Bayesian convolutional neural network models for uncertainty-aware bearing fault detection

A. Mostafavi [1,2], A. Friedmann [1]
[1]Fraunhofer Institute for Structural Durability and System Reliability LBF,
Darmstadt, Hessen, 64289, Germany

[2]Technische Universität Darmstadt, Department of Mechanical Engineering,
Darmstadt, Hessen, 64287, Germany
e-mail: **atabak.mostafavi@lbf.fraunhofer.de**

Abstract
Traditional classifiers are prone to overfitting and may exhibit spuriously high confidence, even when classifying low-quality data. Thus, it is important to have a reliable quantification of data and model uncertainty to make informed decisions about the system's condition. Probabilistic models, such as Bayesian, have been extensively researched to quantify uncertainty. However, recent developments have enabled us to successfully implement Bayesian in the backpropagation algorithm. By leveraging the properties of Bayesian and convolutional neural network, it is possible to classify directly from raw data and have a confidence band for each diagnosis. Here a Bayesian Convolutional Neural Network using back propagation has been implemented to classify common faults in bearings. The methodology has been evaluated using MAFAULDA open dataset which is composed of vibration signals from different experimentally simulated bearing conditions.

1 Introduction

Uncertainty is a fundamental aspect of human experience and a ubiquitous feature of the world we inhabit. It emerges from various sources, including incomplete information, limited observation, and the inherent complexity of the world. Researchers from diverse disciplines have explored the nature of uncertainty and its profound implications for human thought, behavior, and decision-making. Despite the absence of a single, universally accepted definition, uncertainty is generally understood as the state of lacking complete or exact knowledge about a particular subject. Uncertainty can be classified into three distinct categories: ambiguity, probability, and vagueness. Ambiguity arises from phenomena that can possess multiple distinct meanings, potentially impeding comprehension. Probability relates to the laws of chance, while vagueness encompasses a continuum of possible values [1]. Scholars have proposed various definitions of uncertainty, reflecting its multifaceted nature. One argument is that uncertainty is distinct from risk, emphasizing its elements of randomness and unpredictability [2]. Furthermore, risk – especially in a technical sense – is understood as the connection of probability of an event to happen with its severity.

Uncertainty constitutes an inherent component of decisions concerning the future behaviour of complex systems over extended periods. It is indisputable that uncertainties are present and must be considered by decision-makers. However, there is ambiguity regarding the appropriate treatment of these uncertainties to facilitate defensible decisions concerning compliance. The primary challenge in condition assessment lies in translating the outcomes of uncertain predictions, which typically do not offer unequivocal answers, into a format suitable for decision-makers. Ultimately, decision-makers are tasked with making binary (yes or no) decisions regarding whether the system meets performance objectives. [3]

In the context of model prediction, two primary categories of uncertainty can be discerned: aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty also designed as variability, stochastic, irreducible, and type A uncertainty arises from the unpredictable and random characteristics inherent in the physical system under examination. Conversely, epistemic uncertainty also known as the state of knowledge, subjective, reducible, and type B uncertainty is a consequence of an inadequate understanding of the system, particularly concerning its quantities and processes. While aleatoric uncertainty can only be identified and quantified, epistemic uncertainty is amenable to reduction through more extensive study. Employing a thorough sensitivity analysis is an effective means to diminish epistemic uncertainties. [4] Epistemic uncertainty can be further categorized in parameter and model uncertainty:

Parameter uncertainty: This refers to uncertainty derived from the values of input parameters in a model, such as measurement errors, sampling errors, variability, and use of surrogate data. Hence, it is a type of epistemic uncertainty and can be reduced by collecting more reliable evidence to more accurately estimate the parameters used in the model.

Model uncertainty: This indicates uncertainty about a model structure and the mathematical relationships of components defined in the model. Thus, this uncertainty is introduced by missing or incomplete information, which makes it hard to fully define the model. [5]

Process perturbations, measurement noise, model nonlinearities, and various other sources of uncertainty, including parametric and epistemic uncertainty, present significant challenges to fault detection, particularly in the increasingly complex landscape of industrial systems. Some of these sources of uncertainty are inherent and persist regardless of the fault detection method employed or the system under analysis. Disturbances, transient conditions, and signal noise may exhibit varying magnitudes depending on the system but are consistently present and must be carefully considered. Similarly, emergent properties and nonlinearities are constant features of engineering systems. In such cases, fault detection methods must be designed to be robust against these uncertainties while still remain sensitive enough to identify faults in advance. Failure to adequately address these uncertainties can result in their propagation to fault symptoms, impairing not only their detection but also their diagnosis and evaluation. Overly sensitive detection methods may lead to a high rate of false alarms, while insufficient sensitivity may compromise fault diagnosis capacity. Furthermore, detection methods that are excessively sensitive may also detect fault conditions in highly advanced stages, rendering timely remediation actions impossible and ultimately leading to system failure. Moreover, in the case of detection methods reliant on training data, the presence of excessive noise levels in the measurement data may have an opposite effect, reducing the sensitivity of the method as small variations may not be discernible during the training phase [6]. Some causes of uncertainty can be names as:

- Unpredictability: Unpredictability arises when a system or entity exhibits chaotic and variable behavior over space and time, which is often triggered by statistical noise and addressed by confidence intervals. Even as a system learns and adapts to dynamic new conditions, it may continue to exhibit highly variable behaviors. This variability can be caused by unreliability in information, data, or the system itself, influenced by system or network dynamics, non-stationary environmental conditions. Detecting and excluding unreliable sources or data can help reduce this type of uncertainty in the decision-making process.
- Incomplete Knowledge: This may result from a lack of evidence or knowledge due to incomplete theoretical understanding or unreliable information, which can be mitigated by advances in science and technology. Moreover, when decision-makers are overwhelmed with complex information, they may struggle to process it effectively due to cognitive limitations. To address this, individuals often simplify available data, focusing on essential features while neglecting less important or noisy information.
- Multiple Knowledge Frames: Multiple knowledge frames occur when the same information, such as evidence or opinions, is interpreted differently, leading to conflicting views. Ambiguity describes the presence of multiple valid beliefs about a certain phenomenon. Differences in understanding a system or the external world may arise from variations in defining the system's boundaries or determining the focus of attention. Discrepancies can also result from differences in interpreting information about the system. Conflicting evidence may further contribute to this uncertainty, where some information may be incorrect, irrelevant, or the observation model of a system may be incorrect at a given time. Additionally, various observers may offer different opinions based on their subjective perspectives. [5]

Effectively managing uncertainty is essential for navigating the complexities of systems, which encompass various dynamics and interactions. Coping with uncertainty involves seeking information, developing contingency plans, and cultivating resilience. Every decision, whether made by a human or a machine, inherently involves elements of uncertainty which needs to be acknowledged.

1.1 Uncertainty in machine learning

Deep learning has triggered a revolution in machine learning, offering solutions to address traditionally challenging problems. However, deep learning models are susceptible to overfitting, which negatively impacts their ability to generalize. Additionally, these models often exhibit overconfidence in their predictions, particularly when providing a confidence interval. This poses significant challenges in applications where undetected failures can have severe consequences, such as autonomous driving, medical diagnosis, and finance. Consequently, various approaches have been proposed to mitigate this risk. Among these, the Bayesian paradigm offers a robust framework for analyzing and training uncertainty-aware neural networks, and more broadly, for supporting the development of learning algorithms. In contrast with traditional statistical approaches, the Bayesian paradigm provides a comprehensive and systematic method for modeling uncertainty [7].

2 Approach

2.1 Network Weights in Bayesian convolutional Neural network

In traditional neural networks, activation value z_i^l of i^{th} neuron of the layer l for input value of x_i^{l-1} can be calculated knowing the set of related weights w_{ji}^l which connects j^{th} neuron in the layer $l-1$ to the i and biases of b_j^l using the equation (1): [8]

$$z_i^l = \Sigma_{i=1}^{n}(w_{ji}^l \cdot x_i^{l-1} + b_j^l) \tag{1}$$

Subsequently, a cost function is minimized using the parameters of the network. In traditional neural networks, each parameter is a single value that can be easily updated using optimization algorithms such as gradient descent. In contrast, in Bayesian Convolutional Neural Networks (BCNN), each weight or bias is defined by a distribution and its parameter. Similarly, in convolutional layers, filters are defined in terms of a distribution. Bayesian Neural Networks (BNNs) can be conceptualized as an infinite set of ensembles that capture the uncertainty of each network parameter. However, this presents numerous challenges for backpropagation through a stochastic node. Assuming a Gaussian distribution, each weight, bias, or filter would be defined by two parameters: the mean (μ) and the standard deviation (σ), This significantly increases the number of learnable parameters of the network. Therefore, an unbiased network has been designed to further reduce the size of learnable parameters. Figure 1 depicts a simple illustration of the difference between Bayesian traditional neural networks (BNN) and traditional neural networks.

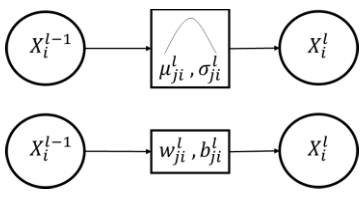


Figure 1 Top: Distributed weights for Bayesian neural network. Bottom: Fixed single-value weights as in traditional neural network.

2.2 Cost function in BNN

In the training phase of a Bayesian neural network (BNN), the objective is to find the best-suited posterior distribution of the weights given the training data. Subsequently, the prediction distribution for unseen data $\hat{x}$ with an unknown labell $\hat{y}$ is determined by the expectation value, as follows in equation (2):

$$p(\hat{y}|\hat{x}) = E_{P(w|D)}[P(\hat{y}|\hat{x}, w)] \tag{2}$$

Where w and D are respectively the set of network weights and the training dataset. This implies that every possible set of weights, weighted by its own prior probability $P(w|D)$, would predict a label for data $\hat{x}$. Therefore, taking an expectation under the posterior distribution on weights is equivalent to using an ensemble of an uncountably infinite number of neural networks. Unfortunately, this is intractable for neural networks of any practical size. Hence, a variational approximation of the Bayesian posterior distribution of the weights has been suggested. This approximation finds the parameters θ of a distribution on the weights $q(w|\theta)$ by minimizes the Kullback-Leibler (KL) divergence with the true Bayesian posterior on the weights. It can be shown that minimizing the negative of the evidence lower bound (ELBO) is equivalent to minimizing the KL divergence. Thus, we define ELBO as:

$$\mathrm{ELBO} = E_{q(w|\theta)}[\log q(w|\theta) - \log P(w)\,(D|W)] \tag{3}$$

The cost function defined in equation (3) consists of two components: a data-dependent part, referred to as the likelihood cost, and a prior-dependent part, referred to as the complexity cost. This cost function represents a trade-off between fitting the complexity of the data D and adhering to the simplicity prior $P(w)$. [9]

