the signal one sample at a time. l is the length of the signal.

$$\begin{gathered} y_1 = \sum_{i=1}^{w} \boldsymbol{x_i j} \\ y_2 = \sum_{i=2}^{w+1} \boldsymbol{x_i j} \\ \cdot \\ \cdot \\ \cdot \\ y_k = \sum_{i=l-w}^{l} \boldsymbol{x_i j} \end{gathered} \tag{1}$$

For n such kernels, the dimension of the data after Convolution will be $n \times k$. Max Pooling is then used to extract the maximum value from the Convoluted data with a window size p.

$$l_1 = max\{y_1, y_2, y_3 \dots . y_p\}$$

$$l_2 = max\{y_2, y_2, y_3 \dots . y_{p+1}\} \tag{2}$$

$$\vdots$$

$$l_m = max\{y_{k-p}, y_{k-p+1}, y_{k-p+2} \dots . y_k\}$$

Corresponding to *n* kernels, after Max Pooling, the dimension of the data will be *n*×*m*. Once the maximum values are extracted, the data is reduced to a single dimensional via a Flattening Layer and forwarded to a Fully Connected Dense Neural Layer with 128 nodes which use the ReLu activation function stated in Eq. 3.

$$f(u) = \max(0, u) \tag{3}$$

A dropout layer between Dense Layer 1 and Dense Layer 2 is employed in order to prevent the data from overfitting **[21]**. It randomly deactivates the nodes of Layer 1, effectively deactivating the weights connecting nodes between Layer 1 and 2.

The Dense layer is further connected to another fully connected Dense Layer with 4 units correlated to Healthy state, and the three other faulty states. The Softmax activation function is used because of its ability to state the probability of each state in a multi-class classification problem. Since 4 fault states are considered for this study, the resultant output will be a vector of dimension 1×4 where each element will represent the probability of the signal from being from one of the four states.

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{4} e^{x_j}} \tag{4}$$

For updating the weights and biases of the neural network, the Adam optimizer was used with a learning rate of 0.001

A schematic of a Conv1D followed by max pooling and flattening layers has been depicted in Fig. 1.

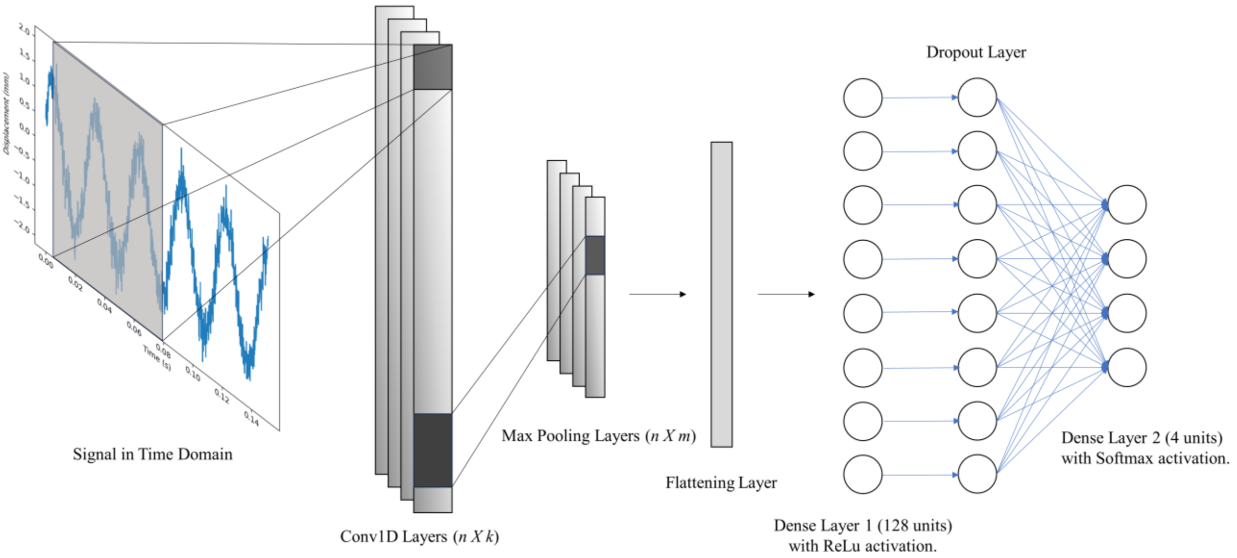

Fig. 2: 1D CNN being applied on noisy data from the Proximity Sensor

Bayesian Optimization is helpful in finding the extrema of functions that are expensive to calculate, since the objective function is not known, traditional optimization techniques such as gradient descent cannot be applied. It is derived from the Bayes Theorem which deduces the optima of the function based on the posterior information combined with an acquisition function. The posterior information being provided by randomly sampling the function to obtain sample information and the prior distribution assumed to be a Gaussian Process. Since the search space is multi-dimensional and depends on numerous parameters, The methodology is explained for a One-Dimensional case [22].

A Gaussian process is denoted by:

$$f(x) \sim GP\big(m(x), k(x, x')\big) \tag{4}$$

The mean *m(x)* is assumed to be 0 and the exponential square function is used as the covariance function $k(x, x')$, represented by:

$$k(x_i, x_j) = \exp\left(-0.5\|x_i - x_j\|^2\right) \tag{5}$$

Where x_i and x_j represent the i^{th} and j^{th} sample respectively, as the values approach closer to each other, a strong correlation is observed, which weakens as the distance between the samples increases.

The distribution of $f(x_{t+1})$ is determined by calculating the covariance matrix of the observation samples till $\boldsymbol{f}_{\boldsymbol{1}:\boldsymbol{t}}$

$$\boldsymbol{K} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_t) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_t) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_t, x_1) & k(x_t, x_1) & \cdots & k(x_t, x_t) \end{bmatrix} \tag{6}$$

Gaussian Process assumes that samples $\boldsymbol{f}_{\boldsymbol{1}:\boldsymbol{t}+\boldsymbol{1}}$ will follow a t+1 dimensional normal distribution

$$\begin{bmatrix} \boldsymbol{f}_{\boldsymbol{1}:\boldsymbol{t}} \\ f(x_{t+1}) \end{bmatrix} \sim N\left(0, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{k} \\ \boldsymbol{k}^{\boldsymbol{T}} & k(x_{t+1}, x_{t+1}) \end{bmatrix}\right) \tag{7}$$

Where $\boldsymbol{f}_{\boldsymbol{1}:\boldsymbol{t}} = [f(x_1), f(x_2), f(x_3) \ldots\ldots f(x_t)]$ and $\boldsymbol{k} = [k(x_{t+1}, x_1), k(x_{t+1}, x_2) \ldots k(x_{t+1}, x_3)]$**.** $f(x_{t+1})$ follows a one-dimensional normal distribution with mean and variance represented as.

$$\begin{gathered} f(x_{t+1}) \sim N(\mu_{t+1}, \sigma_{t+1}^2) \\ \mu_{t+1}(x_{t+1}) = \boldsymbol{k}^{\boldsymbol{T}}\boldsymbol{K}^{-\boldsymbol{1}}\boldsymbol{f}_{\boldsymbol{1}:\boldsymbol{t}} \\ \sigma_{t+1}^2(x_{t+1}) = \boldsymbol{k}^{\boldsymbol{T}}\boldsymbol{K}^{-\boldsymbol{1}}\boldsymbol{k} + k(x_{t+1}, x_{t+1}) \end{gathered} \tag{8}$$

On obtaining the posterior distribution of the function $f(x)$, an acquisition function is employed to determine the maxima of function *f(x)*. The Gaussian Process Upper Confidence Bound (GP-UCB) is used in this study which is capable of finding the global maxima unlike the Probability of Improvement (PI) acquisition function which finds the local maxima.

The GP-UCB is represented as

$$UCB(x) = \mu(x) + \alpha\sigma(x) \tag{9}$$

The GP-UCB determines whether the next sampling point should be in a zone of high confidence, corresponding to higher value of $\mu(x)$ or in a zone of low confidence corresponding to a zone of high $\sigma(x)$. The parameter α is used to bias the search and its value has been set to 2.6.

For the study, validation accuracy was chosen as the argument to be maximized, in order to increase the accuracy of the model on data which was not used in the training dataset. [22] has been employed to implement Bayesian Optimization on the model, and to log the validation accuracy of each hyperparameter set.

Due to an increase in computational time and resource for an increase in the number of Convolutional layers, this study has been conducted on a single layer and a limited search space, and the results obtained are concluded to be the most optimal for the given search space. The hyperparameter search space has been described in Table 1.

Table 1: Hyperparameter Search Space

Parameters	**Range**	**Steps**
Kernel Size (*w*)	8 to 250	2
Number of filters (*n*)	2 to 250	2
Max Pooling size (*p*)	2 to 50	2
Dropout	0 to 0.3	0.1

The data obtained from the proximity probes is converted to displacement in millimeters by dividing the data with the scaling factor of the proximity probes as stated in Section 3. A signal to noise ratio (SNR) of 7 dB is implemented in the original signal to show the robustness of the model. Gaussian noise is added to the signal with the Root Mean Square (RMS) value calculated from Eq. 10. When no noise is added to the signal, the SNR in those cases has been stated as 0 dB.

$$SNR_{dB} = 10\, log_{10} \left(\frac{RMS_{signal}^2}{RMS_{noise}^2} \right) \tag{10}$$

2.2 WAVELET PACKET TRANSFORM

A limitation of the Fourier Transform is it's inability to represent changes in frequency in the time domain and solely represents the frequencies present in the signal over the entire time interval, The Short term Fourier Transform provides the frequency information of a window, the drawback being the width of the window determines the resolution available in time as well as frequency whereas the Wavelet transform represents the signal in time as well as frequency domain.
The WPT is an extension of DWT, which follows dyadic discretization in CWT for the scale and translation parameters.
For a discrete signal *x[k]*, DWT transforms it into detail coefficients via a high pass filter *h* which is associated with the scaling parameter and into approximation coefficients through a low pass filter *g* which is associated with the mother wavelet function. The coefficients obtained through these filters are downsampled by a factor 2. The WPT Decomposition decomposes the Approximation as well as the Detail coefficients unlike DWT, which results in a finer control of coefficients of all available frequencies in the signal till the desired decomposition level.

$$A[n] = \sum_{k=-\infty}^{\infty} x[k] g[2n-k] \tag{11}$$

$$D[n] = \sum_{k=-\infty}^{\infty} x[k] h[2n-k] \tag{12}$$

The wavelet packet decomposition has been depicted in the below schematic.