2.3 Reparameterization trick

Thus far, we have identified a suitable approximation that can address the intractability of the Bayesian cost function, however, to address the challenge of backpropagation through stochastic nodes, we introduce the reparameterization trick. Let z be a continuous random variable, and $z \sim q(z|\hat{x})$ be some conditional distribution. It is then often possible to express the random variable z as a deterministic variable $z = g(\epsilon, x)$, where ϵ is an auxiliary variable. This reparameterization allows us to rewrite an expectation $q(z|x)$ such that the Monte Carlo estimate of the expectation is differentiable. For a detailed proof, please refer to the provided references. [10]

2.4 1D BCNN Architecture

The CNN is a type of feedforward neural network that is able to extract features from data with convolution structures. Different from the traditional feature extraction methods, CNN does not need to extract features manually [11]. This enhances the capability of automation and enables us to process more data. However, an increase in the number of learnable parameters is inevitable. In the application of BCNN, the number of learnable parameters is particularly important, as every weight in the system is defined by distribution parameters, which drastically increases the number of learnable parameters.

Although 1D and 2D CNNs are similar in theory and mechanism, the computational complexity of 1D and 2D convolution calculations differs. Since 1D CNN operates with one dimension less, it results in a significantly lower number of learnable parameters, reducing the computational cost and the amount of necessary data. Here, we introduce a 1D BCNN model (refer to Figure 2) along with the associated hyperparameters (see Table 1). The application of 1D CNN allows us to employ shallower networks and avoids the inclusion of irrelevant information that may result from the conversion of 1D to 2D data.

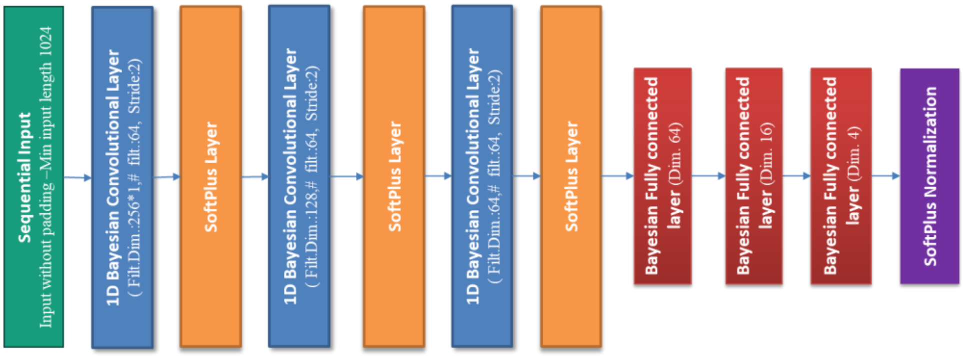


Figure 2 1D BCNN Architecture

Table 1 Model Hyper parameters

Model Hyperparameters		**Bayesian Hyperparameters**	
Mini batch size	10	Posterior	Gaussian
Max epoch	30	Prior	Gaussian Mixture
Network selection	Best Validation Accuracy	Mixture ratio	0.5
optimizer	Adam	Posterior initial parameter	$\mu = 0$ $\sigma_1 = 0.5$ $\sigma_1 = 1$
Learning rate	0.001	Prior learning rate	0.35
		Initial Likelihood Standard deviation	1

The proposed model's architecture differs from the traditional architecture in several aspects. Firstly, a numerically stable version of Softplus, as shown in Equation (4), has been preferred for the activation layer due to its smoothness near zero, which helps prevent vanishing gradients. Secondly, to correctly estimate the uncertainty of the input, normalization is avoided. Scaling input data with various scales would change their distribution. However, if needed, one can use a constant scale for the entire dataset. Similarly, padding has been avoided since adding zeros to the data would affect the distribution shape. Additionally, due to the embedded randomness of the architecture, dropout layers are not necessary anymore since BCNN is quite robust against overfitting. Although Softmax is a common choice for the last layer of classifiers, here, to prevent overconfidence due to the exponential penalization, a normalized Softplus layer, as shown in Equation (5), has been utilized and defined below:

$$\mathrm{Y} = \log\left(1 + \exp -|\mathrm{X}|\right) + \max\left(\mathrm{X}, 0\right) \tag{4}$$

Where X is the input vector to the Softplus layer. Thus, the normalized Softplus can be expressed as:

$$\mathrm{Y}_{\mathrm{n}}^{\mathrm{i}} = \frac{\mathrm{Y}^{\mathrm{i}}}{\Sigma_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{Y}^{\mathrm{i}}} \tag{5}$$

For $\mathrm{Y}_{\mathrm{n}}^{\mathrm{i}}$ representing the normalized Softplus value of the i^{th} class, where $\mathrm{i} = 1, \ldots, \mathrm{n}$. [12]