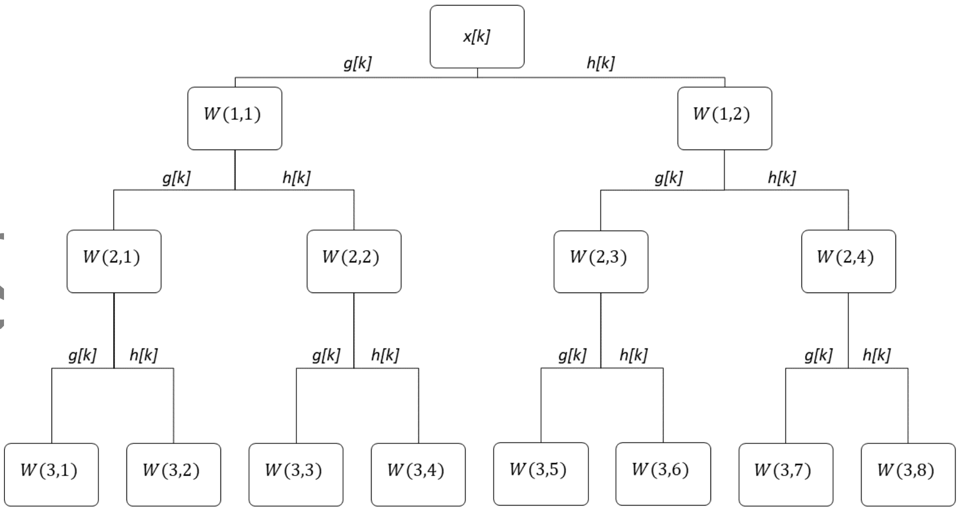

Fig. 3: Wavelet Packet Tree

If the original signal has *N* samples, the number of coefficients in each packet at level *l* will be $\frac{N}{2^l}$. Therefore, for a given packet *W(l,k)*, the coefficients in the packet k can be represented as:

$$W(l,k) = \{w_1(l,k), w_2(l,k), w_3(l,k), \ldots..w_p(l,k)\} \tag{13}$$

Where $p = \frac{N}{2^l}$, and the energy of each packet is calculated as:

$$E_i(l,k) = \sum_{i=1}^{p} \{w_i(l,k)\}^2 \tag{14}$$

The frequency resolution of each packet being:

$$f_p = \frac{F_s}{2^l} \tag{15}$$

Where F_s is half of the sampling frequency. For a sampling frequency of 6.827 kHz, a decomposition level of l = 9 is chosen which results in a frequency resolution ≈ 6.67 Hz of each packet using Eq. (15). The Daubechies 6 wavelet is used on the data acquired from the accelerometer because of it's effectiveness in crack detection as stated in [23].
Due to downsampling, the wavelet coefficients at decomposition level 9 are significantly less as compared the original sample size, the size of the signal being used as input to WPT is set as 13654 which significantly reduces the number of samples available for training and testing the Machine Learning Algorithms, the data has been augmented and number of samples increased using sliding window with 75% overlap in the Time Domain, following which, the energies have been extracted. No noise was added to the accelerometer data due to the inherent noise being present in the signal.

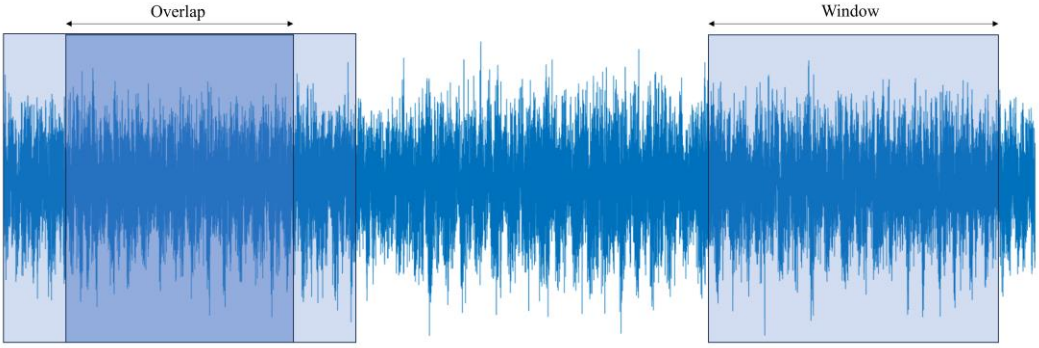

Fig. 4: Overlap and sliding window on the signal

3 EXPERIMENTAL SETUP

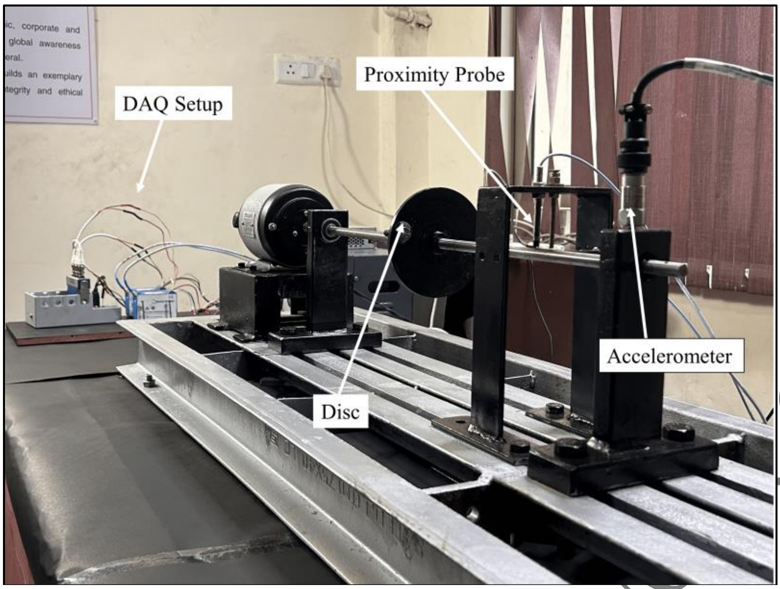

Fig. 5: Experimental setup

The experimental setup consisted of a Standard steel shaft of 10 mm diameter and a span of 0.515m and a slit at the center of the span which simulated a crack, induced using a jeweler's saw of 0.3mm thickness similarly used in [1]. A DC motor of 0.33kW rating and capable of rotating the shaft at 6000 rpm was used.

β has been defined as the ratio of depth of the slit to the diameter of the shaft, the value of β was increased after the data had been collected for each value on the same shaft, without altering the setup. The data is collected for steady state conditions at 1500 (25 Hz) and 2000 (33.33 Hz) rev/min for four values of β which are 0, 0.2, 0.3 and 0.4. The monitoring equipment used consisted of two Bently and Nevada proximity probes (3300 XL 5/8mm) with sensitivity of 7.87 V/mm mounted on the X and Y axis and one accelerometer (AM3100 T2 Z2) of 100mV/g sensitivity mounted on a support. Data from a single proximity probe was used for the study

NI 9232 DAQ card and MATLAB were used to convert the signal from analog to digital with a sampling frequency of 6.827 kHz and the data from all three sensors was saved as 1D arrays. An unbalance disk was used for all shaft conditions to better study the effect of crack and its behavior over time.

Table 2: Specifications of Data acquired at steady state

Crack Depth (*β*)	Rotational Speed (Hz)
0	25, 33.33
0.2	25, 33.33
0.3	25, 33.33
0.4	25, 33.33

4 RESULTS AND DISCUSSION

For all models, the data has been split into a training set, a validation set and a testing set. The number of samples in each set have been described in Table 3. The samples were first randomized before being split into sets. Randomizing the samples before splitting them into sets and training the model yields different results in each randomized iteration as the training set for the model is changed. The results generated are for a single randomized iteration, the samples are randomized once, split into sets and used to train the model. There are no common samples in any set.

For BO CNNs, the validation accuracy was chosen as the objective to be maximized to improve the model's performance on unfamiliar data, data which was not used in training the model. Once the optimal model was obtained, the confusion matrices were obtained on the test set, which was used neither in the training set nor in the validation set.

Table 3: Division of data for training and testing model for BO CNNs

Data	Number of samples
Training set	90
Validation set	25
Test set	25

The following metrics have been used to evaluate the performance of the models. The weighted F1 score is not used since the number of instances of each class is equal. The Macro F1 Score was converted to percentage

$$Macro\ F1\ Score = \frac{\sum_{i=1}^{n} F1\ Score_i}{n}$$

The F1 Score of each class is calculated and depends on the precision and recall.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

The labels of True Positive, False Positive and False Negative are determined from the following reference table (Table 4) where an example of Binary classification has been considered, for a given class, all values present in the same row except from the class are considered as false negatives whereas all values present in the same column excluding the considered class are considered as false positives.

Table 4: Confusion Matrix for Binary Classification

		Predicted Label	
		Positive	**Negative**
Actual Label	**Positive**	*True Positive*	*False Negative*
	Negative	*False Positive*	*True Negative*

4.1 BAYES OPTIMIZED CNNs

The number of trials for each condition were 200, with 250 epochs in each trial. It has been observed that multiple extremes were obtained for conditions with SNR=0. The optimized hyperparameters for each condition have been stated in Table 4 excluding the weights and biases of Layers 1 and 2. Each sample has 1035 data points.

Table 5: Optimized Hyperparameters for each condition

	25 Hz (SNR=0 dB)	25 Hz (SNR=7 dB)	33.33 Hz (SNR=0 dB)	33.33 Hz (SNR=7 dB)
Kernel Size (w)	213	53	77	33
Number of filters (n)	206	58	218	106
Max Pooling size (p)	14	26	36	34
Dropout	0.1	0.3	0.2	0

As compared to signals with noise, the model performed better on signals with no noise (SNR=0) as deducted from Table 5.