In addition to traditional CNN network hyperparameters, specific hyperparameters need to be carefully defined in BCNN. These include the posterior and prior probability distribution functions and their initial parameters. Here, we have chosen a Gaussian mixture probability distribution function as the posterior, which allows us to represent both high and low deviation distributions. Although the parameters of the prior are shared throughout the network, they will be learned during the training phase with their own dedicated learning rate.
As highlighted in several literature sources [9], a weighting value π between the terms of the cost function has been considered according to the equation (6). This weighting heavily influences the first few minibatches by the complexity cost, while the later minibatches are largely influenced by the data. At the beginning of learning, this is particularly useful, as for the first few minibatches, changes in the weights due to the data are slight. As more data are seen, the data become more influential and the prior less influential.

$$\pi_j = \frac{2^{M-j}}{2^M - 1} \tag{6}$$

Where M is the size of minibatch and $j = 1, \dots, M$.

2.5 Test Dataset and Preparation

The methodology has been applied on the MAFAULDA dataset. The dataset consists of 1951 multivariate time-series acquired by sensors on SpectraQuest's Machinery Fault Simulator (MFS) Alignment-Balance-Vibration (ABVT). It includes six different simulated states: normal function, imbalance fault, horizontal and vertical misalignment faults, and inner and outer bearing faults. This heterogeneous dataset involves measuring acoustic and vibration signals, providing comprehensive insights into machinery behavior and fault diagnosis. Each measurement lasts for 5 seconds, with 49 measurements for normal conditions, 197 for horizontal misalignment with angles of 0.5, 1.0, 1.5, and 2.0 degrees, 301 for vertical misalignment with angles of 0.51, 0.63, 1.27, 1.40, 1.78, and 1.90 degrees, and 333 for mass imbalance of 6, 10, 15, 20, 25, 30, and 35 grams. Bearing faults have been combined with 5, 6, 20, and 35 grams of mass imbalance to enhance the effect of the fault. The available experiment specification includes details of used equipment's, including the SpectraQuest Inc. Alignment/Balance Vibration Trainer (ABVT) Machinery Fault Simulator (MFS), Industrial IMI Sensors accelerometers, Monarch Instrument MT-190 analog tachometer, and Shure SM81 microphone. Data acquisition parameters such as sensitivity, frequency range, and measurement range are specified for each sensor. Sequences are categorized based on fault types, with details on the number of sequences per fault category, load values, and degrees of misalignment. The database is accessible online, with links provided in [10] for downloading the entire dataset or specific parts corresponding to different fault types.
Data handling has been conducted in an iterative five step procedure as depict in Figure 3. Raw vibration signals from the tangential direction of the underhang accelerometers (sensor number seven) is inputted along with the tachometer signal for each measurement. These signals are then divided into ten successive parts of half a second each. The first three rotations of each half-second signal are then extracted. The data set is then randomly divided into training (80 %), test (10 %) and valuation (10 %) sets to facilitate model evaluation and validation. Data partitioning has kept the dataset balance between the labels in all training, test and validation sets by randomly drawing from each measurement signal independently, this approach also assures equivalent existence of measurement with different rotational velocity in each set. Additionally, reducing the data to three revolutions per second helps to evaluate the model under more realistic conditions where acquiring large datasets may not be feasible.

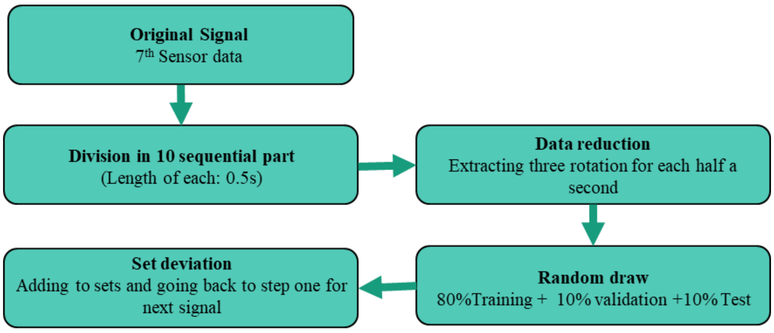


Figure 3 Data preparation scheme

3 Results

The BCNN model (see Figure 2) has been trained on underhang accelerometer data following the data handling procedure outlined in Figure 3 and using the hyperparameters specified in Table 1. The model's performance was evaluated using a test dataset with 100 samples per instance. Interpreting the results requires a more in-depth analysis to understand the uncertainty of the prediction and to establish acceptance criteria relevant to the application and its sensitivity. Statistical analysis can aid in understanding and quantifying the network's confidence in its predictions.