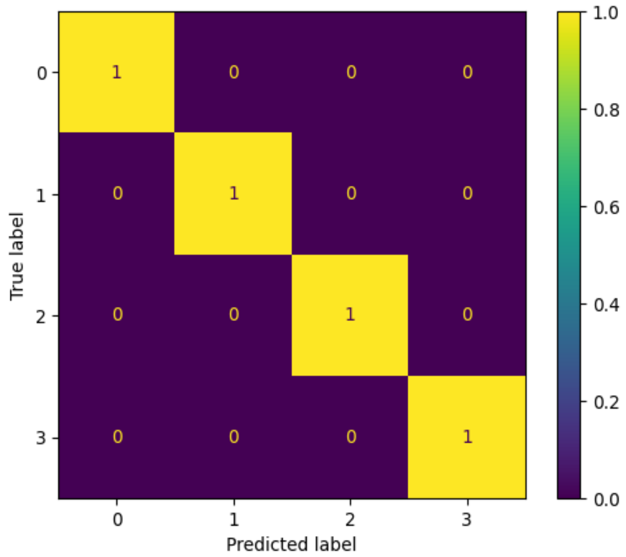

Fig. 6: Normalized confusion matrix for 25 Hz (SNR=0 dB)

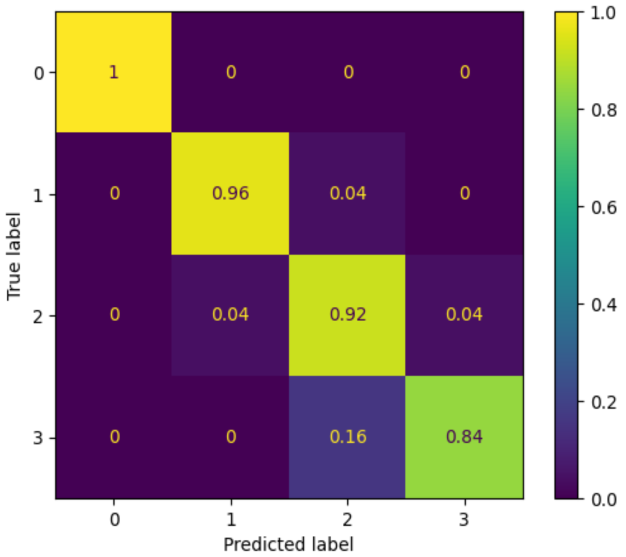

Fig. 7: Normalized confusion matrix for 25 Hz (SNR=7 dB)

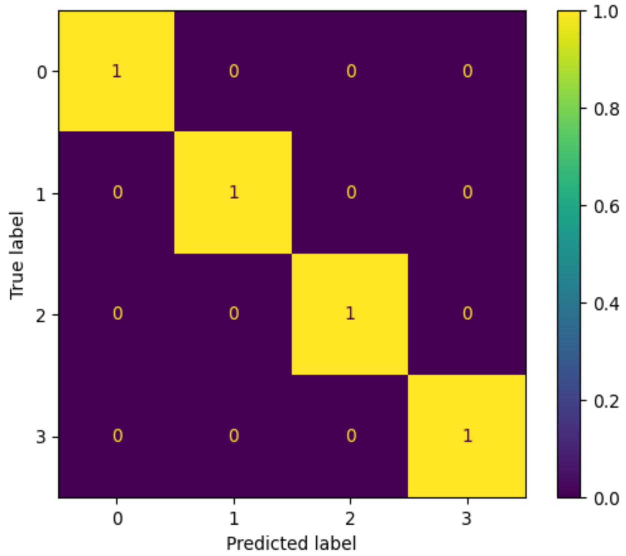

Fig. 8: Normalized confusion matrix for 33.33 Hz (SNR=0 dB)

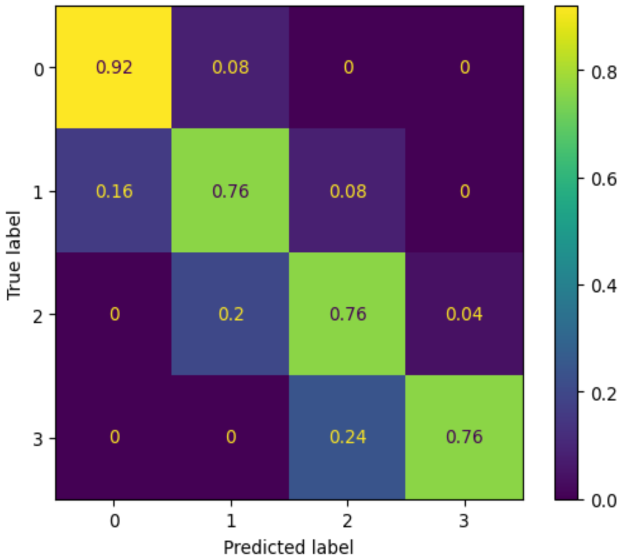

Fig. 9: Normalized confusion matrix for 33.33 Hz (SNR=7 dB)

Table 6: Results for BO CNN

	25 Hz (SNR=0 dB)	25 Hz (SNR=7 dB)	33.33 Hz (SNR=0 dB)	33.33 Hz (SNR=7 dB)
Macro F1 Score	**100%**	**93%**	**100%**	**81%**

It is observed that BO CNNs performed better at lower rotation speed when noise was added. When no noise was added to the signal, the performance was similar for both 25 Hz and 33.33 Hz, However, a decrease in the metric by 7% and 17% is observed on addition of noise in 25 Hz and 33.33 Hz respectively.

4.2 WAVELET PACKET TRANSFORM ENERGIES

The energy of each frequency band is extracted using Wavelet Packet Transform in MATLAB. Since no argument is to be optimized, the accuracy of the model is tested on the test set directly and no validation set is used with the number of samples for training and testing mentioned in Table 3.

The description of the Machine Learning models are as follows:

- SVM with Cubic Kernel.
- KNN where number of neighbours=10 and distance metric is Euclidean.
- ANN, three layers each with 10 nodes and ReLu activation.

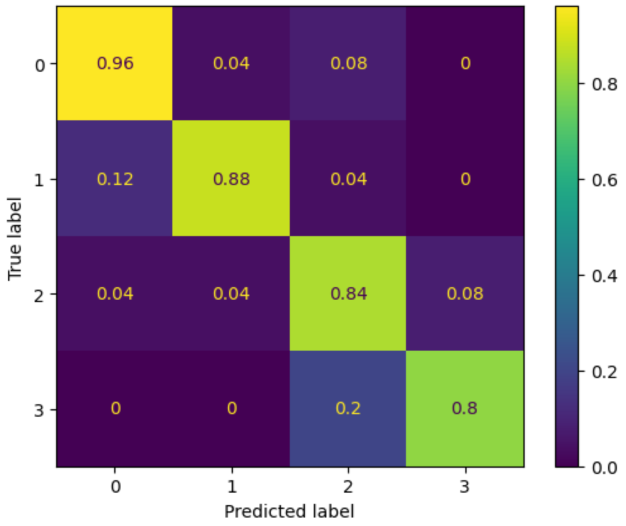

Fig. 10: Normalized Confusion Matrix for Trilayered ANN at 33.33 Hz

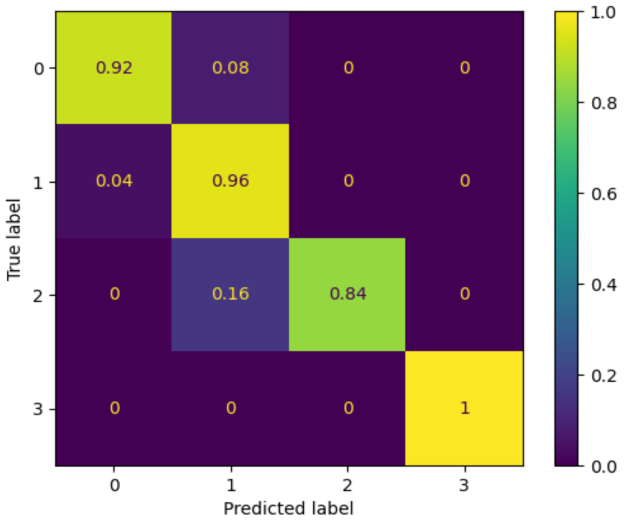

Fig. 13: Normalized Confusion Matrix for Trilayered ANN at 25 Hz

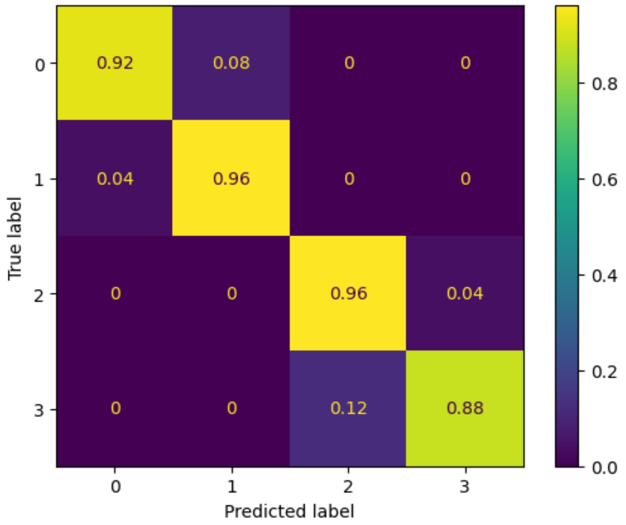

Fig. 11: Normalized Confusion Matrix for Cubic SVM at 33.33 Hz

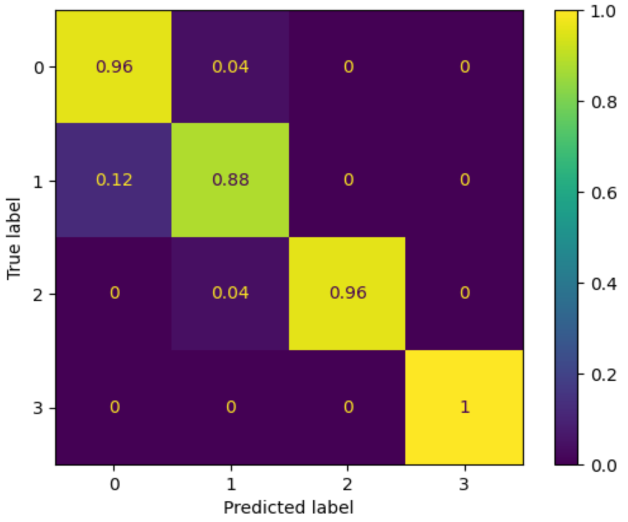

Fig. 14: Normalized Confusion Matrix for Cubic SVM at 25 Hz

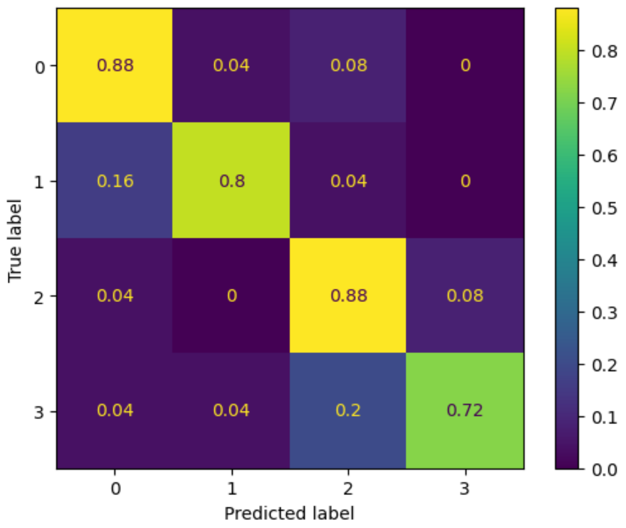

Fig. 12: Normalized Confusion Matrix for Medium KNN at 33.33 Hz

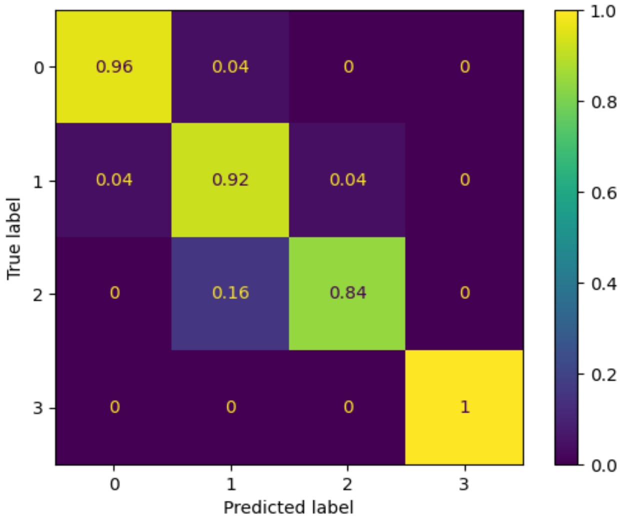

Fig. 15: Normalized Confusion Matrix for Medium KNN at 25 Hz

Table 7: Macro F1 Scores for WPT features

	Cubic SVM	**KNN**	**ANN**
25 Hz	**95%**	**93%**	**93.1%**
33 Hz	**93%**	**82.1%**	**87.1%**

SVM performed better than ANN and KNN at both whirling speeds, the metric score for all considered ML models dropped at a higher rotational speed with KNN showing the highest decline by approximately 11.72%. At a lower rotational speed (25 Hz), the models are able to accurately classify the fault with highest severity (**β**=0.4).

5 CONCLUSION

Compared to WPT for feature extraction on accelerometer data, BO CNNs performed better particularly in cases where there was no added noise in the proximity signal. The conclusion from the study are listed as follows:

- BO CNNs achieved a perfect metric score on signals when no noise was added, the metric dropped on addition of noise to the signal
- BO CNNs performed better on noisy signals at a lower whirling speed as compared to the higher considered speed in the study.
- Number of filters *n* decrease significantly on addition of noise to the sig nal, a similar observation is noted in the kernel size *w* as observed in Table 4.
- The ML algorithms performed better at a lower whirling speed as deducted from Table 6, where **β**=0.4 was correctly classified by all models whereas BO CNN gave better results at both speeds when no noise was added.
- SVM with a cubic kernel outperformed both KNN and ANN irrespective of the whirling speed.

6 ACKNOWLEDGMENTS

The researchers are thankful to Thapar Institute of Engineering and Technology and Dr. Jaskaran Singh for providing the experimental setup, the help and assistance provided by the laboratory staff and Department of Mechanical Engineering is duly acknowledged.

REFERENCES

[1] Darpe AK, Gupta K, Chawla A. Transient response and breathing behaviour of a cracked Jeffcott rotor. Journal of Sound and Vibration. 2004;272(1-2):207-43. https://doi.org/10.1016/S0022-460X(03)00327-4

[2] Mohammed AA, Neilson RD, Deans WF, MacConnell P. Crack detection in a rotating shaft using artificial neural networks and PSD characterisation. Meccanica. 2014;49:255-66. https://dx.doi.org/ 10.1007/s11012-013-9790-z

[3] Prabhakar S, Sekhar AS, Mohanty AR. Detection and monitoring of cracks in a rotor-bearing system using wavelet transforms. Mechanical Systems and Signal Processing. 2001;2(15):447-50. https://doi.org/10.1006/mssp.2000.1381.