Characterization of network uncertainty can be conducted by measuring the distance between the mean of the prediction and the ground truth label, as well as the standard deviation of the prediction. However, in the case of classification, each prediction is bounded in the range of [0,1]. Consequently, when the network confidently assigns a label to an instance, the prediction distribution becomes asymmetric, with accumulated predictions near zero or one. Conversely, in scenarios where the network is not confident enough and there is a higher distance between predicted values and zero or one, the prediction distribution becomes symmetric and follows a Gaussian shape, as defined by a hyperparameter. In the first scenario, with an asymmetric distribution, calculating the mean has lower significance and may cause a spurious distance between the mean and ground truth. To address this issue, various statistical parameters commonly considered in asymmetric data distributions, such as skewness and median, have been investigated and reported in Table 3. However, since the distribution shape could change for each instance, these values may not provide a meaningful understanding. Therefore, the output range of zero to one has been subdivided into ten bins, and the number of members in each bin has been considered to depict the histogram of the data in Figures 4, 5, and 6. It is then suggested that the range mode of the distribution be examined instead of the mean, which is defined as the subdivided bin with the highest number of members. Consequently, the deviation of the data from the center of the mode range has been computed as a measure of data deviation. Table 2 reports these statistical values, which can represent the uncertainty of the output for three chosen cases. In the case of symmetry, the mode, mean, and median of the Gaussian distribution are the same values. Therefore, it is possible to utilize the defined parameters for both symmetric and asymmetric distributions.

Three instances of the network output have been selected to illustrate different conditions of output, including confident true prediction, unconfident true prediction, and ambiguous prediction. Figure 4 depicts an instance where the model predicts one for the second label and zero for the rest with a high level of confidence, and the prediction aligns with the ground truth label. In contrast, Figure 5 illustrates a case in which the model has lower confidence in the output of labels number two and three. This low confidence can be formalized with the high deviation from the center of the mode range reported in Table 2. Figure 6 depicts an ambiguous classification where the model is unable to distinguish between label one or three, while the ground truth is label number three.

Another perspective for result interpretation could be majority voting, which considers the label with the highest number of votes to be the prediction of the model. However, this approach oversimplifies the results and disregards much information about uncertainty, such as deviation. Therefore, the authors do not suggest such an approach and highlight that one of the advantages of the proposed methodology is the capability of keeping humans in the loop. Forcing the model to predict a label regardless of its uncertainty would undermine the value of the methodology. However, to provide a more familiar criterion for model performance, the result of the majority voting approach has been reported in Table 4 and the confusion matrix in Figure 7. Predicted samples for each instance have been rounded to zero or one with a 0.5 threshold, and the label with the highest number of votes has been assigned to the instance.

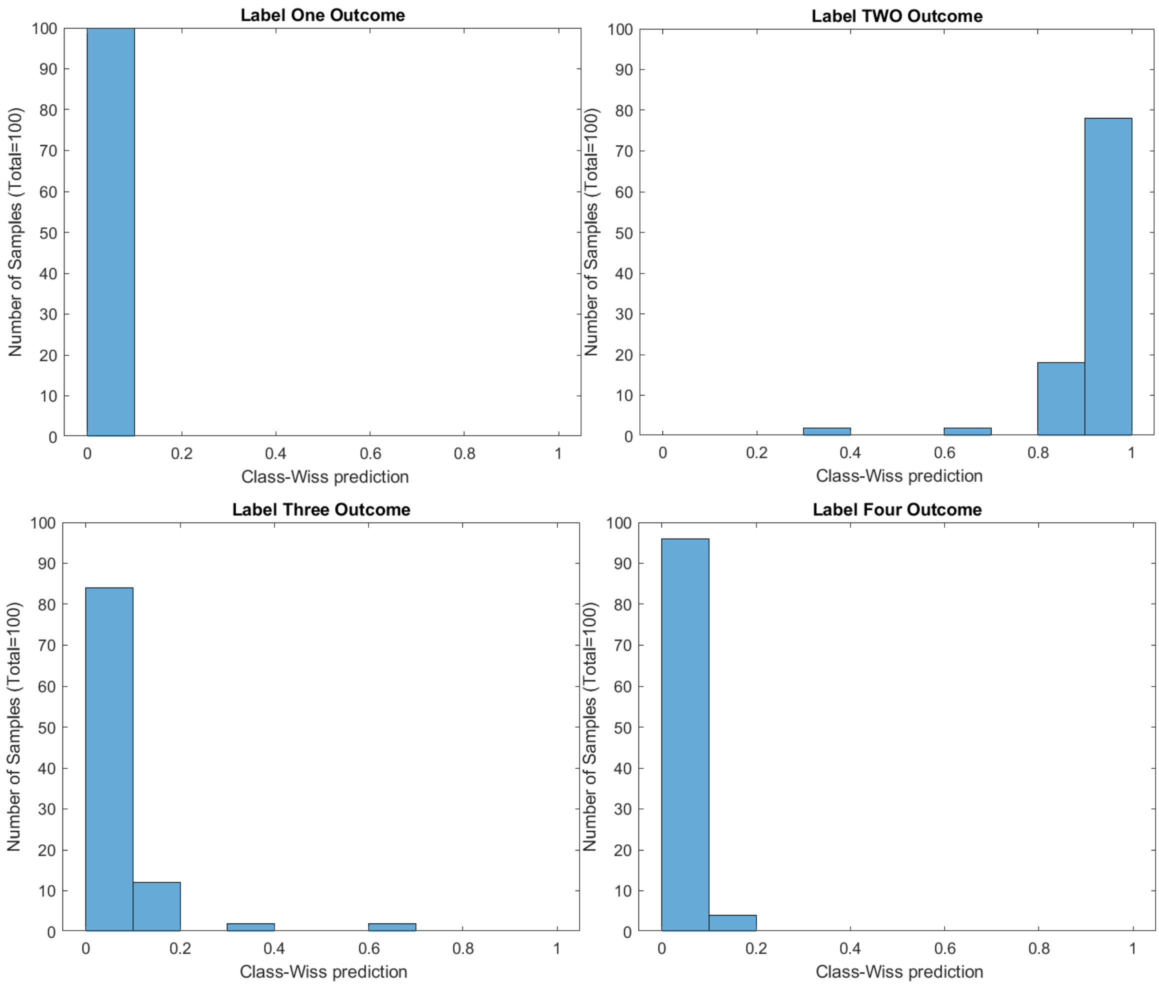


Figure 4 Case one: High confident perdition – Ground truth: label two

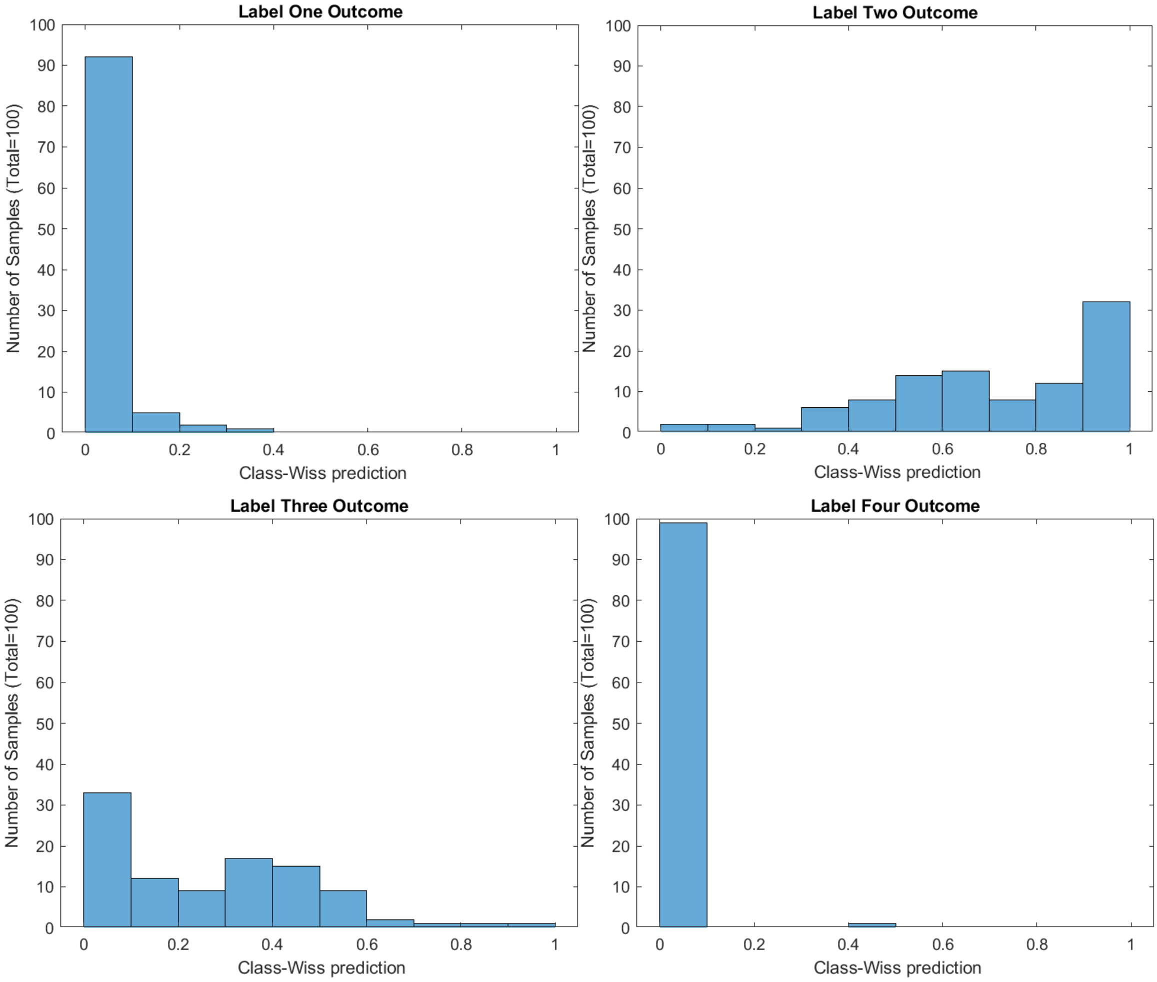
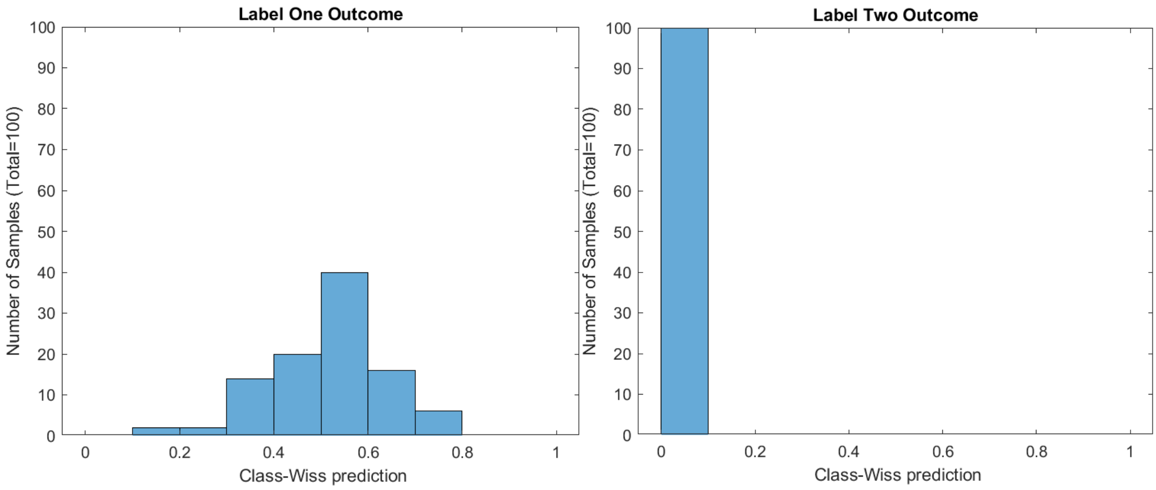