[4] Darpe AK. A novel way to detect transverse surface crack in a rotating shaft. Journal of sound and vibration. 2007;305(1-2):151-71. https://doi.org/10.1016/j.jsv.2007.03.070.

[5] Babu TR, Sekhar AS. Shaft crack identification using artificial neural networks and wavelet transform data of a transient rotor. Adv. Vib. Eng. 2010;9:207-14.

[6] Gómez MJ, Castejón C, García-Prada JC. Crack detection in rotating shafts based on 3× energy: Analytical and experimental analyses. Mechanism and Machine Theory. 2016;96:94-106. https://doi.org/10.1016/j.mechmachtheory.2015.09.009

[7] Gómez MJ, Castejón C, García-Prada JC. Automatic condition monitoring system for crack detection in rotating machinery. Reliability Engineering & System Safety. 2016 Aug 1;152:239-47. https://doi.org/10.1016/j.ress.2016.03.013

[8] Pricop M, Pazara T, Pricop C, Novac G. Crack detection in rotating shafts using combined wavelet analysis. Journal of Physics: Conference Series 2019;1297(1):012031. https://dx.doi.org/10.1088/1742-6596/1297/1/012031

[9] Munoz-Abella B, Ruiz-Fuentes A, Rubio P, Montero L, Rubio L. Cracked rotor diagnosis by means of frequency spectrum and artificial neural networks. Smart Structures and Systems. 2020;25(4):459-69. https://doi.org/10.12989/sss.2020.25.4.459

[10] Deng W, Nguyen KT, Medjaher K, Gogu C, Morio J. Rotor dynamics informed deep learning for detection, identification, and localization of shaft crack and unbalance defects. Advanced Engineering Informatics. 2023;58:102128. https://doi.org/10.1016/j.aei.2023.102128

[11] Jin Y, Hou L, Lu Z, Chen Y. Crack Fault Diagnosis and Location Method for a Dual-Disk Hollow Shaft Rotor System Based on the Radial Basis Function Network and Pattern Recognition Neural Network. Chinese Journal of Mechanical Engineering. 2023;36(1):35. https://doi.org/10.1186/s10033-023-00856-y

[12] Karyofyllas G, Giagopoulos D. Condition monitoring framework for damage identification in CFRP rotating shafts using Model-Driven Machine learning techniques. Engineering Failure Analysis. 2024;158:108052. https://doi.org/10.1016/j.engfailanal.2024.108052

[13] Yuhong JI, Lei HO, Yushu CH, Zhenyong LU. An effective crack position diagnosis method for the hollow shaft rotor system based on the convolutional neural network and deep metric learning. Chinese Journal of Aeronautics. 2022;35(9):242-54. https://doi.org/10.1016/j.cja.2021.09.010.

[14] Zhao W, Hua C, Dong D, Ouyang H. A novel method for identifying crack and shaft misalignment faults in rotor systems under noisy environments based on CNN. Sensors. 2019 Nov 25;19(23):5158. https://doi.org/10.3390/s19235158.

[15] Dao F, Zeng Y, Qian J. Fault diagnosis of hydro-turbine via the incorporation of bayesian algorithm optimized CNN-LSTM neural network. Energy. 2024;290:130326. https://doi.org/10.1016/j.energy.2024.130326.

[16] Kiranyaz S, Avci O, Abdeljaber O, Ince T, Gabbouj M, Inman DJ. 1D convolutional neural networks and applications: A survey. Mechanical systems and signal processing. 2021;151:107398. https://doi.org/10.1016/j.ymssp.2020.107398.

[17] Ince T, Kiranyaz S, Eren L, Askar M, Gabbouj M. Real-time motor fault detection by 1-D convolutional neural networks. IEEE Transactions on Industrial Electronics. 2016;63(11):7067-75. https://doi.org/10.1109/TIE.2016.2582729.

[18] Eren L, Ince T, Kiranyaz S. A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. Journal of Signal Processing Systems. 2019;91(2):179-89. https://doi.org/10.1007/s11265-018-1378-3.

[19] Eren L. Bearing fault detection by one-dimensional convolutional neural networks. Mathematical Problems in Engineering. 2017;2017(1):8617315. https://doi.org/10.1155/2017/8617315

[20] Yu T, Zhu H. Hyper-parameter optimization: A review of algorithms and applications. arXiv preprint arXiv:2003.05689. 2020. https://doi.org/10.48550/arXiv.2003.05689

[21] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research. 2014;15(1):1929-58. https://doi.org/10.5555/2627435.2670313

[22] Wu J, Chen XY, Zhang H, Xiong LD, Lei H, Deng SH. Hyperparameter optimization for machine learning models based on Bayesian optimization. Journal of Electronic Science and Technology. 2019;17(1):26-40.https://doi.org/10.11989/JEST.1674-862X.80904120

[23] O'Malley T, Bursztein E, Long J, Chollet F, Jin H, Invernizzi L, et al. KerasTuner. 2019.

[24] Castejón C, García-Prada JC, Gómez MJ, Meneses J. Automatic detection of cracked rotors combining multiresolution analysis and artificial neural networks. Journal of Vibration and Control. 2015 Nov;21(15):3047-60. https://doi.org/10.1177/1077546313518816