Figure 5 Case two: Low confident prediction – Ground truth: label two



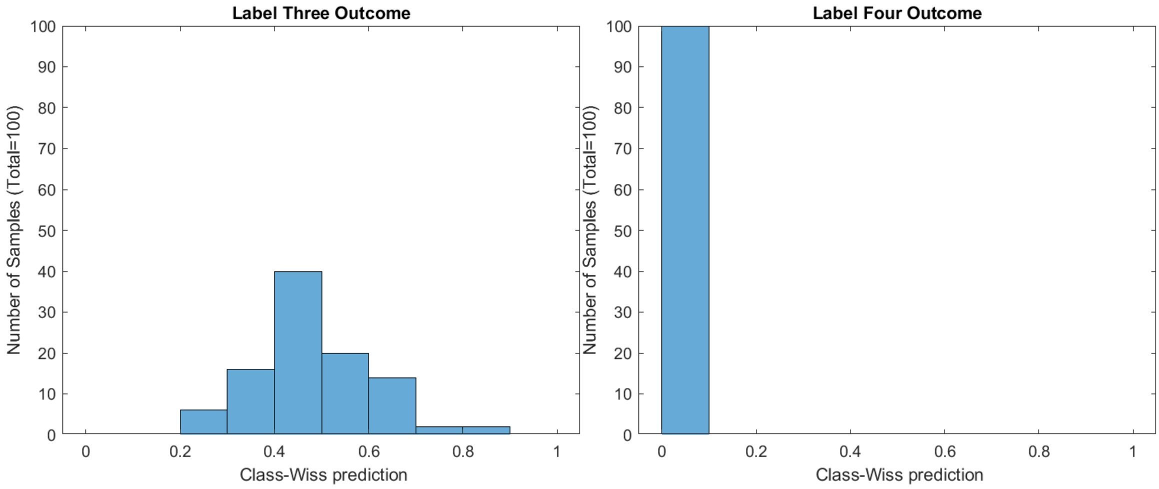


Figure 6 Case three: Uncertain prediction of network – Ground truth: label three

	Class-Wise Deviation				Class-Wise Mode range			
Labels	One	Two	Three	Four	One	Two	Three	Four
Case One	0.05	0.11	0.11	0.05	0-0.1	0.9-1.0	0-0.1	0-0.1
Case Two	0.07	0.34	0.31	0.07	0.0-0.1	0.9-1.0	0.0-0.1	0-0.1
Case Three	0.12	0.05	0.12	0.05	0.5-0.6	0.0-0.1	0.4-0.5	0.0-0.1

Table 2 Labels statistics, Deviation and range

	Class-Wise skewness				Class-Wise median			
Labels	One	Two	Three	Four	One	Two	Three	Four
Case One	0.0	-3.9	4.3	3.6	0	1	0	0
Case Two	3.3	-0.5	0.5	9.7	0	0.7	0.3	0
Case Three	-0.5	0.0	0.5	0.0	0	1	0	0

Table 3 Labels statistics, skewness and median

Label	**Multi-class model (%)**
Outer Race Fault	91.37
Cage Fault	95.94
Ball Fault	88.83
Healthy	95.43
Overall	85.79

Table 4 Class-Wise and total accuracy for majority voting

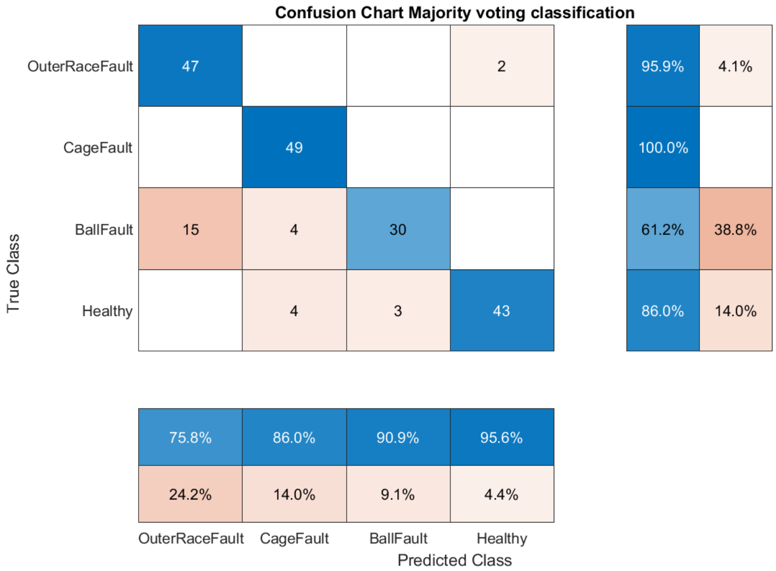


Figure 7 Confusion matrix for majority voting

4 Conclusion

A one-dimensional Bayesian convolutional neural network has been trained with variational inference for bearing condition monitoring. The proposed methodology integrates a human-in-the-loop approach for system diagnosis, providing information about the uncertainty of the diagnosis, thereby preventing overconfidence in model output. The quantification of output uncertainty allows experts to distinguish between high and low confident predictions, thus offering a more informed understanding of the true condition of the system. In instances where the BCNN model produces low-confidence predictions, one may opt to repeat the test, gather more data, or implement a periodic maintenance strategy as alternative solutions. Additionally, informing the operator about the confidence of the model, as opposed to expressing a single condition in a blunt manner, can enhance trust in the methodology by providing clarity.

Given the complexity of the methodology, both in terms of modeling and training cost, it is suitable only for safety-critical applications where false positives or false negatives have non-negligible consequences. The proposed model employs a 1D convolutional layer to reduce the dimensionality of the filters, thereby reducing the complexity and training cost of the model. A normalized Softplus layer has been utilized to prevent overconfidence in the output caused by the Softmax layer, which is a common alternative.

Acknowledgements

The author gratefully acknowledges the European commission for its support of the maria sklodowska curie program through the ETN MOIRA project (GA 955681).

References

[1] W. E. Walker et al., "Defining Uncertainty: A Conceptual Basis for Uncertainty Management in Model-Based Decision Support," Integrated Assessment, vol. 4, no. 1, pp. 5–17, 2003, doi: 10.1076/iaij.4.1.5.16466.

[2] S.-V. Toma, M. Chiriţă, and D. Şarpe, "Risk and Uncertainty," Procedia Economics and Finance, vol. 3, pp. 975–980, 2012, doi: 10.1016/S2212-5671(12)00260-2.

[3] M. W. Kozak, "Safety assessment for near-surface disposal of low and intermediate level wastes," in Geological Repository Systems for Safe Disposal of Spent Nuclear Fuels and Radioactive Waste (Second Edition) : Woodhead Publishing Series in Energy, M. J. Apted and J. Ahn, Eds.: Woodhead Publishing, 2017, pp. 475–498. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780081006429000165

[4] S. Bjarnadottir, Y. Li, and M. G. Stewart, "Chapter Nine - Climate Adaptation for Housing in Hurricane Regions," in Climate Adaptation Engineering, E. Bastidas-Arteaga and M. G. Stewar, Eds.: Butterworth-Heinemann, 2019, pp. 271–299. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128167823000097

[5] Z. Guo et al., "A Survey on Uncertainty Reasoning and Quantification for Decision Making: Belief Theory Meets Deep Learning," 12-Jun-22. [Online]. Available: http://arxiv.org/pdf/2206.05675v2

[6] G. F. M. de Souza, A. Caminada Netto, A. H. de Andrade Melani, M. A. de Carvalho Michalski, and R. F. Da Silva, "Chapter 5 - Engineering systems fault detection methods," in Reliability Analysis and Asset Management of Engineering Systems : Advances in Reliability Science, G. F. M. de Souza, A. Caminada Netto, A. H. de Andrade Melani, M. A. de Carvalho Michalski, and R. F. Da Silva, Eds.: Elsevier, 2022, pp. 119–164. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128235218000037

[7] L. V. Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, "Hands-on Bayesian Neural Networks -- a Tutorial for Deep Learning Users," IEEE Comput. Intell. Mag., vol. 17, no. 2, pp. 29–48, 2022, doi: 10.1109/MCI.2022.3155327.

[8] X. Li, Z. Liu, S. Cui, C. Luo, C. Li, and Z. Zhuang, "Predicting the effective mechanical property of heterogeneous materials by image based modeling and deep learning," Computer Methods in Applied Mechanics and Engineering, vol. 347, pp. 735–753, 2019, doi: 10.1016/j.cma.2019.01.005.

[9] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, "Weight Uncertainty in Neural Networks,"

[10] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," 20-Dec-13. [Online]. Available: http://arxiv.org/pdf/1312.6114v11

[11] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," IEEE transactions on neural networks and learning systems, vol. 33, no. 12, pp. 6999–7019, 2022, doi: 10.1109/TNNLS.2021.3084827.

[12] K. Shridhar, F. Laumann, and M. Liwicki, "Uncertainty Estimations by Softplus normalization in Bayesian Convolutional Neural Networks with Variational Inference," 15-Jun-18. [Online]. Available: http://arxiv.org/pdf/1806.05978v6